

Temporal pattern mining in symbolic time point and time interval data

Fabian Moerchen
Siemens Corporate Research
Princeton, NJ

- I. Introduction**
- II. Symbolic temporal data models**
- III. Temporal concepts and operators**
- IV. Temporal patterns**
 - I. Time point patterns**
 - II. Time interval patterns**
- V. Patterns and algorithms for time interval data**
- VI. Related topics**

Introduction

Temporal Data Mining

Temporal Data Mining

- Any data mining task involving some dimension of time.
 - Includes temporal association rules, evolutionary clustering, spatio-temporal data mining, trajectory clustering, ...
- **Time Series Data Mining**
 - Mining of sequence(s) of observations over time
 - Clustering
 - Classification
 - Indexing
 - Anomaly detection
 - Prediction
 - ...
 - Temporal pattern mining
 - Temporal pattern languages
 - Pattern mining algorithms

Different aspects of temporal pattern mining

Temporal Data Models
How is temporal data represented?
Time point vs. time interval



Temporal Concepts
What are the desired semantics?
Order vs. concurrency

Temporal Operators
How are data elements
compared and combined by
the algorithms?

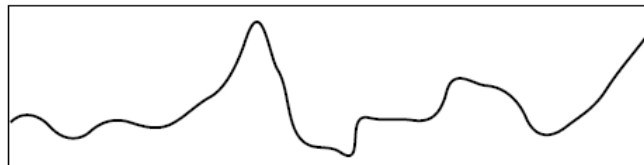
Temporal Data Models

- Time is continuous, computers are binary.
- We assume temporal data is represented with discrete **time points**.
- A **time series** is a set of unique time points.
- A **time sequence** is a multi set of time points.
- A pair of time points defines a **time interval**, inclusively.
- Two intervals **overlap** if there is at least one time point that lies within both intervals.
- An **interval series** is a set of non overlapping time intervals.
- An **interval sequence** can include overlapping and equal time intervals.
- The series data types can be **univariate** or **multivariate**.

Temporal Data Models

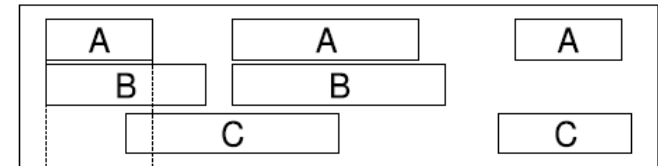
Common models and transformations

A **numeric time series** is a time series with numerical values for each time point.



Sensor data

A **symbolic interval sequence** has overlapping intervals with nominal values.

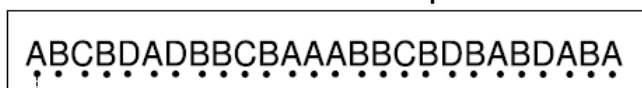


Sign language

Discretization
Segmentation
Motif discovery

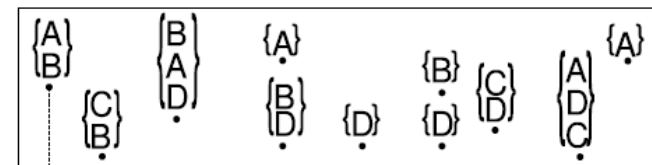
Discretization

A **symbolic time series** is a time series with nominal values for each point.



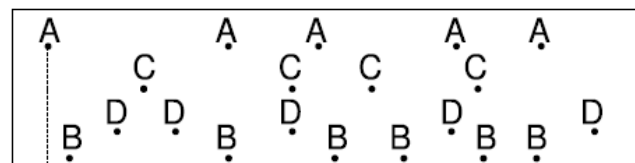
DNA sequences

An **itemset sequence** is a time sequence with sets of nominal values assigned to each time point.

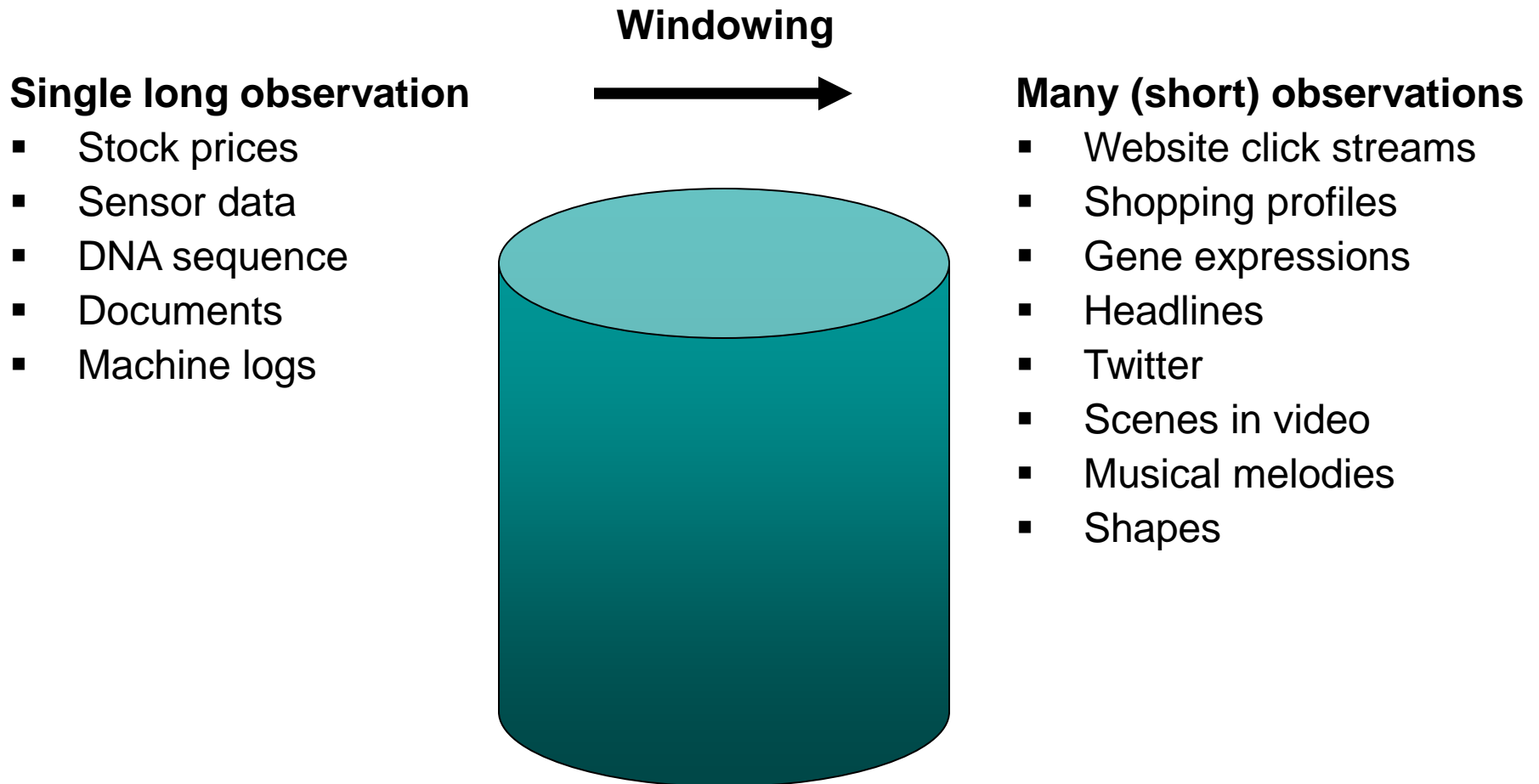


Shopping baskets for same customer

A **symbolic time sequence** has nominal values with possible duplicate time points



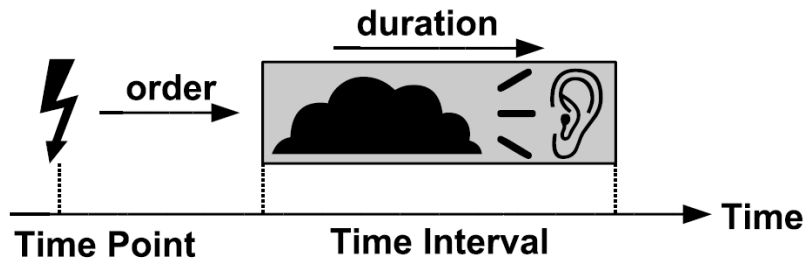
Events from machine service logs



Temporal Concepts and Operators

Temporal Concepts

Semantic categories for temporal operators



Duration is the persistence of an event over several time points.

