



HAL
open science

Admissible generalisation of temporal sequences as chronicles

Thomas Guyet

► **To cite this version:**

Thomas Guyet. Admissible generalisation of temporal sequences as chronicles. *Journal of Applied Non-Classical Logics*, 2023, 33 (3-4), pp.641-653. 10.1080/11663081.2023.2244717 . hal-04383648

HAL Id: hal-04383648

<https://hal.science/hal-04383648>

Submitted on 12 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Admissible Generalization of Temporal Sequences as Chronicles

T. Guyet^a

^aInria, AlstroSight, Lyon, France

ARTICLE HISTORY

Compiled June 1, 2023

ABSTRACT

Machine learning is the art of generalizing a set of examples. Beside the efficiency of the algorithms, the challenge is to define generalizations that make sense for a data scientist. In this article, we consider generalizations of temporal sequences as chronicles. A chronicle is a temporal model that represents a situation occurring in temporal sequences, i.e. a series of event types with timestamps. A chronicle is a collection of event types with metric temporal constraints on their delays of occurrence.

Generalizing sequences by a set of event types can intuitively be the smallest set of events that occur in all sequences. A question arises with the generalization of metric temporal constraints. In the article, we study the admissibility of these generalizations by deriving the notion of rule admissibility to the generalization as chronicles. Through formalization, new insights about the notions of chronicles may lead to conceive original chronicle mining algorithms.

KEYWORDS

Pattern mining; temporal patterns; topological operators;

1. Introduction

Generalizing a given set of examples is essential in many machine learning techniques. In principle, a machine learning algorithm builds an abstract model that represents a set of examples. But representing the examples does not mean to represent the examples at hand strictly (rote learning). This would be of little interest. What is expected from the result of such an algorithm is to find regularities that generalize examples. A difficult question arises: what is a good generalization of examples? As a process, a good generalization can be used to design efficient machine learning algorithms. As an output, a good generalization matches the intuition of users.

In this work, we consider examples that are temporal sequences, i.e. a series of event types with real-valued timestamps. A temporal sequence reports what's happening and at what time. This data type can be seen as the logs of a dynamic process. Such logs are commonly collected from sensors, websites, organizations, etc. Doing machine learning from temporal sequences aims to discover recurrent behaviours in them. It can be used to describe a set of temporal sequences or to predict future events.

One standard machine learning technique to extract valuable information from temporal sequences is temporal pattern mining. In the field of pattern mining, the problem is to discover recurrent patterns in the dataset from structured examples (e.g. itemsets, sequences, graphs). Temporal pattern mining is the branch of pattern mining dedi-

cated to extracting patterns from temporal data, such as temporal sequences. Each extracted pattern abstracts a subset of the structured examples.

We focus on a certain type of temporal pattern called *chronicle*. A chronicle is a collection of event types linked to each other, with metric temporal constraints on the delays between event occurrences. For instance, the chronicle with two event types a , b and a temporal constraint $[2, 3]$ between them occurs in any sequence containing both an a -event and an b -event whose difference between their occurrence date lies in the range $[2, 3]$. Chronicles (Dousson & Duong, 1999) are interesting because of their expressiveness: metric temporal intervals are specified upon pairs of events. It has been investigated in different mining approaches (Álvarez, Félix, Cariñena, & Otero, 2010; Cram, Mathern, & Mille, 2012; Dauxais, Guyet, Gross-Amblard, & Happe, 2017; Sahuguède, Le Corronc, & Le Lann, 2018). Moreover, the model of chronicle subsumes other temporal models such as δ -patterns (Giannotti, Nanni, & Pedreschi, 2006) or TGSP¹ patterns (Yen & Lee, 2013) which also combine collections of event types and metric temporal constraints.

The problem of pattern mining is a problem of generalization of examples. The objective of such algorithm is to identify patterns that are the representatives of a collection of examples in a dataset. Pattern mining algorithms have been widely investigated for domains such as sets of items or sequences of items. Usually, the topological structure of these pattern spaces (lattice or semi-lattice) enables us to define the least general generalizations (lgg). Then, the lgg provides a natural and intuitive candidate for a generalization of examples. The extension of this approach to more complex domains of patterns such as chronicles may have two limitations: 1) the topological structure of the pattern domain makes it difficult to define a least general generalization, and 2) the least general generalization does not match the intuition of a good generalization by experts.

The particular case of a generalization of temporal sequences as a chronicle raises the issue of determining which are the chronicles that generalize well from a collection of sequences. At least, we expect from the chronicle to match all the sequences. But the nature of temporal constraints, with temporal intervals, implies that for a given set of sequences, there is an infinity of chronicles that match them. For computational purposes, chronicle mining algorithms make implicit or explicit choices underlying the task of determining what chronicles abstract a given set of sequences.

