Robust mining of time intervals with semi-interval partial order patterns

Fabian Moerchen, Dmitriy Fradkin
Siemens Corporate Research, Integrated Data Systems
755 College Road East, Princeton, NJ, 08540, USA
firstname.lastname@siemens.com

Abstract

We present a new approach to mining patterns from symbolic interval data that extends previous approaches by allowing semi-intervals and partially ordered patterns. The mining algorithm combines and adapts efficient algorithms from sequential pattern and itemset mining for discovery of the new semi-interval patterns. The semi-interval patterns and semi-interval partial order patterns are more flexible than patterns over full intervals, and are empirically demonstrated to be more useful as features in classification settings. We performed an extensive empirical evaluation on seven real life interval databases totalling over 146k intervals from more than 400 classes demonstrating the flexibility and usefulness of the patterns.

1 Introduction

Temporal data mining is aimed at exploiting temporal information in data sources to improve performance of clustering or classification algorithms or find models that describe the data generating process or patterns that describe local effects. Many data sources under study in business, health-care and scientific applications are dynamic in nature, making them promising candidates for application of temporal mining methods. For an overview of methods to mine time series, sequence, and streaming data see [15, 9].

Sequential patterns [1] are typically extracted from databases with sequences of (sets of) discrete items associated with time stamps. In this study we concentrate on sequences of time intervals with discrete labels that could be observed directly or obtained from numerical time series using abstraction mechanisms such as discretization, segmentation, clustering, or state estimation. When dealing with time interval patterns, the formulation of patterns is much more intricate, because the number of possible binary relations rises from three relations for time points (before, equals, after) to Allen's 13 interval relations [3]. For semi-intervals Freksa identified 10 core relations [11] and, by adding interval to interval mid-point relations, Roddick obtained 49 relations [36]. Pattern mining in interval data has relied almost exclusively on Allen's interval relations.

Many authors have identified problems in the use of Allen's relations for mining patterns from observational data [30, 28, 42]. The relations are not robust to noise because small shifts of time points lead to different patterns describing similar situations observed in the data. The pattern representation is ambiguous because the same pattern can describe quite different situations in the data.

In this work we propose the use of semi-intervals and partial orders as a solution to more flexible matching of interval patterns. This approach has several novel and attractive features:

- The patterns can include complete intervals or only the starting and ending time point expressing a *mixture of intervals and semi-intervals*. By relaxing the constraint that the complete interval must be observed the patterns are more flexible in matching similar situations in the data. This has not been done before and we show that not only more patterns are found but that they are *more predictive on datasets with ground truth labeling of the sequences*.

- The patterns support a *partial order of interval boundaries*. In contrast to sequential patterns, partial orders allow some binary relations among elements of the pattern to be unspecified. This is an elegant way of expressing disjunctions of Allen's relations between intervals in a pattern that required a priori definitions of disjunctive sets of relations in previous work [30, 17].

- The pattern language can naturally incorporate instantaneous events and thus represent patterns that mix intervals, semi-intervals, and time points. This would require allowing degenerate intervals of

unit length in other interval pattern representations [18, 28].

The novelty of our approach is the unified formulation of patterns that can represent intervals, semi-intervals, and time points without the need to explicitly model the complex temporal relations [3, 11, 36]. Previous work has focused almost exclusively on patterns composed of complete intervals [30, 29, 41, 42]. Only [34] has previously used Freksa's semi-interval relations in the data mining context for annotating association rules, but has not considered interval to point or point to point relations that are all covered by our framework. We utilize The proposed interval boundary representation of the data [42] to apply well-developed existing algorithms [5, 32, 35] advancing the field by exploiting previous work.

Extensive experiments are performed on 7 databases of symbolic interval sequences - the largest set of real life interval data we are aware of in the literature. The results highlight the benefits of using semi-intervals over complete interval patterns. We show that a lot more of semi-interval patterns are found and that they provide added value for classification problems.

We include some background information and related work in Section 2. Our novel approach with relevant algorithms is presented in Section 3. Extensive experiments with real data are described in Section 4 comparing different interval pattern representations. The results are discussed in Section 5 before we conclude in Section 6.

2 Related work

For the purpose of temporal reasoning, Allen formalized temporal logic on intervals by specifying 13 interval relations [3] and showing their completeness. Any two intervals are related by exactly one of the relations. The operators are: *before*, *meets*, *overlaps*, *starts*, *during*, *finishes*, the corresponding inverses *after*, *met by*, *overlapped by*, *started by*, *contains*, *finished by*, and *equals* (see Figure 1). These relations are commonly used beyond temporal reasoning, e.g., for the formulation of temporal patterns but this can be problematic, in particular for noisy data where the exact interval boundaries are not reliable or meaningful. The relations are not robust to noise because small shifts of time points lead to different relations for similar situations observed in the data [28], see Figure 2 for an example. Researchers have attempted to remedy this problem by using thresholds [2, 30, 29], fuzzy extensions for temporal reasoning (see [37] and references therein) and different pattern languages that group some of the relations [30, 17] or match against sub-intervals of observed intervals [28]. Our pattern format addresses potential noise in interval boundaries by allowing a partial order of the time points and by allowing interval boundaries to be missing from the pattern. In Figure 3 we give an example of the partial order aspect. The pattern describes all three similar situations in Figure 2 and is a disjunction of the Allen relations overlaps, starts (not shown), during, and finishes. Many different ways of representing patterns of complete intervals using the binary interval relations of Allen have been proposed. Early approaches [20, 8] that used nested combinations of binary relations were shown to be ambiguous [25, 42, 31]. The format of Hoeppner [18], which uses the $\frac{k(k-1)}{2}$ pairwise relations of all intervals in a pattern, is concise and has been adopted by recently proposed efficient algorithms [30, 41, 29]. Equivalent patterns are represented in [42] as a sequence of $2k$ interval boundaries. In [31] nested binary relations are annotated with counter variables, indicating how many intervals of a subpattern interact with an interval joined with a binary relation in different ways.

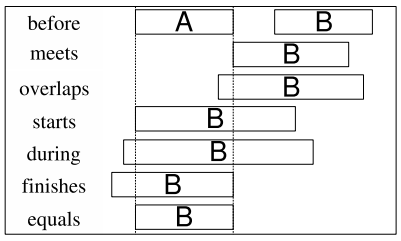


Figure 1: Examples of Allen's interval relations between the intervals A and B. The first six can be inverted.

The different representations require specialized data structures and algorithms for finding patterns expressed by Allen's relations. Early algorithms for mining patterns based on Allen's relations were based on the Apriori principle of building longer patterns by combining frequent short ones [20, 8, 18]. In [18] the transitivity of the relations was used to reduce the number of candidates generated. More recent algorithms use depth-first search strategies with efficient data structures such as enumeration trees [30, 29], prefix trees [42, 21] and bitmaps [41].

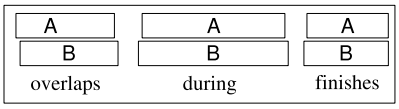


Figure 2: Examples for different patterns according to Allen that are fragments of the same approximate relation *almost equals*.

