Extraction of Classification Rules from Sequences of Crystal Growth Data

Radek Buša[1], Yann Dauxais[2], Stefan Ecklebe[3], Natasha Dropka[4], Martin Holeňa[5,6]

[1] Faculty of Information Technology, Czech Technical University, Thákurova 9, Prague, Czech Republic
[2] KU Leuven, Celestijnenlaan 200a, Leuven, Belgium
[3] Institute of Control Theory, TU Dresden, Georg-Schumann-Str. 7a, Dresden, Germany
[4] Leibniz Institut für Kristalzüchtung, Max-Born Str. 2, Berlin, Germany
[5] Institute of Computer Science, Czech Academy of Sciences, Prague, Czech Republic
[6] Leibniz Institute for Catalysis, Albert-Einstein Str. 29a, Rostock, Germany

Abstract: The paper presents a generalization of a data mining method for the extraction of classification rules for classification of sequences of events, which is called discriminant chronicles mining. The generalization is motivated by the objective to extract classification rules from crystal growth data, for which the original method needs to be extended to events with vectors of attributes and to real-valued attributes. The paper elaborates incorporating both extensions into the theoretical fundamentals of the original method, and describes a corresponding modification of a system for discriminant chronicles mining, which has been developed three years ago to implement the original method. Finally, an application of the generalized method, using the modified system for discriminant chronicles mining, to data from the growth of GaAs crystals by vertical gradient freeze method is briefly sketched.

1 Introduction

This paper deals with data mining of crystal growth data, obtained either experimentally or from simulations. Such data records the crystal growth process, its performance, and conditions in the melt, such as temperatures in various control points or the power of heaters, or parameters of the magnetic fields influencing melt convection [6, 7]. In particular, we consider the common situation that the performance data indicates whether the crystal growth process can be classified as satisfactory according to a given criteria , e.g., according to the shape or position of the solid/liquid interface. Hence, the primary data mining approach to that data is the extraction of classification rules.

Although a plethora of methods for classification rules extraction exist [10, 11], most of them cannot be used for our data. The reason is that crystal growth proceeds sequentially, hence, the data is inheretly sequential. Therefore, we have chosen a specific rules extraction method extracting classification rules for the classification of sequences of events, which was proposed in [3]. It is called "discriminant chronicles mining" because it was originally developed for events described with attributes conveying a temporal meaning. However, it cannot be directly applied to crystal growth data, due to the following two restrictions:

(i) The events in [3] are described with scalar values of the temporal attribute. On the other hand, members of sequences of crystal growth data, which we will for simplicity also call events, are described with vectors of attribute values.

(ii) The temporal attribute describing events in [3] has a finite number of values, thus it can be represented by a finite subset of integers. On the other hand, attributes describing events in sequences of crystal growth data are real-valued.

Therefore, we have extended the method from [3] to sequences of events described by real-valued vector attributes. This extension is the main contribution of the paper.

The next section briefly recalls the original method proposed in [3]. Its extension removing the restrictions i and ii above is described in Section 3. Finally, an application of the proposed method to crystal growth data is sketched in Section 4.

2 Classification Rules Extraction with Discriminant Chronicles

Let $\mathbb{E}$ be a finite set, the elements of which are called *event types*, and let $\mathbb{T}$ be an arbitrary subset of the extended reals, $\mathbb{T} \subset \bar{\mathbb{R}}$. For a multiset of m event types, the rather unusual notation $\{\{e_1, \ldots, e_m\}\}$ has been introduced in [3], and a couple $(e,t) \in \mathbb{E} \times \mathbb{T}$ is called *event*.

Assume further that some ordering is imposed to the event types in the application domain. In [3], where temporal relationships are investigated, a total ordering corresponding to their order of occurrence is considered. For the rules extraction method proposed in Section 3), however, the weaker concept of a partial ordering $\prec$ will be sufficient, with a semantics tailored to a partiucular application (see Section 4 for the real-world application considered in this paper). For $e_1, e_2 \in \mathbb{E}$, $t^-, t^+ \in \bar{\mathbb{R}}$ such that $e_1 \prec e_2, t^- \leq t^+$, a *temporal constraint* is a tuple (e_1, e_2, t^-, t^+), also denoted $e_1[t^-, t^+]e_2$. The semantics of such a temporal constraint is as follows: the difference between the timestamps t_2 of an event (e_2, t_2) of type e_2

and the timestamp t_1 of an event (e_2,t_2) of type e_2 fulfills $t^- \leq t_2 - t_1 \leq t^+$. A temporal constraint $e_1[t^-,t^+]e_2$ is called *satisfied* by a couple of events $((e,t),(e',t'))$ if $e_1 = e, e_2 = e'$ and $t' - t \in [t^-,t^+]$. Because constraining two events to occur in a fixed interval duration is too strict for most applications, the simpliest way to represent temporal constraints is by using duration intervals. These intervals can be interpreted as two constraints defining the lowest and highest accepted duration, respectively.

