Ludwig-Maximilians-Universität München
Lehrstuhl für Datenbanksysteme und Data Mining
Prof. Dr. Thomas Seidl

Knowledge Discovery and Data Mining 1

(Data Mining Algorithms 1)

Winter Semester 2019/20

Agenda

Simple Association Rules: Introduction

Example

Transaction database:

$$D = \{\{butter, bread, milk, sugar\}, \{butter, flour, milk, sugar\}, \{butter, eggs, milk, salt\}, \{eggs\}, \{butter, flour, milk, salt, sugar\}\}$$

Frequent itemsets:

items	support
{butter}	4
{milk}	4
{butter, milk}	4
{sugar}	3
{butter, sugar}	3
{milk, sugar}	3
{butter, milk, sugar}	3

Question of interest

- If milk and sugar are bought, will the customer always buy butter as well? milk, sugar $\Rightarrow$ butter?
- In this case, what would be the probability of buying butter?

Simple Association Rules: Basic Notions

Let *Items, Itemset, Database, Transaction, Transaction Length, k-itemset, (relative) Support, Frequent Itemset* be defined as before. Additionally:

- The items in transactions and itemsets are **sorted** lexicographically: itemset $X = (x_1, \ldots, x_k)$, where $x_1 \leq, \ldots, \leq x_k$
- **Association rule**: An association rule is an implication of the form $X \Rightarrow Y$ where $X, Y \subseteq I$ are two itemsets with $X \cap Y = \emptyset$
- Note: simply enumerating all possible association rules is not reasonable!

What are the interesting association rules w.r.t. D?

Interestingness of Association Rules

Goal

Quantify the interestingness of an association rule with respect to a transaction database D.

Support

- Frequency (probability) of the entire rule with respect to D:

$$supp(X \Rightarrow Y) = P(X \cup Y) = \frac{|\{T \in D \mid X \cup Y \subseteq T\}|}{|D|} = supp(X \cup Y)$$

- "Probability that a transaction in D contains the itemset."

Interestingness of Association Rules

Confidence

- Indicates the strength of implication in the rule:

$$conf(X \Rightarrow Y) = \frac{supp(X \cup Y)}{supp(X)} \stackrel{(*)}{=} \frac{P(X \cap Y)}{P(X)} = P(Y \mid X)$$

  (*) Note that the support of the union of the items in X and Y, i.e. $supp(X \cup Y)$ can be interpreted by the joint probability $P(X \cap Y)$

- $P(Y \mid X) =$ conditional probability that a transaction in D containing the itemset X also contains itemset Y

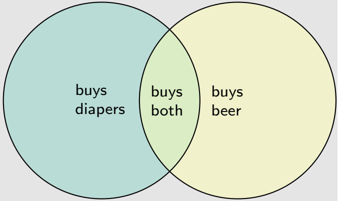
Interestingness of Association Rules

Rule form

"Body $\Rightarrow$ Head [support, confidence]"

Association rule examples

- buys diapers $\Rightarrow$ buys beer [0.5 %, 60%]
- major in CS $\wedge$ takes DB $\Rightarrow$ avg. grade A [1%, 75%]



Mining of Association Rules

Task of mining association rules

Given a database D, determine all association rules having a $supp \geq minSup$ and a $conf \geq minConf$ (so-called *strong association rules*).

Key steps of mining association rules

1. Find frequent itemsets, i.e., itemsets that have $supp \geq minSup$ (e.g. Apriori, FP-growth)
2. Use the frequent itemsets to generate association rules
   - For each itemset X and every nonempty subset $Y \subset X$ generate rule $Y \Rightarrow (X \setminus Y)$ if *minSup* and *minConf* are fulfilled
   - We have $2^{|X|} - 2$ many association rule candidates for each itemset X

Mining of Association Rules

Example

- Frequent itemsets:

1-itemset	count	2-itemset	count	3-itemset	count
{ a }	3	{ a,b }	3	{ a,b,c }	2
{ b }	4	{ a,c }	2		
{ c }	5	{ b,c }	4		

- Rule candidates
  - From 1-itemsets: None
  - From 2-itemsets: $a \Rightarrow b$; $b \Rightarrow a$; $a \Rightarrow c$; $c \Rightarrow a$; $b \Rightarrow c$; $c \Rightarrow b$
  - From 3-itemsets: $a, b \Rightarrow c$; $a, c \Rightarrow b$; $c, b \Rightarrow a$; $a \Rightarrow b, c$; $b \Rightarrow a, c$; $c \Rightarrow a, b$

