Sequential Clustering for Event Sequences and Its Impact on Next Process Step Prediction

Mai Le[1], Detlef Nauck[2], Bogdan Gabrys[1], and Trevor Martin[3]

[1] University of Bournemouth
[2] British Telecommunications
[3] University of Bristol

Abstract. Next step prediction is an important problem in process analytics and it can be used in process monitoring to preempt failure in business processes. We are using logfiles from a workflow system that record the sequential execution of business processes. Each process execution results in a timestamped event. The main issue of analysing such event sequences is that they can be very diverse. Models that can effectively handle diverse sequences without losing the sequential nature of the data are desired. We propose an approach which clusters event sequences. Each cluster consists of similar sequences and the challenge is to identify a similarity measure that can cope with the sequential nature of the data. After clustering we build individual predictive models for each group. This strategy addresses both the sequential and diverse characteristics of our data. We first employ K-means and extent it into a categorical-sequential clustering algorithm by combining it with sequential alignment. Finally, we treat each resulting cluster by building individual Markov models of different orders, expecting that the representative characteristics of each cluster are captured.

1 Introduction

In order to achieve operational excellence, companies must run efficient and effective processes [1], [2]. They must also be able to predict if processes will complete successfully or run into exceptions in order to intervene at the right time, preempt problems and maintain customer service.

It is a real challenge to build such models for business process data due to many reasons. Let us point out two main reasons which we think most dominant. First, business process data is sequential in nature. A business process instance (S_j) is a composition of discrete events (or tasks) in a time-ordered sequence, $S_j = \left\{s_1^{(j)}, s_2^{(j)}, \ldots, s_{n_j}^{(j)}\right\}$, s_k takes values from a finite set of event types $E = \{e_1, \ldots, e_L\}$. Each of these events has its own attributes. For simplicity, we assume that a process does not contain any overlapping events, that means there are no parallel structures [1]. Second, business process data can be very diverse because in practice, processes are typically designed based on knowledge about how a certain objective can be achieved efficiently. When process execution is not enforced by automatic workflow systems, people do not always follow the

A. Laurent et al. (Eds.): IPMU 2014, Part I, CCIS 442, pp. 168–178, 2014.

design. In large companies, many departments can be involved in the execution or maintenance of a process and processes can be implemented across a large number of IT systems. In these environments it can easily happen that over time the company loses track of the original process design and process evolves in an uncontrolled fashion. Consequently, there are many prototypes (different execution sequences) for one process [1].

Even though there is a rich source of mathematical models in data mining, not any sequential approaches seem to be effective in solving this particular problem. The solution is to 'divide' the process into smaller groups of tasks/steps and at each group, build a model accordingly [3], [4]. The shortcoming of these local models is that the coherence and the interaction between events from different event logs are lost. We propose another strategy addressing the complexity and diversity of process data which partitioning process data into groups of sequences of similar characteristics. Mathematical models are chosen according to the properties of the sequences in the resulting groups. The strength of this method is that it keeps the sequential form of the process, discriminates and adequately deals with different representative prototypes.

Existing sequential clustering approaches are found in the works of [5], [6] etc. The principle of these approaches is to build a distance measure matrix by first, modelling the data sequences one by one then comparing the likelihood of a sequence fitting to a chosen model. Any probabilistic model can be used here to describe the data e.g. linear autoregressive, graph-based models etc. HMM based sequential clustering is the most common and has shown its great performance in certain fields where data consists of continuous and/or long sequences. To fit data sequences to descriptive models, the data is assumed to have some properties or prior probabilistic distribution. It might be a better idea comparing the sequences directly based on the events and the order in which they occurred than to build a HMM for each sequence in our data, in particular for short length sequences. Hence, we use local sequence alignment (SA) to match all sequences pairwise and the outcome of the matchings are used as a similarity measure function. The proposed sequential clustering algorithm provides subgroups of sequences which are similar in the way events have occurred. Each resulting cluster is then treated by a couple of hybrid Markov models which are studed in our previous work.