Using a set $\mathbb{E}$ of event types and a set $\mathscr{T}$ of temporal constraints, two complementary concepts can be introduced:

(i) If we are interested in finding events that pairwise satisfy a given set $\mathscr{T}$ of temporal contraints, then the concept of a *simple temporal constraint network* [12], alternatively called *simple temporal problem* [4] is useful, which can be defined as the triple

$$(\mathbb{E}, \prec, \mathscr{T}), \text{ where } \mathscr{T} \text{ is a set of temporal constraints } e[l,u]e', \text{such that } e,e' \in \mathbb{E}, e \prec e'. \quad (1)$$

(ii) If we are interested in mining temporal constraints from given sequences of events, then the concept of a *chronicle* [3, 9] is useful, which can be defined as the couple

$$(\mathscr{E}, \mathscr{T}), \text{ where } \mathscr{E} = \{\{e_1,\ldots,e_m\}\}, e_i \in \mathbb{E} \text{ and } \mathscr{T} \text{ is a set of temporal constraints } e[l,u]e' \text{ such that } (\exists i,j = 1,\ldots,m) i \neq j \,\&\, e_i \prec e_j \,\&\, e = e_i \,\&\, e' = e_j\}. \quad (2)$$

A chronicle is a temporal extension of episodes or partial orders introduced in [8], a type of pattern dedicated to summarize sequential data. Chronicles have proven their usefulness in applications where the temporal dimension is mandatory to differentiate two different behaviors. The first application of them is on alarm log data in [5] where the temporal distance between two alarm events is very important.
Observe that the constraint $e[-\infty,\infty]e'$ holds for any $e,e' \in \mathbb{E}$, meaning that this constraint actually does not constrain anything. In case that no constraint from $\mathscr{T}$ constrains anything, the set $\mathscr{T}$ will be denoted $\mathscr{T}_\infty$. Hence, $(\mathscr{E},\mathscr{T}_\infty)$ is a chronicle for the set of temporal constraints $\mathscr{T}_\infty$ of which, it is only required:

$$(\exists i,j = 1,\ldots,m) i \neq j \,\&\, e_i \prec e_j \,\&\, e = e_i \,\&\, e' = e_j. \quad (3)$$

Because the research reported in this paper concerns data mining, it relies on the latter concept, as well as on several additional concepts concerning chronicles.

Let $m,n \in \mathbb{N}, 2 \leq m \leq n, C = (\{\{e_1,\ldots,e_m\}\},\mathscr{T})$ be a chronicle and $s = ((e_1,t_1),\ldots,(e_n,t_n)), n \geq 2$, be a sequence of events. An *occurence* of C in s is a subsequence $\tilde{s} = ((e_{f(1)},t_{f(1)}),\ldots,(e_{f(m)},t_{f(m)}))$ of s such that:

(i) $f : \hat{m} \to \hat{n}$ is an injective function;

(ii) $e'_i = e_{f(i)}, i = 1,\ldots,m$;

(iii) if $i \neq j$ and $e'_i \prec e'_j$, then $t_{f(j)} - t_{f(i)} \in [a,b]$, where $e'_i[a,b]e'_j \in \mathscr{T}$.

We say that C *occurs* in s if there exists at least one occurrence of C in s.

Let further S be a set of sequences. The *support* of a chronicle C in S is the number of sequences from S in which it occurs:

$$\text{supp}(C,S) = \#\{s \in S | C \text{ occurs in } s\}. \quad (4)$$

If

$$\text{supp}(C,S) \geq \sigma_{\min} \quad (5)$$

for a given $\sigma_{\min} > 0$ or equivalently

$$\frac{\text{supp}(C,S)}{\#S} \geq f_{\min} \quad (6)$$

for a given $f_{\min} = \frac{\sigma_{\min}}{\#S}$, then C is called *frequent* in S on the level $f_{\min}$.

Finally, let S^+ and S^- be two disjoint sets of sequences. The *growth rate* of C for S^+ with respect to S^- is defined:

$$g(C,S) = \begin{cases} \frac{\text{supp}(C,S^+)}{\text{supp}(C,S^-)} & \text{if supp}(C,S^-) > 0 \\ +\infty & \text{if supp}(C,S^-) = 0. \end{cases} \quad (7)$$

If C is frequent and $g(C,S) \geq g_{\min}$ for a given *minimal growth rate* $g_{\min} \geq 1$, then C is called *discriminant* for S^+ with respect to S^- on the level $g_{\min}$.

The sequence sets S^+ and S^- can be viewed as two classes of their union $S = S^+ \cup S^-$. Hence, the algorithm *DCM* for discriminant chronicles mining presented in [3] is actually a sophisticated algorithm for extraction of classification rules. Before searching frequent chronicles satisfying some temporal constraints, it searches frequent chronicles with the set of temporal constraints $\mathscr{T}_\infty$, which is equivalent to the extraction of classification rules without temporal constraints. To this end, any rules extraction algorithm can be used. In the implementation of *DCM* in [3], the algorithm *Ripper*, based on the minimal descrition length principle, [2] has been employed.

