

Efficient Mining Of Understandable Patterns From Multivariate Interval Time Series

Fabian Mörchen¹ and Alfred Ultsch²

¹ Siemens Corporate Research, Princeton, NJ fabian.moerchen@siemens.com

² Databionic Research Group, Philipps-University Marburg, Germany
ultsch@informatik.uni-marburg.de

Abstract. We present a new method for the understandable description of local temporal relationships in multivariate data, called *Time Series Knowledge Mining* (TSKM). We define the *Time Series Knowledge Representation* (TSKR) as a new language for expressing temporal knowledge in time interval data. The patterns have a hierarchical structure, with levels corresponding to the temporal concepts duration, coincidence, and partial order. The patterns are very compact, but offer details for each element on demand. In comparison with related approaches, the TSKR is shown to have advantages in robustness, expressivity, and comprehensibility. The search for coincidence and partial order in interval data can be formulated as instances of the well known frequent itemset problem. Efficient algorithms for the discovery of the patterns are adapted accordingly. A novel form of search space pruning effectively reduces the size of the mining result to ease interpretation and speed up the algorithms. Human interaction is used during the mining to analyze and validate partial results as early as possible and guide further processing steps. The efficacy of the methods is demonstrated using two real life data sets. In an application to sports medicine the results were recognized as valid and useful by an expert of the field.

Keywords: knowledge discovery, time series, interval patterns, Allen's relations

1 Introduction

Temporal data mining has recently received increasing attention, as many processes in business and science have interesting time changing aspects. Applications include the search for patterns in large time series databases (e.g. Palpanas et al. (2004a); Lin et al. (2004)) and streaming time series (e.g. Palpanas et al. (2004b); Papadimitriou et al. (2005)). Symbolic interval time series are an important data format for discovering temporal knowledge that can be easily communicated to human analysts (Guimarães and Ultsch, 1999; Kam and Fu, 2000; Villafane et al., 2000; Höppner, 2001; Cohen, 2001; Last et al., 2001; Höppner, 2003; Mörchen et al., 2004; Bellazi et al., 2005; Papaterou et al., 2005; Winarko

and Roddick, 2007). Numerical time series are often converted to symbolic interval time series by segmentation (Last et al., 2001; Höppner, 2003), discretization (Villafane et al., 2000; Mörchen et al., 2004; Bellazi et al., 2005) or clustering (Guimarães and Ultsch, 1999; Mörchen et al., 2004). Alternatively, interval data can be obtained directly from other temporal data, e.g., video (Fern, 2004) or association rules over time (Rainsford and Roddick, 1999). Patterns mined from symbolic interval data can provide explanation for the underlying temporal processes or anomalous behavior.

Unsupervised pattern discovery in interval time series has largely been performed based on Allen’s interval relations (Allen, 1983), e.g., Kam and Fu (2000); Cohen (2001); Höppner (2001); Bellazi et al. (2005); Papaterou et al. (2005); Winarko and Roddick (2007). The relations were originally developed in the context of temporal reasoning where inference about past, present, and future supports applications in planning, understanding, and diagnosis. The input usually consists of exact but incomplete data and temporal constraints, often expressed by Allen’s relations. Typical problems include determining the consistency of the data and answering queries about scenarios satisfying all constraints (e.g. Schwab and Vila (1997)). But these problems do not occur in the data mining context: almost the complete interval data is given and meaningful and understandable patterns are searched (Höppner, 2003). One may have to cope with some missing data, but more importantly with possibly noisy and incorrect data.

We think that Allen’s relations have severe disadvantages when used for pattern discovery from interval time series, in particular for noisy data and when interval boundaries can only be assumed to approximately describe the underlying temporal phenomena. The patterns are not robust, in the sense that small disturbances of interval boundaries lead to different patterns for similar situations. The representation is ambiguous since the same pattern can have quantitatively widely varying appearances. For all but very simple cases the patterns are not understandable because the textual descriptions are lengthy and unstructured whereas graphical representations do not abstract over the original data.

We propose the Time Series Knowledge Representation (TSKR), a new hierarchical language for the formulation of temporal knowledge based on interval times series that extends the Unification-based Temporal Grammar (UTG) (Ultsch, 1996, 2004). The representation is formally defined and efficient algorithms for mining the patterns are described in detail. A novel pruning technique leads to a manageable amount of patterns while preserving a variety of general and frequent as well as specific and rare patterns.

The TSKR provides a conceptually different alternative to patterns using Allen’s relations that are omnipresent in data mining. Both representations are thoroughly compared and many advantages of the TSKR in robustness, expressivity, and comprehensibility are pointed out. These practical aspects of using Allen’s relations in data mining have been largely neglected in previous studies. Only Cohen (2001); Höppner (2003); Bellazi et al. (2005) actually show some patterns from real life data while Kam and Fu (2000); Papaterou et al.

(2005); Winarko and Roddick (2007) only discuss performance, completely ignoring whether the mined patterns are actually useful. We use two real life datasets to compare the quality of patterns expressed with TSKR and Allen’s relations. In both cases the TSKR creates much fewer patterns that better explain the underlying temporal phenomena.

In Section 2 we discuss existing methods for mining interval time series. Our new knowledge representation is presented in Section 3. In Section 4 we compare the TSKR to a pattern language using Allen’s relations w.r.t. expressivity, robustness, and interpretability. In Section 5 we show how important steps of mining TSKR patterns can be formulated as existing itemset and sequential pattern mining problems. Efficient algorithms are adapted accordingly. The experimental results in Section 6 show that the proposal works well in practice and demonstrate the problems of using Allen’s relations for mining patterns based on interval data with approximate boundaries. We conclude with a discussion and a short summary in Sections 7 and 8.

2 Related work and motivation

There are two qualitatively different approaches that can express many temporal concepts in interval data: patterns based on Allen’s relations and rules expressed with the UTG. Other existing methods are not as powerful. In Villafane et al. (2000) only containments of intervals and in Last et al. (2001) only successive intervals are searched.

2.1 Allen’s interval relations

The 13 interval relations of Allen (1983) are: *before*, *meets*, *overlaps*, *starts*, *during*, *finishes*, the corresponding inverses and *equals*. They can describe any relative positioning of two intervals. They are commonly used for the formulation of temporal rules involving two or more intervals (Kam and Fu, 2000; Siskind, 2001; Cohen, 2001; Höppner, 2003; Mooney and Roddick, 2004; Roddick and Mooney, 2005).

Three methods use Allen’s relations for unsupervised rule mining from symbolic interval data with variants of the Apriori algorithm. In Kam and Fu (2000) interval patterns are restricted to right concatenations of intervals to existing patterns, using one of Allen’s relations to describe the relative positioning of the new interval to the interval of the complete pattern. The support is determined by counting the interval sequences from a set of sequences in which the pattern appears. In Cohen (2001) the patterns are restricted to composites of two already significant patterns or single intervals and support is determined by counting occurrences in sliding windows over a single interval sequence. Both approaches thus use at most $k - 1$ relations for k intervals. In contrast, the patterns of Höppner (2003) list all $\frac{k(k-1)}{2}$ pairwise relations of the intervals within a pattern. Support is determined based on the lifetime of a pattern within a sliding window. In Papaterou et al. (2005) the problem of mining patterns from

interval data was rediscovered. The pattern format of Höppner (2003) is used with a subset of Allen’s relations. The patterns are mined with a tree-based enumeration algorithm adapted from (Bayardo, 1998). In Winarko and Roddick (2007) the patterns of Höppner (2003) are mined with a modified sequential pattern algorithm (Lin and Lee, 2002).

Allen’s relations have severe disadvantages when used for pattern discovery from interval time series, as demonstrated with the following examples. A more detailed analysis is given in Section 4.

(1) *Patterns from noisy interval data expressed with Allen’s interval relations are not robust:* Several of Allen’s relations require the equality of two or more interval boundaries. Small disturbances in interval boundaries can create patterns where a very similar relationship between intervals is fragmented into different relations as defined by Allen’s relations. Figure 1 shows several examples of almost equal intervals.

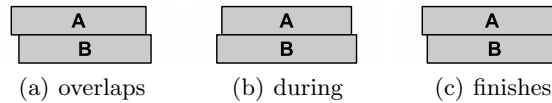


Fig. 1. Examples for different patterns according to Allen that are fragments of the same approximate relation *almost equals*.

(2) *Patterns expressed with Allen’s interval relations are ambiguous:* The same relation of Allen can quantitatively and intuitively represent very different situations. In Figure 2 three very different versions of the *overlaps* relation are shown as an example. Even more ambiguous is the compact representation of patterns from Kam and Fu (2000) or Cohen (2001), several different descriptions are valid for the exact same pattern (see Figure 3). An extension of Allen’s relations using the relative placement of the intervals’ mid points (Roddick and Mooney, 2005) can distinguish the different variants of *overlaps*. Many of the additional conditions require equality of time points, however, increasing fragmentation effects similar to Figure 1.

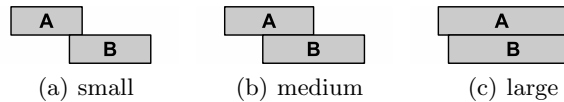


Fig. 2. Three instances of Allen’s *overlaps* relation with large quantitative differences.

(3) *Patterns expressed with Allen’s interval relations are not easily comprehensible:* The representation of patterns with Allen’s relations does not follow the Gricean maxims (Grice, 1989) suggested for the representation of knowledge discovery results to humans (Sripada et al., 2003). For example, the maxim

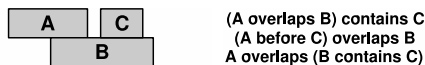


Fig. 3. Pattern that can be described by three different compact rules using Allen’s relations.

of manner is violated. Since the compact format is ambiguous, patterns need to be expressed with an unstructured list of pairwise relations of all intervals (Höppner, 2003) that grows quickly with the number of intervals.

2.2 Unification-based Temporal Grammar

The UTG was proposed by Ultsch (Ultsch, 1996, 2004) and realized in Guimarães and Ultsch (1997); Guimarães (1998). It is a hierarchical pattern language for temporal knowledge discovery. Each level of the hierarchy inhibits an increase in temporal abstraction. After preprocessing the data model is a multivariate symbolic interval time series. The central pattern elements are Events and Sequences shown in Figure 4 for a 3 dimensional time series. Events combine several *more or less simultaneous* intervals, a robust version of Allen’s *equals*. The number of intervals in an Event is restricted, however, to the dimensionality of the interval series. Sequences describe an ordering of several Events with *immediately followed by* or *followed by after at most t time units*. Allowing alternatives for the Events in Sequences leads to the final so-called Temporal Patterns. The UTG was applied in Guimarães and Ultsch (1999) to the analysis of sleeping disorders. Most steps of the discovery were performed manually based on visualizations and/or statistical summaries of the patterns of the previous levels. Algorithms for the discovery of slightly modified UTG patterns are described in Mörchen et al. (2004).

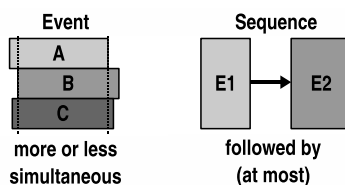


Fig. 4. Core patterns of the UTG: Events describe a group of almost simultaneously starting and ending intervals, Sequences describe an order of Events.

The UTG Event patterns are more robust than Allen’s relations, by matching interval boundaries with a threshold. In contrast, the distinction between two versions of *followed by* is sensitive to small disturbances of interval boundaries. The UTG cannot express all patterns that can be formulated with Allen because

Events require all intervals to begin and end within a small time span, respectively. Consider the overlapping intervals in Figure 2. In Figure 2(c) an Event would describe the almost simultaneous occurrence of A and B . For Figure 2(a), however, the threshold to match the interval boundaries would have to be unreasonably large to detect an Event pattern. The fixed number of intervals in an Event further limits the search space and applicability of the pattern language. The hierarchical structure of the UTG and the separation of temporal concepts over several mining steps, however, offer unique possibilities in relevance feedback during the knowledge discovery process and in the analysis of the results. The expert can zoom into the discovered patterns and guide the mining process by filtering out irrelevant or uninteresting information at lower levels (Ultsch, 2004).