Approaches that do not use Allen's relations for interval mining include containment patterns [39], the UTG with sequence of blocks of almost equal intervals [13], and the TSKR with partial orders of blocks of concurrent sub-intervals [28].

All the above are qualitative intervals patterns. Based on research on quantifying the typical duration of gaps in sequential patterns [44, 6, 12, 16] quantitative interval patterns [14] represented with a symbolic signature indicating the interval labels and a numerical vector representing the temporal positions and differences of the time points.

Algorithms for time interval mining have been inspired by methods for mining time point data, mainly sequential pattern mining [1, 45, 26, 9] where the elements of a pattern have a strict sequential ordering. Episode [23] patterns offer more flexibility allowing some elements of patterns to happen concurrently but in no particular order. In [33] closed partial orders without repeating symbols are mined using itemset mining algorithm on the set of partial order graph edges. In [5] a method to mine closed partial orders (including repeating symbols) from itemset sequences by grouping and merging sequential patterns is presented. This was generalized in [35] to conjunctive groups of sequential patterns that may be closures or generators of an equivalence class.

3 Semi-interval mining

We propose a unified approach to mining interval patterns. The core ideas, in contrast to patterns expressed with Allen's relations, are the admission of semi-intervals and partial order. Both are motivated by the desire to allow more flexible matching of patterns against observations in the data. Pattern languages are used to describe similar situations in observed data in an abstract way. A language that, with a single expression, can match more situations in the data has obvious advantages for data mining. Such a language allows patterns to be found at larger minimum support thresholds, while in less descriptive languages these patterns would be fragmented into less frequent, and thus potentially pruned, patterns. This has been observed on real life interval data using Allen's relations [28].

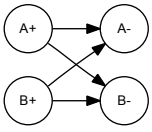


Figure 3: Partial order of interval boundaries representing all three situations of Figure 2.

In the experimental section, we demonstrate that the additional patterns found using our new representation are more useful than Allen patterns in distinguishing ground truth classes of interval sequences. The main influences leading to our proposed framework are listed below.

- **Disjunctive Allen patterns**: Disjunctions of binary relations have been used to express the relation between two intervals in a pattern [30, 17]. When representing Allen patterns with a pairwise matrix of binary relations, some cells are thus not filled with a single relation, but a set of possible relations. A generalization of the previous work would be to allow the set of all relations in a cell, essentially leaving this relation unspecified. This is a possible way of defining a partial ordering of complete intervals using Allen relations similar to the previously used partial orders for time points [5, 32] but it is unclear how existing algorithms would mine such patterns. *By using a time point based representation of time intervals we can apply existing methods for mining partial orders of time points to intervals.*

- **Freksa patterns**: While Allen's relations have been used widely in data mining, Freksa's interval relations were only used in [34] to annotate association rules. This may be due to the lack of semi-interval data as a common use case. Of course Freksa's relations could be applied to complete interval data splitting each interval into two semi-intervals, but this would result in complicated representations of patterns including many complete intervals. *Our representation seamlessly integrates complete intervals, semi-intervals, and even instantaneous time points in a single representation.*

- **Interval boundary representation**: In [42] the use of an interval boundary representation was proposed for mining Allen patterns. The TPrefixSpan algorithm mines frequent patterns composed of complete intervals. *We mine the new class of semi-interval patterns using the same data representation but different with algorithms that find partial orderings and closed patterns.*

In the remainder of this section we define the new classes of patterns, compare their properties with existing approaches with motivating examples, and describe algorithms for efficient mining.

3.1 Definitions In temporal data mining, input data is usually measured at discrete time points of a certain resolution, representing a sample of the generating time

continuous process. Without loss of generality, we make the following definitions based on the natural numbering T of a set of uniformly spaced time points. We begin by defining the data structures that the patterns operate on.

DEFINITION 3.1. *Let the* alphabet Σ *be a set of unique* symbols.

DEFINITION 3.2. *An* itemset *is a subset* $S = \{\sigma_1, ..., \sigma_k\} \subseteq \Sigma$ *of the alphabet.*

DEFINITION 3.3. *A* time interval *is a tuple* $[s, e]$ *with* $[s, e] \in T^2$, $s \leq e$. *The duration of an interval is* $d([s, e]) = e - s + 1$. *The finite set of all time intervals is noted* $I = \{[s, e] \in T^2 | s \leq e\}$.

DEFINITION 3.4. *We define an* order $<$ *of intervals as* $[s_1, e_1] < [s_2, e_2] \Leftrightarrow s_1 < s_2 \vee (s_1 = s_2 \wedge e_1 < e_2)$. *We say that* $[s_1, e_1]$ *is* before $[s_2, e_2]$.

DEFINITION 3.5. *A* symbolic interval *is a triple* $[\sigma, s, e]$ *with* $\sigma \in \Sigma$, $[s, e] \in I$. *For example,* $[$*temperature high*$, 12, 78]$ *describes a state observed starting at time point 12 and lasting until time point 78, inclusively. If* $\{s, ..., e\} \cap \{s', ..., e'\} \neq \emptyset$ *we say that the intervals* $[\sigma, s, e]$ *and* $[\sigma', s', e']$ *overlap.*

DEFINITION 3.6. *An* interval sequence *is an ordered sequence of symbolic intervals* $\mathcal{I} = \{[\sigma_i, s_i, e_i] | \sigma_i \in \Sigma; [s_i, e_i] \in I; i = 1, ..., N; [s_i, e_i] < [s_j, e_j] \forall i < j; [s_i, e_i] = [s_j, e_j] \Leftrightarrow \sigma_i \neq \sigma_j \vee i = j\}$.

DEFINITION 3.7. *An* interval sequence database *is a finite set of interval sequences* $\mathcal{D} = \{\mathcal{I}_i | i = 1, ..., M\}$.

The following definition describes standard properties of any sequential pattern [40]:

DEFINITION 3.8. *The* support *of a pattern is the number of sequences in an interval sequence database that contain the pattern. A pattern is called* frequent *if the support is greater or equal to a given minimum support threshold. A pattern is called* closed *if it cannot be extended with additional elements without decreasing the support.*

All definitions up to this point are standard in the literature, e.g., [23, 40]. The following definition for semi-interval representation of interval data is based on [42].

DEFINITION 3.9. *Let* $\Sigma' = \{\sigma_i^+, \sigma_i^- | i = 1, ...k\}$ *be the* extended alphabet *in which each symbol* σ *from alphabet* Σ *is replaced with* σ^+ *and* σ^- *representing start and end of a symbolic interval with symbol* σ.

Note that the extended alphabet could further include symbols representing instantaneous events if they are available in the data.

DEFINITION 3.10. *A* symbolic semi-interval *is a tuple* $[\sigma, t]$ *with* $\sigma \in \Sigma'$, $t \in T$.

DEFINITION 3.11. *A* semi-interval sequence *is an ordered sequence of itemsets over an extended alphabet with timestamps* $\mathcal{S} = \{[S_i, t_i] | S_i \subseteq \Sigma', t_i < t_j \forall i < j\}$.