3 Proposed Rules Extraction Method

3.1 Discriminant Multi-dimensional Chronicles

Let $d,n \in \mathbb{N}, d,n \geq 2$. For a set $\mathbb{E}$ of event types with a partial ordering $\prec$, $\mathbb{T} \subset \bar{\mathbb{R}}^d$ and a label set $\mathbb{L} = \{+,-\}$, an *event* is a couple (e,t), where $e \in \mathbb{E}, t \in \mathbb{T}$ and a *labelled sequence* of events is defined as a tuple $(SID,(e_1,t_1),\ldots,(e_n,t_n),L)$, where $SID \in \mathbb{N}$ is a *sequence index*, unique among all considered labelled sequences of events, $(e_1,t_1),\ldots,(e_n,t_n)$ are events, and $L \in \mathbb{L}$.

Let $\vec{a} = (a_1,\ldots,a_d), \vec{b} = (b_1,\ldots,b_d), \vec{c} = (c_1,\ldots,c_d) \in \overline{\mathbb{R}}^d$, with $b_i \geq a_i, i = 1,\ldots,d$, and $\mathscr{R}(\vec{a},\vec{b}) = [a_1,b_1] \times$

$[a_2,b_2]\times\ldots\times[a_d,b_d]$. The relation $\vec{c}\in\mathscr{R}(\vec{a},\vec{b}) \iff \forall\ i\ \in\ \hat{d}: c_i\in[a_i,b_i]$ will be called *hyperrectangle test*. A *hyperrectangle constraint* is a tuple $(e_1,e_2,\vec{t_1},\vec{t_2})$, also denoted as $e_1[\![\vec{t_1},\vec{t_2}]\!]e_2$ where $e_1,e_2\in\mathbb{E}$ and $\vec{t_1},\vec{t_2}\in\mathbb{R}^d$. A hyperrectangle constraint $e_1[\![\vec{t_1},\vec{t_2}]\!]e_2$ is said to be *satisfied* by a couple of events $((e,\vec{t}),(e',\vec{t'}))$ if and only if $e=e_1\ \&\ e'=e_2\ \&\ \vec{t'}-\vec{t}\in\mathscr{R}(\vec{t_1},\vec{t_2})$.

A *multi-dimensional chronicle* is a couple $(\mathscr{E},\mathscr{T})$ such that $\mathscr{E}=\{\{e_1,e_2,\ldots,e_n\}\}$, $e_i\in\mathbb{E}$, $i\in\hat{n}$ is a multiset of event types and $\mathscr{T}=m\{e_1[\![\vec{t_1},\vec{t_2}]\!]e_2|e_1,e_2\in\mathscr{E},e_1\prec e_2\}$ is a set of hyperrectangle constraints. If in particular all its constraints are $e[[(-\infty,\ldots,-\infty),(\infty,\ldots,\infty)]]e'$, i.e., they don't constraint anything, then this $\mathscr{T}$ is again denoted $\mathscr{T}_\infty$:

$$\mathscr{T}_\infty=\{e[[(-\infty,\ldots,-\infty),(\infty,\ldots,\infty)]]e'| \\ (\exists i,j\in\hat{m})\ i\neq j\ \&\ e_i\prec e_j\ \&\ e=e_i\ \&\ e'=e_j\}. \quad (8)$$

Let $s=((e_1,\vec{t}_1),\ldots,(e_n,\vec{t}_n))$ be a sequence of events, $m\in\hat{n}$ and $C=(\mathscr{E}=\{\{e'_1,e'_2,\ldots,e'_m\}\},\mathscr{T})$ be a multi-dimensional chronicle. An *occurrence* of the multi-dimensional chronicle C in s is a subsequence $\tilde{s}=((e_{f(1)},\vec{t}_{f(1)}),(e_{f(2)},\vec{t}_{f(2)}),\ldots,(e_{f(m)},\vec{t}_{f(m)}))$, such that $f:\hat{m}\longmapsto\hat{n}$ is an injective function, $\forall i: e'_i=e_{f(i)}$, and if $i\neq j$, then $\vec{t}_{f(j)}-\vec{t}_{f(i)}\in\mathscr{R}(\vec{a},\vec{b})$ where $e'_i[\![\vec{a},\vec{b}]\!]e'_j\in\mathscr{T}$. A multi-dimensional chronicle C is said to *occur* in sequence s if there exists at least one occurrence of C in s.

The *support* of a multi-dimensional chronicle C in a sequence set S is again defined by 2 like for chronicles in Section 2. Finally, also the definition of frequent chronicles and chronicles discriminant for one set of sequences with respect to another transfers to multi-dimensional chronicles.

3.2 Discriminant Multi-dimensional Chronicles Mining

The DCM-MD algorithm illustrated in Listing 1 is a modification of the DCM algorithm for discriminant chronicles mining proposed in [3]. The main aspects of the modification are the data model (substituting scalar integer values for vectors of real numbers) and a new discriminant hyperrectangle constraints mining algorithm (a substitution of an algorithm used for discriminant temporal constraints mining proposed in [3]). It operates with multi-dimensional input data and multi-dimensional chronicles, mining an incomplete set of discriminant multi-dimensional chronicles, determined by user-supplied argument values f_{min} (in the pseudocode as `fmin`) and g_{min} (in the pseudocode as `gmin`).

