

---

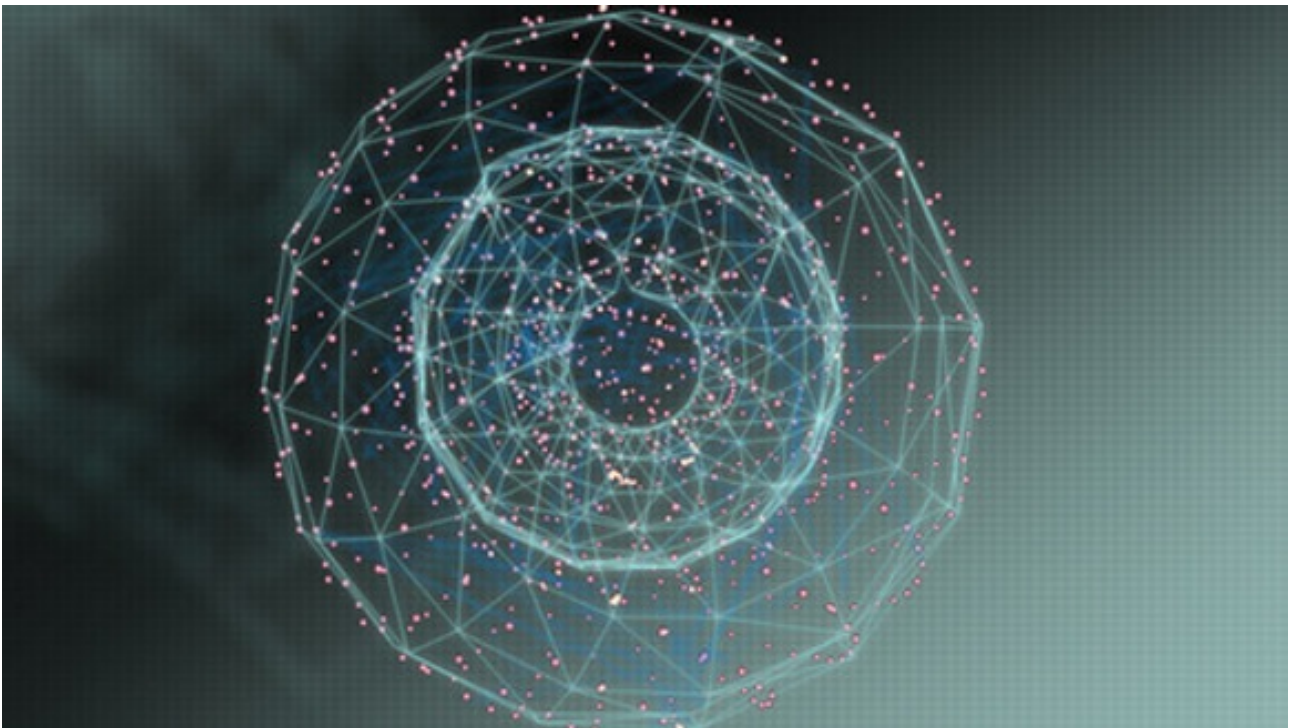
# PATTERN DISCOVERY IN DATA MINING

---

Concepts and challenges in pattern discovery and analysis.  
Pattern evaluation, mining and classification

Course author:

JIAWEI HAN



*University of Illinois at Urbana-Champaign  
&  
Coursera*

2015

# Contents

<b>1</b>	<b>Lecture 2: Pattern Discovery Basic Concepts</b>	<b>3</b>
1.1	Frequent Itemsets (Patterns)	3
1.2	Association Rules	3
1.3	Expressing Patterns in Compressed Form	3
1.4	Recommended readings	3
<b>2</b>	<b>Lecture 3. Efficient Pattern Mining Methods</b>	<b>4</b>
2.1	The Downward Closure Property of Frequent Patterns	4
2.2	The Apriori Algorithm	4
2.2.1	Algorithm pseudocode	4
2.2.2	How to generate candidates?	4
2.3	Extensions or Improvements of Apriori	5
2.3.1	Partitioning	5
2.3.2	Direct Hashing and Pruning (DHP)	5
2.4	Vertical Data Format	5
2.5	A Pattern Growth Approach	6
2.6	CLOSET+: Mining Closed Itemsets by Pattern-Growth	6
2.7	Recommended readings	6
<b>3</b>	<b>Lecture 4: Pattern Evaluation</b>	<b>7</b>
3.1	Interestingness Measures: Lift and $\chi^2$	7
3.1.1	Interestingness Measure: Lift	7
3.1.2	Interestingness Measure: $\chi^2$	7
3.2	Null Invariance Measures	8
3.3	Imbalance Ratio	8
3.4	Recommended Readings	8
<b>4</b>	<b>Lecture 4: Mining Diverse Patterns</b>	<b>8</b>
4.1	Mining Multi-Level Associations	8
4.2	Mining Multi-Dimensional Associations	9
4.3	Mining Quantitative Associations	9
4.4	Mining Negative Correlations	9
4.5	Mining Compressed Patterns	10
4.5.1	Mining Compressed Patterns	10
4.5.2	Redundancy-Aware Top-k Patterns	10
4.6	Mining Colossal Patterns	10
4.6.1	Pattern-Fusion	10
4.6.2	Robustness of Colossal Patterns	11
4.6.3	The Pattern-Fusion Algorithm	11
4.7	Recommended Readings	12
<b>5</b>	<b>Constraint-Based Pattern Mining</b>	<b>12</b>
5.1	Meta-Rule Guided Mining	12
5.2	Kinds of Constraints	12
5.2.1	Pattern space pruning constraints	13
5.2.2	Data space pruning constraints	13
5.3	Recommended Readings	13

<b>6</b>	<b>Sequential Pattern Mining</b>	<b>13</b>
6.1	Sequential Pattern . . . . .	13
6.2	GSP: Apriori-Based Sequential Pattern Mining . . . . .	14
6.3	SPADE: Sequential Pattern Mining in Vertical Data Format . . . . .	14
6.4	PrefixSpan: A Pattern-Growth Approach . . . . .	15
6.5	CloSpan: Mining Closed Sequential Patterns . . . . .	15
6.6	Constraint-Based Sequential-Pattern Mining . . . . .	16
6.6.1	Timing-Based Constraints . . . . .	16
6.7	Recommended Readings . . . . .	16

# 1 Lecture 2: Pattern Discovery Basic Concepts

## 1.1 Frequent Itemsets (Patterns)

$X$  = itemset

- **(absolute) support (count) of  $X$ :** Frequency or the number of occurrences of an itemset  $X$
- **(relative) support,  $s$ :** The fraction of transactions that contains  $X$  (i.e., the probability that a transaction contains  $X$ )
- An itemset  $X$  is **frequent** if the support of  $X$  is no less than a *minsup* threshold (denoted as  $\sigma$ ):  $\text{sup}(X) \geq \sigma$ .

## 1.2 Association Rules

Association rules:  $X \rightarrow Y(s, c)$ :

- **Support ( $s$ ):** the probability that a transaction contains  $X \cup Y$ :

$$\text{sup}(X \rightarrow Y) = P(X \cup Y)$$

- **Confidence ( $c$ ):** the conditional probability that a transaction containing  $X$  also contains  $Y$ :

$$c = P(Y | X) = \frac{\text{sup}(X \cup Y)}{\text{sup}(X)}$$

## 1.3 Expressing Patterns in Compressed Form

**Definition. Closed patterns:** A pattern (itemset)  $X$  is closed if  $X$  is frequent, and there exists no super-pattern  $Y \supset X$ , with the same support as  $X$ .

Closed pattern is a lossless compression of frequent patterns.

**Definition. Max-patterns:** A pattern  $X$  is a max-pattern if  $X$  is frequent and there exists no frequent super-pattern  $Y \supset X$ .

Max-pattern is a lossy compression!

## 1.4 Recommended readings

- R. Agrawal, T. Imielinski, and A. Swami, «Mining association rules between sets of items in large databases», in Proc. of SIGMOD'93
- R. J. Bayardo, «Efficiently mining long patterns from databases», in Proc. of SIGMOD'98
- N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal, «Discovering frequent closed itemsets for association rules», in Proc. of ICDT'99
- J. Han, H. Cheng, D. Xin, and X. Yan, «Frequent Pattern Mining: Current Status and Future Directions», Data Mining and Knowledge Discovery, 15(1): 55-86, 2007

## 2 Lecture 3. Efficient Pattern Mining Methods

### 2.1 The Downward Closure Property of Frequent Patterns

The downward closure (also called «Apriori») property of frequent patterns: **Any subset of a frequent itemset must be frequent.** Apriori pruning principle: **If there is any itemset which is infrequent, its superset should not even be generated!**