To address this issue, we propose a formalization of the problem of chronicle mining, and then we propose different classes of generalization as chronicles derived from the closure-like operators. Our formalization combines two different contributions we jointly proposed with Philippe Besnard: the formalization of the space of chronicles (Besnard & Guyet, 2023) and the formalization of rule learning with metric attributes (Besnard, Guyet, & Masson, 2019). These two works have in common: 1) to investigate formalizations with tools from topology theory and 2) to address a problem of making good generalizations from examples with metric attributes. It turns out they can be merged to propose a new formalization of chronicle mining seen as a generalization from temporal sequences.

The objective of such a work is to present a formalization of chronicle mining that may help to better characterize existing algorithms. These algorithms have implicit or explicit assumptions that specify what is a generalization as chronicle. We do not think that there are a better generalization than the others. Through our formalization, we can relate the technical choices to fundamental notions and thus we could help to

¹TGSP stands for “Temporal Generalized Sequential Pattern”.

choose which algorithm will behave as expected by a data scientist.

2. Temporal sequences and chronicles

In this section, we introduce the formal model of chronicles. We start by introducing the definition of temporal sequence, chronicle, and we give their semantics through the definition of an occurrence of a chronicle in a temporal sequence.

A chronicle is a temporal model that has been introduced by Dousson, Gaborit, and Ghallab (1993) for situation recognition. In short, a chronicle is a set of events and a set of real-valued temporal constraints on the delay between pairs of events. A chronicle can be seen as a generalization of a set of temporal sequences when it is recognized within each of them.

Given a finite alphabet Σ of atomic events, we first introduce the notion of temporal sequence over Σ and then the notion of chronicle. \leq_{Σ} denotes a total order on the elements of Σ . In the sequel, we denote by $[n]$ the set of the n -th first integers $\{1, \dots, n\}$.

Definition 2.1. A **sequence** (or temporal sequence) is a finite list of events $\langle (e_1, t_1), (e_2, t_2), \dots, (e_n, t_n) \rangle$. The events in the temporal sequence are ordered by \prec defined as $\forall i, j \in [n], i < j \Leftrightarrow (e_i, t_i) \prec (e_j, t_j) \Leftrightarrow t_i < t_j \vee (t_i = t_j \wedge e_i \prec_{\Sigma} e_j)$.

A sequence $S' = \langle (e'_1, t'_1), (e'_2, t'_2), \dots, (e'_m, t'_m) \rangle$ is a **subsequence** of another sequence $S = \langle (e_1, t_1), (e_2, t_2), \dots, (e_n, t_n) \rangle$, denoted $S' \preceq S$ iff there exists a strictly increasing function $\tau : [m] \mapsto [n]$ such that $e'_i = e_{\tau(i)}$ and $t'_i = t_{\tau(i)}$ for all $i \in [m]$.

Definition 2.2 (Chronicle). A **chronicle** is an ordered pair $\mathcal{C} = (\mathcal{E}, \mathcal{T})$ where

- \mathcal{E} is a finite **multiset** over Σ , i.e., \mathcal{E} is of the form $\{\{c_1, \dots, c_m\}\}$ such that $c_i \in \Sigma$ for $i = 1, \dots, m$ and $c_1 \leq_{\Sigma} \dots \leq_{\Sigma} c_m$
- \mathcal{T} is a set of **temporal constraints**, i.e., expressions of the form² $(c, o_c)[t^-, t^+](c', o_{c'})$ such that
 - $c, c' \in \mathcal{E}$
 - $t^-, t^+ \in \mathbb{R} \cup \{-\infty, +\infty\}$
 - $o_{\mathcal{E}}(c), o_{\mathcal{E}}(c') \in [m]$ and $o_{\mathcal{E}}(c) < o_{\mathcal{E}}(c')$
 - $c_{o_{\mathcal{E}}(c)} = c$ and $c_{o_{\mathcal{E}}(c')} = c'$.

The size of a chronicle $(\mathcal{E}, \mathcal{T})$ is the size m of its multiset \mathcal{E} .

In a chronicle $(\mathcal{E}, \mathcal{T})$, $o_{\mathcal{E}}(c)$ denotes the position of an event type c in the ordered multiset \mathcal{E} . Considering the fourth condition on the temporal constraints, the notation $(c, o_{\mathcal{E}}(c))$ is redundant ($o_{\mathcal{E}}(c)$ would be sufficient) but it enjoys more readability in examples.

A chronicle can be visualized as a graph with one vertex per event. Edges represent the temporal constraints. Let $\Sigma = \{a, b, c, d\}$, Figure 1 illustrates the chronicle $(\{\{a, b, c\}\}, \{(a, 1)[-3, 2](b, 2), (a, 1)[-2, 1](c, 3), (b, 2)[-1, 5](c, 3)\})$.