Generating Rules from Frequent Itemsets

Rule generation

- For each frequent itemset X:
  - For each nonempty subset Y of X, form a rule $Y \Rightarrow (X \setminus Y)$
  - Delete those rules that do not have minimum confidence
- Note:
  - Support always exceeds *minSup*
  - The support values of the frequent itemsets suffice to calculate the confidence
- Exploit anti-monotonicity for generating candidates for strong association rules!
  - $Y \Rightarrow Z$ not strong $\implies$ for all $A \subseteq D : Y \Rightarrow Z \cup A$ not strong
  - $Y \Rightarrow Z$ not strong $\implies$ for all $Y' \subseteq Y$: $(Y \setminus Y') \Rightarrow (Z \cup Y')$ not strong

Generating Rules from Frequent Itemsets

Example: $minConf = 60\%$

$conf(a \Rightarrow b) = 3/3 = 1$ ✓
$conf(b \Rightarrow a) = 3/4$ ✓
$conf(a \Rightarrow c) = 2/3$ ✓
$conf(c \Rightarrow a) = 2/5$ ✗
$conf(b \Rightarrow c) = 4/4 = 1$ ✓
$conf(c \Rightarrow b) = 4/5$ ✓
$conf(a, b \Rightarrow c) = 2/3$ ✓
$conf(a, c \Rightarrow b) = 2/2 = 1$ ✓
$conf(b, c \Rightarrow a) = 2/4 = .5$ ✗
$conf(a \Rightarrow b, c) = 2/3$ ✓
$conf(b \Rightarrow a, c) = 2/4$ ✗ (pruned wrt. $b, c \Rightarrow a$)
$conf(c \Rightarrow a, b) = 2/5$ ✗ (pruned wrt. $b, c \Rightarrow a$)

itemset	count
{ a }	3
{ b }	4
{ c }	5
{ a,b }	3
{ a,c }	2
{ b,c }	4
{ a,b,c }	2

Interestingness Measurements

Objective measures

Two popular measures:

- Support
- Confidence

Subjective measures [Silberschatz & Tuzhilin, KDD95]

A rule (pattern) is interesting if it is

- *unexpected* (surprising to the user) and/or
- *actionable* (the user can do something with it)

Criticism to Support and Confidence

Example 1 [Aggarwal & Yu, PODS98]

- Among 5000 students
  - 3000 play basketball (=60%)
  - 3750 eat cereal (=75%)
  - 2000 both play basket ball and eat cereal (=40%)
- Rule "play basketball $\Rightarrow$ eat cereal [40%, 66.7%]" is misleading because the overall percentage of students eating cereal is 75% which is higher than 66.7%
- Rule "play basketball $\Rightarrow$ not eat cereal [20%, 33.3%]" is far more accurate, although with lower support and confidence
- Observation: "play basketball" and "eat cereal" are negatively correlated

Not all strong association rules are interesting and some can be misleading.

- Augment the support and confidence values with interestingness measures such as the correlation: "A $\Rightarrow$ B [*supp*, *conf*, *corr*]"

Other Interestingness Measures: Correlation

Correlation

Correlation (sometimes called *Lift*) is a simple measure between two items A and B:

$$corr_{A,B} = \frac{P(A \cap B)}{P(A)P(B)} = \frac{P(B \mid A)}{P(B)} = \frac{conf(A \Rightarrow B)}{supp(B)}$$

- The two rules $A \Rightarrow B$ and $B \Rightarrow A$ have the same correlation coefficient
- Takes both $P(A)$ and $P(B)$ in consideration
- $corr_{A,B} > 1$: The two items A and B are positively correlated
- $corr_{A,B} = 1$: There is no correlation between the two items A and B
- $corr_{A,B} < 1$: The two items A and B are negatively correlated

Other Interestingness Measures: Correlation

Example 2

T	item		
	X	**Y**	**Z**
	1	1	0
	1	1	1
	1	0	1
	1	0	1
	0	0	1
	0	0	1
	0	0	1
	0	0	1

rule	support	confidence	correlation
$X \Rightarrow Y$	25%	50%	2
$X \Rightarrow Z$	37.5%	75%	0.89
$Y \Rightarrow Z$	12.5%	50%	0.57

- X and Y: positively correlated
- X and Z: negatively related
- Support and confidence of $X \Rightarrow Z$ dominates
- But: items X and Z are negatively correlated
- Items X and Y are positively correlated