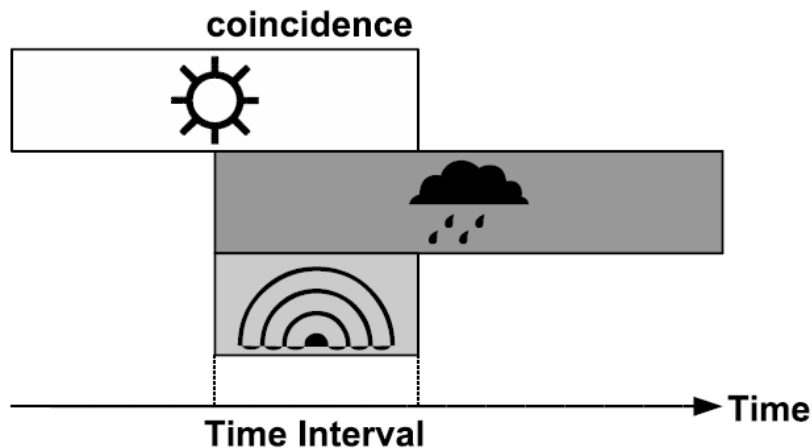
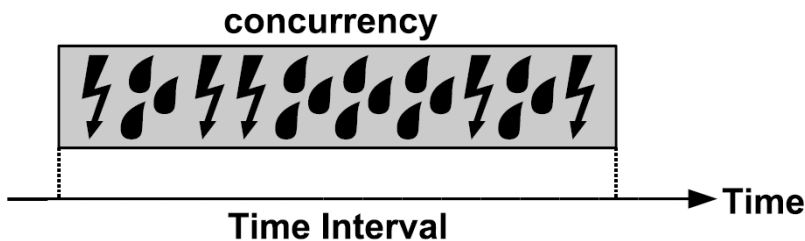
Order is the sequential occurrence of time points or time intervals.

Concurrency is the closeness of two or more temporal events in time in no particular order.

Coincidence describes the intersection of several intervals.

Synchronicity is the synchronous occurrence of two temporal events.

Periodicity is the repetition of the same event with a constant period.



- Time point operators expressing strict order: **before** / **after** (\rightarrow)



- Time point operator expressing concurrency: **close**



- Time point operator expressing synchronicity: **equals**

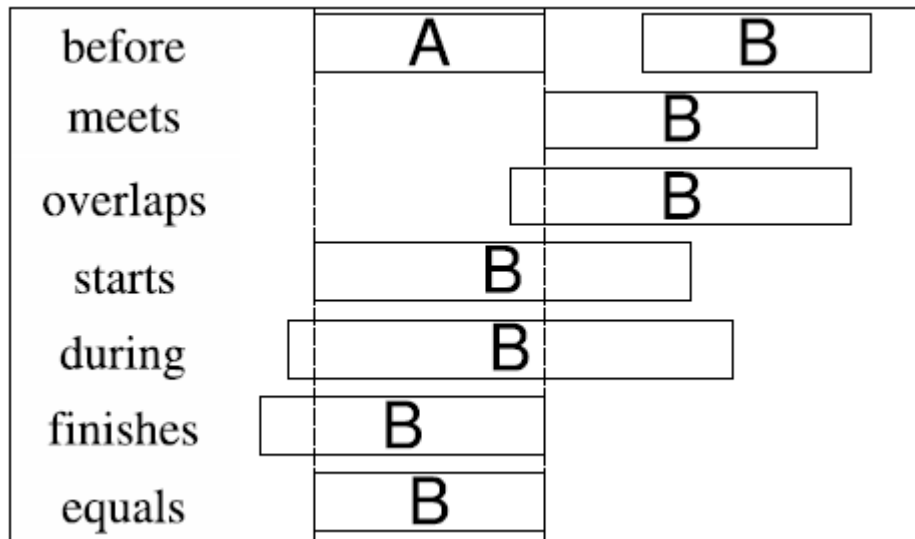


- Special cases
 - With constraints: **shortly before**, **closely after**
 - With specific granularity: **next business day**

- **Time interval operators**
 - Allen's interval relations [Allen 1983]
 - Freksa's semi-interval relations [Freksa 1992]
 - Reich's interval and interval point relations [Reich 1994]
 - Roddick's Midpoint interval relations [Roddick/Mooney 2005]
 - Other operators [Villafane 1999], [Ultsch 1996], [Moerchen 2006]

Temporal Operators

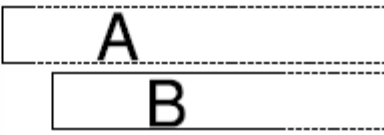
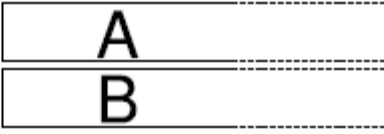
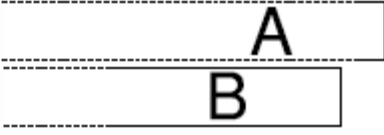
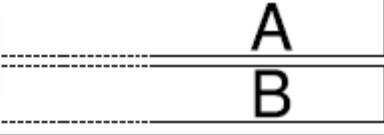
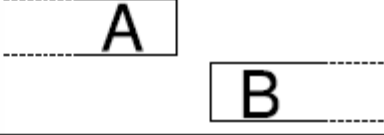
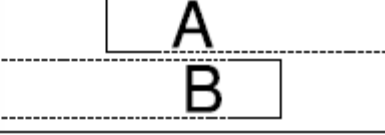
Allen's interval relations



- 13 relations forming an algebra.
- **Any two intervals have exactly one of the relations.**
- Invented in AI for temporal reasoning: given facts associated with time interval derive additional facts or answer specific questions.
- Later widely used in data mining.
- **Disadvantages for knowledge discovery!**
- Thresholds and fuzzy extensions solve some of the problems.

Temporal Operators

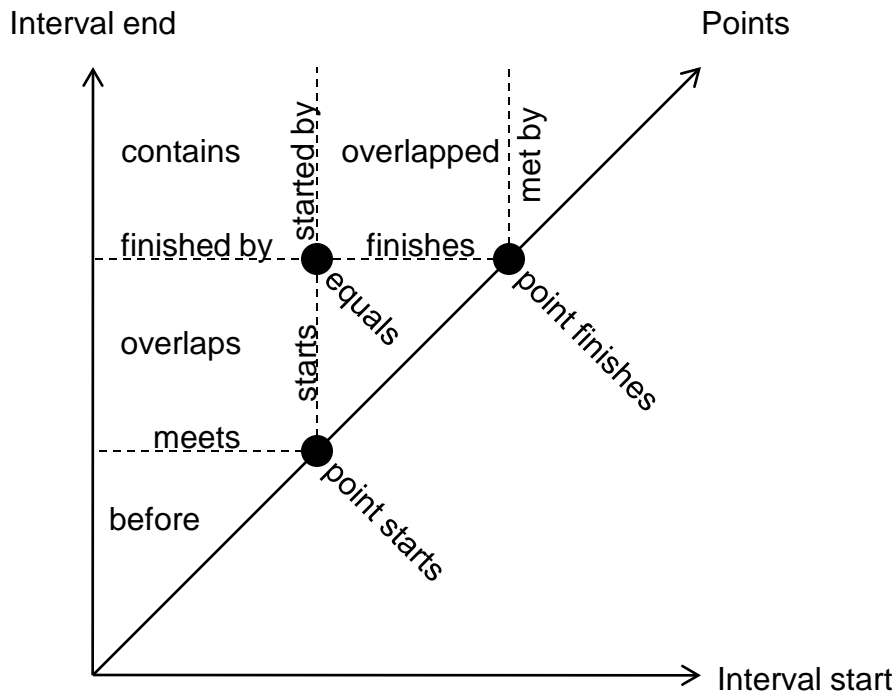
Freksa's semi-interval relations

older <i>younger</i>	
head to head	
survives <i>survived by</i>	
tail to tail	
precedes <i>succeeds</i>	
born before death <i>died after birth</i>	

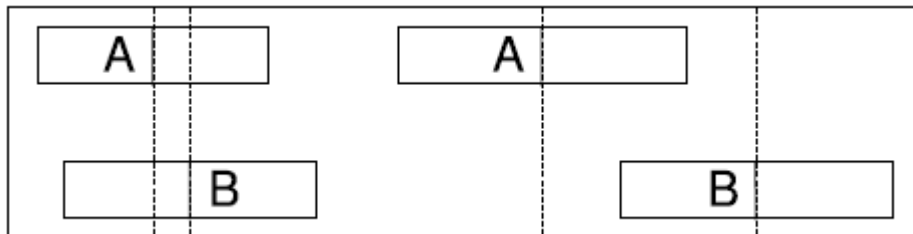
- **Semi-intervals:** one interval boundary unknown.
- Two relations between start or endpoints of the two intervals suffice to uniquely identify the relation.
- Easier to **represent incomplete or coarse knowledge.**
- Not widely used in data mining (yet).
 - [Rainsford/Roddick 1999]
 - [Moerchen/Fradkin 2010]