The branching statement in Listing 1 containing the condition

```
supp(S+,{m,tinf}) > (gmin*supp(S-,{m,tinf}))
```

is used to check whether given frequent multiset without further specific hyperrectangle constraints is discriminant (in the pseudocode, $\mathscr{T}_\infty$ is represented by the `tinf` symbol). If the given condition is true, no discriminant temporal constraints are mined using the `extractDC(...)` function.

```
DCM-MD(S+, S-, fmin, gmin):
M := extractMultiSet(S+,fmin). // M is a set of
                               // frequent multisets
C := emptySet(). // C is a set of resulting
                 // discriminant multi-dimensional
                 // chronicles

for (m of M):
  if supp(S+,{m,tinf}) > (gmin * supp(S-,{m,tinf})):
    C.add({m,tinf}). // adds a discriminant chronicle
                     // without temporal constraints
  else:
    for t of extractDC(S+,S-,m,fmin,gmin):
      C.add({m,t}). // adds a discriminant chronicle
                    // with temporal constraints

  return C.
```

Listing 1: DCM-MD pseudocode

The `extractMultiSet(...)` function extracts a set of frequent multisets from a given sequence set and user-supplied minimal support threshold (f_{min}). It applies a regular *frequent itemset mining algorithm* where an event type $a\in\mathbb{E}$ occurring n times in a sequence is encoded by n items $I^a_1,I^a_2,\ldots,I^a_n$. An intermediate *frequent itemset* of size m denoted as $(I^{e_k}_{i_k})_{1\leq k\leq m}$ is extracted from the supplied sequence set and is further transformed into the resulting multiset. The last phase of the algorithm incorporates converting each frequent itemset $(I^{e_k}_{i_k})_{1\leq k\leq m}$ to a multiset containing mutually different events $e_k,k=1,\ldots,m$, each of them exactly i_k times.

The `extractDC(...)` function is used to mine discriminant hyperrectangle constraints from a given frequent multiset $\mathscr{E}=\{\{a_1,a_2,\ldots,a_n\}\}$, disjoint sequence sets S^+ and S^-, and with user-defined parameters f_{min} and g_{min}. Exact conceptual and implementation details regarding the extraction of discriminant hyperrectangle constraints are further elaborated in [1].

4 Application to Crystal Growth Data

The need for affordable high quality semiconducting crystals such as gallium arsenide GaAs is continuously increasing, particularly for the electronic and photovoltaic applications. Despite GaAs has a number of outstanding physical properties, its production is hampered by challenging processes control due to high melting temperatures (1238°C) and chemically-aggressive environment. Particularly in-situ measurements of the process variables (e.g. temperatures, velocities, concentrations etc.) in the GaAs have high contamination potential and lead to the low crystal quality. Moreover, in-situ visual observations of the crystal growth are not possible. Prediction of the position of the crystallization front, i.e. length of the grown crystal after usage of certain growth recipe (i.e. temporal

profiles of a power of heaters) is a key information for the process monitoring.

Here, we considered Vertical Growth Freeze (VGF) method for the growth of GaAs crystals. VGF growth method involves the progressive freezing of the lower end of a melt upward by moving the desired temperature gradient in a furnace via temporal change of heating power. 1-dimensional model of VGF-GaAs growth is shown in Figure 1.

4.1 Used Data

The above described implementation extending the method proposed in [3] has been applied to data gathered in the German Research Foundation (DFG) project "Model-based control and regulation of the VGF crystal growth process using distributed parametric methods". The data records the *position of the solid/liquid interface* of GaAs crystals grown by the vertical gradient freeze (VGF) method, which involves progressive freezing of the lower end of a melt upward by moving the temperature gradient in a furnace, together with the *evolution of temperatures* in 0th–4th quarter of the GaAs height. They have been obtained by solving the inverse problem for a simplified one dimensional model of the VGF process for different desired growth rates as described in [7], using as input the evolution of 2-dimensional vectors describing the *heat flux in and heat flux out* (Figure 1). All simulations were performed for 100 times, among which the 5th, 10th, ..., 95th, 100th time will in the following serve as *milestone times*.

For an application of the method presented in Section 3, event types and events have been defined as follows. The 2-dimensional inputs of the 500 numeric simulations underlying [7] have been clustered into $k = 20$ clusters using the Matlab implementation of the standard k-means clustering algorithm. The centers of the resulting clusters are listed in Table 1. An *event type* is now the fact that the input belongs to a particular cluster. For each numeric simulation, an event type is recorded at every milestone time. Consequently, the size of any multiset of event types from one numeric simulation is at most 20. . An *event* is a pair (e,T), where e is an event type and $T \in \mathbb{R}^5$ is a vector of temperatures obtained in the numeric simulation and at the milestone time when e was recorded, provided the position of the solid/liquid interface at the end of that simuation was at least 17.25 cm. There were 255 such simulations available, thus we have 255 event sequences of length 20, due to the 20 milestone times. They were divided into two disjoint sequence sets as follows:

$$S^+ = \{((e_1,T_1),\ldots,(e_{20},T_{20})|e_i,T_i,i=1,\ldots,20, \text{ originated in a simulation ending with the position of the solid/liquid interface >25 cm)}\} \quad (9)$$