Hierarchical Association Rules: Motivation

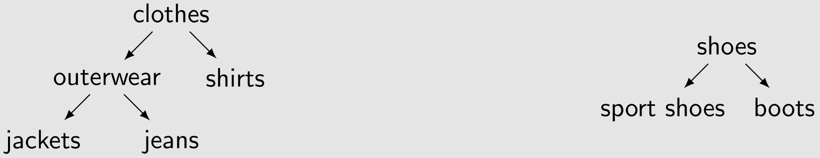
Problem

- High minSup: apriori finds only few rules
- Low minSup: apriori finds unmanagably many rules

Solution

Exploit item taxonomies (generalizations, is-a hierarchies) which exist in many applications

Example



Hierarchical Association Rules

New Task

Find all generalized association rules between generalized items, i.e. Body and Head of a rule may have items of any level of the hierarchy

Generalized Association Rule

$X \Rightarrow Y$ with $X, Y \subset I, X \cap Y = \emptyset$ and no item in Y is an ancestor of any item in X

Example

- Jeans $\Rightarrow$ Boots; supp $<$ minSup
- Jackets $\Rightarrow$ Boots; supp $<$ minSup
- Outerwear $\Rightarrow$ Boots; supp $>$ minSup

Hierarchical Association Rules: Characteristics

$$\begin{tikzcd}[column sep=small] & & Y \arrow[dll] \arrow[dl] \arrow[d] \arrow[dr] \arrow[drr] & & \\ X_1 & \cdots & X_i & \cdots & X_k \end{tikzcd}$$

Characteristics

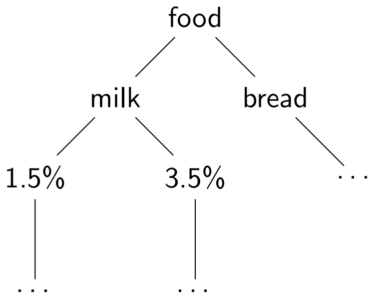
Let $Y = \biguplus_{i=1}^{k} X_i$ be a generalisation.

- For all $1 \leq i \leq k$ it holds $supp(Y \Rightarrow Z) \geq supp(X_i \Rightarrow Z)$
- In general, $supp(Y \Rightarrow Z) = \sum_{i=1}^{k} supp(X_i \Rightarrow Z)$ does not hold (a transaction might contain elements from multiple low-level concepts, e.g. boots *and* sport shoes).

Mining Multi-Level Associations

Top-Down, Progressive-Deepening Approach

1. First find high-level strong rules, e.g. milk $\Rightarrow$ bread [20%, 60%]
2. Then find their lower-level "weaker" rules, e.g. low-fat milk $\Rightarrow$ wheat bread [6%, 50%].



Support Threshold Variants

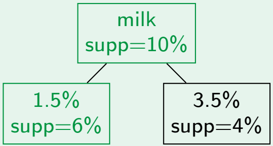
Different minSup threshold across multi-levels lead to different algorithms:

- adopting the same minSup across multi-levels
- adopting reduced minSup at lower levels

Minimum Support for Multiple Levels

Uniform Support

- Search procedure is simplified (monotonicity)
- User only specifies one threshold

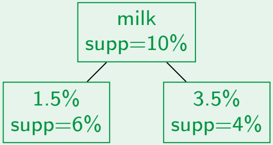


minSup=5%

minSup=5%

Reduced Support (Variable Support)

- Takes into account lower frequency of items in lower levels



minSup=5%

minSup=3%

Multilevel Association Mining using Reduced Support

Level-by-level independent method

Examine each node in the hierarchy, regardless of the frequency of its parent node.

Level-cross-filtering by single item

Examine a node only if its parent node at the preceding level is frequent.

Level-cross-filtering by k-itemset

Examine a k-itemset at a given level only if its parent k-itemset at the preceding level is frequent.

Multi-level Association: Redundancy Filtering

Some rules may be redundant due to "ancestor" relationships between items.

Example

- R_1: milk $\Rightarrow$ wheat bread [8%, 70%]
- R_2: 1.5% milk $\Rightarrow$ wheat bread [2%, 72%]

We say that rule 1 is an ancestor of rule 2.

Redundancy

A rule is redundant if its support is close to the "expected" value, based on the rule's ancestor.

,

Interestingness of Hierarchical Association Rules: Notions

Let $X, X', Y, Y' \subseteq I$ be itemsets.