The TSKR is based on these core ideas inheriting the advantages while fixing the identified discrepancies. In (Mörchen et al., 2004, 2006) we have introduced a modification of the UTG allowing Events to ignore trailing and leading parts of intervals, thus describing coincidence of intervals rather than synchronicity. We introduced a further significant extension of the pattern language (Mörchen, 2006c,b,a) that will be described in detail below. We allow an *arbitrary number* of coinciding intervals and *partial* order of these coincidences. These extensions make the pattern space more complex and require the design of new mining algorithms.

3 Time Series Knowledge Representation

The TSKR is a hierarchical interval language describing the temporal concepts of duration, coincidence, and partial order in interval time series. The basic primitives are labeled time intervals called *Tones* representing duration. Simultaneously occurring Tones form a *Chord*, representing coincidence. Several Chords connected with a partial order form a *Phrase*.

In data mining the 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 define the following patterns based on the natural numbering \mathbb{T} of a set of uniformly spaced time points. Let Σ be a set of unique *symbols*.

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

Definition 2 A symbolic time interval is a triple $[\sigma, s, e]$ with $\sigma \in \Sigma$, $[s, e] \in \mathbb{I}$. For example, $[\text{temperature high}, 12, 78]$ describes a state observed starting at time point 12 and lasting until time point 78, inclusively. We write $[\sigma, s, e] \subseteq [\sigma', s', e']$ if $s' \leq s$ and $e \leq e'$ with equality iff $s = s'$ and $e = e'$. A symbolic time interval $[\sigma, s, e]$ is maximal if $\forall [\sigma', s', e'] \supset [\sigma, s, e]$ $\sigma' \neq \sigma$. If $\{s, \dots, e\} \cap \{s', \dots, e'\} \neq \emptyset$ we say that the intervals $[\sigma, s, e]$ and $[\sigma', s', e']$ overlap.

Definition 3 We define a partial order \prec of intervals as $[s_1, e_1] \prec [s_2, e_2] \Leftrightarrow e_1 < s_2$. We say that $[s_1, e_1]$ is before $[s_2, e_2]$.

Definition 4 A symbolic interval series is a set of non-overlapping symbolic time intervals $\mathcal{I} = \{[\sigma_i, s_i, e_i] | \sigma_i \in \Sigma, [s_i, e_i] \in \mathbb{I}, i = 1, \dots, N; e_j < s_{j+1}, j = 1, \dots, N-1\}$. The duration of an interval series is $d(\mathcal{I}) = \sum_{i=1}^N d([s_i, e_i])$.

Definition 5 A symbolic interval sequence is a finite set of symbolic time intervals $\mathcal{I} = \{[\sigma_i, s_i, e_i] | \sigma_i \in \Sigma, [s_i, e_i] \in \mathbb{I}, i = 1, \dots, N\}$.

Definition 6 An itemset is a subset $S = \{\sigma_1, \dots, \sigma_n\} \subseteq \Sigma$ of all symbols.

Definition 7 An itemset time interval is a triple $[S, s, e]$ with $S \subseteq \Sigma$ $[s, e] \in \mathbb{I}$.

Definition 8 An itemset interval series is a finite set of non overlapping itemset time intervals $\mathcal{I} = \{[S_i, s_i, e_i] | S_i \subseteq \Sigma, [s_i, e_i] \in \mathbb{I}, i = 1, \dots, N; e_j < s_{j+1}, j = 1, \dots, N-1\}$.

At the core of our knowledge representation stand the patterns and the occurrences of patterns in the input data.

Definition 9 Let Λ be a finite set of labels λ . Let $l : \Sigma \mapsto \Lambda$ be the function assigning each symbol a label. Let Φ be a set of characteristic functions $\phi_{\mathcal{X}} : \mathbb{I} \rightarrow \{\text{TRUE}, \text{FALSE}\}$, where \mathcal{X} is some arbitrary input data. A pattern $(\sigma, \lambda, \phi_{\mathcal{X}})$ is a semiotic triple (Ultsch, 1996) composed of:

- $\sigma \in \Sigma$, a unique symbol representing the pattern in higher levels constructs (syntax),
- $\lambda = l(\sigma) \in \Lambda$, a label providing a textual description of the practical meaning of the pattern (pragmatic),
- $\phi_{\mathcal{X}} \in \Phi$, a characteristic function determining when the pattern occurs (semantic).

The semantic of ϕ and the data type of \mathcal{X} will differ for the specific patterns defined below. We write ϕ for $\phi = \text{TRUE}$ and $\neg\phi$ for $\phi = \text{FALSE}$.

Definition 10 An occurrence of a pattern is an interval $[s, e] \in \mathbb{I}$ with $\phi_{\mathcal{X}}([s, e])$. A maximal occurrence is an occurrence $[s, e]$ such that $\forall [s', e'] \supset [s, e] \neg\phi_{\mathcal{X}}([s', e'])$.

The concept of semiotic triples is consistently used on all levels of the TSKR pattern hierarchy. It enables the pragmatic annotation of each pattern with labels to aid the later interpretation when used as parts of larger patterns. In this study we assume a set of Tone patterns given. A Tone pattern (σ, λ, ϕ) describes a property of the temporal process that is observed during time intervals. A value-based Tone obtained by discretizing a univariate time series $v_t \in \mathbb{R} \ t \in \mathbb{T}$ has a characteristic function of the form $\phi_A([s, e]) \leftarrow v_{min} < v_i \leq v_{max} \forall i \in [s, e] \subset \mathbb{T}$ where $v_{min}, v_{max} \in \mathbb{R}$ and could be labeled $\lambda(\sigma) = \text{“temperature high”}$. This

textual label can be used to represent the semantics of this Tone as defined by ϕ_A in more abstract patterns as defined below. Applying time series segmentation can produce Tones like *increasing speed*. See (Mörchen, 2006c) for the straight forward definition of the corresponding characteristic functions.

Definition 11 A Chord pattern is a semiotic triple $c = (\sigma, \lambda, \phi_{\mathcal{T}}^T)$ with $\sigma \in \Sigma$, $\lambda \in \Lambda$, and a characteristic function $\phi_{\mathcal{T}}^T \in \Phi$ indicating the simultaneous occurrences of k Tones $T = \{t_i = (\sigma_i, \lambda_i, \phi_i) | i = 1, \dots, k, k > 0\}$ on a given time interval according to the interval sequence \mathcal{T} with occurrences of the Tones T :

$$\phi_{\mathcal{T}}^T([s, e]) \leftarrow \phi_1([s, e]) \wedge \dots \wedge \phi_k([s, e]) \quad (1)$$

A Chord consisting of k Tones is called a k -Chord, $|c| = k$ is the size of the Chord c . A 1-Chord is simply a copy of a Tone and is called a trivial Chord. We say the Chord $c_i \supset c_j$ is a super-Chord of c_j if c_i describes the coincidence of a superset of the Tones from c_j .

Definition 12 Let the support set of a Chord be the symbolic interval series of all maximal intervals in the interval sequence \mathcal{T} . The support $sup(c, \mathcal{T})$ of a Chord c is the duration of the support set in \mathcal{T} .

Often, the support set will be restricted to intervals with a duration of at least δ , to exclude very short temporal phenomena that are not meaningful for the application under study. We will write $sup_{\delta}(c)$ for short.

In Figure 5 the occurrences of the Tones A , B , and C are shown in the top segment. In the middle segment all maximal Chords of size 2 and 3 are shown. All support sets consist of two intervals. The occurrence of a Chord pattern implies the occurrence of all sub-Chords on the same interval, e.g., AB , BC , and AC for ABC . In general larger Chords can be considered more interesting, because they are more specific. We use the concept of closedness to non-redundantly represent a set of Chords motivated by closed itemsets (Pasquier et al., 1999).

Definition 13 A Chord c_i is closed if there are no super-Chords that have the same support, i.e., $\forall c_j \supset c_i, sup(c_j) < sup(c_i)$.

The definition of closedness considers a Chord as closed even if a larger Chord has only a slightly smaller support set. With Tone patterns mined from possibly inexact and erroneous time series this is a harsh restriction. We therefore introduce the relaxed concept of margin-closedness to prune patterns with very similar support.

Definition 14 A Chord c_i is margin-closed w.r.t. a threshold $\alpha < 1$ if there are no super-Chords that have almost the same support, i.e., $\forall c_j \supset c_i, \frac{sup(c_j)}{sup(c_i)} < 1 - \alpha$.

The third segment in Figure 5 shows all margin-closed Chords ($\alpha = 0.1$). The Chord BC is not closed, because whenever it is observed, so is the super-Chord ABC . The Chord AC is closed, but not margin-closed, because the support set of the super-Chord ABC is only slightly smaller.

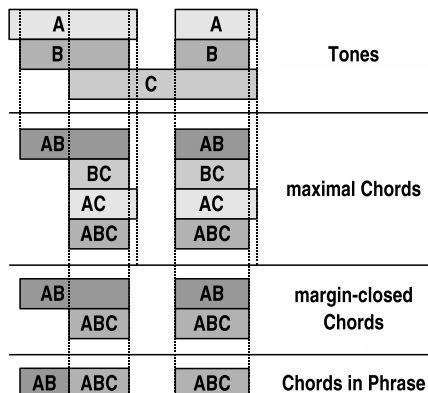


Fig. 5. Simultaneous occurring Tones form maximal Chords. Phrases describe a (partial) order of (not necessarily maximal) Chords.

Definition 15 A Phrase pattern is a semiotic triple $p = (\sigma, \lambda, \phi_C^{C,E})$ with $\sigma \in \Sigma$, $\lambda \in \Lambda$, and a characteristic function $\phi_C^{C,E} \in \Phi$ indicating the occurrences of the k Chords $C = \{c_i = (\sigma_i, \lambda_i, \phi_i) | i = 1, \dots, k, k > 0\}$ according to a partial order $E \subseteq \{\sigma_i\}^2$ on a given time interval.

$$\phi_C([s, e]) \leftarrow (\forall i = 1, \dots, k \exists [s_i, e_i] \subseteq [s, e] \phi_i([s, e])) \quad (2)$$

$$\wedge (\exists i \in \{1, \dots, k\} s_i = s) \quad (3)$$

$$\wedge (\exists i \in \{1, \dots, k\} e_i = e) \quad (4)$$

$$\wedge (\forall i \neq j \in \{1, \dots, k\}^2 [s_i, s_i, e_i] \prec [\sigma_j, s_j, e_j] \Leftrightarrow (\sigma_i, \sigma_j) \in E) \quad (5)$$

A Phrase consisting of k Chords is called a k -Phrase, k is the size of the Phrase. We say the Phrase $p_i \supset p_j$ is a super-Phrase of p_j if p_i describes the partial order of a superset of of the Chords of p_j and all common Chords have the same partial order.

The characteristic function for a Phrase consists of four necessary conditions. Equation 2 ensures that the intervals of all Chords are within the Phrase interval, while Equation 3 and Equation 4 prevent extra room before the first and after the last Chord, respectively. The occurrence of a Phrase is thus maximal by definition. A Phrase occurs on a particular interval but not on the sub-intervals. Equation 5 requires the Chords to be in the partial order specified by E . Note, that any two intervals that have an order relation in E are not allowed to overlap. This restriction makes sense, because Chords already describe the concept of coincidence. Allowing overlapping Chords within a Phrase in general would mean to repeatedly represent the same concept and can in fact be equally represented by a larger Chord on the interval where two Chords overlap. The bottom segment of Figure 5 shows Chord intervals that are part of a Phrase. The interval for AB is non-maximal to avoid overlap with the interval of ABC .

The power of partial order is shown in Figure 6. The two similar Chord sequences in the bottom rows of Figure 6(a) and Figure 6(b) are summarized by the partial order graph of a Phrase shown in Figure 6(c). Note, that similar to Höppner (2003) multiple observation intervals of a Tone (here A) are allowed and treated as distinct intervals in the Phrase. In contrast to Höppner (2003) and similar to Fern (2004) we further allow that only a part of an observed interval is used in a pattern. Different parts of the long Tone interval B in Figure 6(a) are used in the Chords AB , ABC , and BC . The Phrase in Figure 6(c) thus summarizes cases where the Tone B in the three Chords AB , BC , and ABC stems from one (Figure 6(a)) or more (Figure 6(b)) occurrences of the Tone B .

