Evidence Theory Based Combination of Frequent Chronicles for Failure Prediction

Achraf Ben Chrayet[1], Ahmed Samet[2(✉)], and Mohamed Anis Bach Tobji[1]

[1] Univ. Manouba, ESEN, Manouba, Tunisia
{achraf.chrayet,mohamed.tobji}@esen.tn
[2] ICUBE/SDC Team (UMR CNRS 7357)-INSA Strasbourg, Strasbourg, France
ahmed.samet@insa-strasbourg.fr

Abstract. A chronicle is a kind of temporal pattern mined from a set of sequences made-up of time-stamped events. It has been shown recently that such knowledge is effective in sketching machines' behaviours in industry. However, chronicles that describe a same new sequence of events could be multiple and conflictual. To predict nature and time interval of future events, we need to consider all the chronicles that match a new sequence. In this paper, we introduce a new approach, called *FCP*, that uses the evidence theory and chronicle mining to classify sequences. The approach has been evaluated on both synthetic and real-world data sets and compared to baseline state-of-the-art approaches.

Keywords: Chronicle mining · Prediction maintenance · Evidence theory

1 Introduction

In industry 4.0, predictive maintenance relies on analysing sequential data containing time-stamped events. Therefore, data mining and particularly pattern mining techniques [1] turned to be very effective to understand failure sequences [9] by finding recurrent abnormal behaviours before any prediction task.

One type of pattern stands out thanks to its information richness and it is called *chronicle*. A chronicle is a pattern that represents a sequence of events that happened enough frequently to be extracted. Introduced in [6], this new kind of sequences is enriched with the time interval that separates each pair of events, making it possible to predict that an event B will probably happen at a time interval $[t1, t2]$ if event A occurs. If the event B requires an intervention, such as a machine failure, then maintenance may be performed on time avoiding cascading troubles.

Chronicles are complex but highly expressive patterns that enable to take into account the quantitative temporal dimension of the data contrary to classical sequential patterns. Dousson et al. [5] introduced what is called later chronicle

J. Davis and K. Tabia (Eds.): SUM 2020, LNAI 12322, pp. 226–233, 2020.
https://doi.org/10.1007/978-3-030-58449-8_16

mining. They proposed an incomplete algorithm (which does not generate all the patterns) called FACE (Frequency Analyzer for Chronicle Extraction). Then, Cram et al. [3] introduced another complete algorithm to mine the complete set of chronicles. Sellami et al. have introduced a new approach called FADE [9] that mines failure chronicles (chronicles that end with failure event).

In this paper, we tackle the problem of sequences' classification and failure time prediction in the context of predictive maintenance. We aim to understand and predict failures for a target maintenance. Once the failure is predicted, the maintenance is scheduled using failure criticality assessment [2]. Chronicle mining algorithm is used to extract knowledge from the data set: normal and abnormal behaviour patterns. Assuming a set of chronicles representing different machine behaviours with certain level of reliability, a major task is how to classify new incoming sequence.

Therefore, we propose to combine the use of evidence theory and chronicle mining to classify sequence in the context of predictive maintenance. The evidence theory, is a strong mathematical framework that allows to model uncertain knowledge and combine information for decision making. To summarize, this paper introduces two contributions: *(i)* using both normal and failure chronicles for sequence classification and time to failure prediction and finally *(ii)*, a new algorithm called FCP that uses the mined chronicles and evidence theory framework to combine information and predict if a new sequence will lead to a machine failure, and if yes, in which time interval the crash will occur.

2 Background

2.1 Evidence Theory

The evidence theory also called the belief function theory was introduced by Dempster [4]. In this section, we present the main concepts of this theory. The frame of discernment is the set of N possible answers for a treated problem and generally denoted θ. It is composed of exhaustive and exclusive hypotheses: $\theta = (H_1, H_2, \ldots, H_N)$.

