APPLYING HIDDEN MARKOV MODELS TO PROCESS MINING

Gil Aires da Silva, Diogo R. Ferreira
Instituto Superior Técnico, Campus do Taguspark
gil.aires@gmail.com, diogo.ferreira@tagus.ist.utl.pt

ABSTRACT: Process Mining is an area of active and innovative research in recent years, where the goal is to obtain a process model from a log of recorded events. Probabilistic models offer a greater degree of flexibility and are an inspiring promise for their applications in process mining. In this paper, we discuss the use of Hidden Markov Models (HMMs) for process mining, a technique often used in speech recognition and bioinformatics. The focus of this work is the use of HMMs as the underlying framework for a Sequence Clustering algorithm. We discuss the challenges currently being faced, and we present an initial view of the HMM-Based Sequence Clustering algorithm for process mining.

Keywords: Process Mining, Hidden Markov Models, Sequence Clustering, Process Discovery.

1. INTRODUCTION

Business Process Management, or BPM, can be defined as "supporting business processes using methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, organizations, applications, documents and other sources of information" [1]. In a broader sense, BPM concerns the management of the enterprise processes, such as designing process models, implementing workflow engines and supporting application integration, orchestration and choreography of business processes [2]. The supporting systems for BPM are known as *Process-Aware Information Systems* [3], where process models play a key role. The optimization of business processes is of great importance to every organization, due to financial pressures to reduce costs and increase efficiency. Process models are structures which model behavior. One of the preferred notations for modeling the processes is the Petri Net [1].

Case ID	Task Name	Owner	Timestamp
1	Buy Envelope	Gil	20/02/09 10:15
2	Buy Pre-Stamped Envelope	Álvaro	20/02/09 14:45
1	Buy Stamp	Gil	20/02/09 16:10
3	Buy Pre-Stamped Envelope	Diogo	20/02/09 17:22
2	Place Letter	Álvaro	23/02/09 08:40
1	Place Letter	Gil	23/02/09 09:20
2	Mail Regional	Álvaro	23/02/09 13:40
3	Place Letter	Diogo	23/02/09 15:50
1	Mail International	Gil	23/02/09 16:25
3	Mail National	Diogo	24/02/09 08:55

Fig. 1. Example of an event log

Process Mining, also known as Workflow Mining, consists in the extraction of a process model from an event log. This event log records the tasks executed over an information system Process Mining belongs to the wider area of BPM, using techniques to extract knowledge about business processes from event logs recorded by information systems.

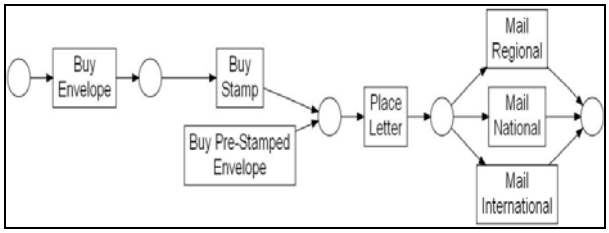


Fig. 2. Example of a process mined from the event log

In Figure 1, we present an example of an event log and a Petri Net model of the process, extracted from that log, in Figure 2. For further information about this process, see [2]. Process Mining algorithms perform well on structured processes with small amounts of noise. However, in reality it is difficult to determine the scope of a process and typically there are all kinds of disturbances in a log.

There are some environments, such as healthcare, where the processes have a high degree of variability and ad-hoc behavior. Traditional Process Mining techniques do not work well under such environments [4], and Hidden Markov Models (HMMs) based techniques offer a good promise due to their probabilistic nature. Therefore, the objective of this work is to study this more advanced probabilistic-based model, and how it can be used in connection with process mining. HMMs can be viewed as stochastic generalizations of finite-state automata, when both transitions between states and generation of output symbols are governed by probability distributions. The use of this model presents several challenges, such as understanding the best ways to estimate the HMM parameters and to define the topological structure of the model.

The remainder of this paper is structured as follows. In Section 2, we refer to existing applications of clustering in process mining. In Section 3, we introduce the algorithm, the HMM-based Sequence Clustering, and give a simple example of how it works. In Section 4, we present preliminary results and in Section 5, we describe the future work. Section 6 concludes the paper.

2. RELATED WORK

Clustering is a statistical data analysis technique for dividing objects into groups, based on the similarity. Each object in a cluster is more similar to the objects in that cluster than the objects in other clusters. Similarity can be defined in several ways, usually through a distance measure. In the context of Process Mining, we have event logs with several traces, and we need to have a technique for separating them into groups representing the behaviors of different processes.