DEFINITION 3.12. *A* semi-interval sequence database *is a finite set of semi-interval sequences* $\mathcal{D}' = \{\mathcal{S}_i | i = 1, ..., M\}$.

Based on the above definitions of data models, we introduce two novel pattern classes:

DEFINITION 3.13. *A* semi-interval sequential pattern *(SISP) is a sequence of itemsets over an extended alphabet* $\mathcal{P} = \{S_i | S_i \subseteq \Sigma', i = 1, ..., k\}$. *A SISP is* contained *in an semi-interval sequence* $\{[S'_j, t_j]\}$ *iff* $\exists j_1 < ... < j_k$ *with* $S_i \subseteq S'_{j_i}$ *for* $i = 1, ..., k$.

The definitions and methods hold for larger itemsets representing intervals with exactly the same endpoints, e.g., *A starts B* represented by $\{A^+, B^+\}\{A^-\}\{B^-\}$. For readability we will only consider itemsets of size one in the following examples and drop the set notation.

EXAMPLE 3.14. *Suppose we have a semi-interval sequence* S=$C^+A^+C^-B^+B^-A^-$. *Then patterns* p_1=A^+C^-, p_2=C^+C^- *and* p_3=$C^+A^+A^-$ *are some of the SISPs that are contained in (match)* S. *We can see that SISPs can include only complete intervals (*p_2*), or only semi-intervals (*p_1*), or a mix thereof (*p_3*).*

DEFINITION 3.15. *A* semi-interval partial order pattern *(SIPO) is a partial order of itemsets over an extended alphabet represented by an directed acyclic graph with nodes* $\mathcal{N} = \{S_i | S_i \subseteq \Sigma', i = 1, ..., k\}$ *and edges* $\mathcal{E} = \{(i, j) | S_i$ *precedes* S_j *in the SIPO* $\}$. *A SIPO is* contained *in a semi-interval sequence* $\{[S'_j, t_j]\}$ *iff* $\exists j_1, ..., j_k$ *with* $S_i \subseteq S'_{j_i}$ *for* $i = 1, ..., k$ *and* $(a, b) \in \mathcal{E} \Rightarrow t_{j_a} < t_{j_b}$.

Figure 4 shows an example of a SIPO. Unlike a SISP, where itemsets are completely ordered in a sequence, in a SIPO some order relations are not specified: i.e. A^+ occurs before B^+ which occurs before B^-, as does C^-. However, the order relation of C^- to A^+ and B^+ is not specified - it can occur before or after or at the same time as either of these events.

COROLLARY 3.16. *For two intervals* $A = [A^+, A^-]$ *and* $B = [B^+, B^-]$ *the relations according to Allen can be expressed via semi-interval representation [11]:*

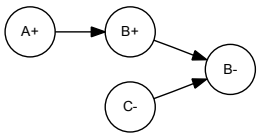


Figure 4: An example of a SIPO pattern.

- *A before B* $\Leftrightarrow A^- < B^+$
- *A overlaps B* $\Leftrightarrow A^+ < B^+ \wedge B^+ < A^- \wedge A^- < B^-$
- *A during B* $\Leftrightarrow B^+ < A^+ \wedge A^- < B^-$
- *A meets B* $\Leftrightarrow A^- = B^+$
- *A starts B* $\Leftrightarrow A^+ = B^+ \wedge A^- < B^-$
- *A finishes B* $\Leftrightarrow A^- = B^- \wedge B^+ < A^+$
- *A equals B* $\Leftrightarrow A^+ = B^+ \wedge A^- = B^-$

and the inverse of the first six analogously (See also Figure 1).

Definition 3.17. *An* Allen pattern *is a SISP* $\{S_i | i = 1, ..., k\}$ *with the following properties*

- $\forall \sigma^+ \in S_i \; \exists j \geq i$ *with* $\sigma^- \in S_j$ *and* $\sigma^+ \notin S_l$ *for* $l = i+1, ..., j$
- $\forall \sigma^- \in S_i \; \exists j \leq i$ *with* $\sigma^+ \in S_j$ *and* $\sigma^- \notin S_l$ *for* $l = j, ..., i-1$

i.e., every interval is represented with both boundaries and no two intervals with the same symbol overlap or meet. An Allen pattern is contained *in a semi-interval sequence* $\{[S'_j, t_j]\}$ *iff* $\exists j_1 < ... < j_k$ *such that:*

1. $S_i \subseteq S'_{j_i}$ *for* $i = 1, ..., k$*; and*
2. *If* $\sigma^+ \in S_i, \sigma^+ \notin S_l, l = i+1, ..., m$ *and* $\sigma^- \in S_m, \sigma^- \notin S_l, l = i, ..., m-1$*, then* $\sigma^+ \notin S'_{j_l}, l = i+1, ..., m$ *and* $\sigma^- \notin S'_{j_l}, l = i, ..., m-1$.

i.e., a pattern forms a subsequence of a semi-interval sequence, and corresponding interval endpoints in the pattern match to endpoints of a single interval in the sequence.

To make the above definition more clear, we show several examples:

- $A^+B^+A^-B^-$ is an Allen pattern (A overlaps B).
- $A^+B^+A^-$ is not an Allen pattern because interval B is not closed (i.e. B^- is missing). Similarly, $B^+A^-B^-$ is not an Allen pattern because A^+ is missing.
- $A^+B^+A^+A^-A^-B^-$ is not an Allen pattern, even though there are two A^+ before two A^- the relation between them is not well defined (one could overlap the other or one could be during the other).
- $A^+B^+A^-A^+A^-B^-$ is an Allen pattern.
- An Allen pattern $p_1 = A^+B^+A^-B^-$ is contained in sequence $S_1 = A^+C^+B^+A^-B^-C^-$, but is not contained in $S_2 = A^+A^-C^+B^+A^+A^-B^-C^-$ because, while p_1 is a subsequence of S_2 (Condition 1 of Definition 3.17 is satisfied), interval A from the pattern is broken into two interval in the sequence S_2, neither of which overlaps interval B, i.e., Condition 2 is not satisfied.

3.2 Comparison In this section we compare the above pattern classes with existing approaches to interval pattern mining.

The data representation we use was proposed in [42] for expressing interval patterns representing Allen's relations. Our Allen patterns are slightly different. In [42] a pattern is represented as chain of interval boundaries connected with binary relations (precedes and equals). Our definition of Allen uses the well known sequential pattern format expressing equality of the timestamp via with itemsets of symbols. In both cases it is ensured that each interval is represented with start and end point.

Similar to [42], it is easy to show that our definition of an Allen pattern can be represented using Allen's relations and the other way around: Consider k intervals with $\frac{k(k-1)}{2}$ binary relations according to Allen. Each Allen relation is defined using one to three binary time point relations (smaller or equal) between interval boundaries (see Definition 3.16). All four relations between the boundaries of the two intervals can be easily derived. We thus have $2k$ interval boundaries and know all pairwise point relations. We group all equal interval boundaries into sets and order the sets such that all inequality relations are preserved obtaining an Allen pattern according to Definition 3.17. This is possible because all interval relations were specified. In reverse, given an Allen pattern in the SISP format (Definition 3.17) we can look up the relation between any two intervals using Definition 3.16.