These elements are assumed to be mutually exclusive and exhaustive. From the frame of discernment θ, we deduce the set 2^θ containing all the 2^N subsets A of θ: $2^\theta = \{A, A \subseteq \theta\} = \{H_1, H_2, \ldots, H_N, H_1 \cup H_2, ..., \theta\}$. A Basic Belief Assignment (BBA) m is the mapping from elements of the power set 2^θ onto [0, 1], having as constraints:

$$\begin{cases} \sum_{A \subseteq \theta} m(A) = 1 \\ m(\emptyset) = 0. \end{cases} \tag{1}$$

The belief function offers many advantages. One of its proposed asset is the information fusion allowing extracting the more veracious proposition from a multi-source context. This benefit is granted by the Dempster rule of combination [4] defined as follows:

$$m_{\oplus}(A) = m_1 \oplus m_2(A) = \frac{1}{1 - \sum_{B \cap C = \emptyset} m_1(B) * m_2(C)} \sum_{B \cap C = A} m_1(B) * m_2(C); \forall A \subseteq \theta, A \neq \emptyset \tag{2}$$

The pignistic transformation allows the decision from a BBA by distributing equiprobably the mass of a proposition A on its sub-hypotheses, formally:

$$BetP(H_n) = \sum_{A \subseteq \theta} \frac{|H_n \cap A|}{|A|} * m(A); \forall H_n \in \theta \tag{3}$$

2.2 Chronicle Mining

To give formal definition of chronicles, this section starts by introducing the concept of event [3].

Definition 1 (Event). *Let $\mathbb{E}$ be a set of event types, and $\mathbb{T}$ a time domain such that $\mathbb{T} \subseteq \mathbb{R}$. $\mathbb{E}$ is assumed totally ordered and is denoted $\leq_{\mathbb{E}}$. According to [3], an event is a couple (e, t) where $e \in \mathbb{E}$ is the type of the event and $t \in \mathbb{T}$ is its time.*

Definition 2 (Sequence). *Let $\mathbb{E}$ be a set of event types, and $\mathbb{T}$ a time domain such that $\mathbb{T} \subseteq \mathbb{R}$. $\mathbb{E}$ is assumed totally ordered and is denoted $\leq_{\mathbb{E}}$. According to the definition in [3], a sequence is a couple $\langle SID, \langle (e_1, t_1), (e_2, t_2), ..., (e_n, t_n) \rangle \rangle$ such that $\langle (e_1, t_1), (e_2, t_2), ..., (e_n, t_n) \rangle$ is a sequence of events, and SID its identifier. For all $i, j \in [1, n], i < j \Rightarrow t_i \leq t_j$. If $t_i = t_j$ then $e_i <_{\mathbb{E}} e_j$ where $<_{\mathbb{E}}$ is the lexical order.*

When the events are time-stamped, how to describe the quantitative time intervals among different events is very important for the prediction of possible future events. To achieve this goal, the notion *temporal constraints* is introduced.

Definition 3 (Temporal constraint). *A temporal constraint is a quadruplet (e_1, e_2, t^-, t^+), denoted $e_1[t^-, t^+]e_2$, where $e_1, e_2 \in \mathbb{E}$, $e_1 \leq_{\mathbb{E}} e_2$ and $t^-, t^+ \in \mathbb{T}$.*

t^- and t^+ are two integers which are called lower and upper bounds of the time interval, such that $t^- \leq t^+$. A couple of events (e_1, t_1) and (e_2, t_2) are said to satisfy the temporal constraint $e_1[t^-, t^+]e_2$ iff $t_2 - t_1 \in [t^-, t^+]$. It is defined that $e_1[a, b]e_2 \subseteq e_1'[a', b']e_2'$ iff $[a, b] \subseteq [a', b']$, $e_1 = e_1'$, and $e_2 = e_2'$. The concept of chronicles [3] is defined as follows.

Definition 4 (Chronicle). *A chronicle is a pair $C = (\mathcal{E}, \mathcal{T})$ such that:*

1. *$\mathcal{E} = \{e_1 ... e_n\}$, where $\forall i, e_i \in \mathcal{E}$ and $e_i \leq_{\mathbb{E}} e_{i+1}$,*
2. *$\mathcal{T} = \{t_{ij}\}_{1 \leq i < j \leq |\mathcal{E}|}$ is a set of temporal constraints on $\mathcal{E}$ such that for all pairs (i, j) satisfying $i < j$, t_{ij} is denoted by $e_i[t_{ij}^-, t_{ij}^+]e_j$.*

Definition 5 (Chronicle support). *An occurrence of a chronicle C in a sequence S is a set $(e_1, t_1) \dots (e_n, t_n)$ of events of the sequence S that satisfies all temporal constraints defined in C. The support of a chronicle C, denoted $Supp(.)$ in the sequence S is the number of its occurrences in a data set of sequences [9]. In this paper, we assume that a sequence could contain at most only one occurrence of any chronicle.*

3 Chronicle Mining and Evidence Theory for Failure Prediction

In this section, we define the notions we use in our approach to combine chronicles for prediction.