The definition of a chronicle is close to the notion of simple temporal constraint network (TCN) (Dechter, Meiri, & Pearl, 1991). However, TCN are oriented toward the reasoning with temporal constraints while chronicles are oriented toward tasks in handling temporal data. More specifically, it makes explicit certain aspects of temporal data, such as repetitions of an event type. The way it is handled by chronicles make the main difference.

²Here and in the sequel, we use a notation with closed intervals, $[t^-, t^+]$, even when t^- and/or t^+ are infinite.

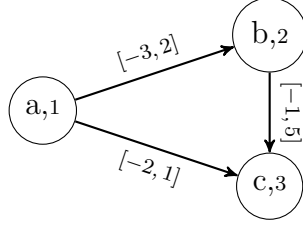


Figure 1. Graphical representation of the chronicle detailed in the text.

One important notion for the generalization is the occurrence of a chronicle in a sequence. Indeed, we require from a chronicle that generalizes a set of sequences \mathcal{S} to occur in every sequence of \mathcal{S} .

Definition 2.3 (Chronicle occurrences and embeddings). An **occurrence** of a chronicle $\mathcal{C} = (\{c_1, \dots, c_m\}, \mathcal{T})$ in a sequence $S = \langle (e_1, t_1), \dots, (e_n, t_n) \rangle$ is a subsequence $\langle (e_{\varphi(1)}, t_{\varphi(1)}), \dots, (e_{\varphi(m)}, t_{\varphi(m)}) \rangle$ of S such that:

- (1) $\varphi : [m] \rightarrow [n]$ is an injective function,
- (2) $\varphi(i) < \varphi(i+1)$ whenever $c_i = c_{i+1}$,
- (3) $c_i = e_{\varphi(i)}$ for $i = 1, \dots, m$,
- (4) $t_{\varphi(j)} - t_{\varphi(i)} \in [t^-, t^+]$ whenever $(c_i, i)[t^-, t^+](c_j, j) \in \mathcal{T}$.

We call φ an **embedding**.

A chronicle \mathcal{C} **occurs** in S iff there is at least one occurrence of \mathcal{C} in S .

The temporal constraints of a chronicle are conjunctive: an embedding of a chronicle has to satisfy all of them. Then, in any chronicle, \mathcal{T} can be assumed to contain at most one temporal constraint per a pair of events. For a deeper understanding of the semantics of chronicles, Guyet and Markey (2022) compared this temporal model with classical temporal logics (MTL and TPTL).

Example 2.4. The chronicle depicted in Figure 1 occurs in the sequence $\langle (b, 0.2), (a, 2.9), (a, 3.5), (b, 3.9), (c, 4.1) \rangle$, as witnessed by the last three events. However, $\langle (b, 0.2), (a, 2.9), (c, 4.1) \rangle$ is not an occurrence of this chronicle because the temporal constraint between a and c is not satisfied ($4.1 - 2.9 \notin [-2, 1]$).

The lgg seems intuitive to formalize a generalization of examples but the topological structure of the space of chronicles does not allow us to define a simple and intuitive lgg for chronicles. For more details about the space of chronicles, we invite the reader to explore the formal account of chronicles by Besnard and Guyet (2023).

This motivates us to propose a formalization of chronicle mining in order to provide a characterization of the ways the existing algorithms generalize temporal sequences as chronicles.

3. Formalizing the process of chronicle mining

In this section, we present a formalization of chronicle mining. Following the approach introduced in (Besnard et al., 2019), the main objective is to capture the general principles of chronicle mining. This is helpful to characterize the behaviour of various algorithms that exist or may be proposed.

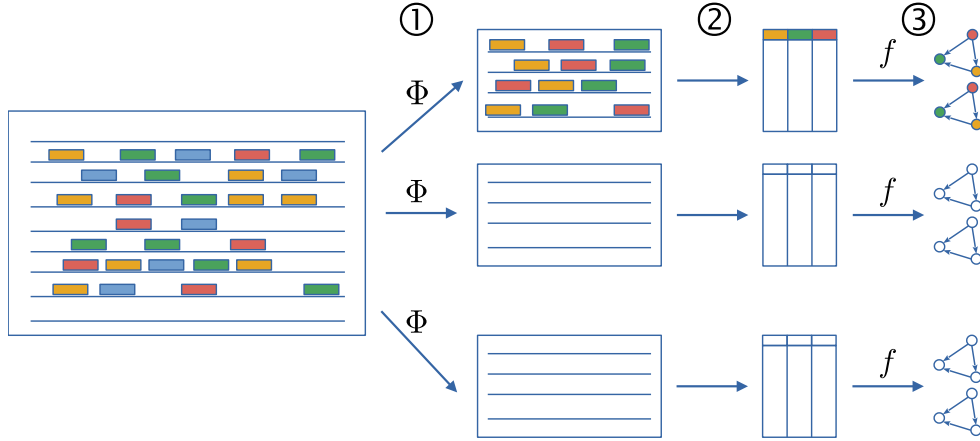


Figure 2. Illustration of the abstract process of chronicle mining. Figures in black point out the three-steps of chronicle mining (see main text). Colours illustrate the event types.