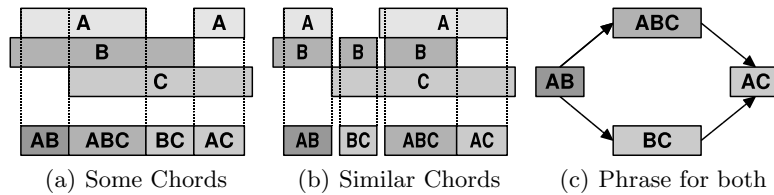


Fig. 6. Chords summarize several overlapping Tones. Phrases summarize similar sequences of Chords.

Definition 16 Let the support set of a Phrase be the symbolic interval series of all maximal intervals. The support $sup(p)$ of a Phrase c is the size of the support set.

Definition 17 A Phrase p_i is closed if there are no super-Phrases that have the same support, i.e., $\forall p_j \supset p_i, sup(p_i) = sup(p_j)$.

Definition 18 A Phrase p_i is margin-closed w.r.t. a threshold $\alpha < 1$ if there are no super-Phrases that have almost the same support, i.e., $\forall p_j \supset p_i, \frac{sup(p_j)}{sup(p_i)} < 1 - \alpha$.

4 Analysis

Having introduced a new language for the description of interval patterns, we compare the TSKR to the patterns of Höppner (2003) listing all pairwise interval relations according to Allen. We show that the pattern language of the TSKR has a higher temporal expressivity, is more robust, and has advantages in interpretability.

4.1 Expressivity

In data mining the purpose of a pattern expressed in a knowledge representation language is to collectively describe many similar or possibly equal situations

observed in the data. We say a pattern is more *expressive* than another, if it summarizes more similar, yet qualitatively different situations.

We first show that all patterns expressible with pairwise Allen’s interval relations can also be expressed with the TSKR. That means that all occurrences of a *single* interval pattern described with the pattern format of Höppner (2003) can also be described by a *single* TSKR pattern. We write AB for a Chord with coinciding Tones A and B . We write $AB \rightarrow CD$ for the Phrase expressing the total order of the Chord AB followed by the Chord CD .

Let $\mathcal{I} = \{(\sigma_i, s_i, e_i) | i = 1, \dots, k\}$ be all involved symbolic intervals with arbitrary pairwise relations according to Allen. Let $B = \{b_j\} = \bigcup_{i=1}^k \{s_i, e_i\}$ the sorted de-duplicated set of all interval boundaries. We construct a Phrase with at most $|B| - 1$ Chords where the j -th Chord describes the non-empty coincidence of all σ_i where $s_i \leq b_j$ and $e_i \leq b_{j+1}$ or is skipped otherwise. The Chords have a complete ordering according to the indices j .

Consider the example pattern in Figure 7 consisting of six intervals and at least one representative of each of Allen’s operators within the pairwise relations. The six resulting Chords are shown in the bottom row.

	A				E
		B			
			C		
	D			F	
AD	ABCD	BC	BCF	ECF	F

Fig. 7. Conversion of complex pattern using all of Allen’s relations to a TSKR pattern.

Note, that a TSKR patterns constructed in this way describes even more similar situations in the data than the original pattern using Allen’s relations. In Figure 8(a) several instances of Allen’s relation A *overlaps* B are shown with the corresponding TSKR pattern $A \rightarrow AB \rightarrow B$. In Figure 8(b) we give some examples of instances that are also covered by the TSKR pattern but not by the Allen pattern. The Phrase also matches observations where the long interval of A and B in Figure 8(a) are interrupted by noise.

On the other hand it is not always possible to find a *single* pattern defined with Allen’s relations to describe all situations covered by a certain *single* TSKR pattern. In particular Phrases utilizing the concept of partial order to summarize similar situations as in Figure 6, cannot be expressed by a single pattern using *all* pairwise relations of Allen. Consider the Chords AB , ABC , BC , and AC as shown in Figure 6(a) at the bottom resulting from the Tone patterns A , B , C in the top rows. In Figure 6(b) the same Tone intervals result in the same Chords but with the middle two Chords exchanged. Both versions can be captured in a *single* Phrase (see Figure 6(c)) using the concept of partial order. The order relation of ABC and BC is simply not specified, both versions of the

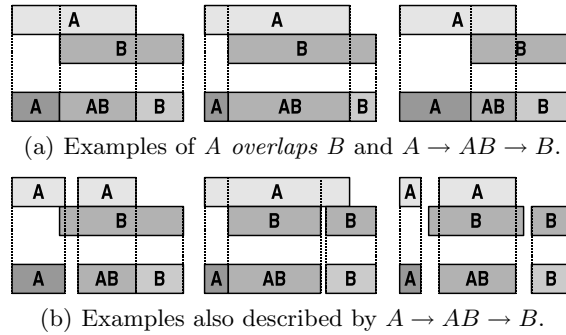


Fig. 8. The TSKR pattern $A \rightarrow AB \rightarrow B$ can describe all instances of Allen's A overlaps B pattern and more.

pattern match this description. The two similar instances cannot be captured with a single pattern using the pattern format of Höppner (2003). Further, the instances of the TSKR pattern $A \rightarrow AB \rightarrow B$ in Figure 8(a) and Figure 8(b) cannot be described with a single pattern of Höppner (2003) because they have a different number of intervals.

Even if we restrict ourselves for the moment to TSKR patterns using only a total order among the Chords in a Phrase and exactly the same set of participating Tone intervals, constructing a pattern with pairwise Allen's relations that describes the same situations in the data is not always possible. For example the simple Phrase $AB \rightarrow C$, i.e., Tones A and B occur simultaneously followed by an interval where the Tones C occurs, covers instances that correspond to very different relations of Allen. The relation between A and B could be *overlaps*, *starts*, *during*, *finished*, *equals* plus the corresponding inverses. The relation between A (or B) and C could be *before*, *meets*, or *overlaps*. See Figure 9 for examples.

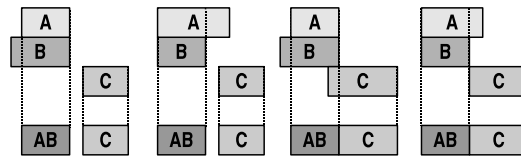


Fig. 9. Example of the same simple TSKR Phrase $AB \rightarrow C$ with different relations according to Allen.

In summary, a single TSKR pattern can express many observations in the data where time intervals have a very similar relative positioning but different relations according to Allen. In contrast each pattern according to Allen can also be described with a single TSKR pattern. The TSKR achieves a higher level of

abstraction over small disturbances in the data while offering clear temporal semantics with the concepts of coincidence and partial order.

4.2 Robustness

Symbolic interval series or sequences obtained from numeric time series inherit the noise present in the original data. The interval boundaries gained from pre-processing steps like discretization of values or segmentation are subject to noise in the measurements. Such time points should thus be considered approximate.

When using Allen’s relations to formulate patterns such slight variations of interval boundaries can create fragmented patterns that describe the same intuitive relationship between intervals with different operators. An example for two almost equal intervals was given in Figure 1. In general any pattern using one of Allen’s relations that requires equality of two interval boundaries can be destroyed by changing one boundary by one time unit only. In Figure 10 we give more examples. If two intervals always appear consecutively but close in time, three different relations can be observed depending on the exact location of the first end point and the second start point (see Figure 10(a)). Intuitively, the relation between the two intervals common to all three examples could be called *largely before* or *roughly meets*. Similar examples are shown for a pattern where the first interval starts almost synchronously with the second but ends earlier (*roughly starts*, Figure 10(b)). The case of *almost equal* intervals is fragmented into nine different patterns of Allen’s relations, the five relations shown in Figure 10(c) and the corresponding inverses for the first four.

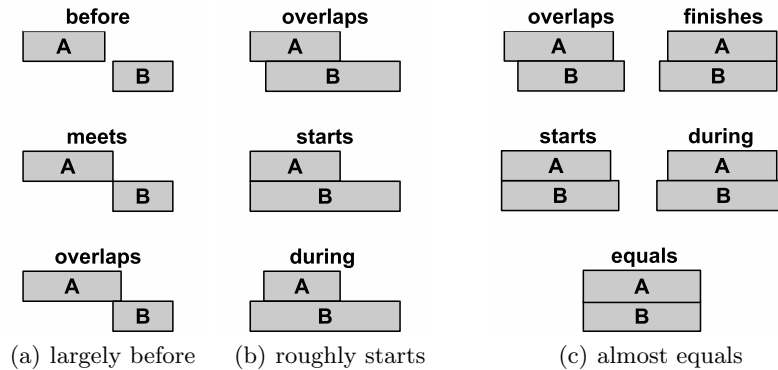


Fig. 10. Three examples where intuitively similar situations are fragmented into different relations of Allen due to slightly shifted interval boundaries.

There are approaches to relax the strictness of Allen’s relations by using a threshold to consider temporally close interval boundaries equal (Aiello et al., 2002). This effectively merges the problematic patterns in each of the cases shown

in Figure 10, but it has not been used in temporal pattern mining. Fragmented patterns are still possible if noise causes interval boundaries to be shifted around the threshold value.

In contrast, the Chord and Phrase patterns of the TSKR are designed to be insensitive to small changes of the interval boundaries. The TSKR operator *coincides* is extremely robust. It only considers the intersection of all participating intervals, any interval can individually be stretched to infinity without changing the pattern at all. All three cases in Figure 1 could be represented well with a Chord describing the coincidence of *A* and *B*. The leading and trailing intervals where only *A* or *B* is observed would not be considered of significant duration for appropriate choices of δ . The robustness of Phrases directly depend on the Chords.

When making intervals smaller, the TSKR patterns break down as soon as the smallest interval disappears - as is true for any pattern in either language. Other aspects of noisy data like missing intervals similarly affect both representations. This can be compensated by extending any of the representations with alternatives as done in the UTG and in Höppner (2003).

4.3 Interpretability

Interpretability is hard to specify or measure. In Sripada et al. (2003) natural language descriptions of time series are generated and interpreted by experts. The authors suggest that the presentation should follow the Gricean maxims of quality, relevance, manner, and quantity. We describe the implications of the maxims in the context of interval patterns and judge how well the different pattern languages support this paradigm.

The maxim of quality states that only well supported facts and no false descriptions should be reported. The former is commonly achieved by reporting only the most frequent patterns. None of the pattern languages can be said to report false patterns. As described above, noisy interval data can lead to fragmented patterns expressed with Allen’s relations, however, that might not be reported even if they are frequent.

The maxim of relevance requires, that only patterns relevant to the expert are listed. This is a rather application dependent requirement, still there are differences in how the pattern languages support this task. The hierarchical structure of the TSKR enables the user to view and filter lower level patterns before the next level constructs are searched. Mining Phrases from only the relevant Chords will be much faster, fewer and smaller Phrase patterns need to be analyzed in the next step. This form of pattern representation supports the human analysis according to the zoom, filter, and details on demand paradigm (Shneiderman, 1996). The patterns expressed with Allen’s relations are commonly mined in a single step, resulting in a much larger set of patterns for manual analysis.

The maxim of manner suggests to be brief and orderly and to avoid obscurity and ambiguity. The TSKR patterns can be longer or shorter than the Allen patterns of Höppner (2003). A pattern of Höppner (2003) with k intervals always

consists of $\frac{k(k-1)}{2}$ pairwise relations. When counting each Chord and each consecutive order relation in a Phrase separately the worst case for the number of atomic relations in a TSKR pattern is $(2k-1)(2k-2)$ because with $2k$ interval boundaries there can be at most $2k-1$ Chords. These Chords would then occur consecutively involving $2k-2$ binary order relations. In the best case there is only a single relation if all intervals are the equal forming a single Chord. The typical size of a TSKR pattern given k intervals strongly depends on the data set at hand.

The textual representation of the TSKR can be argued to be more orderly and to avoid obscurity because it uses a hierarchical structure with a partial order relation on the highest level that offers details on demand. In contrast, there is no inherent order in the list of pairwise Allen relations. They could be ordered by the intervals of the first argument of each relation according to an temporal order of the intervals or an alphabetical order of the interval labels. In either case, the list needs to be considered as a whole to understand that pattern.