- X' is ancestor of X iff there exists ancestors $x'_1, \ldots, x'_k$ of $x_1, \ldots, x_k \in X$ and $x_{k+1}, \ldots, x_n$ with $n = |X|$ such that $X' = \{x'_1, \ldots, x'_k, x_{k+1}, \ldots, x_n\}$
- Let X' and Y' be ancestors of X and Y. Then we call the rules $X' \Rightarrow Y'$, $X \Rightarrow Y'$, and $X' \Rightarrow Y$ ancestors of the rule $X \Rightarrow Y$.
- The rule $X' \Rightarrow Y'$ is a direct ancestor of rule $X \Rightarrow Y$ in a set of rules if:
  1. Rule $X' \Rightarrow Y'$ is an ancestor of rule $X \Rightarrow Y$, and
  2. There is no rule $X'' \Rightarrow Y''$ being ancestor of $X \Rightarrow Y$ and $X' \Rightarrow Y'$ is an ancestor of $X'' \Rightarrow Y''$

R-Interestingness

R-Interestingness

A hierarchical association rule $X \Rightarrow Y$ is called R-interesting if:

- There are no direct ancestors of $X \Rightarrow Y$ or
- The actual support is larger than R times the expected support or
- The actual confidence is larger than R times the expected confidence

Example in tutorial

R-Interestingness: Expected Support

Given the rule for $X \Rightarrow Y$ and its ancestor rule $X' \Rightarrow Y'$ the expected support of $X \Rightarrow Y$ is defined as:

$$\mathbb{E}_{Z'}[P(Z)] = P(Z') \cdot \prod_{i=1}^{j} \frac{P(y_i)}{P(y_i)'}$$

where $Z = X \cup Y = \{z_1, \ldots, z_n\}$, $Z' = X' \cup Y' = \{z'_1, \ldots, z'_j, z_{j+1}, \ldots, z_n\}$ and each $z'_i \in Z'$ is an ancestor of $z_i \in Z$.

R. Srikant, R. Agrawal: Mining Generalized Association Rules. In VLDB, 1995.

R-Interestingness: Expected Confidence

Given the rule for $X \Rightarrow Y$ and its ancestor rule $X' \Rightarrow Y'$, then the expected confidence of $X \Rightarrow Y$ is defined as:

$$\mathbb{E}_{X' \Rightarrow Y'}[P(Y|X)] = P(Y' \mid X') \cdot \prod_{i=1}^{j} \frac{P(y_i)}{P(y_i)'}$$

where $Y = \{y_1, \ldots, y_n\}$ and $Y' = \{y'_1, \ldots, y'_j, y_{j+1}, \ldots, y_n\}$ and each $y'_i \in Y'$ is an ancestor of $y_i \in Y$.

R. Srikant, R. Agrawal: Mining Generalized Association Rules. In VLDB, 1995.

Summary Frequent Itemset & Association Rule Mining

- Frequent Itemsets
  - Mining: Apriori algorithm, hash trees, FP-tree
  - support, confidence
- Simple Association Rules
  - Mining: (Apriori)
  - Interestingness measures: support, confidence, correlation
- Hierarchical Association Rules
  - Mining: Top-Down Progressive Deepening
  - Multilevel support thresholds, redundancy, R-interestingness
- Further Topics (not covered)
  - Quantitative Association Rules (for numerical attributes)
  - Multi-dimensional association rule mining

Agenda

Motivation

Motivation

- So far we only considered sets of items. In many applications the order of the items is the crucial information.
- The ordering encodes e.g. temporal aspects, patterns in natural language.
- In an ordered sequence, items are allowed to occur more than one time.

Applications

Bioinformatics (DNA/protein sequences), Web mining, text mining (NLP), sensor data mining, process mining, ...

Sequential Pattern Mining: Basic Notions I

We now consider transactions having an order of the items. Define:

- **Alphabet** Σ is a set of symbols or characters (denoting items)

  e.g. $\Sigma = \{A, B, C, D, E\}$
- **Sequence** $S = s_1 s_2 \dots s_k$ is an ordered list of a length $|S| = k$ items where $s_i \in \Sigma$ is an item at position i also denoted as $S[i]$.