The chronicle mining process is viewed as a three-step process illustrated in Figure 2:

- (1) Some subsets of subsequences are selected,
- (2) Each subset is transformed into a relational entity (or table) with metric attributes,
- (3) Each relational entity is generalized by a disjunction of chronicles.

It is worth noting that this article details mainly on the third step (see Section 4). The two first steps are described briefly in the remaining of this section and illustrated in Example 3.1.

The input of a chronicle mining algorithm is a set of sequences \mathcal{S} . In the first step, a Φ function yields subsets of sequences, denoted by \mathcal{R} , restricted to some multiset of event types σ .³ More formally, for all $S \in \mathcal{R}$: $\exists S' \in \mathcal{S}$, $S \preceq S'$ and $\{\{x \mid (x, t) \in S\}\} = \sigma$.

It is worth noting that Φ represents a choice among all possible collections of subsequences. More specifically, 1) it chooses a collection of subsequences having a multiset of event types σ , but this subset does not necessarily contain all of them; and 2) if there are more events of a certain type in a dataset sequence S' than in the multiset σ , the Φ function chooses which events of the sequence S' are kept in the sequence S to have the exact number of instances for each event types. For example, it can choose the first occurrences of each event type in S . This means that the Φ function defines a collection of subsequences to generalize. This is an important step in the formalization of a temporal pattern mining algorithm but, in this article, the formalization of Φ is put aside.

Another important remark is that Φ makes use of a multiset σ . Intuitively, a generalization involves the maximum multiset of event types that occurs in all sequences. Thus, all smaller multisets are acceptable multisets to yield chronicles. The problem of generalization arises when temporal constraints have to be defined.

In the second step, the subset of sequences is transformed into a relational entity (or table) by computing delays between events. Each attribute of the entity represents a temporal constraint, denoted $\mathcal{A}_{(c, o_c) \rightarrow (c', o_{c'})}$ between two event types in $(c, c') \in \sigma^2$,

³The notation R has been chosen because the subset can be represented as a *Rectangle*. When working on generalizations as a rule, we make use of the term “square” in our internal discussions to denote similar objects. It was obviously not a square, but we kept the notation S . I hope this notation change from S to R will suit Philippe.

$o_\sigma(c) < o_\sigma(c')$. Each row corresponds to a sequence of \mathcal{S} . The entity contains the temporal delays between the occurrences of each pair of event types, for each sequence. All these values exist and are unique thanks to the definition of a sequence generated by Φ w.r.t to the multiset σ .

The third step processes each attribute $\mathcal{A}_{(c,o_c)\rightarrow(c',o_{c'})}$ independently. It generalizes a set of real values – the time delays between occurrences of the corresponding events in sequence – into a disjunction of intervals $\widehat{\mathcal{A}_{(c,o_c)\rightarrow(c',o_{c'})}}$. We denote by f the function that abstracts this third step. This step is more deeply investigated in the next section. f is named a *choice function* as it makes a choice among all the possible sets of intervals that include all values. Finally, a disjunctive collection of chronicles with the form (σ, \mathcal{T}) is created from the selected sets of intervals. \mathcal{T} is a set of temporal constraints that uses one interval of $\widehat{\mathcal{A}_{c\rightarrow c'}}$ to define a temporal constraint between for each pair of events $(c, c') \in \sigma$, $o_\sigma(c) < o_\sigma(c')$.

Example 3.1 (Illustration of the overall process). This example illustrates the steps of the proposed process. Let $\Sigma = \{a, b, c, d, e\}$ be a set of event types and \mathcal{S} a set of temporal sequences as follows:

$$\begin{aligned} S_1 : & \langle (a, 1), (b, 5), (c, 5), (a, 6), (c, 8) \rangle \\ & \vdots \\ S_4 : & \langle (a, 3), (c, 4), (c, 10), (d, 10) \rangle \\ & \vdots \\ S_{26} : & \langle (b, 1), (b, 3), (e, 3), (c, 8), (a, 10), (c, 12) \rangle \\ & \vdots \end{aligned}$$

A subset \mathcal{S} of temporal sequences with $\sigma = \{\{a, c, c\}\}$, and sequences $\{S_1, S_4, S_{26}\}$ is as follows:

$$\begin{aligned} S_1 : & \langle (a, 1), (c, 5), (c, 8) \rangle \\ S_4 : & \langle (a, 3), (c, 4), (c, 10) \rangle \\ S_{26} : & \langle (c, 8), (a, 10), (c, 12) \rangle \end{aligned}$$

In case of multiple occurrences of an event type, a sequence is here constructed by taking the first occurrence of each event in $\sigma = \{\{a, c, c\}\}$.⁴ One can imagine having another choice: last occurrence, the one with the shortest delay with the previous or even to take all sequences in each temporal sequence. Different choices would lead to subsequences with different timestamps, and thus to different generalizations as chronicles. This would correspond to different algorithm strategies that could be encountered in the literature.