Temporal Operators

Reich's interval/point relations



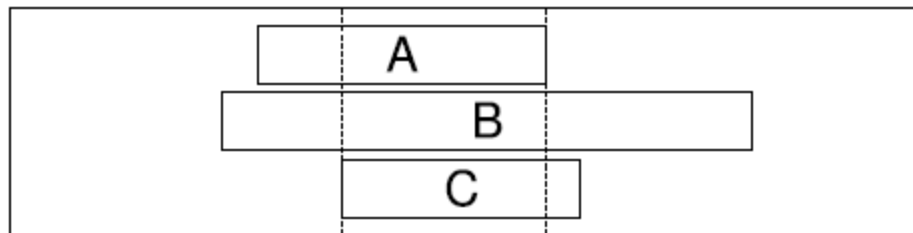
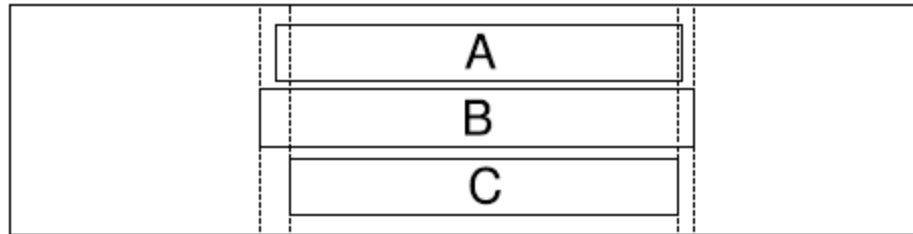
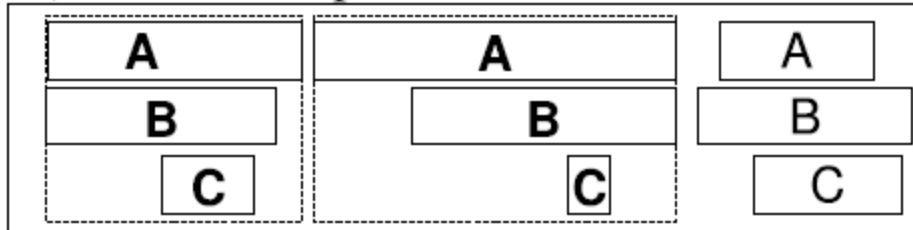
- Extension of Allen's relation to points.
- Only 5 more relations while [Vilain 1982] had 13.
 - point finishes and inverse
 - point starts and inverse
 - point equals
- Also proposed relations for branching time to reason in multiple future worlds.
- Supported in principle by [Moerchen/Fradkin 2010]



- Allen's relation extended by **relation of each interval midpoints to the other interval.**
- Two versions of overlaps shown: midpoints within other intervals (largely overlap) or not (overlap to some extent).
- 9 versions of overlaps
- **Total of 49 relations!**
- Designed to coarse data with arbitrary local order.
- Shares disadvantages with Allen's relations, in some respects even worse.

Temporal Operators

Other interval relations



- **A contains B contains C**
 - Contains is equivalent to (A equals B) or (B starts A) or (B during A) or (B ends A)
 - [Villafane 1999]
- **A,B,C approximately equal**
 - Allow slight variations.
 - N-ary operator
 - [Ultsch 1996]
- **A,B,C coincide**
 - Intersection of intervals
 - N-ary operator
 - [Moerchen 2006]

Temporal patterns Time point patterns

Frequent and closed patterns

- **Frequent patterns**

- Frequency = number of occurrences / size of database.
- Specify minimum frequency for patterns to be significant.
- Redundancy problem: Even if a sub-pattern has exactly same frequency as a super-pattern both are reported as frequent.

- **Closed frequent patterns**

- Cannot make the pattern more specific without decreasing the support.
- Sequence example (adapted from [\[Wang/Han 2004\]](#))

Database
C A A B C
A B C B
C A B C
A B B C A
B C

A B observed 4x
B C observed 5x
A B C also observed 4x
C A B C observed 2x
other extensions of A B C observed 1x



B C is closed
A B C is closed
A B is not

- **Substring patterns**

- Sequence of symbols without gaps
- Expresses concept of **order**
- Example: **B** → **C** → **B**

ABCBDADDBCBAAABCBDBABDABA

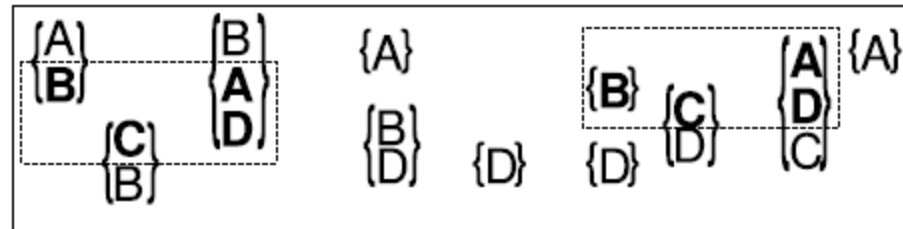
- **Regular expression patterns**

- Extension to allow gaps (via wildcards), negations, repetitions, etc.
- Example: **B** → **¬C** → **A | B**

ABCBDADBBCBAAABBCBDBABDABA

- **Algorithms for substring patterns**
 - Special data structures are used to represent the data and derive frequent patterns efficiently.
 - **Suffix trees**
 - Allowing wildcards [Vilo 2002]
 - Reporting over/under represented patterns [Apostolico et al. 2000]
 - **Suffix arrays**
 - Optimal time [Fischer et al. 2006]
 - Space efficient [Fischer et al. 2007]
 - Many applications in **bioinformatics**
 - Finding discriminative patterns
 - Sequence alignment

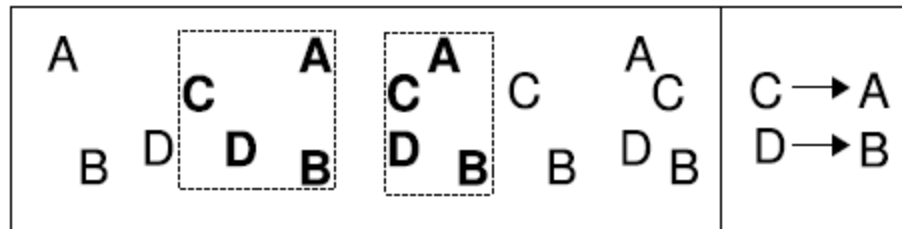
- Sequential patterns [Agrawal/Srikant 1995]
 - Sequence of (sets of) symbols
 - Expresses concept of **order**
 - Example: $\{B\} \rightarrow \{C\} \rightarrow \{A,D\}$



- Observed sets of symbols at each time point can contain more symbols.
- Gaps are allowed.
- Motivated by example of repeated purchases by customers, i.e., database of many short sequences.

- Algorithms for **sequential patterns**
 - AprioriAll [Agrawal/Srikant 1995],
 - SPADE [Zaki 1998]
 - PrefixSpan [Pei et al. 2001]
- Algorithms for **closed sequential patterns**
 - CloSpan [Yan et al. 2003]
 - BIDE (BIDirectional Extension checking) [Wang/Han 2004]
 - BIDEMargin [Fradkin/Moerchen 2010]
 - enforces margin of support difference among reported patterns
- Variations
 - Regular expressions [Garofalakis et al. 1999]
 - Multivariate data [Pinto et al. 2001]
 - Allow mismatches [Kum et al. 2003][Zhu et al. 2007]
 - Generators [Lo et al. 2008]

- **Episode patterns** [Mannila et al. 1995, 1996, 1997]
 - Constrained partial order of (sets of) symbols.
 - Expresses concepts of **order** and **concurrency**.
 - Example: **C → A and D → B within a time window**



- **Serial episodes**: Order relation between symbols (or episodes).
- **Parallel episodes**: Symbols or episodes observed within time window.
- **Episodes**: in principle arbitrary partial order of symbols, but often combination of serial and parallel episodes.
- Motivated by patterns in long message stream from telecommunication equipment.

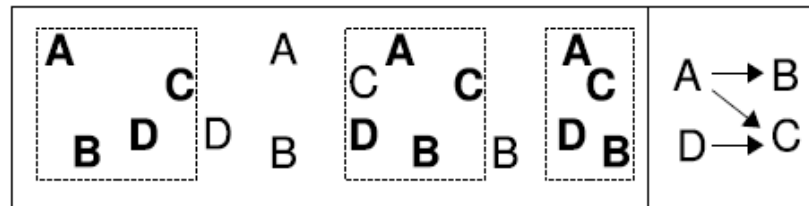
- **Apriori-style** algorithms for episodes.
 - Given maximum window length
 - WINEPI: percentage of windows with the pattern [Mannila et al. 1995]
 - MINEPI: windows that do not contain sub-windows with the pattern [Mannila/Toivonen 1996]
 - Given maximum gap [Meger/Rigotti 2004]
- Algorithms for **closed** (groups of) **episodes** [Harms et al. 2001]
- Determining **significance** beyond the concept of frequency.
 - Significance of episodes against Bernoulli and Markov background models [Gwadera et al. 2005]
 - Formal relation of Episodes with HMM [Laxman et al. 2005]

Time point patterns

Partial order patterns

- **Partial order patterns** [Casas-Garriga 2005]

- Partial order of (sets of) symbols.
- Expresses concepts of **order** and **concurrency**.
- Example: **A → B and A → C and D → C**



- Order between some symbols (A and D) is unspecified (hence partial).
- **Not equivalent to serial-parallel episodes!** [Pei et al. 2006]
 - Example above cannot be expressed as combination of parallel and serial episodes.
 - Definition of [Mannila et al. 1995] captures this but algorithms focus on serial/parallel episodes and many other authors have done the same (or even only deal with serial episodes).