In contrast to Allen patterns, the novel SISPs and SIPOs do not require both endpoints of an interval to be included in a pattern. This allows for more flexible matching of situations where one boundary of an interval has a common relative positioning to other (semi-)intervals but the relation of the other boundary differ. An example is shown in Figure 5. Interval A always starts before C ends and B is observed

during this time. The duration of B varies among the three examples, causing the relations according to Allen between B and the other intervals to differ. Considering only the start point of A and the end point of C, the SIPO $B^+A^+C^-B^-$ can match all three examples.

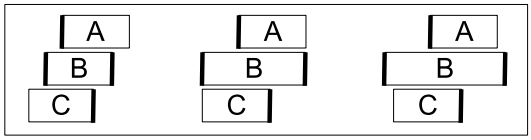


Figure 5: Three instances of the SIPO pattern $B^+A^+C^-B^-$ marked with bold interval boundaries.

The same three situations are difficult to capture with a single pattern using thresholds on Allen's relations. Starting with the center example, the relations would be A finishes B, B starts C, and C overlaps A. Considering close-by interval boundaries as equal using a threshold the same could be said for the left example. On the right, however, we can see a problem with modified relations based on thresholds. The distance between the start points of C and B is about the same as the distance between the start points of A and B. A threshold large enough to obtain B starts C would change the relation between A and B to equals (the threshold would consider both the start and end points of A and B close enough to be equal). This example highlights another problem with thresholds: in patterns with more than 2 intervals they can lead to inconsistencies. From B starts C and A equals B we would expect to be able to deduct A starts C using the transitivity rules of Allen's relations but the difference between the start points of A and C may be up to twice the threshold. Using the TSKR that breaks time intervals into subintervals, all three examples could be matched by the pattern BC, ABC, AB but a minimum duration threshold on the coincidence of intervals must be chosen.

Furthermore, SIPO can represent a partial ordering of interval endpoints. For time point data this concept has been proposed with episodes [23] and partial order patterns [33], but for interval data it was only indirectly used by previous authors through disjunctions of relations [30, 17]. For each disjunction the transitivity rules of Allen's relations [3] could be used to narrow the possibilities and this corresponds to the automated reasoning task the relations were designed for in the first place. Using our semi-interval representation this exercise is much easier, because no reasoning is required. This corresponds to Freksa's observation that *"in no case, more than two relations between beginnings and endings of events must be known for uniquely identifying the relation between the corresponding events"* [11].

A single timepoint event A can be represented as single itemset $(A+, A-)$. It is then handled naturally by our approach described in Section 3.3.

In summary, we conclude that SISPs and SIPOs are a very elegant approach to make interval patterns more flexible. Allowing interval boundaries to be missing from the patterns enables algorithm to discover situations in the data that cannot be represented by Allen patterns. In addition, SIPOs require only a partial ordering of interval boundaries. Depending on the point to point relations of the interval boundaries, two intervals can be completely unrelated, partially related via Freksa's semi-interval relations, or fully related via Allen's relations.

3.3 Algorithms Algorithm 1 shows the main steps required to mine SISPs and SIPOs from an interval sequence database $\mathcal{D}$. We will describe these steps in detail in this section, refering to existing algorithms where a sub-problem is equivalent to a well known data mining problem. In Line 1 the interval sequence

Algorithm 1 Algorithm for mining SISPs and SIPOs from an interval sequence database.

Require: Interval sequence database $\mathcal{D}$ and minimum support $\alpha \in N$

1: Convert $\mathcal{D}$ into semi-interval sequence database $\mathcal{D}'$
2: Find closed SISPs S_i in $\mathcal{D}'$ using a closed sequential pattern mining algorithm.
3: Find closed groups of SISPs using a closed itemset mining algorithm.
4: Merge each group of SISPs into a graph representing a closed SIPO.

database $\mathcal{D}$ is converted into a semi-interval sequence database $\mathcal{D}'$ by converting each sequence $\mathcal{I} \in \mathcal{D}$ as follows:

- Let $SI = \{[\sigma^+, s], [\sigma^-, e] | [\sigma, s, e] \in \mathcal{I}\}$ be all semi-intervals in the interval sequence.
- Let $T = \{s, e | [\sigma, s, e] \in \mathcal{I}\}$ be all unique time stamps in the sequence.
- Let $\mathcal{I}' = \{[S, t] | t \in T, \sigma \in S \Leftrightarrow [\sigma, t] \in SI\}$ be the semi-interval sequence.

In Line 2 a closed sequential pattern mining algorithm, such as BIDE [40], is used to find all closed SISPs with support greater than or equal to α. For each SISP the list of sequences where it occurs should be recorded to support the next mining step.

In Line 3 a closed itemset mining algorithm, such as DCI_Closed [22], is used to find closed groups of SISPs that occur in exactly the same sets of sequences in $\mathcal{D}'$

with support greater than or equal to α. Each SISP S_i is interpreted as an item and each sequence in $\mathcal{D}'$ as an itemset represented by the S_i it contains.

Finally, in Line 4 a partial order over semi-intervals (SIPO) is constructed from each set of SISPs. The construction is based on treating each sequential pattern as a graph, where sets in a sequence are nodes and consecutive nodes are connected with edges. Then the path preserving property [5] is used to find matching positions among individual sequences S_j and merge them [28].

The correctness and completeness of Algorithm 1 follow directly from previously published results on the involved algorithms: BIDE efficiently finds all closed sequential patterns [40] and all closed groups of sequential patterns (closed itemsets efficiently found by DCI_Closed) correspond to all closed partial orders [5].

4 Experiments

We performed experiments on real life data sets, comparing semi-interval patterns (SISPs and SIPOs) with Allen patterns to evaluate two hypotheses:

1. Semi-interval patterns that ignore some interval boundaries *are found in real life data and not superseded by their corresponding Allen patterns* that would include both boundaries for each interval. We measure the total number of patterns given the same minimum support threshold. *We show that significantly more semi-interval patterns than Allen patterns are found in many cases demonstrating that our pattern class is more flexible in describing local structure in real life data.*

2. The semi-interval patterns that ignore some interval boundaries are more useful for data mining tasks. The measure of success is the predictive power of patterns based on ground truth classification of the interval sequences. *We show that semi-interval patterns can better discriminate classes than Allen patterns in many cases.*

Since there is no efficient algorithm for mining closed Allen patterns, we mined the patterns using a step wise approach mining frequent SISP, reducing the result to Allen patterns using Definition 3.17 and applying a brute force closedness check. No runtime or memory experiments were made at this point because the efficiency of our algorithms is inherited from BIDE and DCI_Closed and a comparison with the brute-force approach for mining closed Allen patterns would not be fair.

4.1 Data We evaluated the temporal patterns using the seven datasets with interval data summarized in Table 1. To the best of our knowledge this is largest set of real life interval data used in pattern mining research yet. The data in interval format is available by contacting the first author; its origin and preprocessing steps are described in this section.