Scalable mining Methods: Three major approaches

- Level-wise, join-based approach: Apriori (2.2)
- Vertical data format approach: Eclat (2.4)
- Frequent pattern projection and growth: FPgrowth (2.5)

### 2.2 The Apriori Algorithm

#### 2.2.1 Algorithm pseudocode

$C_k$ : Candidate itemset of size  $k$

$F_k$ : Frequent itemset of size  $k$

TDB = transactional database

---

#### Algorithm 1 The Apriori Algorithm

---

```
 $k := 1$ 
 $F_k :=$  frequent items # frequent 1-itemset
while  $F_k \neq \emptyset$  do
     $C_{k+1} :=$  candidates generated from  $F_k$  # candidate generation
    Derives  $F_{k+1}$  by counting candidates in  $C_{k+1}$  with respect to TDB at minsup
     $k := k + 1$ 
end while
return  $\cup_k F_k$  # return  $F_k$  generated at each level
```

---

#### 2.2.2 How to generate candidates?

- Step1: self-joining  $F_k$
- Step2: pruning

---

#### Algorithm 2 Step1: self-joining $F_k$

---

```
insert into  $C_k$ 
select  $p.item_1, p.item_2, \dots, p.item_{k-1}, q.item_{k-1}$ 
from  $F_{k-1}$  as  $p, F_{k-1}$  as  $q$ 
where  $p.item_1 = q.item_1, \dots, p.item_{k-2} = q.item_{k-2}, p.item_{k-1} < q.item_{k-1}$ 
```

---

---

**Algorithm 3** Step2: pruning

---

```
for all itemsets  $c$  in  $C_k$  do
  for all  $(k-1)$  subsets  $s$  of  $c$  do
    if  $s$  is not in  $F_{k-1}$  then
      delete  $c$  from  $C_k$ 
    end if
  end for
end for
```

---

## 2.3 Extensions or Improvements of Apriori

- Reduce passes of transaction database scans
  - Partitioning
  - Dynamic itemset counting
- Shrink the number of candidates
  - Hashing
  - Pruning by support lower bounding
  - Sampling
- Exploring special data structures
  - Tree projection
  - H-miner
  - Hypercube decomposition

### 2.3.1 Partitioning

**Theorem.** Any itemset that is potentially frequent in TDB must be frequent in at least one of the partitions of TDB

Method: Scan Database Only Twice:

- Scan 1: Partition database (how?) and find local frequent patterns
- Scan 2: Consolidate global frequent patterns (how to?)

### 2.3.2 Direct Hashing and Pruning (DHP)

Observation: A  $k$ -itemset whose corresponding hashing bucket count is below the threshold cannot be frequent

## 2.4 Vertical Data Format

**ECLAT** - Equivalence Class Transformation

Frequent patterns are derived based on vertical intersections. To accelerate data mining you can use **diffset**: only keep track of differences of tids.

## 2.5 A Pattern Growth Approach

FP-tree - frequent pattern tree

TID	Items in the Transaction	Ordered, frequent items
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o, w}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