The instances of a single pattern can quantitatively vary and represent intuitively different patterns. This was already demonstrated for the *overlaps* relation in Figure 2, two more examples for the relations *starts* and *during* are given in Figure 11. This causes ambiguity because the same relation describes several observations that appear visually quite distinct. This high quantitative variation of patterns represented by a single relation of Allen is caused by the mixture of the temporal concepts of order and coincidence. The problem can be weakened by using the Allen’s relations modified with thresholds (Aiello et al., 2002), but the potential ambiguity persist, in particular for intervals of very different lengths.

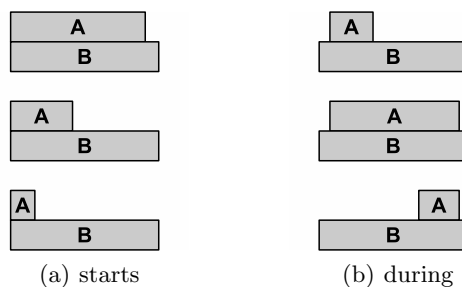


Fig. 11. Three different instances of Allen’s *starts* and *during* relation that have quite distinct appearances.

Whether such different versions of a pattern are semantically equivalent depends on the application. The separation of temporal concepts in the TSKR, however, completely avoids this problem. Chord patterns simply describe the overlapping parts and ignore any variability of the interval boundaries. The parts of the Tone intervals within the Chord occurrence interval always look

exactly the same for all instances. Phrase patterns express the concept of partial order, allowing variation only in the length of Chords and possible gaps. The ambiguity problem is even worse for the compact pattern format used by Kam and Fu (2000) and Cohen (2001). The *exact same* instance of a pattern with k intervals can be written in $\frac{k!}{2}$ different ways by consecutively choosing an interval for the next position in the pattern and excluding duplicates caused by inverse operators. Again, this is not the case for the TSKR, given a set of intervals and a minimum duration for Chords there is only one Phrase of all maximal Chords.

The maxim of quantity states that neither too much nor too few should be reported. Fewer and shorter patterns are easier to read and comprehend. Achieving this maxim is again better supported by the hierarchical approach. Intermediate filtering steps during the pattern mining can be used to prune the search space for the next step and reduce the number of patterns to be inspected. The margin-closedness of the TSKR explicitly offers a powerful mechanism to reduce the number of reported patterns without sacrificing variety. Implication rules generated from Allen patterns can be ranked by interestingness measures Höppner and Klawonn (2002); Höppner (2003), but for the basic pattern considered here, only the frequency can be used. If too many patterns are reported the frequency threshold needs to be raised possibly filtering out rare but interesting patterns.

In summary, the TSKR shows a lot more accordance with the Gricean maxims than Allen's relations. The maxim of quality can be violated by fragmented Allen patterns. The maxim of relevance is better supported by the interactive filtering of TSKR patterns. For the maxim of manner brevity of representation is achieved by the hierarchical structure of the TSKR. The restriction to single temporal concepts for each level of the hierarchical languages avoids ambiguity. The TSKR better supports the maxim of quantity by effective pruning of the results using margin-closedness.

5 Time Series Knowledge Mining

There are basically five steps in the TSKM process shown in Figure 12. If the data to be mined consists of numerical multivariate time series, preprocessing and feature extraction may need to be performed first. Following the divide-and-conquer paradigm it is often advisable to group the dimensions into meaningful subsets, called the Aspects. This task will usually be performed manually. In lack of any information about relations of the different dimensions, a trivial grouping into univariate Aspects or an automated grouping (Gionis et al., 2004) can be performed. Each Aspect is then converted to an interval series of Tones describing persisting properties of the time series. Time series discretization methods (e.g. Mörchen and Ultsch (2005)) can be used to create value-based Tones with descriptions like *high* or *low*. Time series segmentation (e.g. Keogh et al. (2004)) and categorization of the slope values leads to trend-based Tones like *slowly increasing*. Tones representing more complex shapes can be found with time series motif (Lin et al., 2002) methods. For multivariate time series

clustering (e.g. Ultsch (1999)) can generate meaningful Tones. The core part of the TSKM is the mining of coincidence and partial order and will be explained in the next Sections. We show how important steps of the mining can be formulated as well-known problems from frequent itemset and sequential pattern mining.

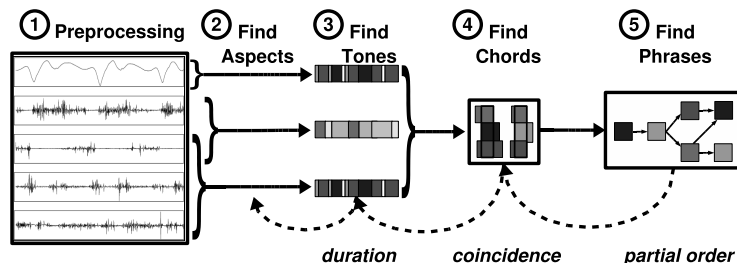


Fig. 12. Data processing steps of Time Series Knowledge Mining.

5.1 Mining coincidence

The input for mining coincidence in form of Chords is a set of Tones with the respective symbolic interval sequence. A Chord is observed whenever at least s_{min} different Tones coincide for a duration of at least δ . A Chord is frequent, if the total duration of all intervals where it is observed exceeds the minimum support sup_{min} . The mining of Chords thus consists of picking a subset of all Tones and determining the support over time considering the minimum duration δ . We further need to compare the support with all super-Chords to ensure (margin-)closedness. This is quite similar to mining closed frequent itemsets, we therefore adapt the CHARM (Zaki and Hsiao, 2002, 2005) algorithm for our purpose. We only describe the basic algorithm and refer to Zaki and Hsiao (2005) for possible performance improvements.

Algorithm 5.1 performs a depth-first search for margin-closed Chords. We write $c_i \cup c_j$ for a Chord describing the coincidence of the Tones from the Chords c_i and c_j . The function EXTEND is recursively called given a prefix Chord c_p and a set of super-Chords S . The recursion is started with an empty Chord and all trivial frequent Chords as possible extensions in Line 1. The prefix Chord c_p is extended by all combinations of super-Chords from the set S in the double loop starting in Line 3. The variable \hat{c}_i (Line 4) stores the initial super-Chord extension by c_i . The extensions with another Chords c_j are filtered by the minimum support in Line 7. The core part of the algorithm is the comparison of the support of the super-Chord $\hat{c}_i \cup c_j$ with the support of both sub-Chords \hat{c}_i and c_j . The following four cases (Lines 8, 11, 13, and 16) are generalizations of the four properties for itemsets in Zaki and Hsiao (2005) to margin-closed Chords and will be explained in detail below. In Line 22 the current Chord \hat{c}_i is added to the set of margin-closed Chords if it is not subsumed by an already

found margin-closed Chord. The function EXTEND is called recursively with the super-Chords \hat{S} of the current Chord \hat{c}_i in Line 25 unless the maximum Chord size is reached. The recursion stops as soon as the set of extensions S is empty.

The four different cases of the support comparison are depicted for an example in Figure 13. Let A, B, C be some Tone patterns. Only one instance of each Tone pattern is shown for demonstration purposes, the lengths and overlaps of the intervals are meant to be representative for the whole data set. For our example, let the arguments to the EXTEND function be the trivial Chord $c_p = A$ and the set of super-Chords be $S = \{AB, AC\}$. For the first extension $\hat{c}_i = A \cup AB = AB$ the support of $\hat{c}_i \cup c_j = AB \cup AC = ABC$ needs to be compared to the support of $\hat{c}_i = AB$ and $c_j = AC$.

The first case in Line 8 is shown in Figure 13(a). The super-Chord ABC has almost the same support as both sub-Chords. When AB or AC are observed, so is almost always ABC . We can effectively replace both sub-Chords with the super-Chord by adding $c_j = AC$ to the current extension \hat{c}_i and excluding it from further processing in the inner loop.

The second case in Line 11 is shown in Figure 13(b). While AB has the same support as ABC , AC has significantly more. Again we can add AC to the current extension \hat{c}_i because whenever AB occurs, so does AC . We can not delete AC from S in this case, however, because combinations with other Chords still need to be considered.

The third case in Line 13 is shown in Figure 13(c). The Chord AB has significantly more support than ABC , while AC does not. The super-Chord ABC is possibly a closed Chord, it needs to be added to the set of immediate super-Chords of $\hat{c}_i = AB$ for the recursive function call. The Chord AC can be excluded from further processing, because whenever it is observed, so is AB .

The last case in Line 16 is shown in Figure 13(d). Both AB and AC have significantly more support than ABC . ABC is added to the set of possibly closed super-Chords for the recursion and no pruning can be performed.

In Zaki and Hsiao (2005) the CHARM algorithm is reported to scale up linearly with the number of transactions. We observed similar behavior with Algorithm 5.1. The parameter δ needs to be chosen according to the application domain. The minimum duration of a Chord needs to be long enough for an expert to consider the coincidence of several properties meaningful. The minimum support and the threshold for margin-closedness directly control the number of Chords found. They should be set to larger values first and be made smaller if the results are too coarse.

5.2 Mining partial order

Phrases are similar to Episodes (Mannila et al., 1995) but describe a partial order of time intervals instead of time points. The mining of margin-closed Phrases can be performed in several steps similar to the closed partial orders of Casas-Garriga (2005). Algorithm 5.2 shows the high level overview of our proposal explained in the forthcoming Sections.

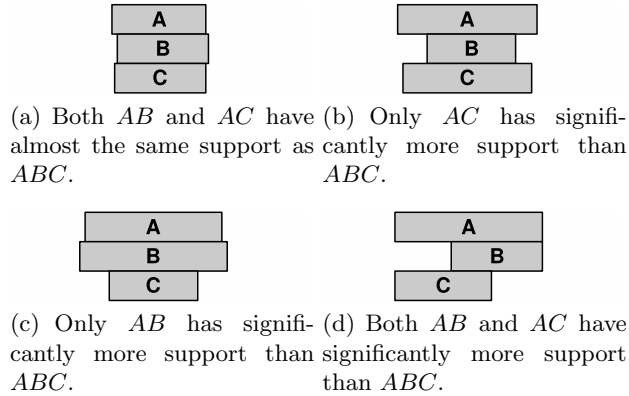


Fig. 13. The four cases of Algorithm 5.1 for mining margin-closed Chords. The support of the Chords AB and AC is compared to the support of the common super-Chord ABC .

Data model conversion In order to apply algorithms from itemset mining, we convert the symbolic interval sequence of Chord occurrences to an itemset interval series. Let \mathcal{C} be the occurrences of a set of Chords $C = \{c_j = (\sigma_j, \lambda_j, \phi_j) | j = 1, \dots, M\}$. We convert \mathcal{C} to an itemset interval series by considering each interval of minimum duration where at least one Chord is valid as an itemset time interval and including the symbols of all currently valid Chords as items to obtain \mathcal{I} .

In Figure 14 we give an example for the conversion of the interval sequence of overlapping Chord occurrences to a series of non-overlapping itemset intervals. The top rows show the occurrence intervals of three Tones indicated by the uppercase letters A , B , and C . The maximal occurrences of all Chords with at least two Tones are shown in the middle rows. Each Chord is assigned a unique symbol in form of a lower case letter ($a-d$). The bottom row shows the intervals between all interval boundaries of all Tones. On each interval an itemset is created including the symbols of all currently active Chords.

Closed sequential pattern mining The first step of the approach in Casas-Garriga (2005) consists of mining closed sequential patterns with an algorithm of choice. Sequential patterns describe a total order among the itemsets in the pattern. Closedness of sequential patterns is defined based upon support, i.e., a sequential pattern is closed if there is no super pattern that occurs in the same transactions. Therefore the single long itemset interval series needs to be implicitly converted to a set of shorter sub series by creating a sequence of intervals. Each itemset sequence is called a transaction and support is defined as the number of transactions in which a pattern occurs.