Definition 6 (Chronicle cover). *Assuming a sequence $S = \langle (e_1, t_1), (e_2, t_2), \dots, (e_n, t_n) \rangle$ and a frequent chronicle C. We say that C covers the sequence S, denoted by $C <\cdot S$, if and only if the events represented by the chronicle belong to the sequence as well as the time intervals between these events in the sequence belong to the temporal constraints extracted by the chronicle, i.e.,*

$$C <\cdot S \Leftrightarrow \forall e_i[t^-, t^+]e_j \in C, \exists ((e,t),(e',t')) \in S \wedge e = e_i, e' = e_j \wedge t' - t \in [t^-, t^+]. \quad (4)$$

Let $\mathscr{C}$ be a set of frequent chronicles, $\mathrm{C}_T \subset \mathscr{C}$, such that $T \in \{F, N\}$ and where $\bar{F} = N$. C_F denotes the set of chronicles that point to the failure event, where C_N is the set of chronicles that do not, and so match normal sequences.

Definition 7 (BBA modeling). *Assuming a chronicle $C_i \in C_T$ that covers a sequence S, we model the BBA m_i of C_i in $\theta = \{T, \bar{T}\}$ as follows:*

$$\begin{cases} m_i(T) = Supp(C_i) \\ m_i(\bar{T}) = 0 \\ m_i(\theta) = 1 - Supp(C_i) \end{cases} \quad (5)$$

Definition 8 (Chronicles combination). *Assuming N chronicles C_i that cover a sequence S, with m_i, $i \in [1, N]$, the mass function relative to the i^{th} chronicle. The joint mass function that combines all the m_i mass functions of the chronicles C_i that cover S using the Dempster Rule of combination is defined as follows:*

$$m_{\oplus}(A) = m_1 \oplus \dots \oplus m_N(A); \forall A \subseteq \theta \quad (6)$$

To make the decision, we compute the pignistic probability $BetP$ for failure (F) and normal (N). The final decision is obtained by retaining the hypothesis that maximized the pignistic probability as follows:

$$x = argmaxBetP_{x_i \in \theta}(x_i). \quad (7)$$

For the prediction task, we developed the FCP method (Fusion of Chronicles for Prediction). It consists in comparing the input sequence (to predict) with every chronicle in terms of events and time constraints. To each matching chronicle, we model a BBA that measures to which degree the chronicle expresses the failure (F) and normal (N) behaviour classes. The level of uncertainty is retained using the support of the chronicle. Once all matching chronicles are modelled, we use the Dempster rule of combination to combine all the BBAs. The joint BBA shows the membership of the input sequence to both classes. The final class is computed using the argmax function. If the final class is failure, we display the failure time by aggregating the time constraints of all matching failure chronicles. Algorithm 1 performs the combination of the covering chronicles to predict the status of a sequence using all aforementioned notions.

Algorithm 1 Fusion of Chronicle for Prediction

Require: S: sequence, $\mathcal{C}$: chronicles set
Ensure: R: result, $min\_time\_failure$: minimum time to failure, $max\_time\_failure$: maximum time to failure

$C_M \leftarrow \{\}$
for all $C \in \mathcal{C}$ **do**
    if (coverage(S,C)) **then**
        $C_M \leftarrow C_M \cup C$
for all $C \in C_M$ **do**
    if $(C.Type == F)$ **then**
        $m \begin{cases} m(F) = Supp(C) \\ m(N) = 0 \\ m(\theta) = 1 - m(F) \end{cases}$
    else
        $m \begin{cases} m(F) = 0 \\ m(N) = Supp(C) \\ m(\theta) = 1 - m(N) \end{cases}$
    $m_\oplus \leftarrow m_\oplus \oplus m$
$R \leftarrow argmax_{x_i \in \theta} BetP(x_i)$
if $(R == F)$ **then**
    Init(C$_M$,$min\_time\_failure$,$max\_time\_failure$)
    ▷ Initialize $min\_time\_failure$ and $max\_time\_failure$
    for all $C \in$C$_M$ **do**
        if ($min\_time\_failure > C.min\_time$) **then**
            $min\_time\_failure \leftarrow C.min\_time$
        if ($max\_time\_failure < C.max\_time$) **then**
            $max\_time\_failure \leftarrow C.max\_time$
    return $R, min\_time\_failure, max\_time\_failure$
else
    return R

4 Experiments and Results

Two kinds of data sets are used to validate our approach. The first one is generated synthetically according to several parameters, such as the number of sequences, the mean size (i.e. width) of a sequence and the number of items (events).[1] In addition, data are generated following a failure model sequence that represents 5% of the entire produced data set. Even such kind of data sets do not include natural patterns of failure/normal events, they are interesting in the way they allow the evaluation of our approach when we vary the data features, which is infeasible with real data sets whose parameters are fixed.