Data	Intervals	Labels	Sequences	Classes
ASL-BU	18250	154	441	7
ASL-GT	89247	47	3493	40
Auslan2	900	12	200	10
Blocks	1207	8	210	8
Context	12916	54	240	5
Pioneer	4883	92	160	3
Skating	18953	41	530	6/7

Table 1: Interval data: Seven databases consisting of many sequences of labeled intervals with class labels for each sequence.

ASL-BU[1] The intervals are transcriptions from videos of American Sign Language expressions provided by Boston University [30]. It consists of observation interval sequences with labels such as *head mvmt: nod rapid* or *shoulders forward* that belong to one of 7 classes like *yes-no question* or *rhetorical question*.

ASL-GT The intervals are derived from 16 dimensional numerical time series with features derived from videos of American Sign Language expressions [38]. The numerical time series were discretized into 2-4 states each using Persist [27]. Each sequence represents one of 40 word like *brown* or *fish*.

Auslan2 The intervals were derived from the high quality Australian Sign Language dataset in the UCI repository [4] donated by Kadous [19]. The x,y,z dimensions were discretized using Persist with 2 bins, 5 dimensions representing the fingers were discretized into 2 bins using the median as the divider. Each sequence represents a word like *girl* or *right*.

Blocks[2] The intervals describe visual primitives obtained from videos of a human hand stacking colored blocks provided by [10]. The interval labels describe which blocks touch and the actions of the hand (*contacts blue red*, *attached hand red*). Each sequence represents one of 8 different scenarios from atomic actions (*pick-up*) to complete scenarios (*assemble*).

Context[3] The intervals were derived from categoric and numeric data describing the context of a mobile device carried by humans in different situations [24]. Numeric sensors were discretized using 2-3 bins chosen

[1] http://www.bu.edu/asllrp/

[2] ftp://ftp.ecn.purdue.edu/qobi/ama.tar.Z

[3] http://www.cis.hut.fi/jhimberg/contextdata/index.shtml

manually based on exploratory data analysis. Each sequence represents one of five scenarios such as *street* or *meeting*.

Pioneer The intervals were derived from the Pioneer-1 datasets in the UCI repository [4]. The numerical time series were discretized into 2-4 bins by choosing thresholds manually based on exploratory data analysis. Each sequence describes one of three scenarios: *gripper*, *move*, *turn*.

Skating The intervals were derived from 14 dimensional numerical time series describing muscle activity and leg position of 6 professional In-Line Speed Skaters during controlled tests at 7 different speeds on a treadmill [28]. The time series were discretized into 2-3 bins using Persist and manually chosen thresholds. Each sequence represents a complete movement cycle and is labeled by skater or speed.

4.2 Numerosity By definition, the number of SISPs is always greater than or equal to that of Allen patterns. Figure 6 shows the number of patterns found by the different methods using different support thresholds. For almost all datasets and minimum support values the number of SISPs is much larger than the number of Allen patterns demonstrating that our relaxed pattern representation uncovers structure in the data that would otherwise not be found. Only for large minimum support values on ASL-BU and Auslan2 and most minimum support values on ASL-GT the numbers are very close, indicating absence of a significant amount of semi-interval patterns that do not include complete intervals. The numbers of SISPs and SIPOs are often comparable. Either one can be larger: Several SISPs could be grouped into a SIPO without loss of frequency reducing the number of patterns. However, when many SIPOs (which are conjunctive combinations of SISPs) have lower frequencies more patterns are observed. The results show that plenty of non-degenerate examples of both newly proposed patterns are found in real-life data.

4.3 Predictiveness Patterns obtained by unsupervised mining can be used for knowledge discovery by ranking and analyzing them directly, for generation of temporal association rules [18], or as features in predictive models [7]. We analyzed the predictiveness of the patterns by evaluating precision and recall for the available classifications for the interval sequences. Full predictive models are beyond the scope of this paper, since they would require classifier learning, validation and parameter tuning. We simply evaluate the usefulness of the patterns for ranking or predictive learning. For each class and each Allen pattern and SISP $\mathcal{P}_i$ we calculated precision p_i, recall r_i, and F1. For each class we analyzed the precision/recall plot and determined the Pareto set of patterns, i.e., all patterns that are not dominated by another pattern in both dimensions. Assuming the (p_i, r_i) of the Pareto set are sorted increasing by precision and decreasing by recall, we draw a curve through the points $(0, r_1), (p_i, r_i), (p_i, r_{i+1}), (p_{i+1}, r_{i+1}), ..., (p_k, r_k), (p_k, 0)$ for $i = 1, ...k$. We calculated the area under the curve (AUC) to summarize the predictive power of the Pareto set of patterns. The best F1 values for each pattern class and the AUC were compared between Allen patterns and SISPs. Figure 7 shows the precision/recall plot for the class representing the word *name* in the Australian sign language dataset annotated with example patterns. The boxes represent Allen patterns and always coincide with a cross representing a SISP because SISPs are a superset of Allen patterns. The Pareto sets with the best patterns for Allen and SISP are shown with the dotted and continuous lines, respectively. Examples for the patterns of each approach are shown. *Several SISPs are clearly better in precision and/or recall than the best Allen patterns.* Three patterns involving complete intervals are found with both approaches: 4^+4^- (precision 0.1/ recall 1.0), $4^+2^+2^-4^-$ (0.67/0.9), and $3^+3^-4^+2^+2^-4^-$ (1.0/0.25). The SISP $4^+6^+2^+2^-4^-$ (1.0/0.60) includes only one boundary of the interval labeled with 6 clearly outperforming the best Allen patterns and causing a gain of 0.12 in the AUC of the Pareto sets. In Figure 8 we show the