Any algorithm for closed sequence mining, e.g., CLOSPAN (Yan et al., 2003) or BIDE (Wang and Han, 2004), can then be used with only one modification. Recall, that we excluded overlapping Chord intervals with an order relation

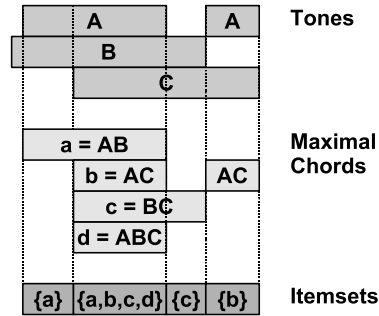


Fig. 14. Conversion of an interval sequence to an itemset interval series.

from Phrase patterns in Definition 15. We therefore need to restrict the mining to picking at most one item per itemset. This will also speed up the mining of closed sequential patterns, because it reduces the search space to total orders of single items. As a result we obtain pairs (s, T) composed of a closed sequential pattern s occurring in the transaction windows T .

Creating margin-closed groups of sequential patterns The mining of margin-closed partial orders (Step 3) is the largest deviation from Casas-Garriga (2005) where pairs are formed consisting of a set of closed sequential patterns and the list of transactions in which they all occur. The pairs were required to be maximal in the sense, that there is no additional sequential pattern, that occurs in all transactions and no additional transaction in which all patterns occur. Furthermore, the sets of closed sequential patterns were required to be non-redundant, i.e., no sequential pattern is allowed to be a sub-pattern of a pattern in the same set. The sequential patterns were grouped by first considering all patterns with the same list of transactions and then adding any pattern with a superset of transactions that do not make the group redundant. Such groups are closed w.r.t. frequency and form a concept lattice. Each group of closed sequences can then be converted to a partial order.

When adding the constraints for margin-closed pairs, this algorithm cannot be used anymore. For each group not only sequential patterns with the same or longer transaction lists need to be considered, but also patterns with only slightly shorter lists. For a valid group of patterns, the intersection of all transaction lists is allowed to be at most 100α percent shorter than the longest. Since all sequential patterns in a group influence the intersection, the search space for the groups consists of all 2^k combinations of k closed sequences. This corresponds to the problem of finding margin-closed itemsets, if we consider each closed sequential pattern an item and each group of sequential patterns an itemset. We can therefore again adapt the CHARM (Zaki and Hsiao, 2005) algorithm to tackle this problem efficiently.

Only minor modifications needed to be made to Algorithm 5.1, we therefore refer the interested reader to Mörchen (2006c) for the complete listing. Accord-

ing to the definition of Phrases in Section 3, the support counting is based on frequency not on duration as for Chords. In Casas-Garriga (2005) each group of closed sequential patterns is required to be non-redundant. Instead of checking this upon each combination of a group with possible extensions, we decided to implicitly include all subsequences of a sequence in a group and remove this redundancy in a post-processing step for each group.

Creating partial order from group of sequential patterns For the last step of converting a set of sequences into a partial order necessary conditions are given in Casas-Garriga (2005) making the construction of the final Phrase representation straightforward, see Mörchen (2006c) for a simple algorithm.

6 Experiments

We demonstrate the efficacy and efficiency of the TSKM on two datasets. We further compare the resulting patterns to results obtained using Allen’s relations. For mining patterns expressed with Allen’s relations we used the format of Höppner (2003) to avoid ambiguity and counted support as occurrences in windows to be comparable with the Phrase mining using the same windows. We further pruned the set of patterns applying the concept of margin-closedness in a brute force post-processing step.

6.1 Skating Data

The Skating data was collected from tests in professional In-Line Speed Skating Hoos (2003). An athlete performed a standardized indoor test at a speed of 7,89m/s on a large motor driven treadmill. EMG (Electromyography) and kinematic data were measured for 30 seconds corresponding to 19 movement cycles. The obtained multivariate time series was converted to a 5-dimensional symbolic interval series. The preprocessing was the same as in Mörchen et al. (2006), except that we used the Persist algorithm (Mörchen and Ultsch, 2005) for time series discretization instead of density estimates. Three interval series describe the activation of leg muscles mainly responsible for forward propulsion: Medial Gastrocnemius (*Gastroc*), Vastus Medialis (*Vastus*), and Gluteus Maximus (*Gluteus*) with Tones *low*, *high*, and *very high*. One series describes the foot contact with Tones *on* and *off* obtained from a pressure sensor in the skate. The last series describes the movement phases obtained by clustering a 6-dimensional time series with angles and angular speeds of the hip, knee, and ankle joints. The Tones were labeled *swing*, *glide+push*, and *recovery*. Non-overlapping windows corresponding to the 19 movement cycles of the experiment were used. The goal was to identify typical muscle activation patterns during the cyclic movement with complex inter-muscular coordination patterns. The knowledge gained from the temporal patterns can be used to further analyze the underlying mechanisms of an athletic performance and, e.g., help to optimize the training process. The experimental results were evaluated by an expert.

For the TSKR patterns we first mined Chords with a minimum duration of 50ms. Shorter phenomena related to muscle activation are physiologically not plausible. The minimum size was set to 3, the maximum size is naturally bounded by the 5 dimensions of the interval series. With a minimum support of 2% a total of 70 Chords was found. Restricting the patterns to closed Chords reduced the number to 60, mining margin-closed Chords with $\alpha = 0.1$ returned 18 patterns. At this point it is difficult to decide automatically whether rare large or small frequent Chords are better. The lattice of these Chords in Figure 15 was annotated with frequency (#) and support (%) and presented to the expert. The 9 bold Chords were selected as the most interesting and labeled accordingly while 3 uninteresting Chords with no relation to other Chords are omitted from the figure.

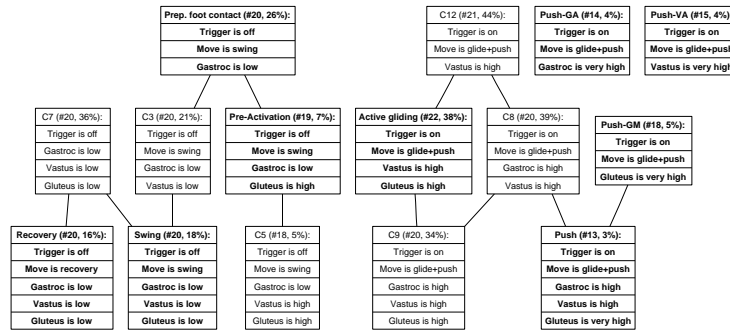
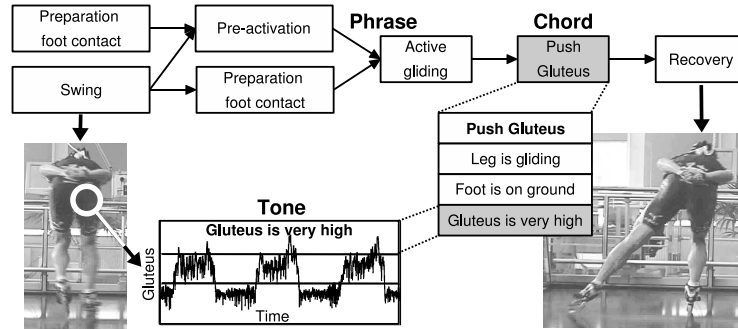


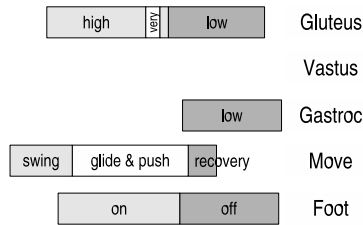
Fig. 15. TSKR representation of coincidences: Lattice of margin-closed Chords for Skating data.

Using these 9 Chords we found 30 closed sequences with a minimum length of 2 and a minimum frequency of 10. Merging groups of closed sequences resulted in 20 closed Phrases, when using $\alpha = 0.1$ 15 margin-closed Phrases were obtained. Again we presented the lattice of Phrases to the expert who selected the most specific and the second most general patterns as the most interesting. The general Phrase present in 17 out of the 19 movement cycles is shown in Figure 16(a) with images from the skating experiments. The advantages of the hierarchical structure of the TSKR is shown by unfolding one Chord from the Phrase to show the coinciding Tones and one Tone to show the original numerical time series with the thresholds for discretization. The more detailed Phrase shown in Figure 17 was observed in 10 cycles, it describes a total order of 8 Chords and provides details on the successive activation of the three major leg muscles.

The pruning by margin-closedness largely reduced the number of patterns so they could easily be analyzed manually to trade off pattern size vs. pattern frequency. We analyzed the search space of Chords in dependence of the parameters to quantify the pruning effects. In Figure 18 the number of Chords found is shown on the left for different values of the parameter α determining



(a) Phrase explaining most observed movement cycles. Chords (e.g. *Push Gluteus*) can be expanded to show the coinciding Tones. Tones (e.g. *Gluteus is very high*) are obtained from the numerical sensor time series.



(b) One instance of the largest Allen pattern. The 10 intervals are connected with 45 pairwise relations.

Fig. 16. Patterns found in inline skating data.

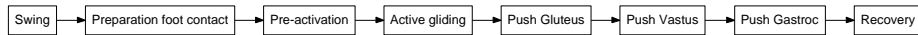


Fig. 17. Detailed Phrase of skating data with additional information on muscle activation.

margin-closedness. The value of 0.1 as used for the analysis above, effectively prunes the size of the result down to a manageable number of patterns that can be analyzed manually. Using lower values would result in a dramatic increase of patterns, using higher values results in less than five patterns for $\alpha > 0.18$. We also measured the number of recursive calls to the EXTEND function as an estimate of the run time of the algorithm independent of implementation and hardware. As is visible on the right of Figure 18 the concept of margin-closedness does not only reduce the size of the result set, it also effectively prunes the search space during mining and speeds up the computation. This is because the notion of margin-closedness more often enables the activation of the cases 1-3 in Algorithm 5.1 that have a pruning effect. For $\alpha \geq 0.2$ less than 10% of the function calls of strict closed Chord mining are needed.

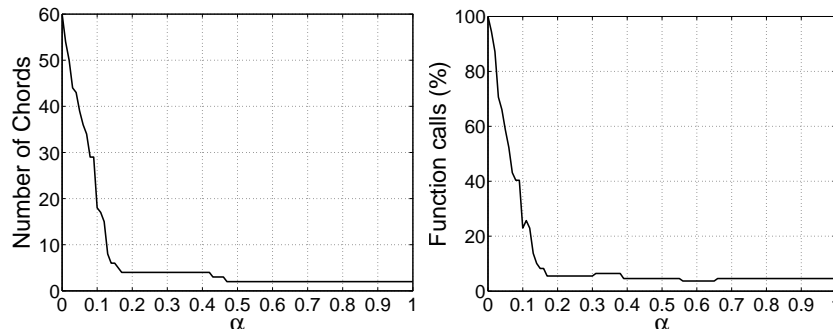


Fig. 18. Effects of different values of α determining margin-closedness on the number of Chords (left) and the complexity of the search (right).

We searched the same data for patterns expressed with Allen’s relations using a minimum frequency of 10 and a minimum size of 2 resulting in 4208 distinct patterns. Pruning by enforcing margin-closed patterns with $\alpha = 0.1$ returned 487 patterns. There were 8 patterns of size 9 and one of size 10. The analysis was performed by the expert looking at an example of each pattern as shown for the size 10 pattern in Figure 16(b) because the listing of all 45 or 36 pairwise relations did not offer a comprehensible description.

Looking at a pattern instance is almost equivalent to looking at the input data before even mining patterns. The only benefit is that the intervals somewhat represent the most dominant structure of the 19 movement cycles. In order to avoid the selection of an ambiguous pattern one has to look at all instances, though. While the relation of the Gluteus muscle activity to the movement phases was recognized as valid, the fact that no information on the Vastus muscle was given and the Gastroc muscle appeared only with low activity rendered the pattern useless to the expert. The fact that the three movement phases appear in the displayed order and the two Foot states alternate is already obvious from the input data and offers no information gain.

Because a correspondence of muscle activity to movement related information was sought we filtered the pattern set explicitly for patterns containing the states *high* and *very high* for the *Vastus* and *Gluteus* muscles. None of the 5 patterns we found contained useful information on the foot contact. Further analysis with less restrictive constraints suggested, that patterns with many muscle states offer only very limited information on movement and foot contact and the other way around. The relation of muscle related interval series to movement related interval series, all of which are subject to noise from the recording process and the discretization procedure, did not seem to be well representable with Allen's relations.