Closed partial order pattern algorithms

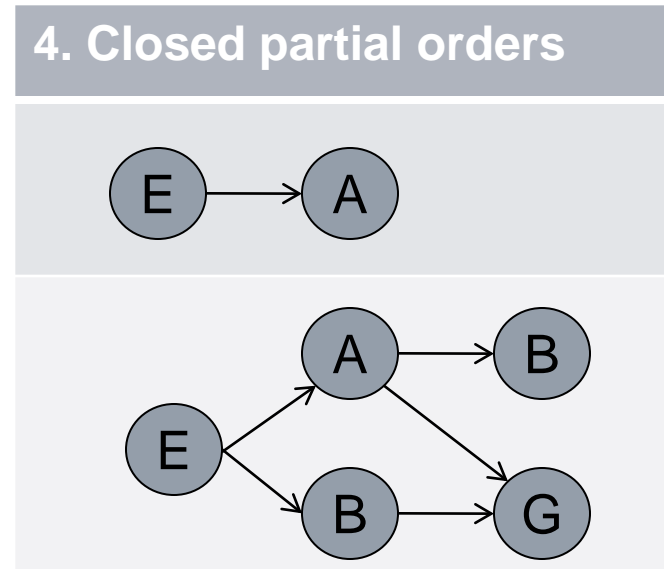
- Three step algorithm [Casas-Garriga 2005] [Moerchen 2006]
 - **Mine closed sequential patterns.**
 - Mine maximal **conjunctive groups** of non-redundant sequential patterns that are observed in the same windows. [Casas-Garriga 2005]
 - [Moerchen 2006]
 - Interpret sequential patterns as items and windows as itemsets.
 - **Mine closed itemsets** = maximal groups.
 - Remove redundant sequential patterns in each group
 - **Convert each group to partial order** [Casas-Garriga 2005]
 - Every sequential patterns is a path in the directed graph.
- Mining closed strict episodes [Tatti, Cule 2010]
 - DFS mining of partial orders where repeating symbols must have order relation.

Time point patterns

Closed partial order example

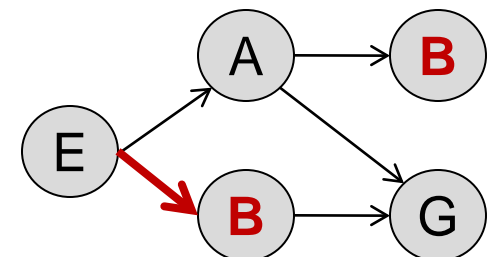
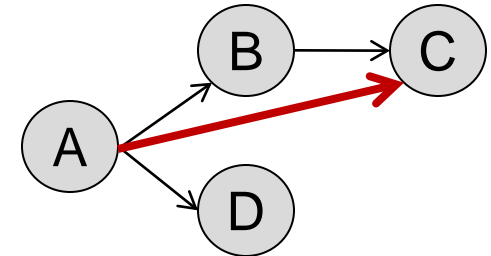
1. Sequences		2. Closed seq. patterns		S1	S2	S3
S1	{D}{E}{A}	{E}{A}	3x	x	x	x
S2	{E}{ABF}{G}{BDE}	{E}{A}{B}	2x		x	x
S3	{E}{A}{B}{G}	{E}{A}{G}	2x		x	x
		{E}{B}{G}	2x		x	x

3. Closed item (=seq. pattern) sets	
{E}{A}	3x
{E}{A}{B}, {E}{A}{G}, {E}{B}{G}	2x



Closed partial order pattern algorithms

- Efficient algorithm for closed partial orders [Pei et al. 2006]
 - Interpret edges in partial order as items.
 - **TranClose**: mine closed item (=edge) sets of transitive closure
 - **ABC** has edges **A** → **B**, **A** → **C**, **B** → **C**
 - Reduce resulting patterns
 - **Frecpo**: PrefixSpan-like algorithms with pruning
 - Directly generates transitive reduction.
 - Forbidden edges: not **A** → **C** if already **A** → **B** and **B** → **C**.
 - Both do not allow for repeating symbols in patterns.
 - **E** → **B** is forbidden but it's not the 'same' B



Conjunctive sequential patterns

- Mining conjunctive groups of sequential patterns [Raïssi et al. 2008] .
- Closed sequential pattern mining algorithms use equivalence classes based on prefix.
- Non-derivable does not work for single sequential patterns and these classes.
 - Lower bound is always 0
- In contrast conjunctive groups of sequential patterns form equivalence classes.
 - see also [Casas-Garriga 2005], [Harms et al. 2001]
- Conjunctive groups can be used just similar to itemsets.
 - Mine closed groups of sequential patterns [Moerchen 2006]
 - Non-derivable groups of sequential patterns [Raïssi et al. 2008]
 - Generate sequential association rules [Raïssi et al. 2008]
 - Mine generators for groups of sequential patterns.

Time point patterns

Sequential patterns vs. Episodes vs. Partial Orders

Supports	Sequential Patterns	Episodes parallel	serial	serial/parallel	Partial Orders
Itemsets	✓	✓ Unordered set of symbols	✗ Typically not discussed	✗ Typically not discussed	✓ Often straightforward extension
Full partial order	✗	✗	✗	✗ Typically not discussed	✓
Closedness	✓ Easy - CloSpan	✓ Easy - Closed itemsets	✓ Easy - Equivalent to seq. patterns	✓ Requires looking at groups of episodes	✓ Requires looking at groups of seq. pat.
Condensed representation	✓ several	✓	✓	✗	✓ Margin-closed

Time point patterns

Summary

- Use substrings if no gaps required.
- Use closed sequential patterns if gaps are and strict ordering are required.
- Partial orders are the most flexible representation but also more complex to mine.

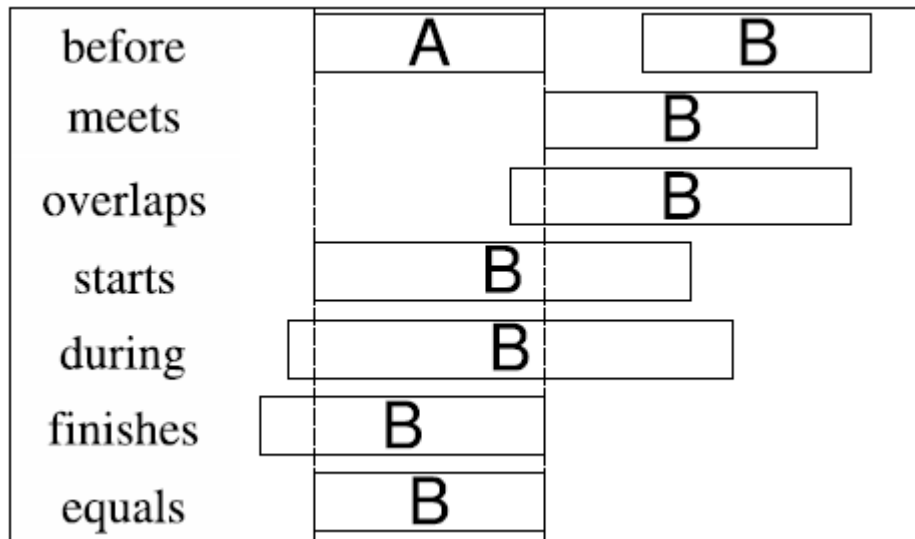
	Univariate symbolic time series	Multivariate symbolic time series	Symbolic time sequence	duration	order	concurrency	synchronicity	partial order
Substrings	✓				✓			
Sequential Patterns	✓	✓	✓		✓		✓	
Episodes	✓	✓	✓		✓	✓		✓
Partial orders	✓	✓	✓		✓	✓	✓	✓

Temporal patterns

Time interval patterns

Temporal Operators

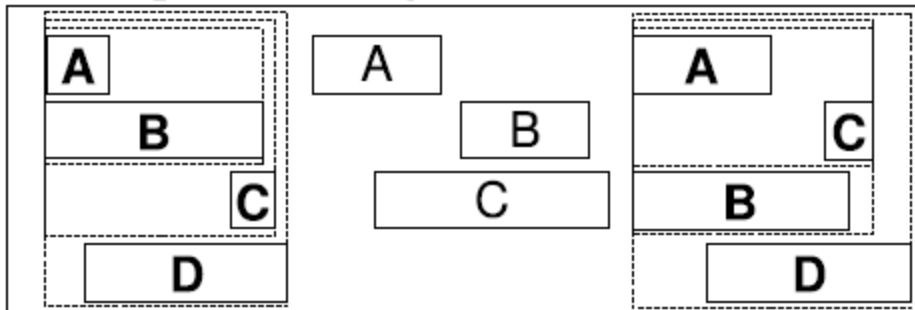
Allen's interval relations



- 13 relations forming an algebra.
- **Any two intervals have exactly one of the relations.**
- Invented in AI for temporal reasoning: given facts associated with time interval derive additional facts or answer specific questions.
- Later widely used in data mining.
- **Disadvantages for knowledge discovery!**
- Thresholds and fuzzy extensions solve some of the problems.