In the second step, the delay between each pair of event types is computed to yield the relational dataset below. Note that there is no ambiguity about how to map the two c of the multiset σ with events in sequences. According to the condition (2) in Def. 2.3, the first (resp. the second) occurrence of c in a sequence is mapped to $(c, 2)$ (resp. $(c, 3)$). The dataset contains only metric attributes. The negative delay indicates that the occurrence of a is before the one of c .

⁴This is a strategy that corresponds to the notion of chronicle reduct (Besnard & Guyet, 2023).

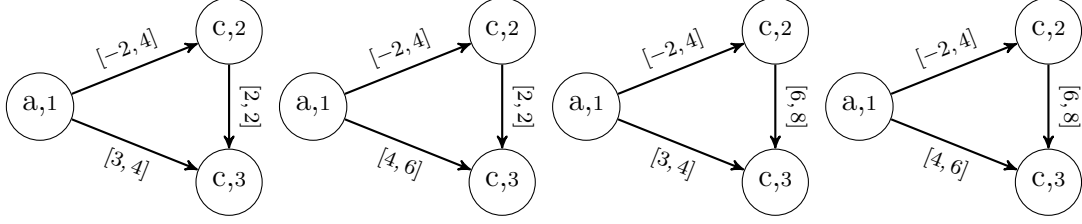


Figure 3. Illustration of the four chronicles extracted in the Example 3.1.

$\widehat{\mathcal{A}}_{(a,1) \rightarrow (c,2)}$	$\widehat{\mathcal{A}}_{(a,1) \rightarrow (c,3)}$	$\widehat{\mathcal{A}}_{(c,2) \rightarrow (c,3)}$
4	7	3
1	7	6
-2	2	4

Finally, the third step of the process generalizes each metric attribute by a set of intervals. For instance, $\widehat{\mathcal{A}}_{(a,1) \rightarrow (c,2)} = \{[-2, 4]\}$, $\widehat{\mathcal{A}}_{(a,1) \rightarrow (c,3)} = \{[2, 2], [6, 8]\}$ and $\widehat{\mathcal{A}}_{(c,2) \rightarrow (c,3)} = \{[3, 4], [4, 6]\}$. The previous intervals illustrate the wide range of possible choices of intervals: we can use a unique interval that contains all values, or several intervals; Boundaries of intervals can be set among the values of the entity or not; etc. The next section will focus on the kinds of generalization that are *admissible*. Then, the resulting collection of chronicles contains the four chronicles illustrated in Figure 3.

Notation 1. For the sake of conciseness, a collection resulting from the abstraction of the temporal delays is denoted by $\mathcal{C} = (\sigma_S, \{(a, 1)\widehat{\mathcal{A}}_{1 \rightarrow 2}(c, 2), (a, 1)\widehat{\mathcal{A}}_{1 \rightarrow 3}(c, 3), (c, 2)\widehat{\mathcal{A}}_{2 \rightarrow 3}(c, 3)\})$. Considering that each $\widehat{\mathcal{A}}_{i \rightarrow j} = \mathcal{A}_{(\sigma_i, i) \rightarrow (\sigma_j, j)}$ is a set of intervals, then each chronicle of \mathcal{C} picks one of these interval for each delay attribute.

As we already noticed, there is an intuitive choice for selecting the multiset of a generalization as chronicles. The question that arises is how to generalize from delays as temporal constraints. In the proposed framework, it amounts to the generalization of a metric attribute. In the next section, we study the admissible generalizations of a metric attribute.

4. Admissible generalizations of a metric attribute

Given the values $\{u_1, \dots, u_n\}$ of a delay attribute, an issue is to determine classes of operator $\widehat{\cdot}$ that yields intuitive intervals, i.e., intuitive temporal constraints.

This question is similar to that of the notion of *rule admissibility* introduced by Besnard et al. (2019). It is here adapted for the admissibility of the generalization of delays as temporal constraints of a chronicle. Note that we do not provide a definition of admissibility, the idea is to propose a framework to define different types of admissibility that can be confronted to user intuition and practical results.

The starting point for defining an admissible generalization is the intuitive notion of closure: all sets of delays S' that are close enough to a given set S have the same (or a close) generalization from the generalization of S . Then, our proposal is to derive different notions of admissibility from closure-like operators.