Typically Process Mining techniques assume that a certain amount of information is present in the event log. Also, many techniques are not noise resistant (they cannot deal with incomplete or irrelevant data), or they only tolerate a small amount of noise and try to find exact process model representations, which makes them inflexible.

Ferreira et al [4] demonstrated the usefulness of applying the Sequence Clustering technique to Process Mining, which can withstand noise and provide added flexibility. The algorithm is a Model Based Clustering technique, which makes use of first-order Markov chains. But in order to capture additional knowledge about business processes, such as their main stages, or their distinctive profiles, there are more elaborated models such as HMMs. Sequence Clustering can also be used as a preprocessing technique. After separating the traces into separate clusters, existing Process Mining techniques can be used to retrieve the process model from each cluster. However, first-order Markov models such as those used in Sequence Clustering algorithms are unable to capture parallel activities, loops, non-local dependencies, or other more intricate workflow patterns [5] that often occur in business processes. HMMs have Markovian behavior between states, but also allow for a distribution of symbols within each state, which provides the ability to capture more flexible behavior.

The definition of the distance metric to measure similarity between HMMs representing a process model is also an open issue of relevance. Rozinat et al [6, 7] developed several useful metrics for analyzing the Conformance of process models, providing the means for a quantitative evaluation of models discovered from event logs.

3. SEQUENCE CLUSTERING WITH HMMS

The use of HMMs as a framework for Sequence Clustering is a relatively unexplored area, and there are relatively few references in the literature. Initial work was presented by Smyth [8], where a distance measure using HMMs was used to cluster the sequences assuming the HMM structure was known *a-priori*, as well as the number of clusters. The algorithm trains an HMM for each sequence, so that the log-likelihood (LL) of each model can be computed, given each sequence. This information is used to build a LL distance matrix to be used to cluster the sequences into the K groups, using a hierarchical algorithm. Work by Li and Biswas [9], address the clustering problem focusing on the model selection issue, i.e. the search of the HMM topology that best represents data, and the clustering structure issue, i.e. finding the most likely number of clusters. The first problem is addressed using Bayesian Information Criterion, and extending to the continuous case the Bayesian Model Merging approach. In the second problem, the sequence-to-HMM likelihood measure is used to enforce the within-group similarity criterion. The optimal number of clusters is then determined maximizing the Partition Mutual Information (PMI), which is a measure of the inter-cluster distances. In [10], Panuccio et al. extends the idea of Smyth by defining a new metric to measure the distance, in the likelihood sense, between sequences. Two clustering algorithms are proposed, one based on the hierarchical agglomerative approach, and the second based on a partition method, a variation of the K-means strategy. The HMM training initialization is made utilizing Kalman filtering, and clustering is made via a mixture of Gaussians. A distance measure between HMMs can also be defined based on the Kullback-Leibler measure [11]. Although it is often interpreted as a distance metric, this divergence is not a true metric since it is not symmetric (hence 'divergence' rather than 'distance'). It measures the difference between 2 probability distributions.

HMM-Based Sequence Clustering Example

To explain how clustering of sequences can be made using HMMs, a small example will be used, using only 3 very basic processes structures and parameter distributions. Let us assume that 3 different HMMs were created to represent 3 different processes. For this example, let us leave aside the details behind the creation of the HMMs (the parameter estimation and the structure). Then, we will assume we have an event log with several traces (10 letters sequences), and we want to discover which process (or which HMM) a trace belongs to. Due to the stochastic nature of

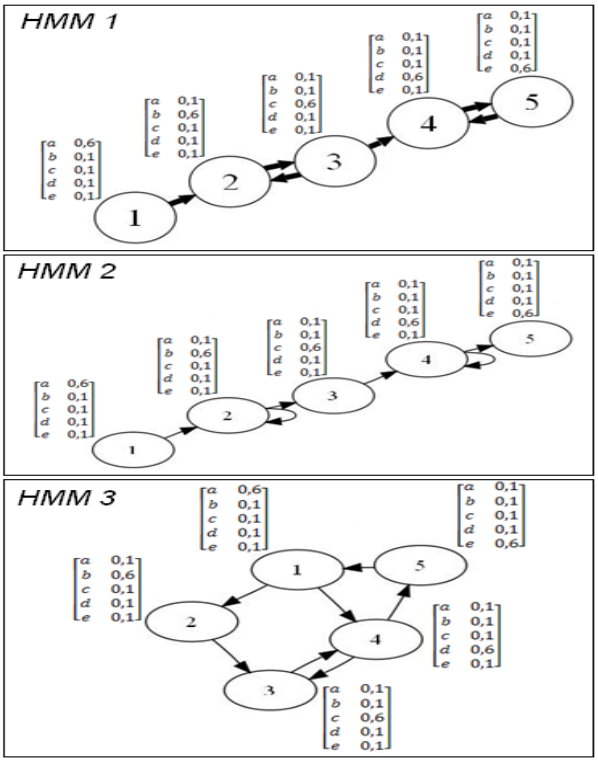
the models, we will find the most likely cluster for a given trace by finding the probability of that trace being generated from each HMM. The probability of a trace belonging to a given HMM can be calculated in the following way:

- first, the initial state (1), will give the probability of the first letter (A), which is 0.6;
- then, we move onto the second state (2), and to the second letter (B), and find the probability in that state, which is 0.6, and multiply it by the first probability, making the accumulated probability P = 0.36;
- then, we move to the third state (3), and the third letter (C), and repeat, making the accumulated probability P = 0.216;
- then, there is the possibility of moving to the forth state (4) or moving back to the second state (2); the multiple possible paths need to be experimented, and the one with the highest accumulated probability will be chosen;
- the previous steps are repeated until the input trace is complete, and there will be an accumulated probability value for the full sequence in the chosen HMM;
- the procedure is repeated for each HMM; the one with the highest result will be chosen as the cluster that input trance will be assigned to.

In Figure 3, the structure and distributions of 3 HMMs are shown, and using those values and the following 3 sequences:

- A-B-C-B-C-D-E-D-E-D
- A-B-B-B-B-C-D-D-D-E
- A-D-E-A-B-C-D-C-D-E

The accumulated probability is calculated and presented in a table format afterwards, also in Figure 3. As the table shows, each trace clearly belongs to a different cluster. Sequence 1 will go to Cluster 1, Sequence 2 will go to Cluster 2 and Sequence 3 will go to Cluster 3. Although all sequences are possible in each HMM, demonstrating the flexibility of this mathematical model, it is quite clear where each sequence is more likely to be produced from. This flexibility of HMMs becomes a very important property for process mining, since the goal is to extract models that are understandable by end users, but not necessarily a perfect match for the input event log. We often want to allow behavior that is not present in any trace of the log, because it might exist in the original model and at the time of application of the Process Mining algorithm, no manifestation of such behavior had occurred (as such, no entry in the log would reflect its existence). This means that HMMs provide models which performs well with respect to *fitness* (the model accommodates all traces in the log), but with a possible lack of *precision* (since the model allows for more behavior than that which is present in the log) [7]. The degree of imprecision in the model can be parameterized when the model is created. In the example, the problems of extracting the process model, of creating the HMM for a process model, and of finding the correct number of clusters were not addressed, but these are also very important issues, that deserve attention in their own right.



	Sequence	HMM 1	HMM 2	HMM 3
1	A-B-C-B-C-D-E-D-E-D	**6,05 x 10^{-3}**	4,67 x 10^{-6}	4,67 x 10^{-6}
2	A-B-B-B-B-C-D-D-D-E	1,30 x 10^{-7}	**6,05 x 10^{-3}**	1,30 x 10^{-7}
3	A-D-E-A-B-C-D-C-D-E	0,6 x 10^{-9}	4,67 x 10^{-6}	**6,05 x 10^{-3}**

Fig. 3. Clustering with HMMs example

4. EXPERIMENTAL SETUPS

The first problem to address is the topology of the HMM. The general problem of HMM topology design is a difficult one. Upon a review of scientific literature relating HMM applications, the following topologies emerged, generally with 3 to 5 hidden states, as shown in Figure 4:

- Linear Model (each state is connected to itself and to its only successor)
- Left-to-Right Model (each state is connected to itself and to all of its successors)
- Bakis Model (each state is connected to itself, to its successor, and to the successor's successor)

- Alternative Paths Model (there is an initial state, connected to several parallel paths, represented by a Linear Model; finally, each last state of each path is connected to a final state)
- Ergodic Model (each state is connected to itself and to every other state)