Time interval patterns

Patterns using Allen's relations



A1 patterns [Kam/Fu 2000]

- Combine two intervals with relation from Allen.
- Combine resulting pattern with another single interval.
- Ambiguous representation [Moerchen 2006]:

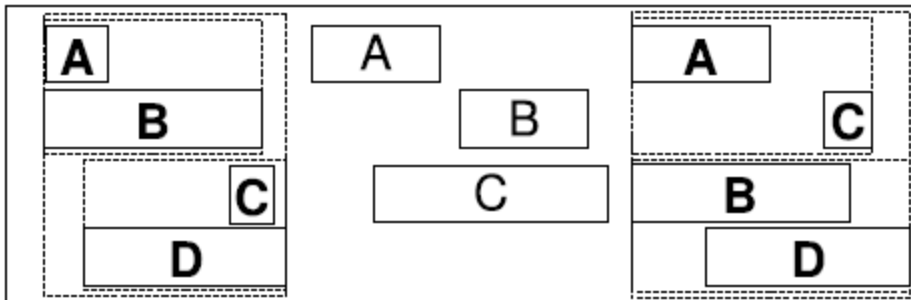
(((A starts B) overlaps C) overlaps D)

- or

(((A before C) started by B) overlaps D)

Time interval patterns

Patterns using Allen's relations



Fluents [Cohen 2001]

- Combine two intervals with relation from Allen.
- Combine two patterns.
- Ambiguous representation [Moerchen 2006]:

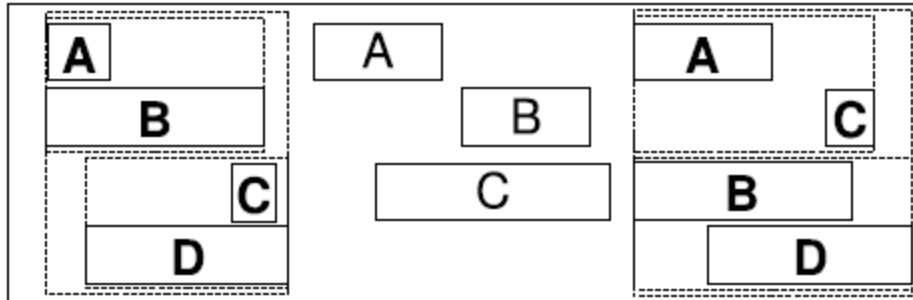
(A starts B) overlaps (C during D)

- or

(A before C) starts (B overlaps D).

Time interval patterns

Patterns using Allen's relations



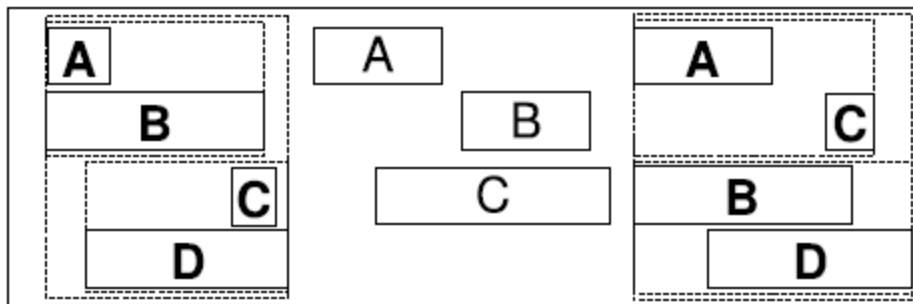
	A	B	C	D
A	Equals	Starts before	before	overlaps
B	Started by	Equals	overlaps	overlaps
C	after	Overlapped by	Equals	during
D	Overlapped by	Overlapped by	contains	Equals

[Hoepfner 2001]

- Set of intervals with pairwise relations of Allen.
- $K(K-1)/2$ relations for K intervals.
- **Not ambiguous** but difficult to write out as rule [Moerchen 2006].
- Transitivity of Allen's relations can be used to derive some relations from others.

Time interval patterns

Patterns using Allen's relations



= both =

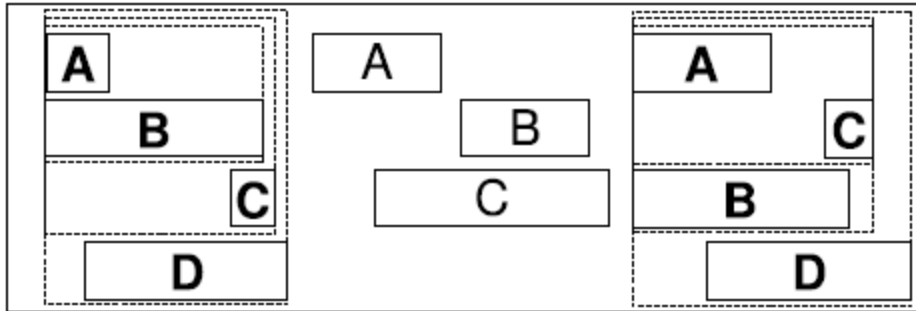
$A^+ = B^+ > D^+ > A^- > C^+ > B^- > C^- > D^-$

Sequence of interval boundaries [Wu/Chen 2007]

- Represent each interval with start and end points (A^+ vs. A^-).
- Pattern with K intervals represented by sequence of $2K$ boundaries.
- Not ambiguous and equivalent to Hoepfner, but more compact:
 $2K+(2K-1)$ vs. $K(K-1)/2$

Time interval patterns

Patterns using Allen's relations



= **both** =

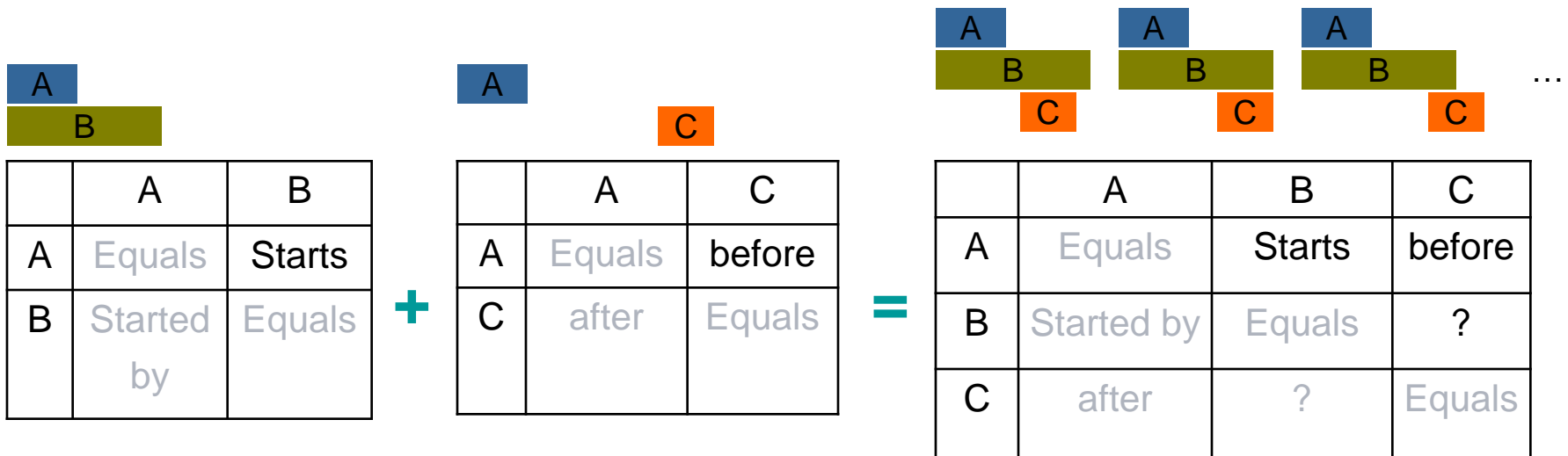
((A starts [0, 0, 0, 0, 1] B)
 overlaps [0, 0, 0, 1, 0] C)
 overlaps [1, 0, 0, 2, 0] D)

Contains
 Finished by
 Meets
 Overlaps
 Starts

Nested representation with counters [Patel et al 2008]

- Fix ambiguity of nested representation by adding counters for 5 relations
- $K+5(K-1)$
- Not as compact as Wu/Chen and not very readable.