4.1. Closure-like operators

The intuition we point out in the latter remark suggests that generalizing a set of values to a superset thereof amounts to applying a closure-like operator $\widehat{\cdot}$.⁵ For any attribute $\mathcal{A}_{i \rightarrow j}$, generalizing the values A_{ij} for $\mathcal{A}_{i \rightarrow j}$ is identified with mapping A_{ij} to $\widehat{\mathcal{A}_{i \rightarrow j}}$ with properties taken from the list of Kuratowski's axioms:

$$\begin{aligned} \widehat{\emptyset} &= \emptyset \\ S \subseteq \widehat{S} &\subseteq \mathbb{R} \\ \widehat{\widehat{S}} &= \widehat{S} \\ \widehat{S \cup S'} &= \widehat{S} \cup \widehat{S'} \quad (\text{pre-closure}) \end{aligned}$$

Actually, we downgrade the last axiom as follows (from the strongest to the weakest operator)⁶:

$$\begin{aligned} \widehat{S} \subseteq \widehat{S'} &\text{ whenever } S \subseteq S' && (\text{closure}) \\ \widehat{S} = \widehat{S'} &\text{ whenever } S \subseteq S' \subseteq \widehat{S} && (\text{cumulation}) \\ \widehat{S \cup S'} \subseteq \widehat{S} &\text{ whenever } S' \subseteq \widehat{S} && (\text{capping}) \end{aligned}$$

We thus arrive at four classes of weaker operators: preclosure, closure, cumulation and capping operators.

An operator is defined through a choice function, denoted f in our formalization. Let Z be a set, a choice function $f : 2^{2^Z} \rightarrow 2^Z$ elicits one subset of Z among all the possible subsets of Z . Letting $\uparrow\{X\} \stackrel{\text{def}}{=} \{Y \mid X \subseteq Y \subseteq \mathbb{R}\}$, a closure-like operator on metric attributes $\widehat{\cdot}$ is defined through a choice function as follows:

$$\widehat{X} = f(\uparrow\{X\})$$

The second Kuratowski's axiom implies that the generalization of a set X has to be taken among the supersets of X , i.e. $\uparrow\{X\}$.

The following sections illustrate three of these operators: we first propose a generalization based on the convex hull of delays (closure), then we introduce the principle of neighbourhoods (pre-closure) and finally the generalization based on splits on gaps (capping).

4.2. Convex hull

An approach uses the convex hull principle of a set of delays. This approach is the one followed in most of the chronicle mining algorithms (Cram et al., 2012; Dousson & Duong, 1999). Formally, the choice function defines, for $X \subseteq \mathbb{R}$,

$$f(\uparrow\{X\}) \stackrel{\text{def}}{=} \begin{cases} \emptyset & \text{if } X = \emptyset \\ [\min_{x \in X}(x), \max_{x \in X}(x)] & \text{otherwise} \end{cases}$$

Extend f to all of 2^{2^Z} by taking $f(\mathcal{X}) \stackrel{\text{def}}{=} [\min \min_{S \in \min \mathcal{X}}(f(\uparrow\{S\})), \max \max_{S \in \max \mathcal{X}}(f(\uparrow\{S\}))]$. Then, f induces a closure operator.

⁵Closure-like operators are topological operators.

⁶A capping is a cumulation, that is a closure, that is in turn a pre-closure.



Figure 4. Illustration of generalizations based on convex hull. Each cross is an example (delay values on the x-axis), the red bars illustrate the generalization as intervals (temporal constraints).

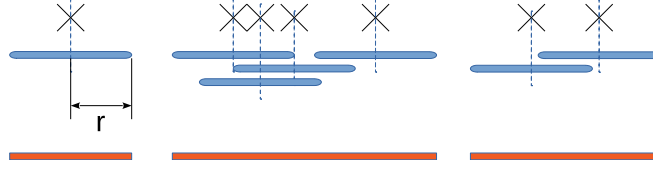


Figure 5. Illustration of generalizations based on example neighbourhoods. Each cross is an example (delay values on the x-axis), a blue bar illustrates the neighbourhood of one example, the red bars illustrate the generalization as intervals (temporal constraints).

Figure 4 illustrates the generalisation based on convex hull.

The notion of convex hull is only sensitive to the extreme values of a dataset. A set of delays $S_1 = \{1, 2, 9, 11\}$ has the same generalization as the set $S'_1 = \{1, 2, 5, 7, 9, 11\}$, i.e. the interval $[1, 11]$.

4.3. Neighbourhoods

In the previous example, we could be interested in considering that there is a large gap between 2 and 9 that may justify splitting the interval into two distinct intervals. Then, we can regard $[1, 2] \cup [9, 11]$ as more admissible temporal constraints than the large $[1, 11]$. In this case, it would take into account the closeness between neighbourhoods, and another notion of admissibility has to be defined.

In this case, the approach would use a neighbourhood principle. Since the delay δ between two events a and b is encountered in examples, it seems rather reasonable to consider that close delays can be abstracted by the same chronicle, meaning that it represents almost the same situation. A temporal constraint exemplifying this would be $(a, 1)[\delta - r, \delta + r](b, 2)$, where r is a parameter that sets the notion of “closeness” between delays.