The rest of the paper is organised as follows: Section 2 presents the sequence alignment technique. It is followed by Section 3 which introduces clustering and the proposed sequential clustering approach. Experiments and experimental results discussion take place in Section 4. Finally, Section 5 will conclude and draw a future research plan based on hints obtained from the former section.

2 Sequence Alignment - Similarity Measure Function

Algorithms used in sequence alignment are mainly categorised into global alignment and local alignment. Global alignment provides a global optimisation solution, which spans the entire length of all query sequences. One such algorithm

was introduced by Needleman and Wunchs [7]. In contrast, local alignment aims to find the most similar segments from two query sequences [8], [9]. In this work we use local alignment. There are two basic matrices associated with local sequence alignment algorithms: substitution matrix and score matrix. The role of the substitution matrix is to generate the degree of matching any two events from the set of event types, or in other words matching subsequences of length 1. This degree which is irrespective of the position of the events then contributes to the matching score in the score matrix that considers the complete sequences, i.e. all events in the order they occur. We now introduce these two matrices.

Substitution Matrix: In biology a substitution matrix describes the rate at which one amino acid in a sequence transforms to another amino acid over time. Regarding business process data no mutation occurs. Therefore, we do not need the complex version of the substitution matrix and we use the identity matrix.

Score Matrix: This matrix's elements are similarity degrees of events from the two given sequences considering the positions.

$$h_{i0} = h_{0j} = h_{00} = 0, \tag{1}$$

These h_{i0}, h_{0j} and h_{00} values are the initial values for the recursive formula that is used to compute h_{ij}.

$$\begin{aligned} h_{ij} = \max \{ & h_{i-1,j} - \delta, h_{i-1,j-1} + s(x_i, y_j), \\ & h_{i,j-1} - \delta, 0 \}, \end{aligned} \tag{2}$$

where $s(x_i, y_j)$ is the element of the substitution matrix and x_i, y_j are events at positions i and j. δ is the penalty for deletion or insertion. The i^{th} event in a sequence can be aligned to the j^{th} event in another sequence, or can be aligned to nothing (deletion). The optimal pair of aligned segments in this case is identified by the highest score in the matrix. The segment is then found by tracking back from that optimal highest score diagonally up toward the left corner until 0 is reached.

3 Data Clustering

Clustering is one of the main constituent elements in data mining. It is known as an unsupervised learning family. The aim of data clustering is to get the data distributed into a finite number of clusters, typically based on the distance between data points. Hence, a distance measure function is required and is vitally important. Clustering aims to group data in a way that each object in one cluster is similar to other objects in the same cluster more than to other objects in other clusters. Clustering approaches mainly are one of the following types:

- hierarchical clustering: agglomerative clustering [10], [11],
- probabilistic clustering: EM algorithm [12],
- partitioning clustering: K means clustering, K modes, K prototypes [10],

- fuzzy clustering [13],
- grid based clustering algorithm [14] (no distance required, only population of the grids is counted),
- graph-based algorithm Click [15].

As sometimes sequential features describe data best, one option is to consider each sequence as a multivariate feature vector and use vector composition based clustering to cluster the given sequences [16]. However, decomposition based approaches require sequences of the same length. Overcoming such issue, there are a large number of HMM-based sequential clustering algorithms and extensions [5], [6], [17], [18] etc. In these publications, the authors model individual data sequences by probabilistic models then use likelihood to build a distance matrix. Traditional clustering techniques are applied to partitioning the data using the obtained distance matrix.

$$d_{ij} = \frac{l(s_i, \lambda_j) + l(s_j, \lambda_i)}{2}, \tag{3}$$

where d_{ij} are the distance between sequence i and sequence j, $l(s_i, \lambda_j)$ is the likelihood of a sequence i belonging to a model $\lambda(j)$.

K Means Clustering. K means clustering in data mining is itself an NP hard problem [19]. However, heuristic K means algorithms exist and provide locally optimal solutions. Some common K means algorithms are Lloyd algorithm [19], Bradley, Fayyad and Reina algorithm [10]. K means is widely used because of its simplicity and competence. Researchers have tried to improve the original approach. One of the alternative algorithms is to modify the original K means to profit from extra information about some specific data points should or should not be assigned to certain clusters [20] (constraints of belonging). Another alternative algorithm is to make the algorithm faster by using triangle inequality to avoid unnecessary computation [21].