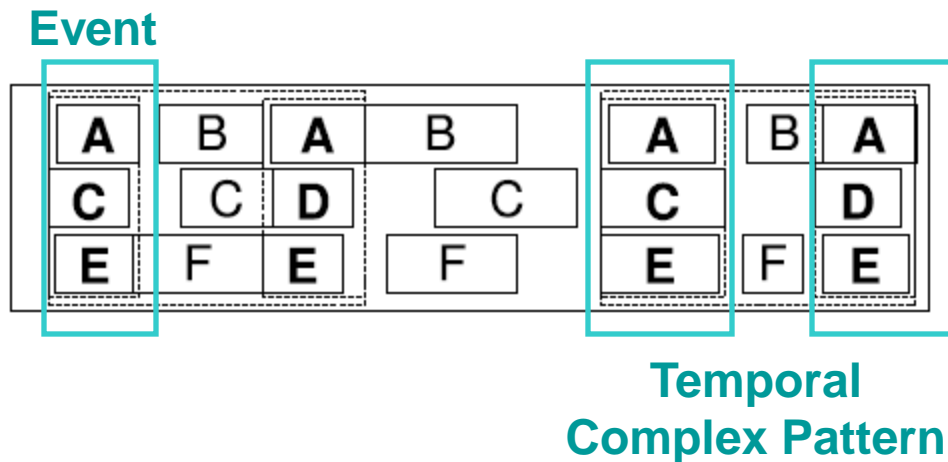
- Apriori-style [Hoeppe 2001]
 - Combine two length k patterns with common k-1 prefix.
 - Use transitivity of Allen's relations to prune some candidates for the relations of the two kth intervals.



- **B {contains, ended by, overlaps, meets, before} C**
- Pruned relations: {after, met by, overlapped by, started by}

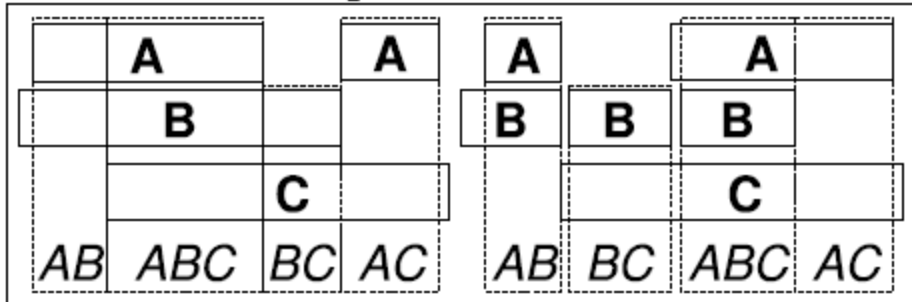
Allen's relations pattern mining algorithms

- Early work using only Apriori-style algorithms. Recently more efficient algorithms have been proposed:
 - H-DFS (Hybrid Depth First Search) [Papaterou et al. 2005, 2009]
 - Enumeration tree
 - ARMADA [Winarko/Roddick 2006]
 - Adaptation of sequential pattern mining algorithm to intervals.
 - TPrefixSpan [Wu/Chen 2007]
 - PrefixSpan using interval boundaries pruning patterns that are not valid interval patterns.
 - IEMiner [Patel et al 2008] (compares to TPrefixSpan and H-DFS)
 - Apriori using nested representation with counters and pruning.
 - KarmaLego [Moskovitch/Shahar 2009] (compares to Armada and H-DFS)
 - Enumeration tree exploiting transitivity.

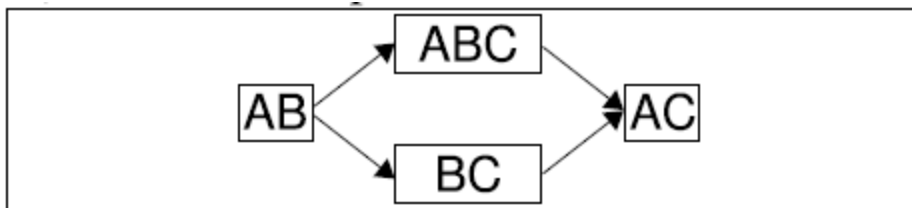


Unification-based Temporal Grammar (UTG) [Ultsch 1996]

- Events: several intervals occur **more or less simultaneous**.
- Temporal Complex Patterns: sequence of Events.
- Annotations for duration of intervals and gaps.
- Detection of patterns can be formulated Prolog (hence unification-based)



Chords



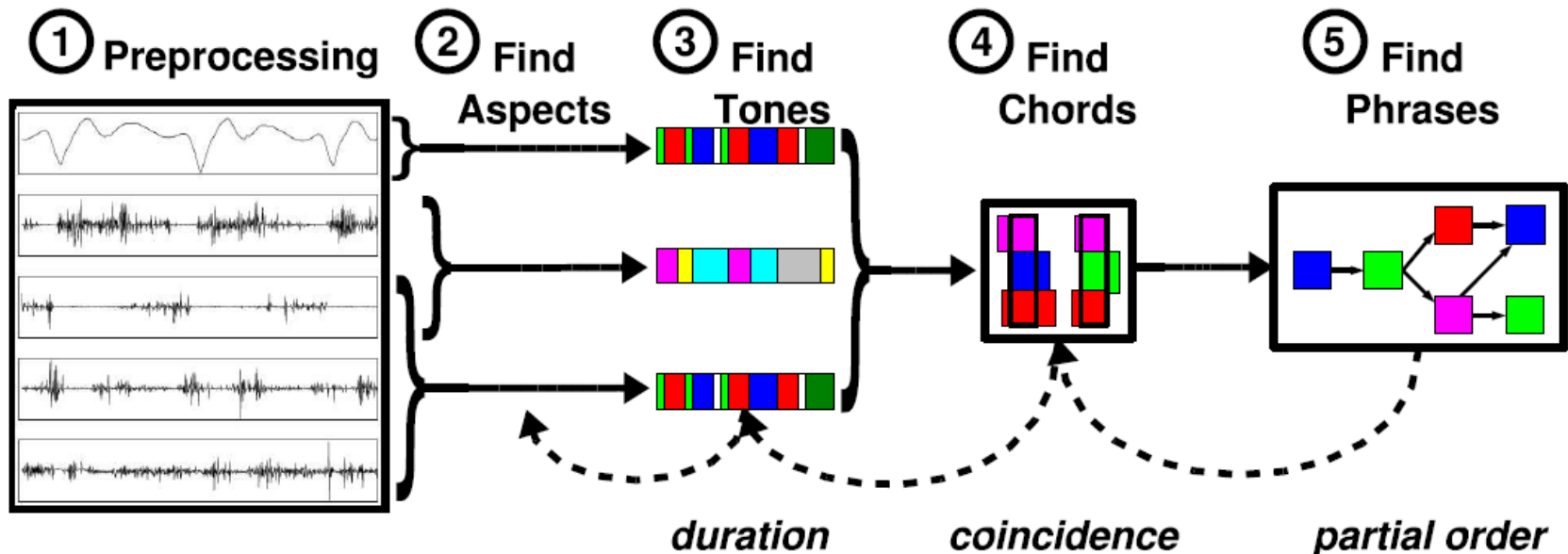
Phrase

Time Series Knowledge Representation (TSKR)

[Moerchen 2006]

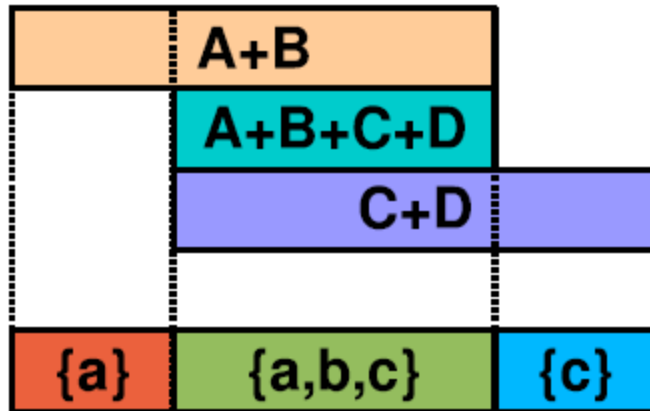
- Extension of UTG
- Tones represent **duration** with intervals.
- Chords represent **coincidence** of Tones.
- Phrases represent **partial order** of Chords
- Compact, unambiguous representation with details on demand.
- Robust against noise in the interval boundaries.

- **Time Series Knowledge Mining** [Moerchen 2006]
 - Methodology for mining from multivariate time series.
 - Tone mining: discretization, segmentation, clustering.
 - **Chord mining**: variation of itemset mining.
 - **Phrase mining**: variation of partial order mining.