For a datum $u \in \mathbb{R}$, we look at a generalization for u in the form of the neighbourhood centred at u of radius r , for a given $r > 0$. Figure 5 illustrates the construction of generalizations (set of intervals) based on neighbourhoods.

Formally, the choice function defines, for $X \subseteq \mathbb{R}$,

$$f(\uparrow\{X\}) \stackrel{\text{def}}{=} \begin{cases} \emptyset & \text{if } X = \emptyset \\ \bigcup_{x \in X} [x - r, x + r] & \text{otherwise} \end{cases}$$

Extend f to all of $2^{2^{\mathbb{Z}}}$ by taking $f(\mathcal{X}) \stackrel{\text{def}}{=} \bigcup_{S \in \min \mathcal{X}} f(\uparrow\{S\})$.

Then, f induces a pre-closure operator (Besnard et al., 2019). Obviously, it is not a closure operator. A counterexample was given in the previous section. As expected, we obtain a generalization $[1 - r, 2 + r] \cup [9 - r, 11 + r]$ for S'_1 and $[1 - r, 11 + r]$ for $S_1 \subset S'_1$.

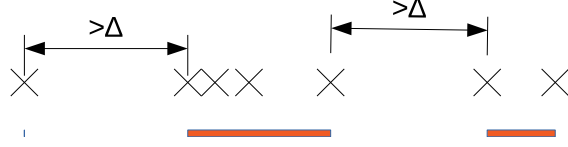


Figure 6. Illustration of a generalization based on examples split on gaps. Each cross is an example (value on the x-axis), Δ is a threshold for splitting intervals, the red bar illustrates the generalization as intervals (temporal constraints). The red tick at the bottom-left is a singleton interval.

One can note that the predefined radius r does not take into account the actual distribution of values when it comes to finding intervals. The next section offers interpolation from pairs (u, v) of values.

4.4. Splits on gaps

Now, for generalization from X as interpolation of the kind: If $u \in C$ and $v \in X$ such that the gap between u and v is smaller than some threshold, then generalize u and v to all values between u and v . Figure 6 illustrates the principle of this construction of the generalization as temporal constraints. Again, generalizing X amounts to applying some closure-like operator $\hat{\cdot}$, giving \hat{X} .

Let f selects, among all supersets of $X = \langle x_1, \dots, x_m \rangle$,⁷ the union of the intervals over \mathbb{R} that have both endpoints in X and length (denoted by l) bounded by a threshold value $\Delta(X)$ as follows:

$$\varphi X(x_j) \stackrel{\text{def}}{=} \begin{cases} [x_j, x_{j+1}] & \text{if } l([x_j, x_{j+1}]) \leq \Delta(X) \\ [x_j, x_j] & \text{otherwise (including } j = m) \end{cases} \quad (1)$$

where Δ is a function on the increasing sequences over \mathcal{R} .

For all finite $X \subseteq \mathbb{R}$, define

$$f(\uparrow\{X\}) \stackrel{\text{def}}{=} \bigcup_{x \in X} \varphi X(x) \quad (2)$$

where the $\varphi X : X \rightarrow 2^{\mathbb{R}}$ functions can be required to satisfy, for all $x \in X$ and all finite $X' \subseteq \mathbb{R}$, the following constraints:

- (1) $x \in \varphi X(x)$,
- (2) $X \subseteq X' \subseteq \bigcup \varphi X(X) \Rightarrow \bigcup \varphi X'(X') \subseteq \bigcup \varphi X(X)$.

Extend f to all of 2^{2^Z} by taking

$$f(X) \stackrel{\text{def}}{=} \bigcap \mathcal{X} \text{ whenever } \mathcal{X} \neq \uparrow\{S\} \text{ for all } S \subseteq Z. \quad (3)$$

The following instances to Δ give choice functions, as per eq. (1)-(3), generating rules according to capping (Besnard et al., 2019):

- Geometric mean: $\Delta(S) = \left(\prod_{i=1}^{m-1} (x_{i+1} - x_i) \right)^{\frac{1}{m-1}}$
- Higher power means: $\Delta(S) = \sqrt[p]{\frac{1}{m-1} \sum_{i=1}^{m-1} (x_{i+1} - x_i)^p}$

⁷ S is identified with its enumeration in increasing order as this simplifies the formulation in eq. 1.

In this notion of admissible generalization, the gap between two consecutive values is to be turned into an interval unless the gap is much greater than most of the other gaps in the series. For instance, let $S = \{3.1, 3.3, 3.5, 3.6, 5.0, 5.5, 6.0\}$ and $S' = \{3.1, 3.6, 5.0, 5.5, 6.0, 6.8, 7.1\}$ the gaps in S from 5.0 to 6 have lengths equals to 0.5 but all other gaps from 3.1 to 3.6 have lengths lower than 0.2. It is a counterexample to cumulation because, according to the principle of the splitting on gaps, the generalization of S would be $[3.1, 3.6] \cup [5.0, 6.0]$ but the generalization would be $[3.1, 6.0]$. Then, it turns out to be only a capping operator.