Table 1: Centers c of the 20 clusters in $\mathbb{R}^2$ defining event types. They were obtained through clustering the first 500 numeric simulations underlying [7] using the k-means algorithm in Matlab

Cluster	c_1	c_2	Cluster	c_1	c_2
A	13400	8560	K	14300	8720
B	12300	8690	L	11900	8650
C	15200	8840	M	12700	8660
D	13900	8590	N	12100	8670
E	13600	8730	O	13100	8720
F	14900	8800	P	13700	8550
G	13200	8570	Q	14100	8710
H	12900	8590	R	13300	8730
I	13800	8740	S	14600	8760
J	12500	8680	T	12900	8720

$$S^- = \{((e_1,T_1),\ldots,(e_{20},T_{20})|e_i,T_i,i=1,\ldots,20, \text{ originated in a simulation ending with the position of the solid/liquid interface 17.25–25 cm)}\} \quad (10)$$

As to the number of sequences in both sets, $\#S^+ = 90, \#S^- = 165$.

Finally, the considered partial ordering $\prec$ of event types is given by the order of ocurrence of events of those types in any of the event sequencesin S^+ or in S^-, i.e.,

$$e \prec e' \text{ iff } (\exists((e_1,T_1),\ldots,(e_{20},T_{20}) \in S^+ \cup S^-) \; (\exists i,j=1,\ldots,20)\; i<j \;\&\; =e_i \;\&\; e'=e_j. \quad (11)$$

4.2 Experimental Setup

The experimental setup aimed at a chronicle set containing about 20-30 elements and including both chronicles discriminant for S^+ with respect to S^- and chronicles discriminant for S^- with respect to S^+. Each chronicle $(\mathscr{E},\mathscr{T}) \in \mathbb{C}$ should contain only a minimal number of $\mathscr{T}_\infty$ constraints.

Assume that $C = (\mathscr{E},\mathscr{T})$ is a chronicle, $\mathbb{C}$ is a set of chronicles and $t_s \in [0,1]$. *Chronicle specificity* denoted as $\mathrm{s}(C)$ is defined as:

$$\mathrm{s}(C) = \frac{\#\{e[[t,t']]e' \in \mathscr{T} | e[[t,t']]e' \notin \mathscr{T}_\infty\}}{\#\mathscr{T}}.$$

Chronicle set $\mathbb{C}$ *specific for a specificity threshold* t_s denoted as $\mathrm{s}(\mathbb{C},t_s)$ is defined as $\mathrm{s}(\mathbb{C},t_s) = \{C|C \in \mathbb{C} \;\&\; \mathrm{s}(C) \geq t_s\}$.

The metrics used for evaluating the convenience of parameters passed to the *DC-PBC* component are described in the rest of this paragraph. $\#\mathbb{M}$ is the size of the set of frequent multisets set as introduced in the pseudocode of the `DCM-MD` algorithm in Listing 1. $\#\mathbb{E}$ is the count of distinct frequent multisets which occurred in some discrimi-

