ICIC Express Letters
Volume **10**, Number **3**, March **2016**

ICIC International ©2016 ISSN 1881-803X
pp. 535–540

UTILIZATION OF SEQUENTIAL DATA FOR MACHINE LEARNING IN PROCESS CONTROL

Byeongjun Joo[1], Sunghyun Shim[2] and Hyerim Bae[1,*]

[1]Deparment of Industrial Engineering
[2]Department of Statistics
Pusan National University
Busandaehak-ro 63 Beon-gil, Geumjeong, Busan 609-735, Korea
{ jbj; ssh3608 }@pusan.ac.kr; *Corresponding author: hrbae@pusan.ac.kr

Received August 2015; accepted November 2015

Abstract. *Correct data collection and reasonably timely data processing are very important in Big Data analysis. Furthermore, interpreting the analyzed result is also an interesting issue. Although many sophisticated data mining techniques are already available, they cannot be applied directly to process mining, due to the input-data format differences. For example, whereas data mining techniques focus on the relations between attributes without considering the process, formatting data as row-based instances, process mining finds flow-patterns among instances, formatting data as column-based instances in the MXML/XES format. In the present study, we utilized a sequential dataset to enable the use of more enhanced statistical methods and to broaden the utilization of process analysis to many sophisticated data mining techniques. We experimented on artificial data to calculate the activities probability distribution using machine learning and the probability density function (PDF).*

Keywords: Sequence analysis, Machine learning, Probability density function (PDF), K-means clustering, Process analysis

1. **Introduction.** One of the major challenges facing corporate management has been process management. In order to meet the diverse needs of customers, the process must be efficient; otherwise, operational performance will suffer, eventually affecting the company in adverse ways. In the 2000s, with the advance of IT technology, Business Process Reengineering (BPR) was transformed into Business Process Management (BPM), the importance of which to process-oriented management and innovation has been emphasized by Weske [1]. Subsequently, over the succeeding years, various process management methodologies have been introduced and utilized, one of which is process mining [2].

In this Big Data era, huge amounts of data, holding information on who, when, where and what in the form mostly of log histories or transaction databases, are generated in seconds – and frequently discarded because we are unable to take advantage of it. Data analysis therefore has emerged as an important issue. Effectively accessing data using process mining techniques, process models and other means of improvement can be discovered. However, process mining encounters the well-known problem of pre-processing [3]. Circumventing it requires either adaptation of off-the-shelf data mining and statistical analysis methods or exhaustive data pre-processing, neither of which is a practical solution. All of this difficulty boils down to a basic data-input format difference: in data mining, the row-type dataset is used, whereas in process mining, a column-type dataset compliant with the *MXML/XES* format is utilized.

In this study, we got inspiration from sequential pattern mining. Sequential pattern mining is a data mining method that finds statistically related patterns in the form of sequential data by adding the element of 'time' in the function of the Association rule [4]. This means that if there are sequence rules in a continuously occurring process, they can be calculated by statistical or data mining techniques. Sequential data are employed in various analytical techniques, such as statistical distribution, machine learning and data mining, to find hidden knowledge and patterns [5]. A sequential dataset is a form of data that shows all, continuous activities for an instance (i.e., its *CaseID*) in one row. In process mining, we can discover, within a *CaseID*, the activities execution pattern (i.e., the temporal pattern) for a particular time. From this concept, we used a sequential dataset to minimize the pre-processing step while using off-the-shelf data mining techniques. To that end, we experimented on artificial data to calculate the activities probability distribution using machine learning and the probability density function (PDF).

This paper is presented as follows. Section 2 presents the problem definition. Section 3 discusses the experiment, and Section 4 draws conclusions.

