Actas de XIV Jornadas de ARCA
Sistemas Cualitativos y sus Aplicaciones en
Diagnosis, Robótica e Inteligencia Ambiental
F. J. Ruiz, N. Agell, J. A. Ortega (Eds.)

An extended chronicle discovery approach to find temporal patterns between sequences

Alvarez, M.A.[a]; Subias, A.[b,c]; Travé-Massuyès, L.[b,c]; Gonzalez-Abril, L.[d]; Ortega, J.A.[a]
[a]Department of Computer Science, University of Seville, 41500 Seville, Spain
[b]CNRS, LAAS, 7, avenue du Colonel Roche, F-31400 Toulouse, France
[c]Univ de Toulouse, INSA, LAAS, F-31400 Toulouse, France
[d]Department of Applied Economics I, University of Seville, 41500 Seville, Spain
maalvarez@us.es, {subias, louise}@laas.fr, {luisgon, jortega}@us.es

Abstract

Sequences of events describing the behavior and actions of users or systems can be collected in several domains. An episode is a collection of events that occurs relatively close to each other in a given partial order. Also, chronicles are a special type of temporal patterns, where temporal orders of events are quantified with numerical bounds and reflect the temporal evolution of the system over the time. In this paper, the problem of finding rules for describing or predicting the behavior of the sequences with the intention of characterizing some interesting tasks is considered. Obtaining these patterns is the main objective of this work, where an automatic method to learn relevant and discriminating chronicles is proposed. The method extends existing algorithms that have been proposed to find frequent episodes/chronicles in a single event sequence to the case of multiple sequences.

1 Introduction

In some application areas of knowledge like data mining or machine learning, the data to be analyzed is made up of a sequence of events. So, the data can be viewed as a sequence of events, where each event has an associated time of occurrence. An example of an event sequence is shown in Figure 1. Here A to F are events and they are represented on a time line. In the last years, there have been many authors interest in knowledge discovery from sequential data [Dousson *et al.*, 2008; Le Guillou *et al.*, 2008; Pencolé and Subias, 2009; Bertrand *et al.*, 2009; Saddem *et al.*, 2010; Bauer *et al.*, 2011] because the technology have been applied in a lot of areas.

Analysing human activities is required in many domains, like ergonomics, safety diagnosis, process design, and more generally for understanding cognitive and social processes. In this article, we propose an approach to support the process of activity analysis with the help of interactive discovery of temporal patterns named chronicles.

The first task for describing the behavior of systems from sequences of events is to find frequent episodes, i.e., collections of events occurring frequently together. In the Figure 1, the event E is followed by F several times and it is an episode, and ordered set of events. From the same sequence in the figure, the observation that whenever A and B occur, in either order, C occurs soon can be done.

Taking into account the last definition, a set of maximum episode rules can be obtained from a event sequence. The main motivation of this paper is to find a minimal set of rules from some event sequences. This set must contain the maximum episode rules that have been found in all sequences, i.e. the set is an intersection of the set of each one.

In this paper the following problem is considered. Given some input sequences of events, find all episodes that occur frequently in all sequences. To achieve this goal some extended techniques from [Mannila *et al.*, 1997], [Mannila and Ronkainen, 1997] and [Cram *et al.*, 2011] are proposed.

The rest of this paper is organized as follows: first, the main problem and the motivation of this paper are presented in Section 2. Later, the definition of the problem is presented in Section 3 to establish the notation of the rest of the paper. In Section 4, the existing algorithms to discover chronicles in a sequence are explained. Section 5 contains the Mannila's approach to get chronicles and Section 6 presents a methodology to discover chronicles that must be exist in all event sequences. Section 7 reports the obtained results of applying the methodology. The paper is finally concluded with a summary of the most important points in Section 8.

2 Problem and motivation

This work aims to reach an estimation of the state of a network from chronicle recognition. The chronicles describe behaviors or situations to be recognized from temporal patterns.

The main difficulty of this approach focuses on how to develop chronicles. One of the possible solutions can be obtained from learning, so in this paper, one of the principal chronicle learning approaches existing in the literature is studied. From this approach, a solution to the problem of the adaptation of the communication protocols is proposed. These problems can be network congestions or packet loss. These patterns are based on generated signals from routers.