K is usually chosen in the first place or estimated by trial but there are also a number of studies on how to select a reasonable value for K [22]. Given the number of clusters K, first, the corresponding centers are initialised. These centers can be data points randomly taken from the available dataset. Second, each data point is assigned to the closest cluster based on the distance between the data point and K centers. Once all the data points are assigned, a new center for each cluster is determined. For numerical data, such centers are mean values of the elements in the corresponding clusters. The procedure of assigning data points to clusters and recomputing centers is repeated until it converges.

K Means Variant for Event Sequences. The sequence matching degree presented earlier is used as similarity measure in our proposed K means clustering which will be called SA based clustering from now on through out this paper. Because we directly use ordered events of data sequences to compare sequences, there is no mean value for each cluster. We resort to choose the sequence in each cluster whose total distance (similarity) to other members of the cluster is smallest (largest) as the new center.

4 Evaluation

We carried out a number of experiments based on records from two real processes ($DS1-2$) from a multi-national telecommunications company. $DS1$ is a legacy process, it consists of 2367 process instances of different lengths, varying from 1 to 78, 285 unique tasks (events) and about 20 process attributes. $DS2$ is also a real process with 11839 entries, 576 process instances with different lengths, and also has hundreds of unique tasks. The lengths of process instances in $DS2$ vary considerably. We would like to illustrate our process data and its complex property as mentioned in the introduction before getting into the main experiments. The diagram in Figure 1 shows the complexity of the subset of $DS1$. It is basically impossible to visually analyse or understand the process from this figure.

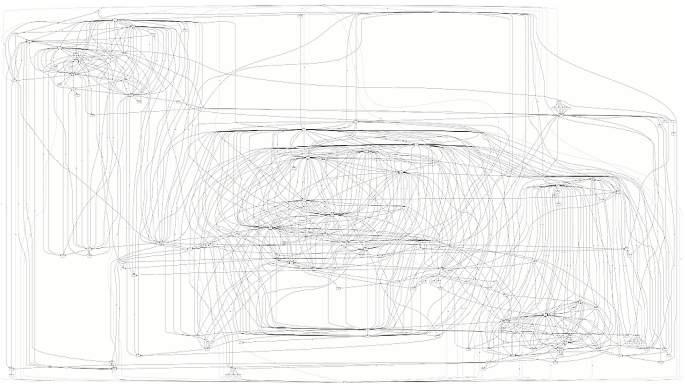

Fig. 1. Process model obtained by using Aperture visualising a highly complex process

To benchmark our proposed clustering we use HMM based sequential clustering. Each data sequence is described by one HMM. The distance between any pair of sequences is computed based on the loglikelihood of fitting the one sequence to the descriptive HMM of the other sequence. Once the distance matrix is built, K-means clustering is used. We use an available HMM toolbox in Matlab to develop the HMM based clustering.

The final goal of clustering business process data in this work is to improve the performance whilst predicting the data. We use Markov models and one of its extensions to predict the next process step in the experiments. The performances of these models applied to clustered data are means to evaluate the clustering strategy. In other words, they are proofs for verifying the impact of the strategy on the predictive capability.

- *MM - Markov Models*: in order to find the next task following the current one, we build transition matrices of different order Markov models.

– *MSA - Hybrid Markov Models*: a default prediction improvement module is added to higher order Markov models to obtain better accuracy. The default prediction is improved by comparing a new pattern to all the patterns from the transition matrix using sequence alignment. The most similar pattern found from the matrix is used as a substitution for the given one to contribute the prediction.