2. **Problem Definition.** Although process mining uses data sources in the form of a simple flat file database, Excel spreadsheet or transaction log data, these data need to be formatted to comply with the *MXML/XES* standard before process analysis can be initiated [6]. However, in order to use such data in off-the-shelf data mining, machine learning and enhanced statistical methods, extensive data pre-processing must be performed. For example, to use statistical package R, the data has to be rearranged into a time sequence before analyzing it. We used artificial data consisting of 2,000 *CaseIDs*, 200,000 *Activities* and time values within the $T_1 - T_{100}$ range. A *CaseID* is composed of 7 activities denoted as Act_i, where $i = \{1, 2, \ldots, 7\}$, Act_1 and Act_7 being assigned as the first and last activities, respectively, and the remaining activities being assigned randomly. We utilized three event types: (1) *Ready*, signifying the activity preparation time before it begins; (2) *End*, signifying the end time, and (3) *Idle time*, signifying the idle time during the execution of the activity.

Table 1. Transaction log dataset

CaseID	*Activity*	*Start Time*	*End Time*
1	*Ready*	T_1	T_2
1	Act_1	T_3	T_4
⋮	⋮	⋮	⋮
1	Act_i	T_n	T_n
2	*Ready*	T_1	T_1
⋮	⋮	⋮	⋮

Table 1 represents the column-type dataset usually used in process mining; Table 2 shows the row-type (sequential) data reconstructed from Table 1 data. Using Table 2 data, we seek to find which activity occurs at a specific point in time (T_n).

Table 2. Sequential dataset

CaseID	T_1	T_2	T_3	T_4	⋯	T_n
1	*Ready*	*Ready*	Act_1	Act_1	⋯	Act_i
2	*Ready*	Act_1	Act_1	Act_3	⋯	Act_j
⋮	⋮	⋮	⋮	⋮	⋮	⋮

According to Aggrawal and Srikant [7], the classification and prediction of an activity that occurs at a specific point in time can be used to accomplish the following: calculate the employee/equipment/process/customer utilization at that time; analyze the time pattern that probably will recur in executing a certain set of activities; determine and predict a process occurrence through machine learning or statistical analysis; analyze activities from other perspectives using multi-dimensional visualization; analyze real-time data quickly and accurately.

3. **Experiment.** As explained in this section, we used the sequential data in Table 2 to calculate, via the PDF, the distribution of activities at a particular point in time. Additionally, we employed classification and prediction using machine learning methods. For classification, we used unsupervised learning with the Support Vector Machine (SVM) algorithm, while for prediction, we used supervised learning with Artificial Neural Networks (ANN). We also used the statistical package tool in R. Finally, we modeled the clustered process model by K-means clustering.

For the SVM and ANN algorithms, we assigned 1,400 cases as training data and 600 cases as validation data.

3.1. **Probability density function (PDF).** We utilized the PDF to check the distribution between Act_2, Act_5, and *Idle time*, which is plotted in Figure 1.

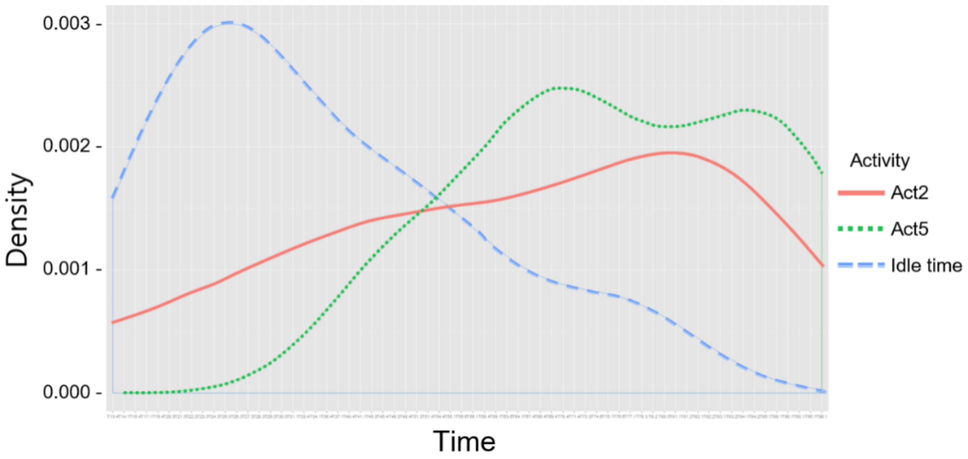

FIGURE 1. Probability density function (PDF)

In the results, we can see that some activities are more likely to execute at the beginning of the process, while others are more likely to execute at the end of the process. This fact confirms the possibility of conducting time-series analysis with this type of data.