5. Conclusions

We proposed a framework to formalize the generalization of temporal sequences as chronicles. It is at the crossroad of two contributions of Philippe Besnard in the field of machine learning: a contribution to the formalization of the space of chronicles and a contribution to the formalization of admissible rules.

The objective of this work is not to propose an operational framework for chronicle mining but a formalization to analyse the consequences of choices made by algorithm designers. The different notion of admissible generalizations as chronicles can be used to discuss the outputs of algorithms. Our intuition is that most of the pattern mining algorithms behave more as a closure-admissibility, an algorithm like TGSP (Yen & Lee, 2013) yields more preclosure-admissible chronicles, and DCM (Dauxais et al., 2017) yields more cumulation or capping-admissible chronicles.⁸

The strength of the formalizations that have been proposed by Philippe Besnard lies in the fact that they capture intuitions. For instance, the operators that have been presented capture the idea that there are two different strategies to create the temporal intervals from collections of delays: a merge vs split approach. This has been proposed as an alternative algorithmics (Guyet & Quiniou, 2011). What makes this work interesting is to bring the light on the underlying implications and to make them intuitive. This is undoubtedly a meaningful and an original approach that may be pursued for proposing an in-depth formalization of machine learning.

Acknowledgements

My acknowledgements go to Philippe for sharing his experience and knowledge about a broad range of topics coming from sports to research administration. I have been very lucky to share an office with Philippe during more than five years. His patience, his determination, his kindness and his commitment to collaborative work are impressive and inspiring.

References

- Álvarez, M. R., Félix, P., Cariñena, P., & Otero, A. (2010). A data mining algorithm for inducing temporal constraint networks. In *Proceedings of the international conference on information processing and management of uncertainty in knowledge-based systems (IPMU)* (Vol. 6178, pp. 300–309). Springer.

⁸DCM – Discriminant Chronicle Mining – is based on the CN2 rule learning algorithms (Clark & Niblett, 1989), and our analysis of CN2 shows that it behaves more like a cumulation than a capping.

- Besnard, P., & Guyet, T. (2023). *Chronicles: Formalization of a temporal model*. Springer briefs.
- Besnard, P., Guyet, T., & Masson, V. (2019). Admissible generalizations of examples as rules. In *Proceedings of the international conference on tools with artificial intelligence (ICTAI)* (pp. 1488–1493). IEEE.
- Clark, P., & Niblett, T. (1989). The CN2 induction algorithm. *Machine learning*, 3(4), 261–283.
- Cram, D., Mathern, B., & Mille, A. (2012). A complete chronicle discovery approach: application to activity analysis. *Expert Systems*, 29(4), 321–346.
- Dauxais, Y., Guyet, T., Gross-Amblard, D., & Happe, A. (2017). Discriminant chronicles mining. In *Proceedings of the conference on artificial intelligence in medicine (AIME)* (Vol. 10259, pp. 234–244). Springer.
- Dechter, R., Meiri, I., & Pearl, J. (1991). Temporal constraint networks. *Artificial Intelligence*, 49, 61–95.
- Dousson, C., & Duong, T. V. (1999). Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems. In *Proceedings of the international joint conference on artificial intelligence (IJCAI)* (pp. 620–626). Morgan Kaufmann.
- Dousson, C., Gaborit, P., & Ghallab, M. (1993). Situation recognition: Representation and algorithms. In *Proceedings of the international joint conference on artificial intelligence (IJCAI)* (Vol. 93, pp. 166–172).
- Giannotti, F., Nanni, M., & Pedreschi, D. (2006). Efficient mining of temporally annotated sequences. In *Proceedings of the siam international conference on data mining (SDM)* (pp. 348–359). SIAM.
- Guyet, T., & Markey, N. (2022). Logical forms of chronicles. In *Proceedings of the international symposium on temporal representation and reasoning (time)* (Vol. 247, pp. 7:1–7:15). Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- Guyet, T., & Quiniou, R. (2011). Extracting temporal patterns from interval-based sequences. In *Proceedings of the international joint conference on artificial intelligence (IJCAI)* (pp. 1306–1311). AAAI Press.
- Sahuguède, A., Le Corronc, E., & Le Lann, M.-V. V. (2018). An ordered chronicle discovery algorithm. In *Proceedings of the ECML/PKDD workshop on advanced analytics and learning on temporal data*.
- Yen, S.-J., & Lee, Y.-S. (2013). Mining non-redundant time-gap sequential patterns. *Applied Intelligence*, 39, 727–738.