Our aim is to improve the accuracy of the prediction on the original data by clustering the data into groups of sequences with similar characteristics. Each group requires a suitable predictive model. To verify if we can cluster the data into such groups and if we can improve the accuracy by finding a suitable model for each group, we first present the MMs and MSAs performance on $DS1$ and $DS2$:

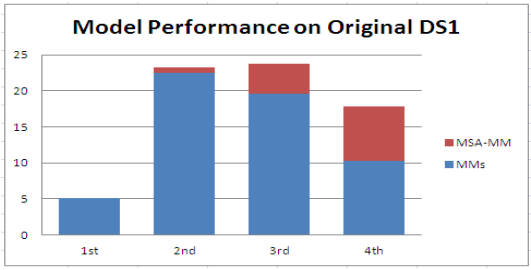

Fig. 2. Percentage correct of MM and MSA in predicting next process step using DS1

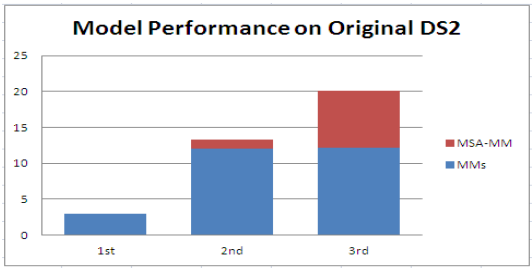

Fig. 3. Percentage correct of MM and MSA in predicting next process step using DS2

The results from Figures 2 and 3 show that the performances of these predictive models are quite low, only about 25%. We then introduce the performances of the same models applied to the clustered data. Both sequential methods, HMM based and our proposed clustering (SA based), are implemented to gain distance matrices. Therefore, K means clustering becomes sequential K means using such matrices as distance measure. Sequential K means HMM based and

SA based are used to cluster $DS1$, $DS2$ into 3 (or 6) and 2 clusters respectively. Figures 4 and 6 illustrate the performances of MMs and MSAs applied to 3 and 6-cluster-$DS1$, which we obtained by SA based clustering, respectively and Figures 5 and 7 illustrate the same with HMM based clustering.

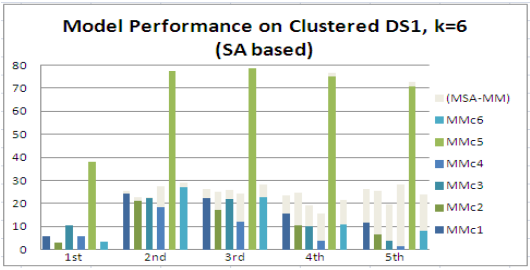

Fig. 4. Percentage correct of different order MMs and MSAs in predicting next process step using 6 clusters obtained by applying SA based clustering to dataset DS1

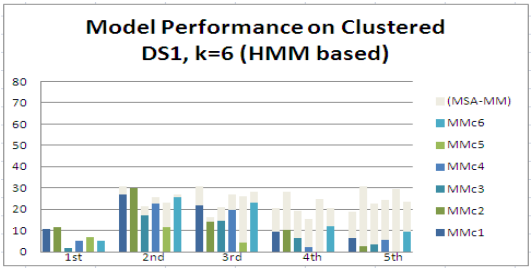

Fig. 5. Percentage of correct next process step predictions of different order MMs and MSAs using 6 clusters obtained by applying HMM based clustering to dataset DS1

As can be seen, in the case of K means SA based with $K = 6$, MMs and MSAs applied to cluster 5 have significantly high performance. The highest performance is 78.69 % (third order MSA) which is almost four times greater than the performance of the same models applying to the whole $DS1$ and 2.5 times comparing to these of the other clusters. Applying MMs and MSAs on clusters 4 and 6 provides better accuracy (27.27% and 28.94% respectively) than applying these on the original $DS1$ (23.76%).

In contrast, there is not much difference in terms of performance of these predictive models applying to the original data set $DS1$ or the clustered data using K means HMM based. With $K = 3$, in both cases HMM based ans SA based, there is not much change in terms of the accuracy of the process next step prediction regarding to the accuracy of the same models applied on the original $DS1$.