Time interval patterns

Phrase mining algorithm



where

$$a = A+B$$

$$b = A+B+C+D$$

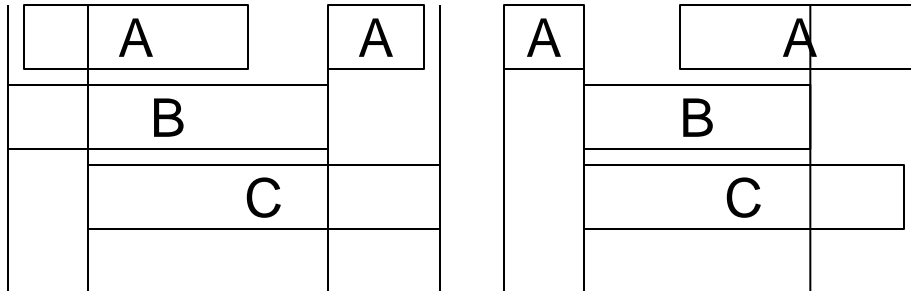
$$c = C+D$$

Phrase mining

- Transform Chords to itemset sequence: set of Chords observed over time interval.
- Mine sequential patterns with additional constraints: Only one Chord can be picked per time interval.
- Mine closed groups of sequential patterns.
- Convert to partial order.

Time interval patterns

Templates for (partial) presence/absence



ed	ed	ed	pe	ed	pe
pe	pe	ad	ad	pe	ad
ad	pe	pe	ad	pe	ed

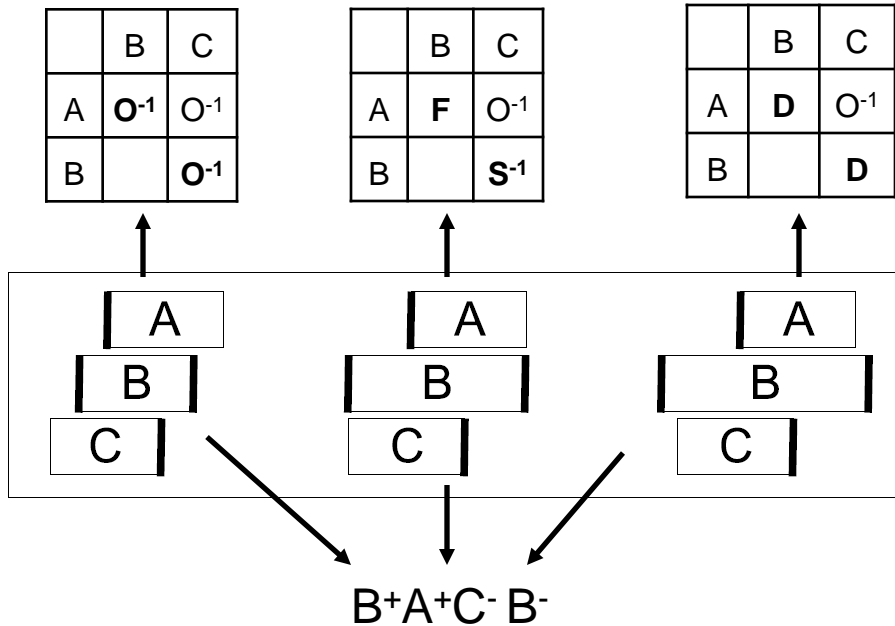
=

e	e	e
*	p	a
a	p	e

Templates [Peter/Hoeppner 2010]

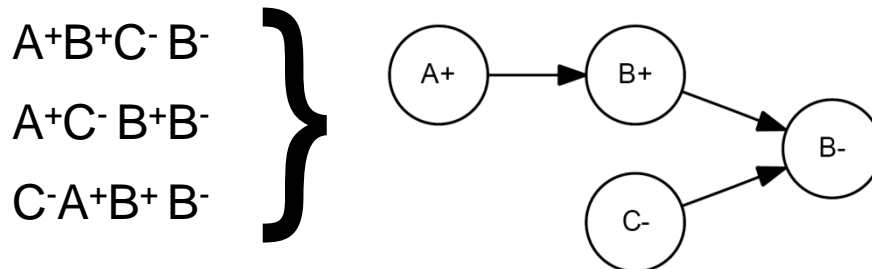
- Segment time axis
- 5 Interval predicates
 - present (\forall)
 - absent ($\forall \neg$)
 - unconstrained (*)
 - exists (\exists)
 - disappears ($\exists \neg$)
- Algorithm to search classification rules (not frequent patterns)
- Picks time segments and captures constraints on duration

Semi-interval Partial Order (SIPO) patterns



Semi-Interval Partial Order (SIPO) [Moerchen/Fradkin 2010]

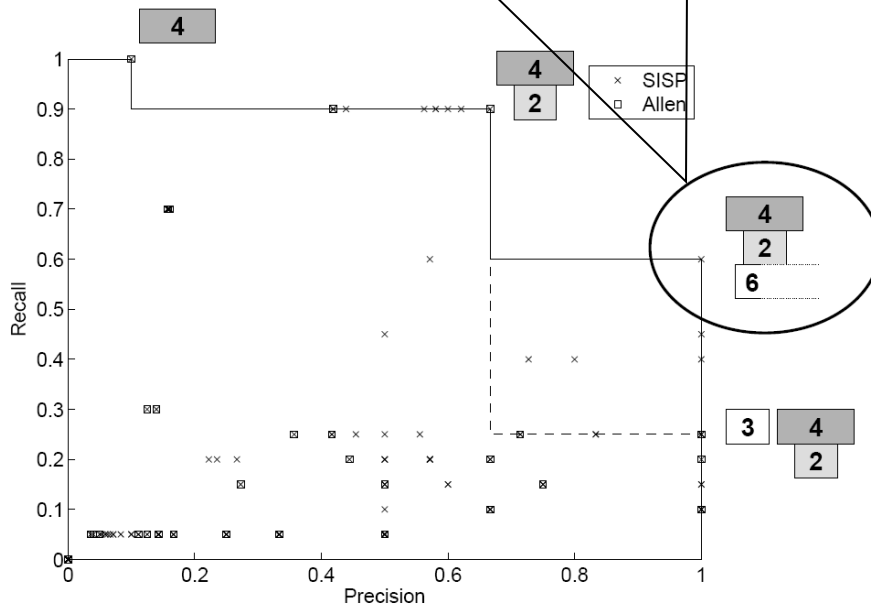
- Sequential pattern of interval boundaries **without constraints**.
 - Superset of full interval (Allen) patterns.
 - Matches more similar situations in the data.
- **Partial order patterns** of interval boundaries.
 - Even less constraints.
 - Even more matches.
- Can be applied to **mixed time point / time interval data!**



Time interval patterns

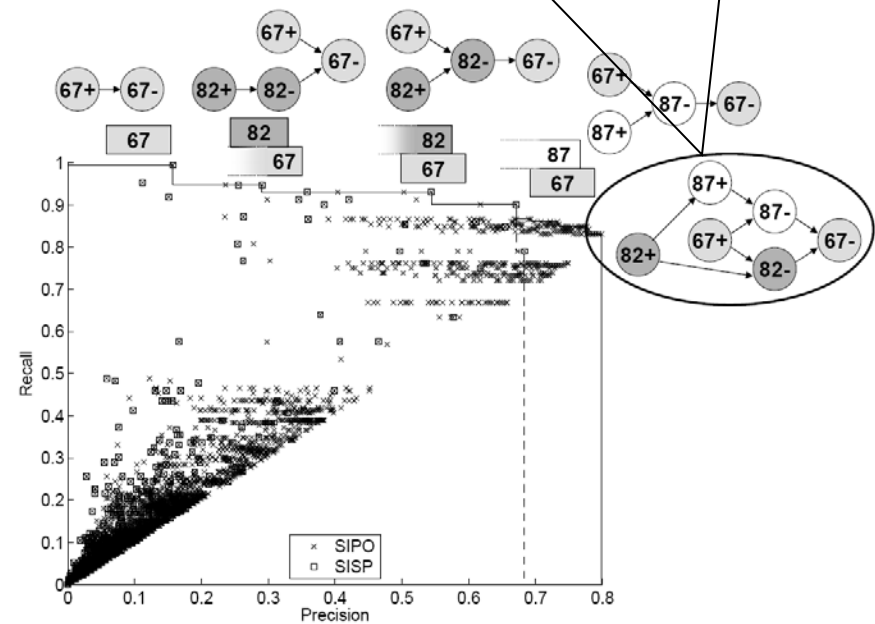
Semi-interval patterns for classification

Semi interval sequential pattern (SISP) with **better precision/recall than any full interval pattern.**



'Name' class in Australian Sign language dataset.