The second experiment is made on an industrial real data set, denoted SECOM (semi-conductor manufacturing process), introduced in [8]. It's a data set that records 1567 measurements of 590 sensors installed in manufacturing

[1] Reader may refer to https://gitlab.inria.fr/tguyet/pychronicles for further details about data sets generation.

machines. Each record has a timestamp (the instant at which the 590 measurements are taken), and also a general state; 1 for a normal state, and −1 for a failure.

4.1 The Performance Evaluation

The performance of our approach is evaluated on different synthetic data sets to assess the effect of several parameters mainly on the run-time and the memory usage. Figure 1 shows the execution time of FCP according to the number of sequences and the vocabulary size. The execution time increases when the number of sequences and their sizes increase. Indeed, when number and size of sequences are large, the number of extracted frequent chronicles increases accordingly. The Dempster rule of combination is the most consuming part of our approach. The more we find matching chronicles, the more we model and combine BBAs.

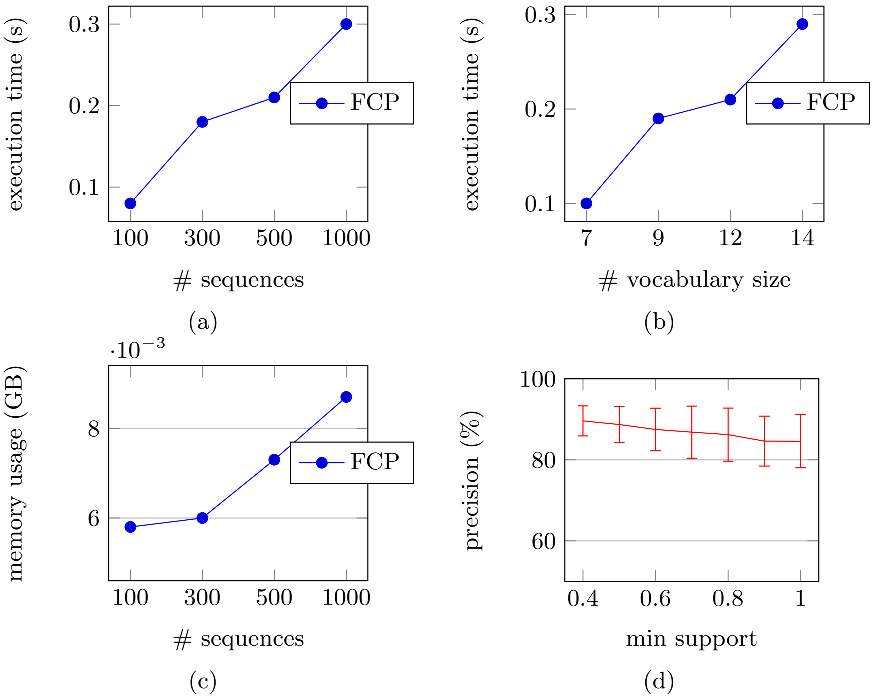

Fig. 1. FCP experiments on synthetic data sets

As part of the performance evaluation, we also assessed the memory consumption of both algorithms. Figure 1 shows the amount of memory used according to the sequence number variation. For FCP, the use of memory increases when the number of covering chronicles increases, especially because of the operation of combination that uses matrix structures within the evidence theory for mass functions.

4.2 The Prediction Quality Experiments

To evaluate the prediction quality of our approach, we used the 10-fold-cross-validation method [10] to compute the precision, the recall and the F-measure. A failure sequence is considered correctly classified, if we predict the failure state and also the time interval into which the breakdown will occur. A normal sequence is correctly classified if we predict the normal state.

First, we evaluate the prediction quality of our approach *FCP* on different synthetic data sets. We carried out experiments to assess the precision of *FCP* by varying the minimum support, denoted *minsup*, of the mining algorithm [9]. The Fig. 1 (d) pictures the results. It shows that the precision decreases as long as *minsup* increases. In fact, precision and *minsup* are both linked. Indeed, the more we increase *minsup*, less chronicles we mine. Then, unfortunately, several sequences could not be covered by any chronicle. We note that best prediction results are observed when *minsup* is set to 0.4.