3 Definitions

An input as a sequence of events is considered where each event has a tag and a time stamp represented by an integer. Given a set E of event types, an event e is a pair (A, t) where

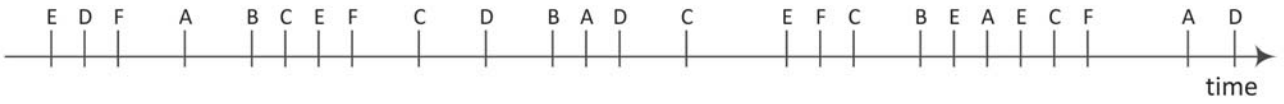

Figure 1: A sequence of events

$A \in E$ is an event type and t is the time of the event.

$$e = (A, t \mid A \in E)$$

A sequence s is a list of events between an interval of time stamps

$$s(T_s, T_e) = \langle (A_1, t_1), (A_2, t_2), \ldots, (A_n, t_n) \rangle \quad (1)$$

where

$$A_i \in E, \forall i = 1, 2, \ldots, n$$
$$t_i \leq t_{i+1}, \forall i = 1, 2, \ldots, n$$
$$T_s \leq t_i < T_e, \forall i = 1, 2, \ldots, n$$

A event sequence $s(29, 68)$ is represented in Figure 1.

$$s(29, 68) = \langle (E, 31), (D, 32), (F, 33), (A, 35), (B, 37), (C, 38), \ldots, (D, 67) \rangle$$

A window ω on a sequence is defined as a set of events between an interval of time stamps

$$\omega(t_s, t_e) = \{(A, t) \in s \mid t_s \leq t < t_e\} \quad (2)$$

where $t_s < T_e$ and $t_e > T_s$.

Also, two windows ω_a and ω_b are similar if $|\omega_a| = |\omega_a|$ and $|(A_i, t_i)| = |(A_j, t_j)|$ where $A_i = A_j, \forall A_i \in \omega_a, \forall A_j \in \omega_b$.

A width can be defined as the difference between two time stamps. In Figure 1, the width of the sequence is the difference between the start time stamp and the end time stamp:

$$width(s) = T_e - T_s = 68 - 29 = 39$$

Also, the width of a window ω can be defined as $width(\omega) = t_e - t_s$.

Now, from 1 and 2, the set of all windows with the same width in a sequence is represented below:

$$\mathcal{W}(s, win) = \{\omega(t_s, t_e) \in s \mid width(\omega) = win\}$$

For example, in Figure 1, the number of windows with the width equals to 5 is 43, $|\mathcal{W}(s, 5)| = 43$, and $(\emptyset, 25, 30)$ and $(\langle (D, 67) \rangle, 25, 30)$ are the first and the last windows of $\mathcal{W}(s, 5)$.

Finally, an episode α is a set of nodes with a partial order and a relation between them:

$$\alpha = (V, \leq, g : V \to E)$$

Also, an episode β is a subepisode of α if the following conditions are fullfiled:

$$\beta = (V', \leq', g' : V' \to E) \preccurlyeq \alpha = (V, \leq, g : V \to E) \quad (3)$$
$$\exists f : V' \to V \mid g'(\nu) = g(f(\nu)), \forall \nu \in V'$$
$$\nu \leq' \varpi, \forall \nu, \varpi \in V'$$
$$f(\nu) \leq f(\varpi), \forall \nu, \varpi \in V'$$

In the other hand, an episode α is a superepisode of β, if and only if $\beta \preccurlyeq \alpha$.

From 3, an episode rule is defined as $\beta \Rightarrow \alpha$ if $\beta \preccurlyeq \alpha$.

In Figure 2, α, β and γ are episodes and γ is a subepisode of β.

Figure 2: Three different episodes

4 State of the art on discovering chronicles

[Dousson and Ghallab, 1994] defines chronicle as a temporal pattern intended to represent a pattern of evolution of a system, it is mainly composed of a set of events and temporal constraints between their dates of occurrences. A chronicle C is presented as $C = S, T$ where S represents all the events and T represents all the constraints.

In Figure 3, the e_3 event occurs between 3 and 6 time units after the e_1 event or between 3 and 9 time units after the e_2 event. This chronicle is represented below:

$$C = \{S, T\}$$
$$S = \{e_1, e_2, e_3\}$$
$$T = \{t_1 < t_3 \wedge t_2 < t_3 \mid 3 \leq t_3 - t_1 \leq 6 \wedge 3 \leq t_3 - t_2 \leq 9\}$$

Some considerations are needed to taking into account to recognize a chronicle:

i A set of dated events: this set is the input of the recognition system.

ii A set of constraints: these constraints are time intervals between the dates of occurrences of events.

iii A set of formulas or a general scheme: it is the chronic patterns without considering timing constraints between their dates of occurrence.