6.2 Video Data

The video data³ consists of symbolic intervals data describing different scenarios involving a hand moving colored blocks that were generated by the LEONARD system for recognition of visual events from video camera input (Siskind, 2001). The scenes include simple actions like putting one block on another and more complex scenes where a whole stack of blocks is built. Using image segmentation and motion tracking algorithms a description of each video in terms of primitive events valid on time intervals was generated (Siskind, 2001). In Fern (2004) each scene is described with a logical formula found by inductive logic programming in a supervised process. We use the data in an unsupervised manner and use the ground truth on the scenarios for evaluation purposes only. We preprocessed the descriptions by normalizing the argument order, filtering out some redundant descriptions, and merging scenarios that were equivalent.

We mined Chords with a minimum support of 1%, minimum size of 1, and a minimum count of 10. We found 19 closed Chords, with $\alpha = 0.1$ this reduced to 15 margin-closed Chords. The co-occurrences of the Chords with the known scenario were striking, a selection is listed in Table 1. The first two Chords each describe a stack of three blocks with two *contacts* relations. Without looking at the original Video data, it is unclear, however, which block sits on top and which at the bottom. Further, the Chord patterns are not sufficient, to discriminate the scenarios, because they always appear in at least two different scenes that are reversed versions of one another, e.g., *stack* and *unstack*. In order to distinguish each pair, we need patterns expressing an order among the Chords.

Using a minimum frequency of 12, 20 closed sequential patterns were found and grouped into the 10 margin-closed Phrases ($\alpha = 0.1$). The co-occurrences of the Phrases with the known scenarios showed an almost perfect correspondence, see Table 2 for a selection. They summarize the relative positions of 7-10 symbolic intervals.

The Phrases further *explain* the actions in the videos. The Phrase for the *disassemble* scenario is shown with pictures from the original Videos in Figure 19. All other scenarios are described similarly, see Mörchen (2006c) for all Phrases.

³ <ftp://ftp.ecn.purdue.edu/qobi/ama.tar.Z>

Scenes/Chords	BLUE-CONTACTS-GREEN	BLUE-CONTACTS-GREEN	BLUE-CONTACTS-GREEN
	BLUE-CONTACTS-RED	GREEN-CONTACTS-RED	HAND-HOLDS-RED
<i>stack</i>	0	28	30
<i>unstack</i>	0	29	30
<i>assemble</i>	14	0	34
<i>disassemble</i>	30	0	30

Table 1. Co-occurrences of Chords and scenes.

Scenes/Phrases	P1	P2	P3	P4
<i>stack</i>	0	0	28	0
<i>unstack</i>	0	28	0	0
<i>assemble</i>	0	0	0	28
<i>disassemble</i>	30	0	0	0
intervals	10	7	7	8

Table 2. Co-occurrences of Phrases and scenes.

With Allen’s relations, a minimum size of two, and a minimum frequency of 10, 363 patterns were found, 174 of which were margin-closed for $\alpha = 0.1$. This large amount of patterns could not be analyzed manually, some filtering needed to be applied.

We first looked at the 14 largest patterns with five or more intervals. Patterns of these sizes were *only* observed within the (*dis*)*assemble* scenes with less than 21 out of 30 repetitions. The patterns for the *assemble* scene further described only fragments of the true temporal phenomena. None of them contained any information about the hand taking the red block and placing it on top of the stack. We alternatively filtered the mining results by a minimum frequency of 24 and a minimum size of three. A selection of these frequent patterns is given in Table 3. The correspondence with the known scenarios is decent but mostly worse than that of the Phrases in Table 2.

The Allen patterns are further only composed of three symbolic intervals each, not sufficiently explaining the complex events of these scenes. As an example we show pattern A1 found in 26 out of 30 of the *disassemble* scenes in Figure 20. The intervals only correspond to the first three video frames of Fig-

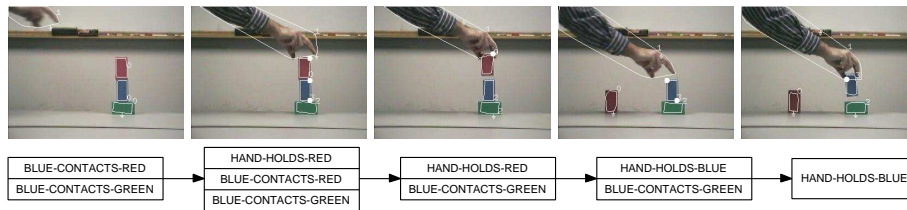


Fig. 19. Video frames (Fern 2004) and Phrase for *disassemble* scene.

Scenes/Allen	A1	A2	A3	A4	A5
<i>stack</i>	0	20	16	0	0
<i>unstack</i>	0	0	0	27	0
<i>assemble</i>	0	5	0	0	29
<i>disassemble</i>	26	0	0	0	0
intervals	3	3	3	3	3

Table 3. Co-occurrences of frequent Allen patterns and scenes.



Fig. 20. Example of frequent Allen pattern A1 explaining only fragments of the *disassemble* scene in Figure 19.

Figure 19 and fail to mention important parts of the scene. The TSKR Phrases consist of up to 10 intervals providing much more information.

Only for simpler scenes like *pick-up* or *put-down* three intervals can adequately describe the actions in the video. A detailed analysis of the pattern for *put-down*, however, revealed four very similar patterns with the same intervals but slightly different relations (see Figure 21). Only the first combination is frequent enough to appear in the mining result, the others would commonly be filtered out from the large amount of patterns found.

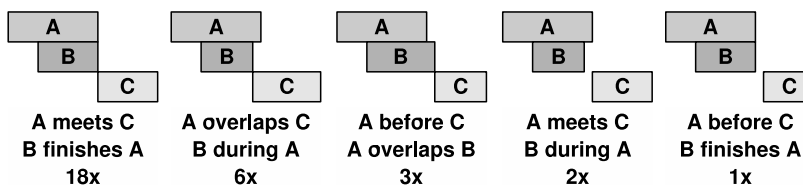


Fig. 21. Five similar patterns for *put-down* only one of which is frequent. *A* is the hand holding the red block, *B* is the placing of the red on top of the green block, *C* is the red on the green block without the hand.

Apart from the more specific explanations that the TSKR patterns provide, they are also almost always more precise and more distinctive than the Allen patterns of Höppner (2003), as shown in Table 4. For each video scene, precision and recall of the best pattern from each representation was calculated. Only for the most complex scenarios in the last two rows the numbers of Allen/Höppner are better. In these cases the TSKR patterns are much more complex and provide better insight into the temporal phenomena of the scene sacrificing some discrimination power for explanation.

Scenes	TSKR		Allen/Höppner	
	Precision	Recall	Precision	Recall
<i>pick-up</i>	100.0	100.0	42.3	36.7
<i>put-down</i>	97.8	97.8	41.9	60.0
<i>stack</i>	100.0	93.3	80.0	66.7
<i>unstack</i>	100.0	93.3	100.0	90.0
<i>assemble</i>	100.0	93.3	100.0	100.0
<i>disassemble</i>	100.0	80.0	76.5	86.7

Table 4. Precision and recall for Allen and TSKR patterns.

7 Discussion

For a successful application of the TSKR the temporal process producing the symbolic intervals should not be chaotic such that the properties described by Tones can be repeatedly observed during intervals of a minimum length meaningful to the analyst. When starting with numerical data and using discretization or segmentation, the parameters of these methods should be chosen with care incorporating prior knowledge when possible. This and the further manual interaction during the mining steps and possible revision of parameters for mining Chords and Phrases can be seen as a weakness of our method. We believe that the discovery of knowledge is inherently interactive as has also been acknowledged for other data mining tasks like classification (Ankerst et al., 2000) and clustering (Aggarwal, 2001).

The most important parameters of the TSKM evolve around the concept of the minimum duration of a temporal phenomenon to be considered meaningful. Tones need to coincide for a certain time to be considered a Chord. An occurrence of a Chord within a Phrase must have a certain duration. We believe that an intuition on the selection of these parameters will be available in many applications. Other parameters like the minimum support or the margin threshold influence the number of patterns found but not the qualitative aspects thereof. Once a good combination of preprocessing methods and parameters has been found, the TSKM steps can be automated.

The Phrases found in the Skating data were considered valid and interesting by the expert. Of particular importance is the connection of external variables describing the movement with the internal observations on muscle activation that could not be discovered with Allen’s relations. The application of the TSKM with the final graphical representation of the Phrases offered a great benefit over simply viewing the input time series, as is common practice in the analysis of such data in sports medicine. The descriptions found by the TSKM in the Video data in an unsupervised process directly explained the known experimental setups. The Allen patterns found mostly explained only fragments of the scenarios and showed less correspondence to the ground truth. This was partly explained by pattern fragmentation.

In Höppner (2003) the rule sets are defragmented with disjunctions of patterns but this makes the result even longer and harder to understand. Take for

example a rule disjunctively combining the 5 cases of Figure 21. Another solution to the problem is to use thresholds, effectively merging the problematic patterns. This does not solve the problems of ambiguity and comprehensibility, however, and poses the problem of threshold selection. Ambiguity could be reduced by splitting patterns with potential high variability into several different patterns, e.g., using the mid points (Roddick and Mooney, 2005). But using 49 instead of 13 relations with many additional conditions requiring equality of time points will in turn increase effects of pattern fragmentation.

The reasons for the TSKR being more expressive than Höppner’s representation using Allen’s relation is the permission of partial order and of subintervals in patterns. One could also mine partially related Allen patterns by allowing blanks in the matrix of pairwise relations. This would make the mining even more costly. Further, imagine being given a listing of 10 pairwise relations of 5 intervals and the task to draw a qualitative example of the pattern. This is not at all a trivial task, many dependencies need to be considered. In fact, this corresponds to the constraint satisfaction problem of temporal reasoning which is NP-complete (Vilain et al., 1989). The same task is rather easy given a description in the TSKR language and shouldn’t this be the case in order to call a pattern *understandable*? It would be interesting to evaluate this with psychological studies.

There might be applications where the semantics of Allen’s relations are explicitly desired. The disadvantages in robustness are not an issue if the data is not noisy and the exact boundaries of intervals are important. Ambiguity caused by quantitative variation is in a way also application dependent. If the three different versions of *overlaps* show in Figure 2 actually do describe different observation of the same meaningful concept, Allen’s relations offer a valid abstraction for them. If the examples in Figure 8(b) truly have a different meaning than the examples in Figure 8(a), because they are not caused by noise or inaccurate interval boundaries but rather by some property of the data generating process, Allen patterns would be more accurate. The corresponding TSKR pattern would have the same recall but lower precision because it is more general. To avoid ambiguity of the pattern language, the format of Höppner (2003) should be used. Many of the disadvantages in interpretability still apply, but for patterns involving just a few intervals a graphical representation of an exemplary pattern observation as in Figure 16(b) could be used to represent the patterns intuitively. Note that the TSKR can also be used for data where the interval boundaries matter, e.g., by using a minimum Chord duration of one.

Early approaches of mining patterns with Allen’s relations are all based on the Apriori principle from association rule mining (Agrawal et al., 1993). Recently, more efficient algorithms adapted from itemset and sequential pattern mining have been proposed (Papaterou et al., 2005; Winarko and Roddick, 2007). Similarly, the TSKM patterns can be found with modified versions of efficient depth-first algorithms like CHARM (Zaki and Hsiao, 2005), CloSpan (Yan et al., 2003), and DCI-Closed (Lucchese et al., 2006). CHARM was one of the best methods in a recent evaluation of closed itemset algorithms (Yahia

et al., 2006). The adaption of other algorithms to the interval data models and margin-closedness could be worthwhile on some datasets. Similar, the recently proposed FRECPO algorithm for direct mining of partial order (Pei et al., 2005, 2006) could also be used to mine Phrases.