The prediction approaches that use chronicles for prediction are limited. In this paper, we compare our approach to *FADE* [9]. The latter consists in mining frequent chronicles. Then, it uses the highest support matching chronicle to predict. As for *FCP*, *FADE* classifies the sequence and predicts when it is going to happen using time constraint of the chronicle failure event. For these reasons *FADE* is a natural comparative reference to *FCP*. In addition, we adapted the k-NN algorithm introduced in [7] for evaluation. In our adapted version, we choose the k most similar chronicles to our sequence among the chronicles that cover it. So we do not consider all chronicles, just the k nearest chronicles that correspond to the top 30% of the covering chronicles. Second, we combine the obtained classes using the weighted majority vote method, so the weight of a class is proportional to the distance between the sequence and the chronicles that represent the class in question. Knowing that, our approach uses mined patterns, classifies sequences and predict time to failure. We also compare FCP to other neural network based classification approaches [11]. Table 1 shows the results in terms of recall, precision and F-measure on the SECOM data set.

Table 1. Quality of prediction on the SECOM data set

Approach	Parameters	Recall	Precision	F-measure
FCP	minsup = 0.4	**0.78**	**0.81**	**0.79**
FADE [9]	minsup = 0.4	0.72	0.70	0.71
k-NN [7]	k equivalent to 30% of chronicles	0.69	0.71	0.69
LSTM [11]	1 shared layer; 2 prediction layers	0.73	0.74	0.73

5 Conclusion

In this paper, we are interested in prediction of failures as well as their time of occurrence, in the context of predictive maintenance of industrial machines. To resolve this problem, we rely on frequent chronicle mining, which allows not only the extraction of patterns, but also the time constraint between events for each sequence in the data set. We used evidence theory to combine chronicles. Experiments show that our FCP approach is more effective than existing methods. As future work, we intend to work on improving prediction of the occurrence time. As current works predict a large time interval, we intend to be more precise by predicting the most probable instant of occurrence.

Acknowledgements. This work has received funding from INTERREG Upper Rhine (European Regional Development Fund) and the Ministries for Research of Baden-Wrttemberg, Rheinland-Pfalz (Germany) and from the Grand Est French Region in the framework of the Science Offensive Upper Rhine HALFBACK project.

References

1. Borgelt, C.: Frequent itemset mining. In: Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery, pp. 35–50, November 2012
2. Cao, Q., Samet, A., Zanni-Merk, C., de Bertrand de Beuvron, F., Reich, C.: Combining evidential clustering and ontology reasoning for failure prediction in predictive maintenance. In: Proceedings of the 12th International Conference on Agents and Artificial Intelligence, ICAART 2020, Valletta, Malta (to appear, 2020)
3. Cram, D., Mathern, B., Mille, A.: A complete chronicle discovery approach: application to activity analysis. Exp. Syst., 321–346, May 2011
4. Dempster, A.: Upper and lower probabilities induced by multivalued mapping. AMS **38** (1967)
5. Dousson, C., Duong, T.V.: Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems. In: Proceedings of the International Joint Conference on Artificial Intelligence, San Francisco, CA, USA, July 31–August 06, pp. 620–626 (1999)
6. Dousson, C., Gaborit, P., Ghallab, M.: Situation recognition: Representation and algorithms. In: Proceedings of the International Joint Conference on Artificial Intelligence, Chambéry, France, August 28–September 3, pp. 166–174 (1993)
7. Fan, Y., Ye, Y., Chen, L.: Malicious sequential pattern mining for automatic malware detection. Exp. Syst. Appl. **52**, 16–25 (2016)
8. McCann, M., Li, Y., Maguire, L., Johnston, A.: Causality challenge: Benchmarking relevant signal components for effective monitoring and process control. In: Proceedings of the International Conference on Causality: Objectives and Assessment, Whistler, Canada, pp. 277–288 (2008)
9. Sellami, C., Miranda, C., Samet, A., Bach Tobji, M.A., de Beuvron, F.: On mining frequent chronicles for machine failure prediction. J. Intell. Manuf., 1 (2019)
10. Stone, M.: Cross-validatory choice and assessment of statistical predictions. J. Roy. Stat. Soc. Ser. B Methodol., 111–147 (1974)
11. Tax, N., Verenich, I., La Rosa, M., Dumas, M.: Predictive business process monitoring with LSTM neural networks. In: Dubois, E., Pohl, K. (eds.) CAiSE 2017. LNCS, vol. 10253, pp. 477–492. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59536-8_30