  e.g. $S = CAB, \quad s_3 = B$
- A **k-sequence** is a sequence of length k

  e.g. $S = CAB$ is a 3-sequence
- **Consecutive subsequence** $R = r_1 r_2 \dots r_m$ of $S = s_1 s_2 \dots s_n$ is also a sequence in Σ s.t. $r_1 r_2 \dots r_m = s_j s_{j+1} \dots s_{j+m-1}$, with $1 \leq j \leq n - m + 1$. We say S contains R and denote this by $R \subseteq S$

  e.g. $R_1 = AB \subseteq S = CAB$

Sequential Pattern Mining: Basic Notions II

- In a more general **subsequence** R of S we allow for gaps between the items of R, i.e. the items of the subsequence $R \subseteq S$ must have the same order of the ones in S but there can be some other items between them

  e.g. $R_2 = CB$ is a subsequence of $S = CAB$
- A **prefix** of a sequence S is any consecutive subsequence of the form $S[1:i] = s_1 s_2 \dots s_i$ with $0 \leq i \leq n$, $S[1:0]$ is the empty prefix

  e.g. $R_3 = C, R_4 = CA, R_5 = CAB$ are prefixes of $S = CAB$
- A **suffix** of a sequence S is any consecutive subsequence of the form $S[i:n] = s_i s_{i+1} \dots s_n$ with $1 \leq i \leq n+1$, $S[n+1:n]$ is the empty suffix.

  e.g. $R_4 = AB$ is a suffix of $S = CAB$
- **(Relative) support** of a sequence R in D: $supp(R) = |\{S \in D \mid R \subseteq S\}|/|D|$

Sequential Pattern Mining: Basic Notions III

- S is *frequent (or sequential)* if $supp(S) \geq minSup$ for threshold $minSup$.
- A frequent sequence is *maximal* if it is not a subsequence of any other frequent sequence
- A frequent sequence is *closed* if it is not a subsequence of any other frequent sequence with the same support

Sequential Pattern Mining

Task

Find all frequent subsequences occuring in many transactions.

Difficulty

The number of possible patterns is even larger than for frequent itemset mining!

Example

There are $|\Sigma|^k$ different k-sequences, where $k > |\Sigma|$ is possible and often encountered, e.g. when dealing with DNA sequences where the alphabet only comprises four symbols.

Sequential Pattern Mining Algorithms

Breadth-First Search Based

- GSP (Generalized Sequential Pattern) algorithm[6]
- SPADE[7]
- ...

Depth-First Search Based

- PrefixSpan[8]
- SPAM[9]
- ...

[6] Sirkant & Aggarwal: *Mining sequential patterns: Generalizations and performance improvements*. EDBT 1996

[7] Zaki M J. *SPADE: An efficient algorithm for mining frequent sequences*. Machine learning, 2001, 42(1-2): 31-60.

[8] Pei at. al.: *Mining sequential patterns by pattern-growth: PrefixSpan approach*. TKDE 2004

[9] Ayres, Jay, et al: *Sequential pattern mining using a bitmap representation*. SIGKDD 2002.

GSP (Generalized Sequential Pattern) algorithm

- Breadth-first search: Generate frequent sequences ascending by length
- Given the set of frequent sequences at level k, generate *all* possible sequence extensions or candidates at level $k+1$
- Uses the Apriori principle (anti-monotonicity)
- Next compute the support of each candidate and prune the ones with $supp(c) < minSup$
- Stop the search when no more frequent extensions are possible

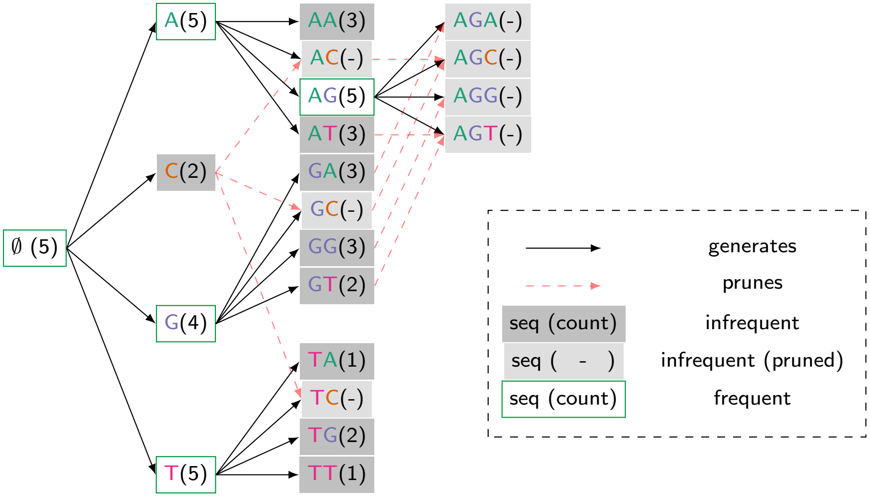
Projection-Based Sequence Mining: PrefixSpan: Representation