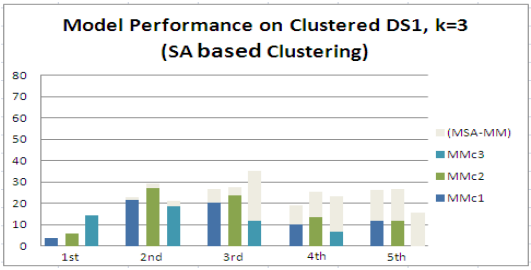

Fig. 6. Percentage of correct process next step predictions of different order MMs and MSAs using 3 clusters obtained by applying SA based clustering to dataset DS1

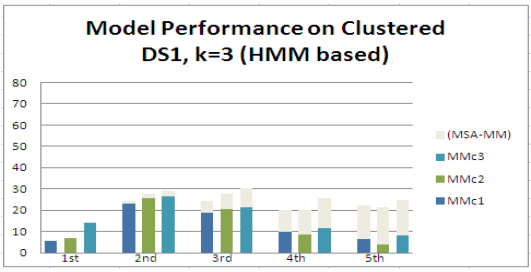

Fig. 7. Percentage of correct next process step predictions of different order MMs and MSAs using 3 clusters obtained by applying HMM based clustering to dataset DS1

The significant accuracy improvement in cluster 5 is the proof for our initial intuition that (1) if we manage to have subsets of data which consist of similar sequences then there exists a suitable model which performs well in each subset. Also, these models perform better in certain subsets than others and than the whole data set before being divided. (2) It indirectly proves that our proposed sequential clustering performs well, similar sequences are naturally clustered into clusters. (3) Our SA based clustering is more suitable for this type of data than the common HMM based clustering. In the case of data set $DS2$, there is not much improvement in terms of prediction accuracy after clustering the data with both clustering approaches. The highest performance of the fourth order MSA is about 27% applied to cluster 2 obtained by SA based and HMM based clusterings comparing to 20% to whole $DS2$. The performances of the first to fifth order MMs and MSAs applied to clusters 1 and 2 obtained by clustering $DS2$ in the case SA based clustering are illustrated in Figure 8.

When clustering $DS2$ using the two methods, only two clusters are formed, when we decrease the clustering goodness, more clusters are obtained but most of them have very low populations. The results of the experiments on clustered $DS1$ and $DS2$ show that different clusters need different predictive models. Higher order MMs and MSAs are especially good for data sequences in cluster 5

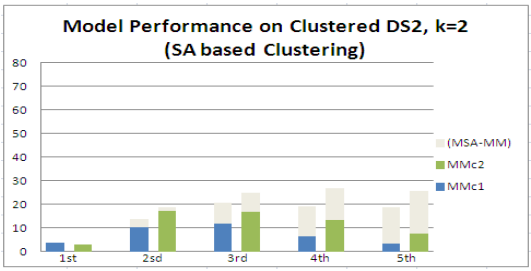

Fig. 8. Percentage of correct next process step predictions of different order MMs and MSAs using 2 clusters obtained by applying SA based clustering to data set DS2

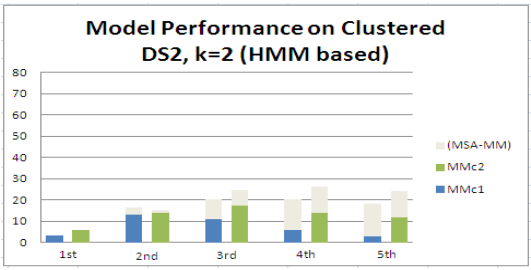

Fig. 9. Percentage of correct next process step predictions of different order MMs and MSAs using 2 clusters obtained by applying HMM based clustering to data set DS2

generated from $DS1$ using sequential K means clustering with $K = 6$. None of these models works well on other clusters from both data sets $DS1$ and $DS2$. It is sensible saying that the experimental models are not good for clustered data of $DS2$ as this data set is relatively small.

5 Conclusions

In order to deal with individual process prototypes differently, we first attempt to cluster process data into different groups of similar sequences. Such data consists of discrete symbolic sequences. After studying a number of available sequential clustering approaches, in this paper we introduce a new sequential clustering approach which is suitable for business process data. We also use the common HMM based sequential clustering in order to compare to our proposed approach. We then use predictive models to predict next process step and we significantly improve the next process step prediction in one cluster of one of the used data sets. This implies the data has been successfully clustered in a natural way and proves our strategy right.