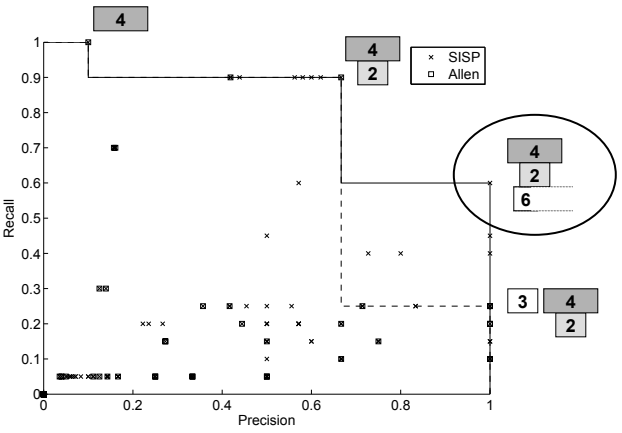
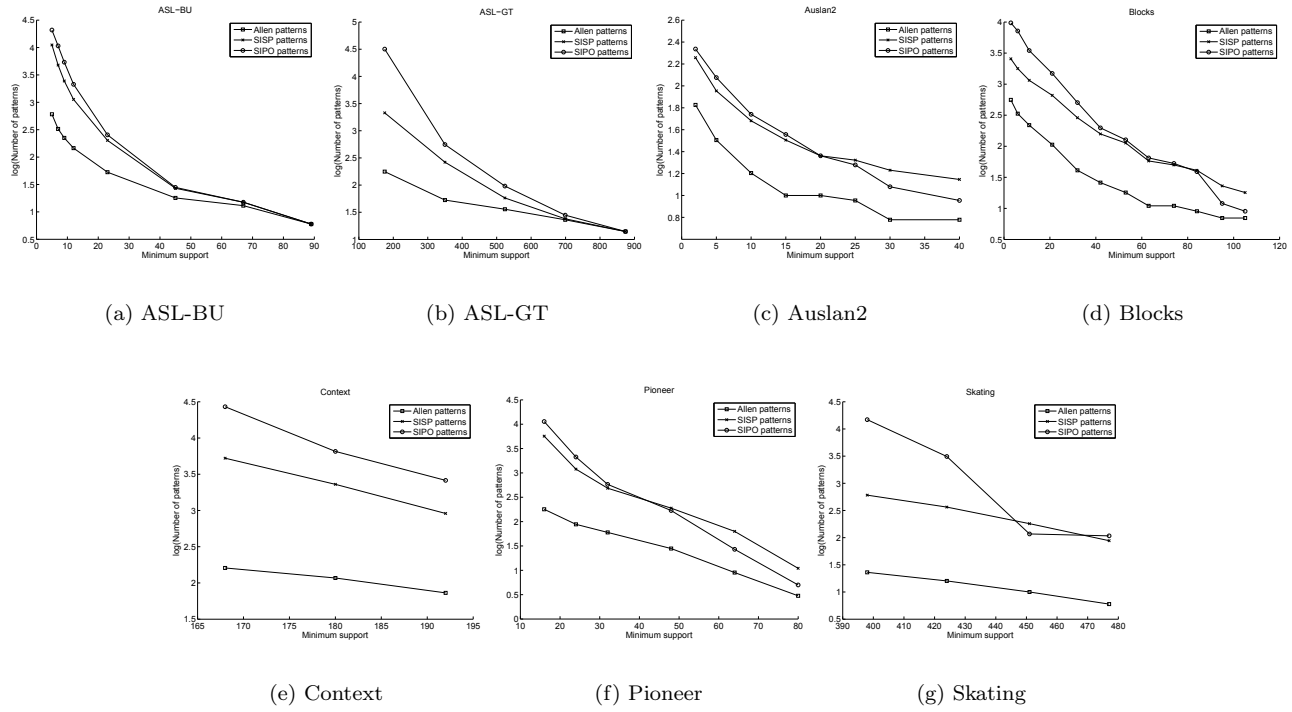


Figure 7: Precision vs. recall for the class *name* (6) in the Auslan dataset. The continuous and dotted lines indicate the Pareto sets for SISPs and Allen patterns, respectively. The difference in AUC is shown in Figure 8c. The best patterns are shown and include the circled SISP with only the opening of interval 6 that is better than any Allen pattern.

difference in AUC between SISPs and Allen patterns for



(a) ASL-BU (b) ASL-GT (c) Auslan2 (d) Blocks

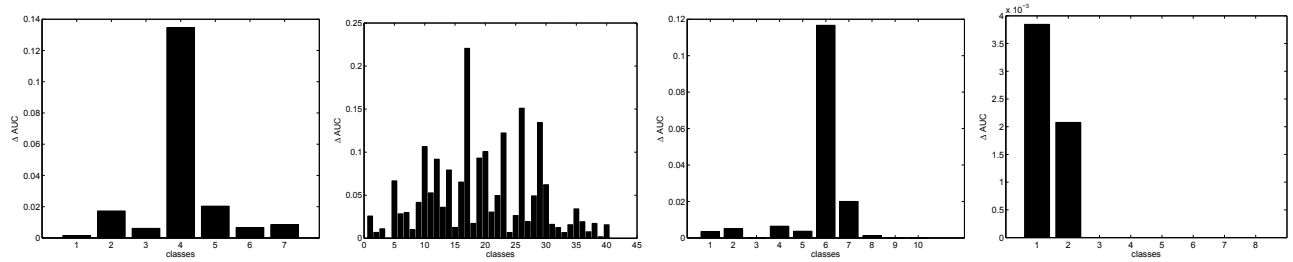
(e) Context (f) Pioneer (g) Skating

Figure 6: Comparison of the number of patterns for different minimum support thresholds in intervals datasets. For most datasets there are many more SISPs and SIPOs than Allen patterns.

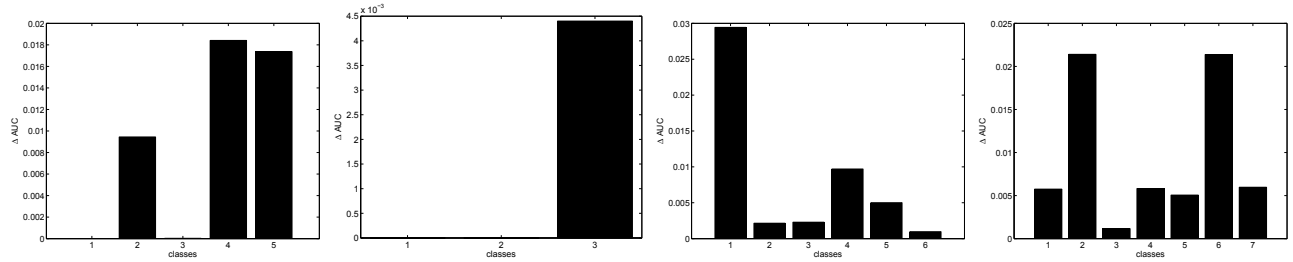
all datasets and classes. Qualitatively, SISPs are always better or equal in predictive power to Allen patterns and show large improvements for ASL-GT, ASL-BU, and Auslan2. Very small quantitative improvements are observed on Pioneer and Blocks. A comparison of difference in F1 of the best patterns showed similar results. *These results demonstrate that SISPs can uncover relationships among semi-intervals in the data that correlate better with known classes than patterns limited to complete intervals.* For the Skating dataset, two ground truth classifications were available: by individual skater and by speed. The best F1 values for SISPs were between 0.29 and 0.43 for the six skaters. For the seven different speeds the F1 values ranged much lower from 0.22 to 0.33. This indicates that *regularities found in the movement cycles of the skaters are stronger for individuals than for speeds.* This is evidence that personal style persists over different speeds. The patterns from better performing skaters can be analyzed for clues regarding their techniques.

For most datasets SIPOs showed similar predictive performance to SISPs. On the ASL-GT dataset, however, AUC improvement of up to 0.1 were observed as shown in Figure 9. The precision/recall plot for the class 17 representing the word *I* with the largest difference in performance is shown in Figure 10. The best SISP are shown with open and closed boxes representing intervals and the best SIPO are shown with directed graphs of interval boundaries. We discuss the patterns from left to right:

- The interval 67 represented by the identical SISP and SIPO $67^{+}67^{-}$ has almost perfect recall but low precision.
- The second SISP $82^{+}82^{-}67^{-}$ adds the complete interval 82 but does not include the start point of 67. The precision is much increased while recall drops slightly. The corresponding SIPO has the same precision and recall but also includes the start point 67. *This demonstrates that even SIPO that are not more predictive than any of the SISP they contain can offer a better explanation by including more order relations.* Since the data contains only complete intervals, the start point of 67 is observed in all sequences with the endpoint of 67. It is not part of the SISP because there is no consistent order relation between 67^{+} and the boundaries of interval 87. The SIPO is able to express this explicitly in the directed graph by only connecting it to 67^{-}.