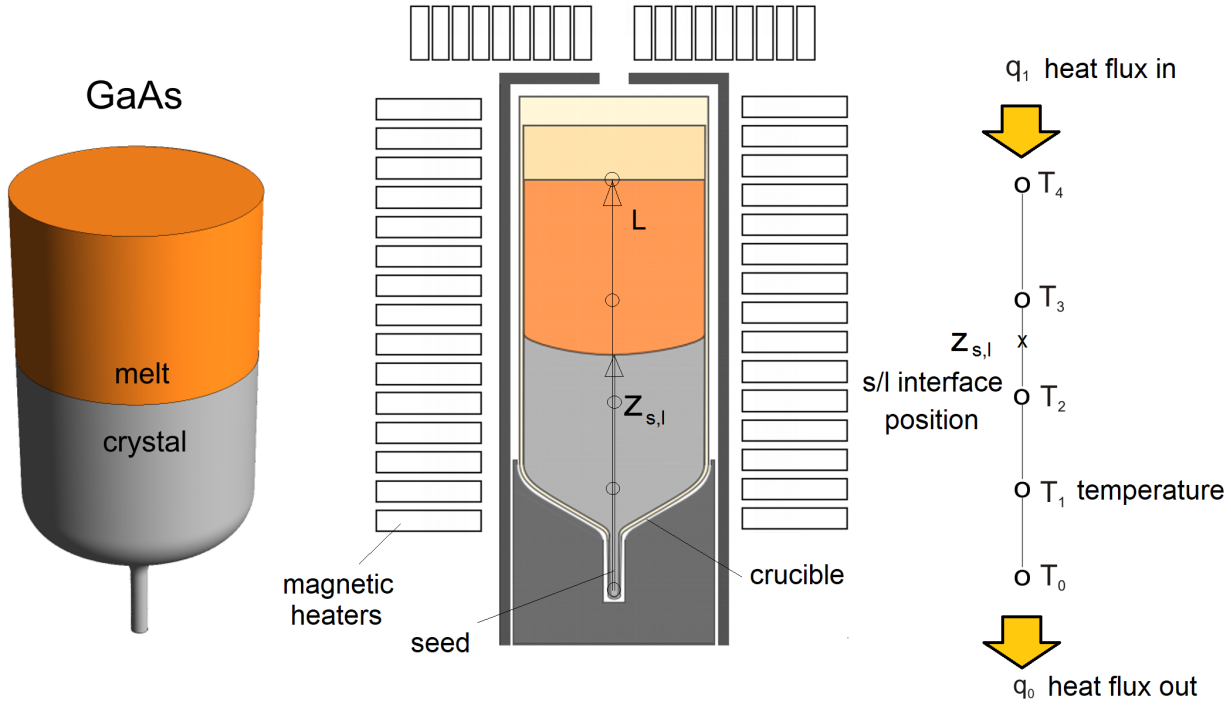

Figure 1: Illustration explaining the used crystal growth data

nant chronicle of the resulting chronicle set $\mathbb{C}$:

$$\#\mathbb{E} = \#\{\mathscr{E} | (\exists \mathscr{T} - \text{a set of hyperrecrtangle constraints}) (\mathscr{E}, \mathscr{T}) \in \mathbb{C}\}.$$

$\max s(\mathbb{C}) = \max\{s(C) | C \in \mathbb{C}\}$ is the maximal specificity value found among the chronicles in $\mathbb{C}$. $\#s(\mathbb{C}, t_s)$ is the count of chronicles specific for t_s found in $\mathbb{C}$.

The following parameters were tuned: f_{min} implemented by the `--fmin` parameter representing *minimal support threshold*. g_{min} implemented by the `--gmin` parameter representing the *minimal growth rate threshold* parameter of the `DCM-MD` algorithm as introduced in Listing 1. $\min(\#\mathscr{E})$ implemented by the `--mincs` parameter representing *minimal chronicle event multiset size*. $\max(\#\mathscr{E})$ implemented by the `--maxcs` parameter representing *maximal chronicle event multiset size*. t_s representing the *specificity threshold* for a custom tool implemented for extracting *specific discriminant chronicles* from a set of discriminant chronicles.

After evaluating the metrics for each parameter tuning step, the argument values $f_{min} = 0.1$, $g_{min} = 5000$, $\min(\#\mathscr{E}) = 2$, $\max(\#\mathscr{E}) = 5$, $t_s = 0.7$ proved sufficient for retrieving a set of specific discriminant chronicles with the desired properties.

4.3 Examples of Extracted Rules

The implementation of generalized *DC-PBC* available at `github.com/busarade-itat/md-dc-pbc` was invoked for the data described above with argument values `--mincs 2`, `--maxcs 5`, `--fmin 0.1`, `--gmin 5000`.

The resulting set of discriminant chronicles was afterwards filtered to include only specific discriminant chronicles. To this end, a tool `chronicle_statgen` available at `github.com/busarade-itat` was invoked with arguments `--minspec 0.7`, `--vecsize 5`.

The final result is presented in Tables 2 and 3, counting a total of 26 specific discriminant chronicles – 18 of them discriminant for S^+ with respect to S^-, the remaining 8 discriminant for S^- with respect to S^+.

Proposed method enables prediction of the conditions for reaching targeted crystal length by following the differences among segments in temporal profiles of temperatures in characteristic points in the GaAs. If the same approach is further applied on the experimental temperature profiles measured by thermocouples in heaters (outside of the melt and crystal) as in real experiments, it will be possible to determine moment of reaching desired crystal length without visual observations and GaAs contamination. From that moment on, crystal growth process step terminates and cooling down of the furnace starts. Such accurate prediction of the end of solidification step will be very beneficial for the process economy and the final crystal quality.

Table 2: Resulting set of specific chronicles discriminant for S^+ with respect to S^-, rounded to 3 significant digits