Figure 1: Transational DB

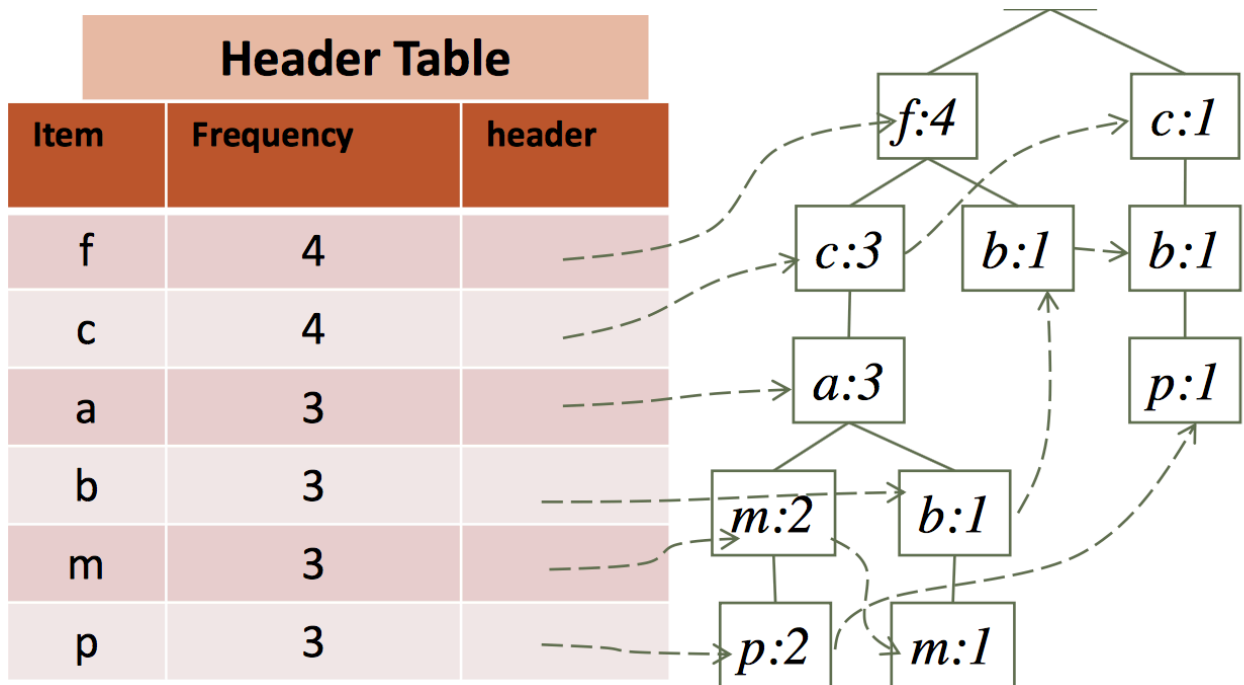


Figure 2: FP-tree

## 2.6 CLOSET+: Mining Closed Itemsets by Pattern-Growth

Itemset merging: *If Y appears in every occurrence of X, then Y is merged with X*

## 2.7 Recommended readings

- R. Agrawal and R. Srikant, «Fast algorithms for mining association rules», VLDB'94
- A. Savasere, E. Omiecinski, and S. Navathe, «An efficient algorithm for mining association rules in large databases», VLDB'95
- J. S. Park, M. S. Chen, and P. S. Yu, «An effective hash-based algorithm for mining association rules», SIGMOD'95

- S. Sarawagi, S. Thomas, and R. Agrawal, «Integrating association rule mining with relational database systems: Alternatives and implications», SIGMOD'98
- M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li, «Parallel algorithm for discovery of association rules», Data Mining and Knowledge Discovery, 1997
- J. Han, J. Pei, and Y. Yin, «Mining frequent patterns without candidate generation», SIGMOD'00
- M. J. Zaki and Hsiao, «CHARM: An Efficient Algorithm for Closed Itemset Mining», SDM'02
- J. Wang, J. Han, and J. Pei, «CLOSET+: Searching for the Best Strategies for Mining Frequent Closed Itemsets», KDD'03
- C. C. Aggarwal, M.A., Bhuiyan, M. A. Hasan, «Frequent Pattern Mining Algorithms: A Survey», in Aggarwal and Han (eds.): Frequent Pattern Mining, Springer, 2014

### 3 Lecture 4: Pattern Evaluation

#### 3.1 Interestingness Measures: Lift and $\chi^2$

##### 3.1.1 Interestingness Measure: Lift

**Lift** - measure of dependent/correlated events:

$$\text{lift}(B, C) = \frac{c(B \rightarrow C)}{s(C)} = \frac{s(B \cup C)}{s(B) \times s(C)}$$

Lift(B, C) may tell how B and C are correlated:

- $\text{Lift}(B, C) = 1$ : B and C are independent
- $\text{Lift}(B, C) > 1$ : positively correlated
- $\text{Lift}(B, C) < 1$ : negatively correlated

##### 3.1.2 Interestingness Measure: $\chi^2$

$$\chi^2 = \sum \frac{(\text{Observed} - \text{Expected})^2}{\text{Expected}}$$

General rules:

- $\chi^2 = 0$ : independent
- $\chi^2 > 0$ : correlated, either positive or negative, so it needs additional test

Too many null transactions may lead to invalid correlation result!