Semi interval partial order (SIPO) pattern with **better precision/recall than any SISP or full interval pattern.**



'l' class in American Sign language dataset.

Time interval patterns

Summary

- Use **Allen** only if exact interval boundaries matter and Allen semantics are required.
- SIPO**: superset of Allen patterns allowing for missing boundaries (Freksa) and partial order.
- TSKR** and **Templates**: alternative approaches that match partial intervals.

	Univariate symbolic time series	Multivariate symbolic time series	Symbolic time sequence	duration	order	coincidence	synchronicity	partial order
Allen		✓	✓	✓	✓	✓	✓	
UTG		✓	✓	✓	✓		✓	
TSKR		✓	✓	✓	✓	✓	✓	✓
Templates		✓	✓	✓	✓	✓		
SIPO		✓	✓	✓	✓	✓	✓	✓

Temporal patterns

Quantitative temporal patterns

Quantitative temporal patterns

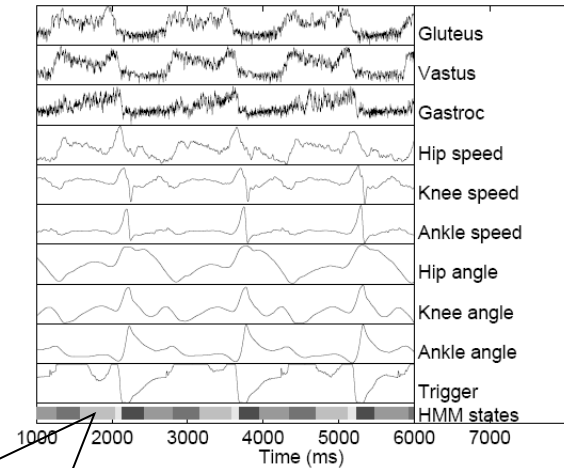
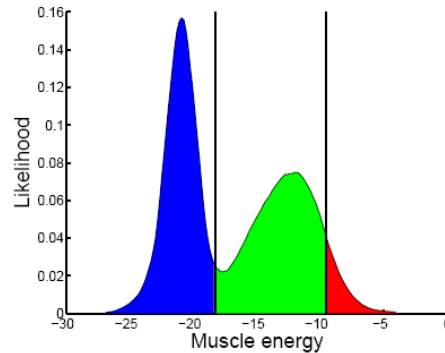
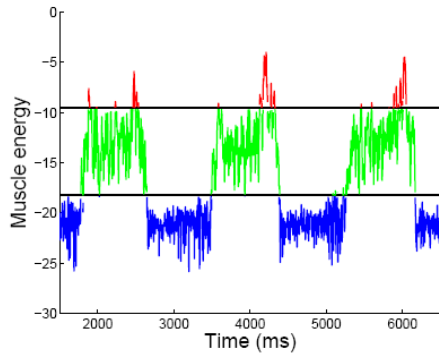
Quantitative sequential patterns

- Extend sequential patterns to capture typical durations between sequence elements.
- Different from constraints on maximum duration, gaps, etc. [Pei et al 2002] [Li 2008].
- Find time interval [min,max] between elements of sequence found by clustering [Yoshida 2000].
- Insert pseudo items corresponding to time intervals of user defined granularity [Chen 2003][Hirate/Yamana 2006].
 - Extension to fuzzy characterization of gaps [Chen/Huang 2005].
- Exhaustive for Temporally Annotated Sequences (TAS) with typical duration between elements of sequence using clustering [Gianotti et al. 2006].
- Trees representation of sequential patterns with split for different typical duration intervals [Nanni/Rigotti 2007].

- Episode patterns for intervals where each sequence element allows an interval starting at this time point to have a duration of [min, max] [[Laxman et al. 2007](#)]
- A priori pattern generation interleaving symbolic patterns generation with density based clustering [[Guyet/Quiniou 2008](#)]
 - K Intervals in a pattern for hypercube in K dimensions
 - Points are generated in the hypercube
 - EM clustering finds typical quantitative patterns.
- Interval templates of [[Peter/Hoeppe 2010](#)] capture duration of time axis segmentation in each pattern.

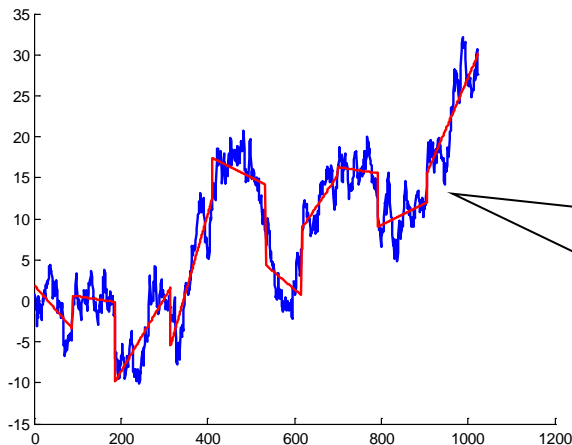
Preprocessing

Discretization, segmentation, clustering



Discretization by partitioning the value axis.

Clustering time points using multivariate vector (k-means, GMM, HMM, ...)



Segmentation at change points and approximation with constant or linear functions models

Applications

Temporal patterns at work



Sports

Meteorology

Predictive maintenance

Recommender systems

Patient records

Intrusion detection

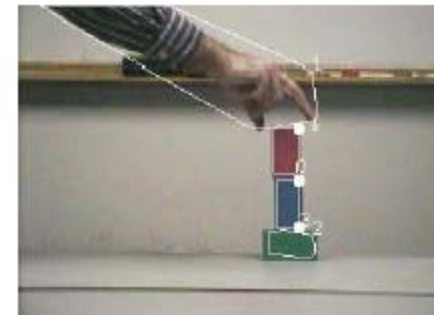
Assisted living

Software engineering

Video analytics

Network monitoring

Workflow analysis



Time interval data sets

- Benchmark dataset with labeled time interval sequences introduced in [Moerchen/Fradkin 2010]. Available by email.


Data	Description	Intervals	Labels	Sequences	Classes	Source
ASL-BU	transcriptions from videos of American Sign Language	18250	154	441	7	[Papaterou et al. 2005]
Auslan2	derived from sensor time series of Australian Sign Language (UCI)	900	12	200	10	[Kadous 2002] UCI ML Repository
Blocks	visual primitives detected in videos of a human hand stacking colored blocks	1207	8	210	8	[Fern 2004]
Context	derived from sensor time series of a mobile device	12916	54	240	5	[Maentyjaervi et al. 2004]
Pioneer	derived from sensor time series of Pioneer-1 robot (UCI)	4883	92	160	3	UCI ML Repository
Skating	derived from sensor time series of speed skaters during controlled tests	18953	41	530	6/7	[Moerchen et al. 2007]

Some unlabeled datasets available: smart home, weather, ...

More to come, **please donate!**






Some datasets used in literature

- Gene sequences (UCI)
- User traces (UCI)
- Web server logs (UCI, Internet Traffic Archive)
- Gazelle (KDD CUP 2000)



UCI Machine Learning Repository
Center for Machine Learning and Intelligent Systems

Browse Through: 5 Data Sets

Default Task	Name
Classification (3) Regression (0) Clustering (0) Other (2)	 Entree Chicago Recommendation Data
Attribute Type - Undo Categorical (5) Numerical (7) Mixed (0)	 Molecular Biology (Promoter Gene Sequences)
Data Type - Undo Multivariate (28) Univariate (0) Sequential (5) Time-Series (0) Text (2) Domain-Theory (2) Other (3)	 Molecular Biology (Protein Secondary Structure)
Area Life Sciences (3)	 Molecular Biology (Splice-junction Gene Sequences)
	 UCI MSNBC.com Anonymous Web Data

Many publications heavily rely on artificially generated data

More real life data with different characteristics are needed.

Temporal pattern mining in symbolic time point and time interval data

Summary

- Introduced **data models, patterns, and algorithms** for point and interval data.
- Online version: additional material on **preprocessing** to convert other data into symbolic temporal data and **applications** using temporal patterns for data mining.

Research opportunities

- **Algorithms**: efficiency, error tolerant patterns, data streams, ...
- **Interestingness**: pattern significance, expert knowledge, ...
- **Classification**: directly mine patterns predict events (early).
- **Anomaly detection**: learn to distinguish normal from abnormal behavior for symbolic temporal data based on patterns.
- **Applications** in medicine, finance, maintenance, meteorology, ...

Temporal pattern mining in symbolic time point and time interval data

SIEMENS

Thank you!

Please send *any* feedback to fabian.moerchen@siemens.com

www.siemens.com

www.usa.siemens.com/en/about_us/research/home.htm

www.timeseriesknowledgemining.org (bibliography, slides, etc.)

Fabian Moerchen: **Unsupervised pattern mining from symbolic temporal data**, SIGKDD Explorations 9(1), ACM, pp. 41-55, 2007

We are hiring!