Phrases are similar in structure to sequential patterns. Each time point of a symbolic interval sequence could be considered a transaction with an itemset holding one item for each symbolic interval that contains the time point. This leads to very long transactions (e.g. about 1,500 itemsets per transaction in the Skating data) that are highly redundant. Naively applying standard sequential pattern mining techniques to this data would be a large waste of computing resources. Using our novel data model conversion to itemset intervals greatly reduces this redundancy. Test with algorithms for sequential pattern were, however, still extremely slow compared to the search for Chords and Phrases that seems to highly profit from the semantically motivated search space restrictions.

The search for (closed) frequent itemsets as applied to mining Chords and Phrases can become infeasible for a large number of items. For Chords this corresponds to the dimensionality of the time series and the number of Tones for each dimension. For Phrases it is the number of closed sequential patterns found. The novel concept of margin-closedness helps to partly alleviate this complexity. It serves as a sampling of the results and a pruning of the search space to avoid redundancy. A manageable amount of patterns is returned representing a variety ranging from general and frequent to specific and rare patterns. Coarse results can be found very fast and refined if necessary.

To the best of our knowledge margin-closedness (Mörchen, 2006c,a) represents a novel way of lossy compression for frequent itemset mining. The δ -tolerance itemsets of Cheng et al. (2006) represent the same concept and were discovered simultaneously. Condensed itemset representations were developed to derive the support of all itemsets from a compact summary exactly (Bykowski and Rigotti, 2001; Kryszkiewicz, 2001; Calders and Goethals, 2003) or approximately (Pei et al., 2002; Boulicaut et al., 2003). The basic idea is to derive the support of an itemset given the support of all subsets (Calders and Goethals, 2003). If this is possible, the larger set does not need to be stored in the result. Similarly, in (Pudi and Haritsa, 2003) all supersets with approximately the same support as a smaller itemset are pruned. In contrast, we prune the smaller subsets with support similar to a larger, more specific itemset. This favors more detailed, thus generally more interesting patterns. Another line of work is motivated by the fact, that transaction data is often noisy. The strict definition of support, requiring all items of an itemset to be present in a transaction, is relaxed (Pei et al., 2001; Yang et al., 2001; Seppänen and Mannila, 2004; Afrati et al., 2004; Yan et al., 2005). All these approaches have one thing in common - they report false information and thus violate the Gricean maxim of quality. They only report approximate support values and possibly list itemsets that are not present in the collection at all (Afrati et al., 2004) or with much smaller support. Margin-closed itemsets present true information, i.e., exact patterns with exact support. We merely reduce the amount of patterns reported.

In Casas-Garriga (2005) it is shown how grouping of closed sequential patterns leads to closed partial orders. It is tempting to assume that margin-closed sequential patterns will also lead to margin-closed partial orders, but simple examples show that this is not the case (see Mörchen (2006c) for details).

The mining of Phrases requires a windowing of the input data. If no windowing is given by the application overlapping sliding windows can be used instead. To avoid redundancy, the sequential patterns should then be restricted to start with an item from the first itemset in the window (Chen et al., 2005). For the grouping of sequential patterns the windows defined by the occurrences of the closed sequential patterns found can be used as transaction windows.

The TSKR was designed for unsupervised learning in noisy interval data. We believe that other data mining tasks like classification or anomaly detection can also profit from the high robustness and high understandability of the TSKR. The interval language used for supervised learning in Fern (2004) is similar as it considers total orders of coinciding intervals without gaps. It would be interesting to see if the inductive learning method would also work with the more general TSKR. Other possible extensions include the use of quantitative information (Höppner, 2003), generation of implication rules (Agrawal et al., 1993; Höppner, 2003; Bellazi et al., 2005; Winarko and Roddick, 2007), and fuzzy itemsets (Dubois et al., 2006).

8 Summary

We have presented a new language for the formulation of temporal knowledge in interval series. It is more robust, more expressive, and better interpretable than previously proposed approaches using Allen’s relations, in particular when dealing with noisy or inaccurate data. The TSKR patterns can be efficiently mined with methods adapted from itemset mining. The novel concept of margin-closedness enables the creation of concise pattern sets that explain various aspects of the temporal data ranging from more frequent and less specific to less frequent but more detailed descriptions. We have further demonstrated the superiority and usefulness of the TSKM by presenting two detailed practical examples from different application domains. Our methods are very general and will most likely be useful in other domains and for other data mining tasks like classification or prediction.

Acknowledgements: This research was done in the Databionic Research Group, Philipps-University Marburg, Germany. The first author was partly supported by Siemens Corporate Research, Princeton, NJ during this time. We thank Olaf Hoos for providing the Skating data and interpreting the results, and Alan Fern for providing and explaining the video data. We are also grateful for the constructive criticism of the reviewers for this paper and previous conference submissions.

Reproducible Results Statement: All datasets and code used in this work are available by emailing the first author.

Algorithm 5.1 Depth-first margin-closed Chord mining.

Input:

- Set of Tones T .
- Minimum Chord duration δ .
- Minimum Chord support sup_{min} (default 0.01).
- Minimum Chord size s_{min} (default 2).
- Maximum Chord size s_{max} (default ∞).
- Threshold α for margin-closedness (default 0.1).

Output:

- Set R of margin-closed Chords.

```

1:  $S := \{t \in T | sup(t) \geq sup_{min}\}$ 
2:  $R := EXTEND(\emptyset, S, \emptyset)$ 

    $EXTEND(c_p, S, R)$ 
3: for all  $c_i \in S$  do
4:    $\hat{c}_i := c_p \cup c_i$ 
5:    $\hat{S} := \emptyset$ 
6:   for all  $c_j \in S$  with  $j > i$  do
7:     if  $sup_{\delta}(\hat{c}_i \cup c_j) \geq sup_{min}$  then
8:       if  $\frac{sup_{\delta}(\hat{c}_i \cup c_j)}{\max(sup_{\delta}(\hat{c}_i), sup_{\delta}(c_j))} \geq 1 - \alpha$  then
9:          $\hat{c}_i := \hat{c}_i \cup c_j$ 
10:         $S := S \setminus \{c_j\}$ 
11:       else if  $\frac{sup_{\delta}(\hat{c}_i \cup c_j)}{sup_{\delta}(\hat{c}_i)} \geq 1 - \alpha$  and  $\frac{sup_{\delta}(\hat{c}_i \cup c_j)}{sup_{\delta}(c_j)} < 1 - \alpha$  then
12:          $\hat{c}_i := \hat{c}_i \cup c_j$ 
13:       else if  $\frac{sup_{\delta}(\hat{c}_i \cup c_j)}{sup_{\delta}(\hat{c}_i)} < 1 - \alpha$  and  $\frac{sup_{\delta}(\hat{c}_i \cup c_j)}{sup_{\delta}(c_j)} \geq 1 - \alpha$  then
14:          $\hat{S} := \hat{S} \cup \{\hat{c}_i \cup c_j\}$ 
15:          $S := S \setminus \{c_j\}$ 
16:       else
17:          $\hat{S} := \hat{S} \cup \{\hat{c}_i \cup c_j\}$ 
18:       end if
19:     end if
20:   end for
21:   if  $|\hat{c}_i| \geq s_{min}$  and  $\forall c \in R$  with  $\hat{c}_i \subseteq c$  holds  $\frac{sup_{\delta}(c)}{sup_{\delta}(\hat{c}_i)} < 1 - \alpha$  then
22:      $R := R \cup \{\hat{c}_i\}$ 
23:   end if
24:   if  $|\hat{c}_i| < s_{max}$  then
25:      $R := EXTEND(\hat{c}_i, \hat{S}, R)$ 
26:   end if
27: end for

```

Algorithm 5.2 High level algorithm to find margin-closed Phrases.

Input:

- Set of Chords C .
- Minimum duration ϵ of Chords in Phrase (default $\frac{\delta}{2}$).
- Minimum Phrase support sup_{min} (default 1).
- Minimum path length in Phrase s_{min} (default 2).
- Threshold α for margin-closedness (default 0.1).
- Transaction windows $W = \{[s_i, e_i] | i = 1, \dots, w\}$.

Output:

Set of Phrases.

- 1: Convert C to itemset interval series \mathcal{I} using ϵ .
 - 2: Mine pairs (s, T) composed of a closed sequential pattern s occurring within the transaction windows $T \subseteq W$ using sup_{min} and s_{min} .
 - 3: Create margin-closed maximal pairs (S, T) where S is the set of all closed sequential patterns occurring in all transaction windows $T \subseteq W$ using sup_{min} and α .
 - 4: Build partial order of Chords for each set S .
-

Bibliography

- F. Afrati, A. Gionis, and H. Mannila. Approximating a collection of frequent sets. In W. Kim, R. Kohavi, J. Gehrke, and W. DuMouchel, editors, *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*, pages 12–19. ACM Press, 2004.
- C. C. Aggarwal. A human-computer cooperative system for effective high dimensional clustering. In F. Provost and R. Srikant, editors, *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01)*, pages 221–226. ACM Press, 2001.
- R. Agrawal, T. Imielinski, and A. N. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia, editors, *Proceedings of the 1993 ACM SIGMOD International Conference on Management of data*, pages 207–216. ACM Press, 1993.
- M. Aiello, C. Monz, L. Todoran, and M. Worring. Document understanding for a broad class of documents. *International Journal on Document Analysis and Recognition*, 5(1):1–16, 2002.
- J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26(11):832–843, 1983.
- M. Ankerst, M. Ester, and H.-P. Kriegel. Towards an effective cooperation of the user and the computer for classification. In R. Ramakrishnan, S. Stolfo, R. Bayardo, and I. Parsa, editors, *Proceedings of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'00)*, pages 179–188. ACM Press, 2000.
- R. J. Bayardo. Efficiently mining long patterns from databases. In A. Tiwary and M. Franklin, editors, *Proceedings of the 17th ACM SIGMOD symposium on Principles of database systems (PODS'98)*, pages 85–93. ACM Press, 1998.
- R. Bellazi, C. Larizza, P. Magni, and R. Bellazi. Temporal data mining for the quality assessment of hemodialysis services. *Artificial Intelligence in Medicine*, 34:25–39, 2005.
- J.-F. Boulicaut, A. Bykowski, and C. Rigotti. Free-sets: A condensed representation of boolean data for the approximation of frequency queries. *Data Mining and Knowledge Discovery*, 7(1):5–22, 2003.
- A. Bykowski and C. Rigotti. A condensed representation to find frequent patterns. In W. Fan, editor, *Proceedings of the 20th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS'01)*, pages 267–273. ACM Press, 2001.
- T. Calders and B. Goethals. Minimal k -free representations of frequent sets. In N. Lavrac, D. Gamberger, H. Blockeel, and L. Todorovski, editors, *Proceedings of the 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'03)*, pages 71–82. Springer, 2003.
- G. Casas-Garriga. Summarizing sequential data with closed partial orders. In H. Kargupta, J. Srivastava, C. Kamath, and A. Goodman, editors, *Proceedings*