E(C)	T(C)	supp(C,S^+)	supp(C,S^-)
$\{\{e_1 = A, e_2 = A, e_3 = Q\}\}$	$\{e_1[\![(-\infty,-\infty,-\infty,-\infty,-\infty),(-33.7,-51.3,-68.9,-86.7,-105)]\!]e_2,$ $e_1[\![(-\infty,-\infty,-\infty,-\infty,-\infty),(73.4,73.5,73.7,73.8,74.0)]\!]e_3,$ $e_2[\![(-\infty,-\infty,-\infty,-\infty,-\infty),(159,160,161,162,180)]\!]e_3\}$	23	0
$\{\{e_1 = A, e_2 = I, e_3 = Q\}\}$	$\{e_1[\![(-\infty,-\infty,-\infty,-\infty,-\infty),(15.5,17.9,18.0,18.1,18.2)]\!]e_2,$ $e_1[\![(43.1,60.6,73.7,73.8,74.0),(\infty,\infty,\infty,\infty,\infty)]\!]e_3,$ $e_2[\![(-\infty,-\infty,-\infty,-\infty,-\infty),(144,145,146,146,163)]\!]e_3\}$	10	0
$\{\{e_1 = A, e_2 = I, e_3 = Q\}\}$	$\{e_1[\![(-\infty,-\infty,-\infty,-\infty,-\infty),(36.9,37.0,37.1,37.1,37.2)]\!]e_2,$ $e_1[\![(43.1,60.6,73.7,73.8,74.0),(\infty,\infty,\infty,\infty,\infty)]\!]e_3,$ $e_2[\![(-\infty,-\infty,-\infty,-\infty,-\infty),(144,145,146,146,163)]\!]e_3\}$	13	0
$\{\{e_1 = G, e_2 = A, e_3 = I\}\}$	$\{e_1[\![(-\infty,-\infty,-\infty,-\infty,-\infty),(-128,-128,-129,-130,-130)]\!]e_2,$ $e_1[\![(55.9,73.8,91.6,109,127),(\infty,\infty,\infty,\infty,\infty)]\!]e_3,$ $e_2[\![(398,412,414,416,418),(\infty,\infty,\infty,\infty,\infty)]\!]e_3\}$	13	0
$\{\{e_1 = G, e_2 = A, e_3 = I\}\}$	$\{e_1[\![(-\infty,-\infty,-\infty,-\infty,-\infty),(-374,-376,-378,-380,-382)]\!]e_2,$ $e_1[\![(-\infty,-\infty,-\infty,-\infty,-\infty),(88.6,107,125,144,162)]\!]e_3,$ $e_2[\![(398,412,414,416,418),(\infty,\infty,\infty,\infty,\infty)]\!]e_3\}$	13	0
$\{\{e_1 = G, e_2 = A, e_3 = Q\}\}$	$\{e_1[\![(139,140,141,142,142),(\infty,\infty,\infty,\infty,\infty)]\!]e_2,$ $e_1[\![(-\infty,-\infty,-\infty,-\infty,-\infty),(249,268,287,306,325)]\!]e_3,$ $e_2[\![(43.5,61.3,73.7,73.8,74.0),(\infty,\infty,\infty,\infty,\infty)]\!]e_3\}$	9	0
$\{\{e_1 = G, e_2 = I, e_3 = Q\}\}$	$\{e_1[\![(104,117,134,151,163),(\infty,\infty,\infty,\infty,\infty)]\!]e_2,$ $e_1[\![(-\infty,-\infty,-\infty,-\infty,-\infty),(189,205,222,239,256)]\!]e_3,$ $e_2[\![(-60.4,-60.5,-60.6,-60.7,-60.8),(\infty,\infty,\infty,\infty,\infty)]\!]e_3\}$	18	0
$\{\{e_1 = G, e_2 = Q, e_3 = Q\}\}$	$\{e_1[\![(-\infty,-\infty,-\infty,-\infty,-\infty),(249,267,286,305,324)]\!]e_2,$ $e_1[\![(60.4,78.3,96.3,114,132),(\infty,\infty,\infty,\infty,\infty)]\!]e_3,$ $e_2[\![(-32.5,-32.5,-32.6,-32.7,-32.7),(-31.8,-31.8,-31.9,-32.0,-32.0)]\!]e_3\}$	14	0
$\{\{e_1 = A, e_2 = A\}\}$	$\{e_1[\![(-135,-135,-136,-137,-137),(-127,-128,-128,-129,-130)]\!]e_2\}$	38	0
$\{\{e_1 = A, e_2 = I\}\}$	$\{e_1[\![(431,451,470,490,510),(\infty,\infty,\infty,\infty,\infty)]\!]e_2\}$	12	0
$\{\{e_1 = A, e_2 = K\}\}$	$\{e_1[\![(286,305,324,343,362),(\infty,\infty,\infty,\infty,\infty)]\!]e_2\}$	30	0
$\{\{e_1 = A, e_2 = Q\}\}$	$\{e_1[\![(372,391,411,430,449),(\infty,\infty,\infty,\infty,\infty)]\!]e_2\}$	21	0
$\{\{e_1 = G, e_2 = A\}\}$	$\{e_1[\![(-\infty,-\infty,-\infty,-\infty,-\infty),(-375,-377,-379,-381,-383)]\!]e_2\}$	16	0
$\{\{e_1 = G, e_2 = G\}\}$	$\{e_1[\![(-124,-125,-125,-126,-127),(-122,-123,-123,-124,-124)]\!]e_2\}$	8	0
$\{\{e_1 = G, e_2 = K\}\}$	$\{e_1[\![(26.5,43.7,44.6,44.8,45.1),(161,180,198,217,236)]\!]e_2\}$	17	0
$\{\{e_1 = I, e_2 = K\}\}$	$\{e_1[\![(72.5,72.7,72.8,73.0,73.1),(137,144,162,180,198)]\!]e_2\}$	12	0
$\{\{e_1 = I, e_2 = Q\}\}$	$\{e_1[\![(68.6,68.8,68.9,69.1,69.2),(69.0,69.1,69.3,69.4,69.5)]\!]e_2\}$	7	0
$\{\{e_1 = Q, e_2 = K\}\}$	$\{e_1[\![(-27.3,-27.4,-27.4,-27.5,-27.5),(\infty,\infty,\infty,\infty,\infty)]\!]e_2\}$	72	0

Table 3: Resulting set of specific chronicles discriminant for S^- with respect to S^+, rounded to 3 significant digits