(a) ASL-BU (b) ASL-GT (c) Auslan2 (d) Blocks



(e) Context (f) Pioneer (g) Skating (Individuals) (h) Skating (Speeds)

Figure 8: Comparison of Allen patterns and SISPs using the difference in AUC of the Pareto set in a precision/recall plot. SISPs are always equal to or better than Allen patterns.

- The next two SISP are similar in structure and further increase precision with small loss in recall.
- Finally, the circled SIPO clearly outperforms any SISP and includes all previously discussed SISP as sub-graphs. *This demonstrates that SIPO can be more predictive than SISP by conjunctively combining sequences of interval boundaries.* The SIPO can also be interpreted as a combination of the last two SIPO with the additional order relation of 82^+ and 87^+ (interval 82 starts before interval 87).

5 Discussion

We present the first approach to mining of semi-interval patterns from interval databases. To the best of our knowledge [34] is the only previous work where semi-intervals were considered when using Allen's and Freksa's relations to add temporal semantics to association rules. The less restrictive nature of SISPs and SIPOs helps fight pattern fragmentation [28, 17], caused by small shifts in interval boundaries, that leads to similar situations being represented by different (possibly infrequent) Allen patterns.

The patterns also support equality of interval boundaries represented by itemsets of size greater than one. Such patterns were observed during the experiments but did not belong to the best patterns for the examples considered. Our approach further *easily generalizes to datasets with mixed time interval and time point data.* This was not investigated because the datasets consisted only of intervals. When converting numerical data, peaks or valleys could be converted to instantaneous events. The pattern representation has further shown promising results for use in predictive models such as [7], [17], [31], [43].

We found the interval boundary data model of [42] to be very useful in generalizing interval patterns to semi-interval patterns and applying the concepts of partial order. Efficient algorithms like BIDE and DCI can be used with almost no change. In [17] an argument is made against representing intervals with boundaries, because it looses the interval semantics. In [42] a sequential pattern mining algorithm is extended to keep the interval semantics. We let the data speak for itself and generate patterns that can contain complete intervals and semi-intervals and leave the interpretation to the user.

In future work we will compare semi-intervals to the Time Series Knowledge Representation (TSKR) [28] that is also aimed at more robust handling of interval

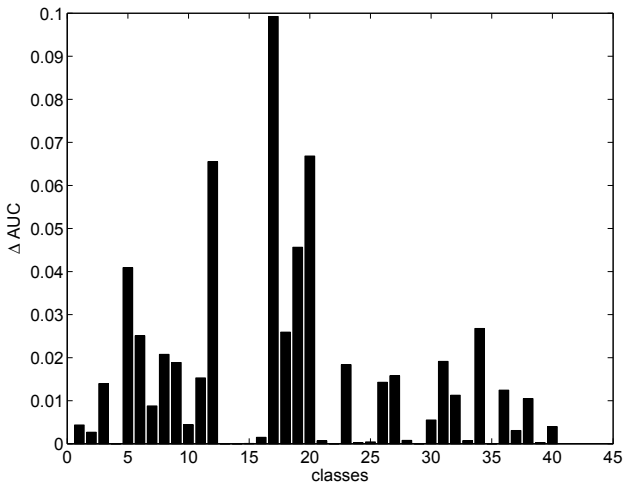


Figure 9: Comparison of SISPs and SIPOs for the classes of the ASL-GT dataset using the difference in AUC of the Pareto set in a precision/recall plot. SIPO are always equal to or better than SISP patterns.

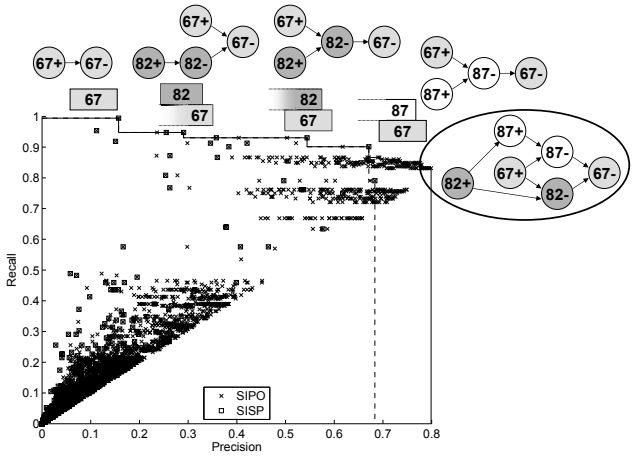


Figure 10: Precision vs. recall for the class I (17) in the ASL-GT dataset. The continuous and dotted lines indicate the Pareto sets for SIPOs and SISPs, respectively. The difference in AUC is shown in Figure 9. The best patterns are shown and including the circled SIPO that combines several SISPs to achieve better performance.

data than Allen patterns. A fundamental difference of TSKR vs. both interval and semi-interval patterns is that subintervals of observed intervals are part of the patterns. More parameters need to be tuned and some depend on domain knowledge that was not available for all datasets, e.g., the minimum duration or interval intersections.

6 Summary

We proposed a novel approach to interval data mining using two kinds of semi-interval patterns, SISPs and SIPOs. This approach differs significantly from existing approaches based on Allen patterns in two ways: (i) interval boundaries are allowed to be missing from a pattern; and (ii) in SIPO, a partial ordering of interval boundaries can be modelled. These characteristics allow for more flexible matching of situations in the data that correspond to different Allen patterns. We demonstrated in an extensive empirical evaluation that such patterns exist in real life data and that they are useful for explaining or predicting known classes of interval sequences in applications such as sign language, robotics, and medicine.

References

[1] R. Agrawal and R. Srikant. Mining sequential patterns. In *Proc. IEEE ICDE*, pages 3–14. IEEE Press, 1995.

[2] M. Aiello, C. Monz, L. Todoran, and M. Worring. Document understanding for a broad class of documents. *Intl. Journal on Document Analysis and Recognition*, 5(1):1–16, 2002.

[3] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.

[4] A. Asuncion and D. Newman. UCI Machine Learning Repository. University of California, Irvine `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

[5] G. Casas-Garriga. Summarizing sequential data with closed partial orders. In *Proc. of the 5th SIAM Intl. Conf. on Data Mining (SDM)*, pages 380–391. SIAM, 2005.

[6] Y.-L. Chen and T.-K. Huang. Discovering fuzzy time-interval sequential patterns in sequence databases. *IEEE Trans. Systems, Man, and Cybernetics*, 35(5):959–972, 2005.

[7] H. Cheng, X. Yan, J. Han, and C.-W. Hsu. Discriminative frequent pattern analysis for effective classification. In *Proc. IEEE ICDE*, 2007.

[8] P. R. Cohen. Fluent learning: Elucidating the structure of episodes. In *Proc. of the 4th Intl. Conf. in Intelligent Data Analysis (IDA)*, pages 268–277. Springer, 2001.