The experimental results encourage and motivate us to continue and extend our work. The future work dicrections will explore different predictive

approaches, for example, decision trees, neural networks etc. to profit from their abilities in clustered data. We are ultimately interested in recovering the process logic even though our recover process can be the combination of a number of representative prototypes.

References

1. Ruta, D., Majeed, B.: Business process forecasting in telecommunication industry. In: 2011 IEEE GCC Conference and Exhibition (GCC), pp. 389–392 (2011)
2. Tsui, K., Chen, V., Jiang, W., Aslandogan, Y.: Data mining methods and applications. In: Pham, H. (ed.) Handbook of Engineering Statistics, pp. 651–669. Springer (2005)
3. Trcka, N., Pechenizkiy, M.: From local patterns to global models: Towards domain driven educational process mining. In: 9th International Conference on Intelligent Systems Design and Applications, pp. 1114–1119 (2009)
4. van der Aaslt, W., Weijters, A.: Process mining: Research agenda. Computers in Industry 53(3), 231–244 (2004)
5. Smyth, P.: Clustering sequences with hidden markov models. In: Advances in Neural Information Processing Systems, pp. 648–654. MIT Press (1997)
6. Garcia, D., Parrado, E., Diaz-de Maria, F.: A new distance measure for model-based sequence clustering. IEEE Transactions on Pattern Analysis and Machine Intelligent 1(7), 1325–1331 (2009)
7. Needleman, S., Wunsch, C.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. Journal of Molecular Biology 48, 443–453 (1970)
8. Waterman, M.: Estimating statistical significance of sequence alignments. Philosophical Transactions of the Royal Society of London, Series B: Biological Sciences 344, 383–390 (1994)
9. Smith, T., Waterman, M.: Identification of common molecular subsequences. Journal of Molecular Biology 147, 195–197 (1981)
10. Rajaraman, A., Ullman, J.: Mining of Massive Datasets. Cambridge University Press, Cambridge (2011)
11. Duda, R., Hart, P., Stork, D.: Pattern Classification. Wiley, New York (2001)
12. Berry, M., Linoff, G.: Data Mining Techniques: for Marketing, Sales, and Customer Relationship Management. Wiley, Newyork (2004)
13. Gabrys, B., Bargiela, A.: General fuzzy min-max neural network for clustering and classification. IEEE Transactions on Neural Networks 11(3), 769–783 (2000)
14. Anitha Elavarasi, S., Akilandeswari, J., Sathiyabhama, B.: A survey on partition clustering algorithms. International Journal of Enterprise Computing and Business Systems 1 (2011)
15. Zaki, M., Peters, M., Assent, I., Seidl, T.: Clicks: An effective algorithm for mining subspace clusters in categorical datasets. Data Knowl. Eng. 60(1), 51–70 (2007)
16. Dhillon, S., Modha, S.: Concept decompositions for large sparse text data using clustering. Machine Learning 42, 143–175 (2001)
17. Li, C., Biswas, G.: Clustering sequence data using hidden markov model representation. In: Proceedings of the SPIE 1999 Conference on Data Mining and Knowledge Discovery: Theory, pp. 14–21 (1999)
18. Porikli, F.: Clustering variable length sequences by eigenvector decomposition using hmm. In: Fred, A., Caelli, T.M., Duin, R.P.W., Campilho, A.C., de Ridder, D. (eds.) SSPR&SPR 2004. LNCS, vol. 3138, pp. 352–360. Springer, Heidelberg (2004)

19. Kanungo, T., Mount, D., Netanyahu, N., Piatko, C., Silverman, R., Wu, A.: An efficient k-means clustering algorithm: Analysis and implementation. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 881–892 (2002)
20. Wagstaff, K., Cardie, C., Rogers, S., Schrodl, S.: Constrained k-means clustering with background knowledge. In: 18th International Conference on Machine Learning, pp. 577–584 (2001)
21. Elkan, C.: Using the triangle ilequality to accelerate k-means. In: 20th International Conference on Machine Learning (ICML-2003), Washington DC, pp. 2–9 (2003)
22. Pham, D., Dimov, S., Nguyen, C.: Selection of k in k-means clustering. I MECH E Part C Journal of Mechanical Engineering Science 219(1), 103–119 (2005)