- The sequence search space can be organized in a prefix search tree
- The root (level 0) contains the empty sequence with each item $x \in \Sigma$ as one of its children
- A node labelled with sequence: $S = s_1 s_2 \ldots s_k$ at level k has children of the form $S' = s_1 s_2 \ldots s_k s_{k+1}$ at level $k+1$ (i.e. S is a prefix of S' or S' is an extension of S)

Prefix Search Tree: Example

ID	Sequence
S_1	CAGAAGT
S_2	TGACAG
S_3	GAG
S_4	AGTT
S_5	ATAG

$minSup = .8$



Projected Database

- For a database D and an item $s \in \Sigma$, the projected database w.r.t. s is denoted D_s and is found as follows: For each sequence $S_i \in D$ do
  - Find the first occurrence of s in S_i, say at position p
  - $suff_{S_i,s} \leftarrow suffix(S_i)$ starting at position $p+1$
  - Remove infrequent items from $suff_{S_i,s}$
  - $D_s = D_s \cup suff_{S_i,s}$

Example

$minSup = .8$ (i.e. 4 transactions)

ID	Sequence	D_A	D_G	D_T
S_1	CAGAAGT	GAAGT	AAGT	$\emptyset$
S_2	TGACAG	AG	AAG	GAAG
S_3	GAG	G	AG	-
S_4	AGTT	GTT	TT	T
S_5	ATAG	TAG	$\emptyset$	AG

Projection-Based Sequence Mining: PrefixSpan Algorithm

- The *PrefixSpan* algorithm computes the support for only the individual items in the projected databased D_s
- Then performs recursive projections on the frequent items in a depth-first manner

Initialization: $D_R \leftarrow D, R \leftarrow \emptyset, \mathcal{F} \leftarrow \emptyset$
procedure $\textsc{PrefixSpan}(D_R, R, minSup, \mathcal{F})$
    for all $s \in \Sigma$ such that $supp(s, D_R) \geq minSup$ **do**
        $R_s \leftarrow R + s$ ▷ append s to the end of R
        $\mathcal{F} \leftarrow \mathcal{F} \cup \{(R_s, sup(s, D_R))\}$ ▷ calculate support of s for each R_s within D_R
        $D_s \leftarrow \emptyset$
        for all $S_i \in D_R$ **do**
            $S_i' \leftarrow$ projection of S_i w.r.t. item s
            Remove all infrequent symbols from S_i'
            if $S' \neq \emptyset$ **then**
                $D_s \leftarrow D_s \cup S_i'$
        if $D_s \neq \emptyset$ **then**
            PrefixSpan$(D_s, R_s, minSup, \mathcal{F})$

PrefixSpan: Example

minSup $= 0.8$ (i.e. 4 transactions)

$D_\emptyset$

ID	Sequence
S_1	CAGAAGT
S_2	TGACAG
S_3	GAG
S_4	AGTT
S_5	ATAG
	A(5) ~~C(2)~~ G(5) T(4)

D_G

ID	Sequence
S_1	AAGT
S_2	AAG
S_3	AG
S_4	TT
S_5	$\emptyset$
	~~A(3) G(3) T(2)~~

D_T

ID	Sequence
S_1	$\emptyset$
S_2	GAAG
-	-
S_4	T
S_5	AG
	~~A(2) G(2) T(1)~~

D_A

ID	Sequence
S_1	GAAGT
S_2	AG
S_3	G
S_4	GTT
S_5	TAG
	~~A(3)~~ G(5) ~~T(3)~~

D_{AG}

ID	Sequence
S_1	G
S_2	$\emptyset$
S_3	$\emptyset$
S_4	$\emptyset$
S_5	$\emptyset$
	~~G(1)~~

Hence, the frequent sequences are: $\emptyset$, A, G, T, AG

Interval-based Sequential Pattern Mining

Interval-Based Representation

- Deals with the more common interval-based items s (or events).
- Each event has a starting t_s^+ and an ending time point t_s^-, where $t_s^+ < t_s^-$

Application

Health data analysis, Stock market data analysis, etc.

Relationships

Predefined relationships between items are more complex.

- Point-based relationships: before, after, same time.
- Interval-based relationships: Allen's relations[10], End point representation[11], etc.