3.2. **Using SVM for classification of activities.** Using the SVM algorithm [8], we undertook to classify activity that had occurred at time T_{80} according to the parameter-settings syntax shown in the following Figure 2.

```
train.svm < −nn[1:1400,]
test.svm < −nn[1401:2000,]
svm < −ksvm(T80∼., data = train.svm, kernel = “vanilladot”)
svm.predict < −round(predict(svm,test.svm,type = “response”))
```

FIGURE 2. SVM algorithms in R

Table 3 provides the experimental results. The first row shows the classified activity at time T_{80}, while the first column shows the actual activity occurrence at time T_{80}. Each cell shows the number of cases having a certain activity as both a classified and an actual result. For example, there were 51 cases having Act_2 as its classified and actual value (i.e., 51 cases of correct classification). From the cells, we calculated that 99.5% of the

Table 3. SVM results

T_{80}	**Act_2**	**Act_5**	**Act_6**	**Act_8**	***Idle time***
Act_2	**51**	0	0	0	0
Act_4	0	2	0	0	0
Act_5	0	**174**	0	0	0
Act_6	0	0	**181**	0	0
Act_7	0	0	0	1	0
Act_8	0	0	0	**176**	0
Idle time	0	0	0	0	**15**

actual activities occurring at time T_{80} were matched with the classification result (i.e., 0.5% were mismatched on classification).

3.3. **Using ANN for activity prediction.** We used ANN [9] to predict the activity at time T_{25} according to the parameter-settings syntax shown in Figure 3.

```
train < −sample(1:2000,1400)
test < −setdiff(1:2000,train)
ideal < −class.ind(nn$T25)
NN = nnet(nn[train,−101],ideal[train,],size = 9,softmax = TRUE)
predict(NN, nn[train,−101], type = “class”)
```

Figure 3. ANN algorithms in R

Table 4 provides the experiment results. The first row shows the predicted activity at time T_{25}, and the first column shows the actual activity occurrence at time T_{25}. As indicated, there were 595 cases where the actual and predicted values were consistent (99.3%), while 5 cases were incorrectly predicted (0.7%).

Table 4. ANN results

T_{25}	Act_3	Act_4	*Idle time*
Act_3	**394**	3	1
Act_4	0	**177**	0
Idle time	0	1	**24**

Both the SVM and ANN results show a high precision value (i.e., more than 90%). This is due to the simple example that was used throughout the experiment. And in this case, the number of *CaseID* is much larger than *Activity*. So the performance of the machine learning showed good results. If more complex data were used or the difference between the number of *CaseID* and *Activity* is little bit, the precision value might have been lower. This means that it is necessary to improve the machine learning performance for more accurate results.

3.4. **Process clustering through K-means.** Lee and Bae [10] asserted that a more meaningful and accurate model can be discovered by process model clustering. In the present study, therefore, we used K-means algorithms to cluster our sequential data according to the syntax shown in Figure 4.

In the results shown in Figure 5, the 4 clusters contain 790, 190, 420, and 600 cases, respectively. The relationship between activities and *Idle time* is confirmed. In <Cluster 1> and <Cluster 2>, *Idle time* is generated mainly before Act_3 and after Act_4. In the case of <Cluster 3>, *Idle time* is incurred after Act_4, Act_5 and before Act_6. Whereas in <Cluster 4>, Act_1 is an activity that starts immediately without going through the

```
cluster < −as.data.frame(lapply(example,scale))
cluster[is.na(cluster)] < −0
km < −kmeans(cluster, centers = 4, nstart = 10)
```