[9] G. Dong and J. Pei. *Sequence Data Mining*. Morgan Kaufmann, 200y.

[10] A. Fern. *Learning Models and Formulas of a Temporal Event Logic*. PhD thesis, Purdue University, West Lafayette, IN, USA, 2004.

[11] C. Freksa. Temporal reasoning based on semi-intervals. *Artificial Intelligence*, 54(1):199–227, 1992.

[12] F. Gianotti, M. Nanni, and D. Pedreschi. Efficient mining of temporally annotated sequences. In *Proc. of the 6th SIAM Intl. Conf. on Data Mining (SDM)*, pages 346–357. SIAM, 2006.

[13] G. Guimarães, J. Peter, T. Penzel, and A. Ultsch. A

method for automated temporal knowledge acquisition applied to sleep-related breathing disorders. *Artificial Intelligence in Medicine*, 23(3):211–237, 2001.

[14] T. Guyet and R. Quiniou. Mining temporal patterns with quantitative intervals. In *Proc. IEEE Intl. Conf. on Data Mining Workshops*, pages 218–227. IEEE, 2008.

[15] J. Han and M. Kamber. *Data Mining - Concepts and Techniques, 2nd edition*. Morgan Kaufmann, 2006.

[16] Y. Hirate and H. Yamaha. Generalized sequential pattern minin with item intervals. *Journal of computers*, 1(3):51–60, 2006.

[17] F. Hoeppner and A. Topp. Classification based on the trace of variables over time. In *Proc. Int. Conf. Intelligent Data Engineering and Automated Learning (IDEAL)*, pages 739–749. Springer, 2007.

[18] F. Höppner. Discovery of temporal patterns - learning rules about the qualitative behaviour of time series. In *Proc. of the 5th European Conf. on Principles of Data Mining and Knowledge Discovery (PKDD)*, pages 192–203. Springer, 2001.

[19] M. W. Kadous. *Temporal Classification: Extending the Classification Paradigm to Multivariate Time Series*. PhD thesis, University of New South Wales, 2002.

[20] P.-S. Kam and A. W.-C. Fu. Discovering temporal patterns for interval-based events. In *Proc. DaWaK*, pages 317–326. Springer, 2000.

[21] S. Kempe, J. Hipp, and R. Kruse. Fsmtree: An efficient algorithm for mining frequent temporal patterns. In *Proc. Conf. of the Gesellschaft fr Klassifikation*, pages 253–260. Springer, 2008.

[22] C. Lucchese, S. Orlando, and R. Perego. Fast and memory efficient mining of frequent closed itemsets. *IEEE TKDE*, 18(1):21–36, 2006.

[23] H. Mannila, H. Toivonen, and I. Verkamo. Discovery of frequent episodes in event sequences. In *Proc. of the 1st Intl. Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 210–215. AAAI Press, 1995.

[24] J. Mäntyjärvi, J. Himberg, P. Kangas, U. Tuomela, and P. Huuskonen. Sensor signal data set for exploring context recognition of mobile devices. In *Proc. of 2nd Int. Conf. on Pervasive Computing (PERVASIVE 2004)*, pages 18–23. Springer, 2004.

[25] F. Mörchen. A better tool than allen's relations for expressing temporal knowledge in interval data. In *TDM Workshop, ACM SIGKDD*, pages 25–34, 2006.

[26] F. Mörchen. Unsupervised pattern mining from symbolic temporal data. *SIGKDD Explor. Newsl.*, 9(1):41–55, 2007.

[27] F. Mörchen and A. Ultsch. Optimizing time series discretization for knowledge discovery. In *Proc. ACM SIGKDD*, pages 660–665. ACM Press, 2005.

[28] F. Mörchen and A. Ultsch. Efficient mining of understandable patterns from multivariate interval time series. *Data Min. Knowl. Discov.*, 2007.

[29] R. Moskovitch and Y. Shahar. Karmalego - fast time intervals mining. Technical Report 23, ISE-TECH-REP Ben Gurion University, 2009.

[30] P. Papaterou, G. Kollios, S. Sclaroff, and D. Gunopoulos. Discovering frequent arrangements of temporal intervals. In *Proc. of the 5th IEEE Intl. Conf. on Data Mining (ICDM)*, pages 354–361, 2005.

[31] D. Patel, W. Hsu, and M. Lee. Mining relationships among interval-based events for classification. In *Proc. SIGMOD*, pages 393–404, 2008.

[32] J. Pei, J. Liu, H. Wang, K. Wang, P. S. Yu, and J. Wang. Efficiently mining frequent closed partial orders. In *Proc. of the 5th IEEE Intl. Conf. on Data Mining (ICDM)*, pages 753–756. IEEE Press, 2005.

[33] J. Pei, H. Wang, J. Liu, K. Wang, J. Wang, and P. S. Yu. Discovering frequent closed partial orders from strings. *IEEE TKDE*, 18(11):1467–1481, 2006.

[34] C. Rainsford and J. Roddick. Adding temporal semantics to association rules. In *Proc. of the 3rd European Conf. on Principles of Data Mining and Knowledge Discovery (PKDD)*, pages 504–509. Springer, 1999.

[35] C. Raïssi, T. Calders, and P. Poncelet. Mining conjunctive sequential patterns. *Data Min. Knowl. Discov.*, 17(1):77–93, 2008.

[36] J. F. Roddick and C. H. Mooney. Linear temporal sequences and their interpretation using midpoint relationships. *IEEE TKDE*, 17(1):133–135, 2005.

[37] S. Schockaert, M. De Cock, and E. Kerre. Fuzzyfying allen's temporal relations. *IEEE Trans. Fuzzy Systems*, 16(2):517–533, 2008.

[38] T. Starner, J. Weaver, and A. Pentland. Real-time American Sign Language recognition using desk and wearable computer-based video. *IEEE TPAMI*, 20(12), 1998.

[39] R. Villafane, K. A. Hua, D. Tran, and B. Maulik. Mining interval time series. In *Proc. DaWaK*, pages 318–330. Springer, 1999.

[40] J. Wang and J. Han. BIDE: Efficient mining of frequent closed sequences. In *Proc. ICDE*, pages 79–90. IEEE Press, 2004.

[41] E. Winarko and J. F. Roddick. Armada - an algorithm for discovering richer relative temporal association rules from interval-based data. *Data & Knowledge Engineering*, 2007.

[42] S.-Y. Wu and Y.-L. Chen. Mining nonambiguous temporal patterns for interval-based events. *IEEE TKDE*, 19(6):742–758, 2007.

[43] Z. Xing, J. Pei, G. Dong, and P. S. Yu. Mining sequence classiers for early prediction. In *Proc. IEEE ICDM*, 2008.

[44] M. Yoshida, T. Iizuka, H. Shiohara, and M. Ishiguro. Mining sequential patterns including time intervals. In *Proc. of SPIE*, volume 4057, pages 213–220, 2000.

[45] Q. Zhao and S. Bhowmick. Sequential pattern mining: A survey. Technical report, Nanyang Technichal University, Singapore, 2003.