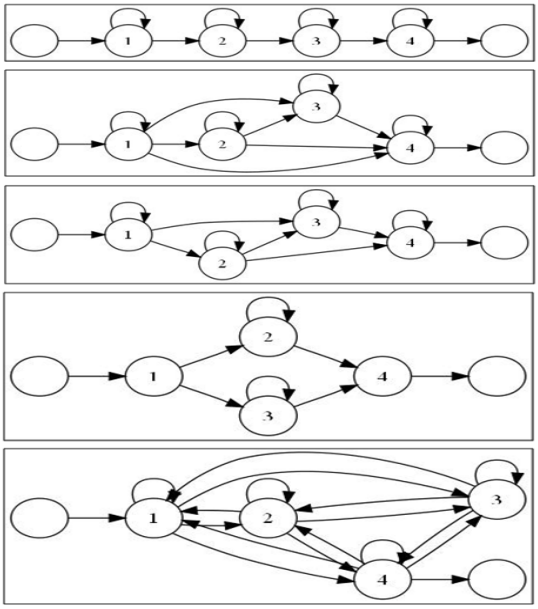


Fig. 4. The 5 HMMs topologies

The second problem to be address is the initialization and estimation of the HMMs. For estimating the HMM parameters, we use the Viterbi algorithm, and we use an Expectation-Maximization algorithm for clustering, initializing the HMM parameters with equiprobable values, and for each of the topologies, assign each sequence to each cluster, recalculate the parameters of the HMMs, and readjust the clusters until the model converges. The results we have obtained performing clustering with the 3 first topologies are very similar, and very different from the other 2 topologies, suggesting that there are fundamentally two approaches to choose from. Furthermore, using the Kullback–Leibler distance to measure similarity between the trained models, the first 3 were found to be very close to each other, and very far away for the other 2.

5. FUTURE WORK

Currently we are experimenting with different ways of estimating the parameters of the HMMs, and evaluating the results. Using the *Noise Experimenter* component developed by Rozinat et al. [7] it is possible to generate logs with large amount of noise in order to confirm the advantages of HMMs in handling noise, and also to derive statistical measures for analysis. Afterwards, a case study will be performed using highly unstructured behavior recorded in a hospital environment.

6. CONCLUSION

In this paper, we have discussed the use of HMMs to perform Sequence Clustering of event log traces and the motivation for using such models. Then we presented an initial view of the HMM-Based Sequence Clustering algorithm. We have also focused on several problems, namely defining the structure of the HMM and the estimation of its parameters, and provided our findings regarding the topology issue. The next steps will concern other methods for parameter estimation.

REFERENCES

[1] Weske, M., Business Process Management – Concepts, Languages, Architectures, Springer-Verlag, 2007

[2] van der Aalst, W.M.P. et al., Workflow Mining: A Survey of Issues and Approaches, Data and Knowledge Engineering, 47(2), pp. 237-267, 2003.

[3] Dumas, M., van der Aalst, W.M.P., ter Hofstede, A.H.M. (Eds), *Process-Aware Information Systems: Bridging People and Software through Process Technology*, pages 179-203. Wiley-Interscience, Hoboken, NJ, USA, 2005.

[4] Ferreira, D., Zacarias, M., Malheiros, M., Ferreira, P.: Approaching Process Mining with Sequence Clustering: Experiments and findings, *in* Alonso, G., Dadam, P., Rosemann, M. (Eds) BPM 2007. *LNCS*, vol. 4714, pp. 360–374. Springer, Heidelberg, 2007.

[5] W.M.P van der Aalst, A.H.M. ter Hofstede, B. Kiepuszewski, and A.P. Barros. Workflow Patterns, Distributed and Parallel Databases, 14(3), pp.5-51, July 2003.

[6] Rozinat, A., van der Aalst, W., Conformance Testing: Measuring the Fit and Appropriateness of Event Logs and Process Models, *in* Bussler, C. et al. (Eds.), Business Process Management 2005 Workshops, vol. 3812 of *LNCS*, pp. 163–176, Springer-Verlag, Berlin, 2006.

[7] Rozinat, A., Veloso, M., van der Aalst, W.M.P., Using Hidden Markov Models to Evaluate the Quality of Discovered Process Models, Extended Version, BPM Center Report, BPMcenter.org, 2008.

[8] Smyth, P., Clustering Sequences with HMM, *in* Advances in Neural Information Processing, Mozer, M., Jordan, M., Petsche, T. (Eds), MIT Press 9, 1997.

[9] Li, C., Biswas, G., Clustering Sequence Data using Hidden Markov Model Representation, Proceedings SPIE99 Conference on Data Mining and Knowledge Discovery: Theory, Tools, and Technology, pp. 14–21, 1999.

[10] Panuccio, A., Bicego, M., Murino, V., A Hidden Markov Model-Based Approach to Sequential Data Clustering, T. Caelli et al. (Eds), SSPR & SPR 2002, of *LNCS*, vol. 2396, pp. 734–743, Springer-Verlag Berlin Heidelberg 2002.

[11] Kullback, S., The Kullback-Leibler Distance, The American Statistician, vol. 41, pp. 340-341, 1987.