Recognizing is to explain the input events (i) using the temporal patterns (iii) while respecting the constraints of the field (ii). So, a chronicle is a set of constraints on events with their dates of occurrence, where each characteristic of a chronicle situation is based on the observed consequences.

In our context, an abnormal communication system is characterized by a set of chronicles, where each chronicle describes the situation of interest according to the evolution of the system. For example, a situation of data loss has many causes, such as network congestion, loss of connectivity or changing access interface. Each of these situations or states must be represented by at least one chronicle.

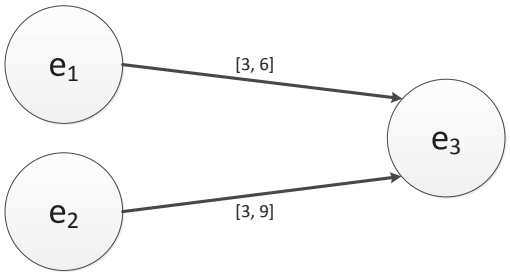

Figure 3: A chronicle with three events

5 Mannila's approach

Mannila's approach finds all episodes that occur frequently in a event sequence. It is based on the idea of first finding small frequent episodes and then progressively looking for larger frequent episodes.

Apart from the definitions in Section 3, Mannila defines the frequency fr of an episode as the fraction of windows in which the episode occurs. That is, given an event sequence s and a window width win, the frequency of an episode α in s is

$$fr(\alpha, s, win) = \frac{|w \in \mathcal{W}(s, win) \mid \alpha \, occurs \, in \, w|}{|\mathcal{W}(s, win)|}$$

Furthermore, α is frequent if $fr(\alpha, s, win) \geq min\_fr$ where min_fr is a frequency threshold.

The main objective is to discover all frequent episodes from a set of episodes. Mannila denotes the collection of frequent episodes with respect to s, win and min_fr by $\mathcal{F}(s, win, min\_fr)$.

From episode rule, the confidence is defined as $\frac{fr(\gamma, s, win)}{fr(\beta, s, win)}$.

The next Mannila algorithm describes how rules and their confidences can be computed from the frequencies of episodes where the input is a set E of event types, a sequence s, a set of $\mathcal{E}$ of episodes, a window width win, a frequency threshold min_fr and a confidence threshold min_conf; and the output is the episode rules that hold in s with respect to win, min_fr and min_conf.

```
/* Find frequent episodes */
compute F(s, win, min_fr)

/* Generate rules */
for all α ∈ F(s, win, min_fr) do
  for all β ≺ α do
    if fr(α) / fr(β) ≥ min_conf then
      output the rule β → α and
      the confidence fr(α) / fr(β);
```

6 Discovering episodes common to several sequences

In this section, a methodology to discover episodes in some sequences is developed. It is based on Mannila's approach because it finds the episode with the maximum width that occurs in all sequences.

The similarity between two windows is necessary to understand the developed approach. Two windows are similar if the number of all existing event types in the windows is equal. For example, in the next sequences, the windows $\omega(0, 2)$ over sequences s_1 and s_2 are not similar, but $\omega(1, 4)$ yes.

$$s_1 = \langle (A, 0), (B, 1), (B, 2), (C, 3), (C, 4), (B, 5), (D, 6) \rangle$$

$$s_2 = \langle (A, 0), (C, 1), (B, 2), (C, 3), (B, 4), (B, 5), (D, 7) \rangle$$

$$s_3 = \langle (A, 1), (B, 2), (B, 3), (C, 4), (C, 6), (D, 8) \rangle$$

Let S a set of sequences, l an integer and W a set of episodes.

```
/* Set variables */
S = {s_1, s_2, ..., s_n};
l = min{width(s_1), width(s_2), ..., width(s_n)};
W = ∅;

/* Get maximum similar windows */
while |W| = 0 and l > 0 do
  W = {ω(t_s, t_e) ∈ s | ∀s ∈ S, width(ω) = l};
  l = l − 1;

if |W| = 0 then
  there is not a common episode
else

  /* Construct chronicle */
  k = width(ω_0);
  nodes = ∅;

  while k > 0 do
    nodes = ∅;

    for all ω ∈ W do
      nodes = nodes ∩ event(ω, k);

    episode = episode ∩ nodes;
    k = k − 1;
```

In the above code, $nodes = nodes \cap event(\omega, k)$ adds events to existing in serial and $episode = episode \cap nodes$ adds events to existing in parallel. Furthermore, $event(\omega, k)$ gets the event in the ith position.