- of the 5th SIAM International Conference on Data Mining (SDM'05), pages 380–391. SIAM, 2005.
- G. Chen, X. Wu, and X. Zhu. Mining sequential patterns across data streams. Technical Report CS-05-04, University of Vermont, Burlington, VT, USA, 2005.
- J. Cheng, Y. Ke, and W. Ng. δ -tolerance closed frequent itemsets. In *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM'06)*, pages 139–148. IEEE Press, 2006.
- P. R. Cohen. Fluent learning: Elucidating the structure of episodes. In F. Hoffmann, D. Hand, N. Adams, D. Fisher, and G. Guimarães, editors, *Proceedings of the 4th International Conference in Intelligent Data Analysis (IDA'01)*, pages 268–277. Springer, 2001.
- D. Dubois, E. Hüllermeier, and H. Prade. A systematic approach to the assessment of fuzzy association rules. *Data Mining and Knowledge Discovery*, 13(2):167–192, 2006.
- A. Fern. *Learning Models and Formulas of a Temporal Event Logic*. PhD thesis, Purdue University, West Lafayette, IN, USA, 2004.
- A. Gionis, H. Mannila, and E. Terzi. Clustered segmentations. In *Workshop on Mining Temporal and Sequential Data, 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*, 2004.
- H. Grice. *Studies in the Way of Words*. Harvard University Press, 1989.
- G. Guimarães. *Eine Methode zur Entdeckung von komplexen Mustern in Zeitreihen mit Neuronalen Netzen und deren Überführung in eine symbolische Wissensrepräsentation*. PhD thesis, Philipps-University Marburg, Germany, 1998. German.
- G. Guimarães and A. Ultsch. A symbolic representation for pattern in time series using definitive clause grammars. In R. Klar and O. Opitz, editors, *Proceedings of the 20th Annual Conference of the German Classification Society (GfKI'96)*, pages 105–111. Springer, 1997.
- G. Guimarães and A. Ultsch. A method for temporal knowledge conversion. In D. J. Hand, J. N. Kok, and M. R. Berthold, editors, *Proceedings of the 3rd International Conference in Intelligent Data Analysis (IDA'99)*, pages 369–380. Springer, 1999.
- O. Hoos. *Bewegungsstruktur, Bewegungstechnik und Geschwindigkeitsregulation im ausdauerorientierten Inline-Skating*. Görlich & Weiershäuser, Marburg, Germany, 2003. German.
- F. Höppner. Discovery of temporal patterns - learning rules about the qualitative behaviour of time series. In L. D. Raedt and A. Siebes, editors, *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD'01)*, pages 192–203. Springer, 2001.
- F. Höppner. *Knowledge Discovery from Sequential Data*. PhD thesis, Technical University Braunschweig, Germany, 2003.
- F. Höppner and F. Klawonn. Finding informative rules in interval sequences. *Intelligent Data Analysis*, 6(3):237–255, 2002.
- P.-S. Kam and A. W.-C. Fu. Discovering temporal patterns for interval-based events. In Y. Kambayashi, M. K. Mohania, and A. M. Tjoa, editors, *Proceed-*

- ings of the 2nd International Conference on Data Warehousing and Knowledge Discovery (DaWaK'00), pages 317–326. Springer, 2000.
- E. Keogh, S. Chu, D. Hart, and M. Pazzani. Segmenting time series: A survey and novel approach. In M. Last, A. Kandel, and H. Bunke, editors, *Data Mining In Time Series Databases*, chapter 1, pages 1–22. World Scientific, Singapore, 2004.
- M. Kryszkiewicz. Concise representation of frequent patterns based on disjunction-free generators. In N. Cercone, T. Lin, and X. Wu, editors, *Proceedings of the 1st IEEE International Conference on Data Mining (ICDM'01)*, pages 305–312. IEEE Press, 2001.
- M. Last, Y. Klein, and A. Kandel. Knowledge discovery in time series databases. *IEEE Transactions on Systems, Man, and Cybernetics*, 31(1):160–169, 2001.
- J. Lin, E. Keogh, S. Lonardi, and P. Patel. Finding motifs in time series. In D. Hand, D. Keim, and R. Ng, editors, *Workshop on Temporal Data Mining, 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'02)*, 2002.
- J. Lin, E. Keogh, S. Lonardi, J. P. Lankford, and D. M. Nystrom. Visually mining and monitoring massive time series. In W. Kim, R. Kohavi, J. Gehrke, and W. DuMouchel, editors, *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*, pages 460–469. ACM Press, 2004.
- M.-Y. Lin and S.-Y. Lee. Fast discovery of sequential patterns by memory indexing. In Y. Kambayashi, W. Winiwarter, and M. Arikawa, editors, *Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery (DaWaK'02)*, pages 150–160. Springer, 2002.
- C. Lucchese, S. Orlando, and R. Perego. Fast and memory efficient mining of frequent closed itemsets. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):21–36, 2006.
- H. Mannila, H. Toivonen, and I. Verkamo. Discovery of frequent episodes in event sequences. In U. M. Fayyad and R. Uthurusamy, editors, *Proceedings of the 1st International Conference on Knowledge Discovery and Data Mining (KDD'96)*, pages 210–215. AAAI Press, 1995.
- C. Mooney and J. F. Roddick. Mining relationships between interacting episodes. In M. W. Berry, U. Dayal, C. Kamath, and D. B. Skillicorn, editors, *Proceedings of the 4th SIAM International Conference on Data Mining (SDM'04)*. SIAM, 2004.
- F. Mörchen. Algorithms for time series knowledge mining. In T. Eliassi-Rad, L. H. Ungar, M. Craven, and D. Gunopulos, editors, *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'06)*, pages 668–673. ACM Press, 2006a.
- F. Mörchen. A better tool than Allen's relations for expressing temporal knowledge in interval data. In T. Li, C. Perng, H. Wang, and C. Domeniconi, editors, *Workshop on Temporal Data Mining at the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 25–34, 2006b.
- F. Mörchen. *Time Series Knowledge Mining*. PhD thesis, Philipps-University Marburg, Germany, 2006c.

- F. Mörchen and A. Ultsch. Optimizing time series discretization for knowledge discovery. In R. Grossman, R. Bayardo, and K. P. Bennett, editors, *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'05)*, pages 660–665. ACM Press, 2005.
- F. Mörchen, A. Ultsch, and O. Hoos. Discovering interpretable muscle activation patterns with the Temporal Data Mining Method. In J.-F. Boulicaut, F. Esposito, F. Giannotti, and D. Pedreschi, editors, *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'04)*, Lecture Notes in Computer Science, pages 512–514. Springer, 2004.
- F. Mörchen, A. Ultsch, and O. Hoos. Extracting interpretable muscle activation patterns with Time Series Knowledge Mining. *International Journal of Knowledge-Based & Intelligent Engineering Systems*, 9(3):197–208, 2006.
- T. Palpanas, M. Cardle, D. Gunopulos, E. Keogh, and V. B. Zordan. Indexing large human motion databases. In M. A. Nascimento, M. T. Özsu, D. Kossmann, R. J. Miller, J. A. Blakeley, and K. B. Schiefer, editors, *Proceedings of the 30th International Conference on Very Large Data Bases (VLDB'04)*, pages 780–791. Morgan Kaufmann, 2004a.
- T. Palpanas, M. Vlachos, E. Keogh, D. Gunopulos, and W. Truppel. Online amnesic approximation of streaming time series. In *Proceedings of the 20th International Conference on Data Engineering (ICDE'04)*, pages 338–349. IEEE Press, 2004b.
- S. Papadimitriou, J. Sun, and C. Faloutsos. Streaming pattern discovery in multiple time-series. In K. Böhm, C. S. Jensen, L. M. Haas, M. L. Kersten, P.-Å. Larson, and B. C. Ooi, editors, *Proceedings of the 31st International Conference on Very Large Data Bases (VLDB'05)*, pages 697–708. Morgan Kaufmann, 2005.
- P. Papaterou, G. Kollios, S. Sclaroff, and D. Gunopoulos. Discovering frequent arrangements of temporal intervals. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM'05)*, pages 354–361. IEEE Press, 2005.
- N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *Proceeding of the 7th International Conference on Database Theory (ICDT'99)*, pages 398–416. Springer, 1999.
- J. Pei, A. K. Tung, and J. Han. Fault-tolerant frequent pattern mining: Problems and challenges. In *Workshop on Research Issues in Data Mining and Knowledge Discovery, 20th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems (PODS'01)*, 2001.
- J. Pei, G. Dong, W. Zou, and J. Han. On computing condensed frequent pattern bases. In *Proceedings of the 2nd IEEE International Conference on Data Mining (ICDM'02)*, pages 378–385. IEEE Press, 2002.
- J. Pei, J. Liu, H. Wang, K. Wang, P. S. Yu, and J. Wang. Efficiently mining frequent closed partial orders. In *Proceedings of the 5th IEEE International Conference on Data Mining (ICDM'05)*, pages 753–756. IEEE Press, 2005.

- J. Pei, H. Wang, J. Liu, K. Wang, J. Wang, and P. S. Yu. Discovering frequent closed partial orders from strings. *IEEE Transactions on Knowledge and Data Engineering*, 18(11):1467–1481, 2006.
- V. Pudi and J. R. Haritsa. Generalized closed itemsets for association rule mining. In U. Dayal, K. Ramamritham, and T. M. Vijayaraman, editors, *Proceedings of the 19th International Conference on Data Engineering (ICDE'03)*, pages 714–716. IEEE Press, 2003.
- C. Rainsford and J. Roddick. Adding temporal semantics to association rules. In J. M. Zytkow and J. Rauch, editors, *Proceedings of the 3rd European Conference on Principles of Data Mining and Knowledge Discovery (PKDD'99)*, pages 504–509. Springer, 1999.
- J. F. Roddick and C. H. Mooney. Linear temporal sequences and their interpretation using midpoint relationships. *IEEE Transactions on Knowledge and Data Engineering*, 17(1):133–135, 2005.
- E. Schwalb and L. Vila. Temporal constraints: A survey. Technical report, ICS, University of California at Irvine, CA, USA, 1997.
- J. K. Seppänen and H. Mannila. Dense itemsets. In W. Kim, R. Kohavi, J. Gehrke, and W. DuMouchel, editors, *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'04)*, pages 683–688. ACM Press, 2004.
- B. Shneiderman. The eyes have it: A task by data type taxonomy for information visualizations. In *Proceedings of the 1996 IEEE Symposium on Visual Languages*, page 336. IEEE Press, 1996.
- J. M. Siskind. Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. *Journal of Artificial Intelligence Research*, 15: 31–90, 2001.
- S. G. Sripada, E. Reiter, and J. Hunter. Generating English summaries of time series data using the Gricean maxims. In L. Getoor, T. E. Senator, P. Domingos, and C. Faloutsos, editors, *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*, pages 187–196. ACM Press, 2003.
- A. Ultsch. Unification-based temporal grammar. Technical Report 37, Department of Mathematics and Computer Science, Philipps-University Marburg, Germany, 2004.
- A. Ultsch. Eine unifikationsbasierte Grammatik zur Beschreibung von komplexen Mustern in multivariaten Zeitreihen. personal notes, 1996. German.
- A. Ultsch. Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 33–46. Elsevier, 1999.
- M. Vilain, H. A. Kautz, and P. G. van Beek. Constraint propagation algorithms for temporal reasoning: A revised report. In *Readings in Qualitative Reasoning about Physical Systems*, pages 373–381. Morgan Kaufmann, San Francisco, USA, 1989.
- R. Villafane, K. A. Hua, D. Tran, and B. Maulik. Knowledge discovery from series of interval events. *Journal of Intelligent Information Systems*, 15(1): 71–89, 2000.

- J. Wang and J. Han. BIDE: Efficient mining of frequent closed sequences. In *Proceedings of the 20th International Conference on Data Engineering (ICDE'04)*, pages 79–90. IEEE Press, 2004.
- E. Winarko and J. F. Roddick. ARMADA - An algorithm for discovering richer relative temporal association rules from interval-based data. *Data & Knowledge Engineering*, 2007.
- S. B. Yahia, T. Hamrouni, and E. Mephu Nguifo. Frequent closed itemset based algorithms: A thorough structural and analytical survey. *ACM SIGKDD Explorations Newsletter*, 8(1):93–104, 2006.
- X. Yan, J. Han, and R. Afshar. CloSpan: Mining closed sequential patterns in large datasets. In D. Barbará and C. Kamath, editors, *Proceedings of the 3rd SIAM International Conference on Data Mining (SDM'03)*, pages 166–177. SIAM, 2003.
- X. Yan, H. Cheng, J. Han, and D. Xin. Summarizing itemset patterns: a profile-based approach. In R. Grossman, R. Bayardo, and K. P. Bennett, editors, *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'05)*, pages 314–323. ACM Press, 2005.
- C. Yang, U. Fayyad, and P. S. Bradley. Efficient discovery of error-tolerant frequent itemsets in high dimensions. In F. Provost and R. Srikant, editors, *Proceedings of the 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'01)*, pages 194–203. ACM Press, 2001.
- M. J. Zaki and C.-J. Hsiao. CHARM: An efficient algorithm for closed itemset mining. In R. L. Grossman, J. Han, V. Kumar, H. Mannila, and R. Motwani, editors, *Proceedings of the 2nd SIAM International Conference on Data Mining (SDM'02)*, pages 457–473. SIAM, 2002.
- M. J. Zaki and C.-J. Hsiao. Efficient algorithms for mining closed itemsets and their lattice structure. *IEEE Transaction on Knowledge and Data Engineering*, 17(4):462–478, 2005.