[10] Allen: Maintaining knowledge about temporal intervals. In Communications of the ACM 1983

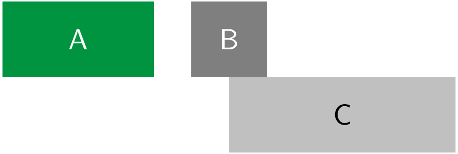
[11] Wu, Shin-Yi, and Yen-Liang Chen: Mining nonambiguous temporal patterns for interval-based events. TKDE 2007

Allen's Relations

Before	Overlaps	Contains	Starts	Finished-By	Meets	Equal
After	Overlapped-By	During	Started-By	Finishes	Met-by	Equal

Problem

- Allen's relationships only describe the relation between two intervals.
- Describing the relationship between k intervals unambiguously requires $\mathcal{O}(k^2)$ comparisons.



Interval-based Sequential Pattern Mining

- *TPrefixSpan*[12] converts interval-based sequences into point-based sequences:

$$\{A^+\}, \{A^-\}, \{B^+\}, \{B^-\}$$

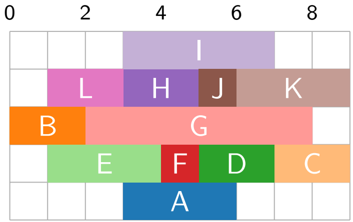
$$\{A^+\}, \{B^+\}, \{A^-\}, \{B^-\}$$

$$\{A^+\}, \{A^-, B^+\}, \{B^-\}$$

- Similar prefix projection mining approach as PrefixSpan algorithm.
- Validation checking is necessary in each expanding iteration to make sure that the appended time point can form an interval with a time point in the prefix.

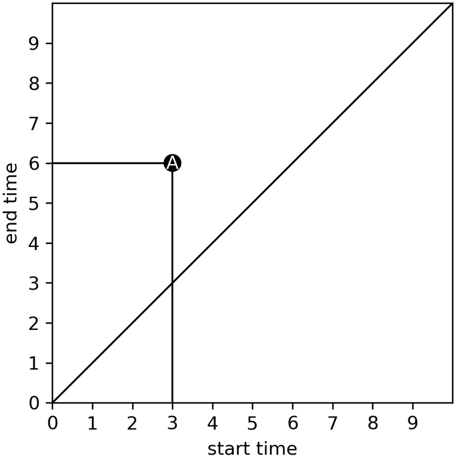
[12] Wu, Shin-Yi, and Yen-Liang Chen: Mining nonambiguous temporal patterns for interval-based events. TKDE 2007

Allen's Relations with *Point Transformation*: Example

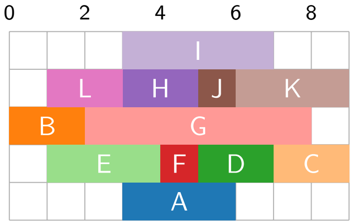


A is the interval starting at time 3 and ending at time 6.
→ Point Transformation maps it in the 2-dim space with $A = (3, 6)$.

A is the reference point in this example!

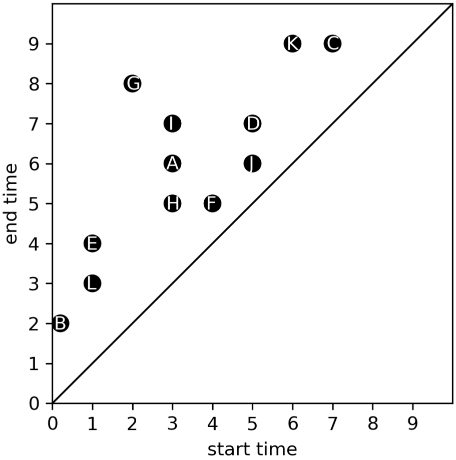


Allen's Relations with *Point Transformation*: Example

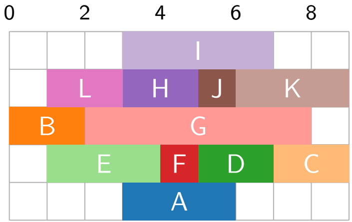


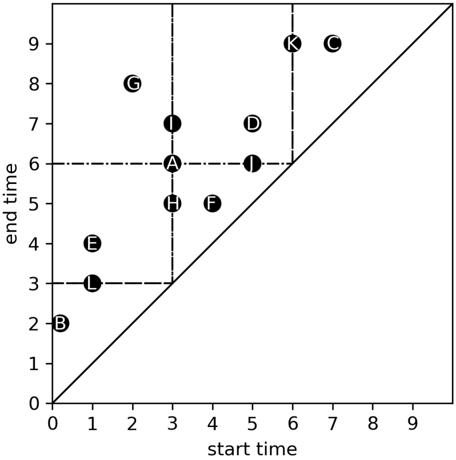
Before: BA
After: CA
Overlaps: DA
Overlapped-By: EA
During: FA
Contains: GA