Note that the time constraints between two related events is the minimum between all similar pairs of events in the same position.

7 Experimentation

The algorithm does not been applied over a real data set, but it has been proven over the sequences s_1, s_2 and s_3 in Section 6.

The maximum similar windows are below:

$$\omega_1(0, 4) = \{(A, 0), (B, 1), (B, 2), (C, 3), (C, 4)\}$$

$$\omega_2(0, 4) = \{(A, 0), (C, 1), (B, 2), (C, 3), (B, 4)\}$$

$$\omega_3(1, 6) = \{(A, 1), (B, 2), (B, 3), (C, 4), (C, 6)\}$$

Finally, the chronicle can be viewed in Figure 4.

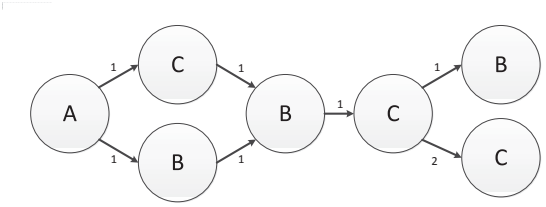

Figure 4: The generated chronicle

8 Conclusions

We have studied a method to find all episodes that occur frequently in a event sequence, Mannila's approach, whose objective is to find small frequent episodes at first and then progressively looking for larger frequent episodes. After that, we have developed a methodology to discover episodes in some sequences based on Mannila's approach. Thus, a new similarity between two windows has been defined to determine the maximum chronicle. To test the development of the methodology, it has been applied to three sample sequences to obtain the maximum common chronicle. Nevertheless, we think that this approach can be satisfactorily applied to data sets that it is the next step to complement this development.

Acknowledgments

This research is partially supported by the projects of the Spanish Ministry of Economy and Competitiveness ARTEMISA (TIN2009-14378-C02-01) and Simon (TIC-8052) of the Andalusian Regional Ministry of Economy, Innovation and Science.

References

[Bauer *et al.*, 2011] A. Bauer, A. Botea, A. Grastien, P. Haslum, and J. Rintanen. Alarm processing with model-based diagnosis of event discrete systems. In *Proceedings of the AI for an Intelligent Planet*, page 2. ACM, 2011.

[Bertrand *et al.*, 2009] O. Bertrand, P. Carle, and C. Choppy. Modelling chronicle recognition for distributed simulation processing with coloured petri nets. In *Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, page 42. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009.

[Cram *et al.*, 2011] D. Cram, B. Mathern, and A. Mille. A complete chronicle discovery approach: application to activity analysis. *Expert Systems*, 2011.

[Dousson and Ghallab, 1994] C. Dousson and M. Ghallab. Suivi et reconnaissance de chroniques. *Revue dintelligence artificielle*, 8(1):29–61, 1994.

[Dousson *et al.*, 2008] C. Dousson, F. Clerot, and F. Fessant. Method for the machine learning of frequent chronicles in an alarm log for the monitoring of dynamic systems, June 17 2008. US Patent 7,388,482.

[Le Guillou *et al.*, 2008] X. Le Guillou, M.O. Cordier, S. Robin, and L. Rozé. Chronicles for on-line diagnosis of distributed systems. In *Proceeding of the 2008 conference on ECAI 2008: 18th European Conference on Artificial Intelligence*, pages 194–198. IOS Press, 2008.

[Mannila and Ronkainen, 1997] H. Mannila and P. Ronkainen. Similarity of event sequences. In *Temporal Representation and Reasoning, 1997.(TIME'97), Proceedings., Fourth International Workshop on*, pages 136–139. IEEE, 1997.

[Mannila *et al.*, 1997] H. Mannila, H. Toivonen, and A. Inkeri Verkamo. Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery*, 1(3):259–289, 1997.

[Pencolé and Subias, 2009] Y. Pencolé and A. Subias. A chronicle-based diagnosability approach for discrete timed-event systems: Application to web-services. *Journal of Universal Computer Science*, 15(17):3246–3272, 2009.

[Saddem *et al.*, 2010] R. Saddem, A. Toguyeni, and M. Tagina. Consistency's checking of chronicles' set using time petri nets. In *Control & Automation (MED), 2010 18th Mediterranean Conference on*, pages 1520–1525. IEEE, 2010.