E(C)	T(C)	supp(C,S^+)	supp(C,S^-)
$\{\{e_1 = G, e_2 = I, e_3 = Q\}\}$	$\{e_1[\![(-\infty,-\infty,-\infty,-\infty,-\infty),(80.3,80.5,80.6,80.7,80.9)]\!]e_2,$ $e_1[\![(-\infty,-\infty,-\infty,-\infty,-\infty),(50.2,50.2,50.3,50.4,50.5)]\!]e_3,$ $e_2[\![(-\infty,-\infty,-\infty,-\infty,-\infty),(67.4,67.5,67.6,77.0,93.3)]\!]e_3\}$	0	19
$\{\{e_1 = A, e_2 = Q\}\}$	$\{e_1[\![(12.1,12.1,12.1,12.1,12.1),(13.6,13.6,13.6,13.6,13.7)]\!]e_2\}$	0	13
$\{\{e_1 = G, e_2 = A\}\}$	$\{e_1[\![(316,333,351,368,385),(\infty,\infty,\infty,\infty,\infty)]\!]e_2\}$	0	16
$\{\{e_1 = G, e_2 = G\}\}$	$\{e_1[\![(-121,-122,-122,-123,-123),(\infty,\infty,\infty,\infty,\infty)]\!]e_2\}$	0	39
$\{\{e_1 = G, e_2 = I\}\}$	$\{e_1[\![(318,334,351,367,384),(\infty,\infty,\infty,\infty,\infty)]\!]e_2\}$	0	31
$\{\{e_1 = G, e_2 = Q\}\}$	$\{e_1[\![(393,411,429,447,465),(\infty,\infty,\infty,\infty,\infty)]\!]e_2\}$	0	18
$\{\{e_1 = I, e_2 = I\}\}$	$\{e_1[\![(-30.9,-31.0,-31.0,-31.1,-31.1),(\infty,\infty,\infty,\infty,\infty)]\!]e_2\}$	0	37
$\{\{e_1 = Q, e_2 = Q\}\}$	$\{e_1[\![(-30.8,-30.8,-30.9,-31.0,-31.0),(-30.1,-30.2,-30.2,-30.3,-30.3)]\!]e_2\}$	0	14

5 Conclusion

The paper has presented a generalization of the method for discriminant chronicles mining proposed in [3]. This generalization has been motivated by the objective to extract classification rules from crystal growth data, bringing two additional problems not pertaining to the data to which the original method had been applied: the events are described with a vector of attributes instead of a single scalar attribute, and the attributes are real-valued instead of integer-valued. The theoretical fundamentals of the method in [3] have been extended to tackle those two problems and the system for discriminant chronicles mining based on [3] has been adapted to accomodate those extensions, together with some additional implementation improvements such as refactoring. As a proof of concept of the presented generalization, it has been applied, using the modified system, to real-world data with events characterizing the heat fluxes for the growth of GaAs crystals by vertical gradient freeze method, and with a vector of 5 attributes recording the temperatures in different heights. Although most of the hyperrectangles in Tables 2 and 3 are not very restrictive, the extracted classification rules nevertheless show that the proposed approach allows to assess whether the grown crystal will have a desired length based solely on the temperature profiles. Regarding future research, it would be interesting to assess how small changes to the mined hyperrectangle constraints affect the manufacturing process of the VGF-GaAs crystals.

Acknowledgement

The research reported in this paper has been supported by the Czech Science Foundation (GAČR) grant 18-18080S.

References

[1] Radek Buša. Implementation of a generalized version of a system for discriminant chronicles mining. Czech Technical University in Prague. Computing and Information Centre., Cham, 2020.

[2] W.W. Cohen. Fast effective rule induction. pages 115–123, 1995.

[3] Yann Dauxais, Thomas Guyet, David Gross-Amblard, and André Happe. Discriminant chronicles mining. 2017.

[4] R. Dechter, I. Meiri, and J. Pearl. Temporal constraint networks. *Artifical Intelligence*, 49:61–95, 1991.

[5] Christophe Dousson and Thang Vu Duong. Discovering chronicles with numerical time constraints from alarm logs for monitoring dynamic systems. In *IJCAI*, pages 620–626, 1999.

[6] N. Dropka and M. Holeňa. Optimization of magnetically driven directional solidification of silicon using artificial neural networks and Gaussian process models. *Journal of Crystal Growth*, 471:53–61, 2017.

[7] N. Dropka, M. Holeňa, S. Ecklebe, C. Frank-Rotsch, and J. Winkler. Fast forecasting of VGF crystal growth process by dynamic neural networks. *Journal of Crystal Growth*, 521:9–14, 2019.

[8] Gemma C. Garriga. Summarizing sequential data with closed partial orders. In *SDM*, 2005.

[9] M. Ghallab, D. Nau, and P. Traverso. *Automated Planning and Acting*. Cambridge University Press, Cambridge, 2016.

[10] D.J. Hand. *Construction and Assessment of Classification Rules*. John Wiley and Sons, New York, 1997.

[11] M. Holeňa, P. Pulc, and M. Kopp. *Classification Methods for Internet Applications*. Springer, 2020.

[12] H.C. Lau, T. Ou, and M. Sim. Robust temporal constraint networks. In *International Conference on Tools with Artificial Intelligence*, pages 82–88, 2005.