Started-By: HA
Starts: IA
Finished-By: JA
Finishes: AJ
Met-By: KA
Meets: LA
Equal: AA



Allen's Relations with *Point Transformation*: Example

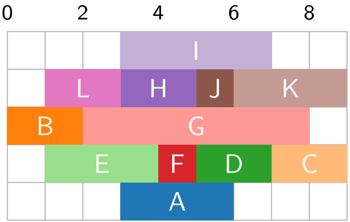


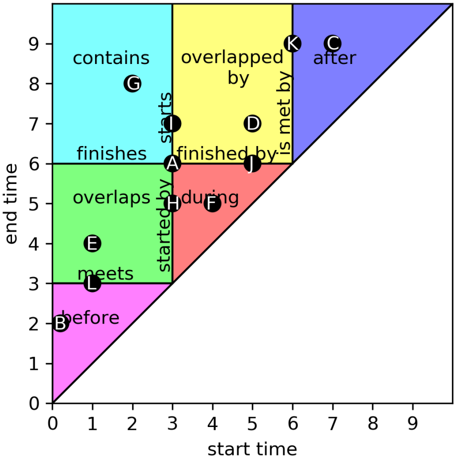


Before: BA
After: CA
Overlaps: DA
Overlapped-By: EA
During: FA
Contains: GA

Started-By: HA
Starts: IA
Finished-By: JA
Finishes: AJ
Met-By: KA
Meets: LA
Equal: AA

Allen's Relations with *Point Transformation*: Example

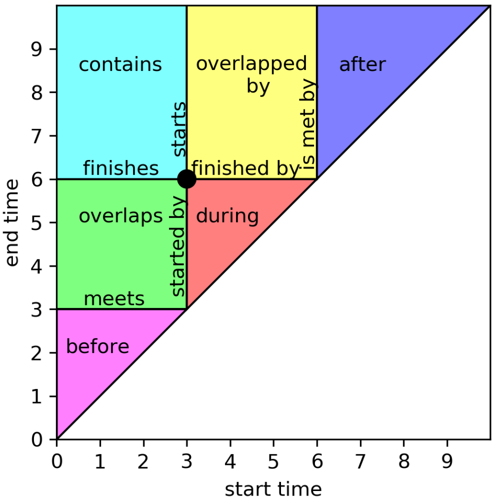




Before: BA
After: CA
Overlaps: DA
Overlapped-By: EA
During: FA
Contains: GA

Started-By: HA
Starts: IA
Finished-By: JA
Finishes: AJ
Met-By: KA
Meets: LA
Equal: AA

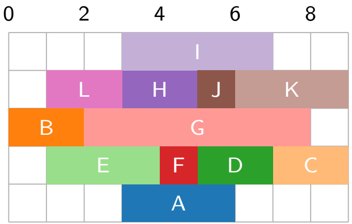
Allen's Relations with *Point Transformation*: Example

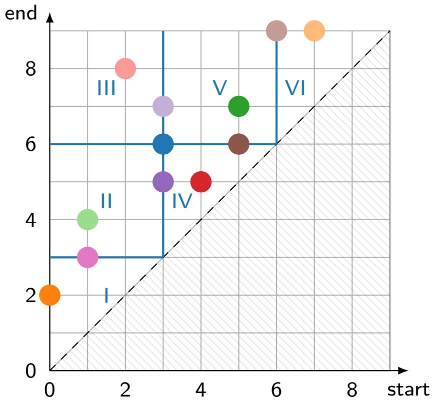


An Open Issue: Considering Timing Information

Idea

Learn pattern from data by clustering, e.g. QTempIntMiner[13], Event Space Miner[14], PIVOTMiner[15]





[13] Guyet, T., & Quiniou, R.: *Mining temporal patterns with quantitative intervals*. ICDMW 2008

[14] Ruan, G., Zhang, H., & Plale, B.: *Parallel and quantitative sequential pattern mining for large-scale interval-based temporal data*. IEEE Big Data 2014

[15] Hassani M., Lu Y. & Seidl T.: *A Geometric Approach for Mining Sequential Patterns in Interval-Based Data Streams*. FUZZ-IEEE 2016