Figure 4. K-means algorithms in R

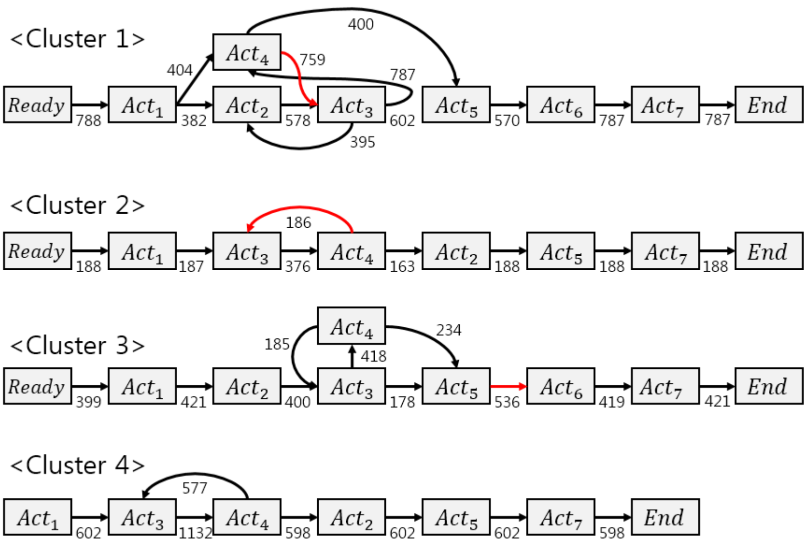

Figure 5. Clustering process modeling

Ready state, it can be known that <Cluster 4> is the simplest model. Since the data is sequentially ordered based on time, we discovered the interesting fact that the clusters were formed based on time standards as well.

4. **Conclusions.**

This paper addresses the utilization of a sequential dataset in machine learning along with statistical analysis for column-type data normally used in process mining. Previously, the analyses that could be run using column-type data (e.g., transactional data) were limited due to the incompatible input-data format. Our experimentation confirmed the meaningfulness of sequential dataset analysis. Thus, with a sequential dataset, efficient use of off-the-shelf data mining, machine learning and enhanced statistical analysis tools is now possible.

A limitation of this study is its analysis only of linear relations in a sequential dataset. For future work, we will consider the inclusion of non-linear relations among instances in sequential data. Additionally, we would like to extend the present analysis to real-time sequential process analysis of actual data.

Acknowledgment. This research was supported by a National Research Foundation of Korea (NRF) grant funded by the Korean government (MEST) (No. NRF-2015R1D1A1A 09061331).

REFERENCES

[1] M. Weske, *Business Process Management: Concepts, Languages, Architectures*, Springer, Berlin, 2007.

[2] I. H. Witten, E. Frank and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques,* Elsevier, 2011.

[3] D. Pyle, *Data Preparation for Data Mining*, Morgan Kaufmann, 1999.

[4] C. H. Mooney and J. F. Roddick, Sequential pattern mining: Approaches and algorithms, *ACM Computing Surveys*, vol.45, no.19, 2013.

[5] N. R. Mabroukeh and C. I. Ezeife, A taxonomy of sequential pattern mining algorithms, *ACM Computing Surveys*, vol.43, no.3, 2010.

[6] W. M. P. van der Aslst, *Process Mining: Discovery, Conformance and Enhancement of Business Process*, Springer, Berlin, 2011.

[7] R. Aggrawal and R. Srikant, Mining sequential patterns, *Proc. of the 11th International Conference on Data Engineering*, pp.3-14, 1995.

[8] A. Karatzoglou, A. Smola and K. Hornik, *Kernel-Based Machine Learning Lab*, 2015.

[9] B. Ripley and W. Venables, *Feed-Forward Neural Networks and Multinomial Log-Linear Models*, 2015.

[10] D. Lee and H. Bae, Analysis framework using process mining for block movement process in shipyards, *ICIC Express Letters*, vol.7, no.6, pp.1913-1917, 2013.