## 3.2 Null Invariance Measures

$$\begin{aligned}\text{AllConf}(A, B) &= \frac{s(A \cup B)}{\max\{s(A), s(B)\}} \\ \text{Jaccard}(A, B) &= \frac{s(A \cup B)}{s(A) + s(B) - s(A \cup B)} \\ \text{Cosine}(A, B) &= \frac{s(A \cup B)}{\sqrt{s(A) \times s(B)}} \\ \text{Kulczynsky}(A, B) &= \frac{1}{2} \left( \frac{s(A \cup B)}{s(A)} + \frac{s(A \cup B)}{s(B)} \right) \\ \text{MacConf}(A, B) &= \max \left\{ \frac{s(A)}{s(A \cup B)}, \frac{s(B)}{s(A \cup B)} \right\}\end{aligned}$$

## 3.3 Imbalance Ratio

IR (Imbalance Ratio): measure the imbalance of two itemsets A and B in rule implications:

$$\text{IR}(A, B) = \frac{|s(A) - s(B)|}{s(A) + s(B) - s(A \cup B)}$$

Kulczynski and Imbalance Ratio (IR) together present a clear picture

## 3.4 Recommended Readings

- C. C. Aggarwal and P. S. Yu. A New Framework for Itemset Generation. PODS'98
- S. Brin, R. Motwani, and C. Silverstein. Beyond market basket: Generalizing association rules to correlations. SIGMOD'97
- M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. I. Verkamo. Finding interesting rules from large sets of discovered association rules. CIKM'94
- E. Omiecinski. Alternative Interest Measures for Mining Associations. TKDE'03
- P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the Right Interestingness Measure for Association Patterns. KDD'02
- T. Wu, Y. Chen and J. Han, Re-Examination of Interestingness Measures in Pattern Mining: A Unified Framework, Data Mining and Knowledge Discovery, 21(3):371-397, 2010

# 4 Lecture 4: Mining Diverse Patterns

## 4.1 Mining Multi-Level Associations

Items often form hierarchies. How to set min-support thresholds? **Level-reduced min-support:** items at the lower level are expected to have lower support.

Efficient mining: **shared** multi-level mining. Use the lowest min-support to pass down the set of candidates.

Redundancy<sup>1</sup> filtering: some rules may be redundant due to «ancestor»<sup>2</sup> relationships between items. A rule is **redundant** if:

- its support is close to the «expected» value, according to its «ancestor» rule
- it has a similar confidence as its «ancestor».

It is necessary to have customized min-support settings for different kinds of items: group-based «individualized» min-support.

## 4.2 Mining Multi-Dimensional Associations

Rules can be single-dimensional or multi-dimensional:

- Single-dimensional:

$$\text{buys}(X, \text{«milk»}) \Rightarrow \text{buys}(X, \text{«bread»})$$

- Inter-dimension association rule:

$$\text{age}(X, \text{«18-25»}) \wedge \text{occupation}(X, \text{«student»}) \Rightarrow \text{buys}(X, \text{«coke»})$$

- Hybrid-dimension association rules:

$$\text{age}(X, \text{«18-25»}) \wedge \text{buys}(X, \text{«popcorn»}) \Rightarrow \text{buys}(X, \text{«coke»})$$

Attributes can be categorical or numerical

## 4.3 Mining Quantitative Associations

Methods:

- Static discretization based on predefined concept hierarchies
- Dynamic discretization based on data distribution
- Clustering: distance-based association
- Deviation analysis

## 4.4 Mining Negative Correlations

- Rare patterns = very low support but interesting
- Negative patterns = negatively correlated, unlikely to happen together

A support-based definition: if itemsets A and B are both frequent but rarely occur together, i.e.,  $\text{sup}(A \cup B) \ll \text{sup}(A) \times \text{sup}(B)$  then A and B are negatively correlated.

The support-based definition is not null-invariant!

A Kulczynski measure-based definition: if itemsets A and B are frequent but  $\frac{P(A|B) + P(B|A)}{2} < \varepsilon$ , where  $\varepsilon$  is a negative pattern threshold, then A and B are negatively correlated.

---

<sup>1</sup>Redundancy - избыточность

<sup>2</sup>Ancestor – предок

## 4.5 Mining Compressed Patterns

### 4.5.1 Mining Compressed Patterns

Pattern distance measure:

$$Dist(P_1, P_2) = 1 - \frac{|T(P_1) \cap T(P_2)|}{|T(P_1) \cup T(P_2)|}$$

**$\delta$ -clustering.** For each pattern P, find all patterns which can be expressed by P and whose distance to P is within  $\delta$  ( $\delta$ -cover). All patterns in the cluster can be represented by P = compressed patterns.<sup>3</sup>

### 4.5.2 Redundancy-Aware Top-k Patterns

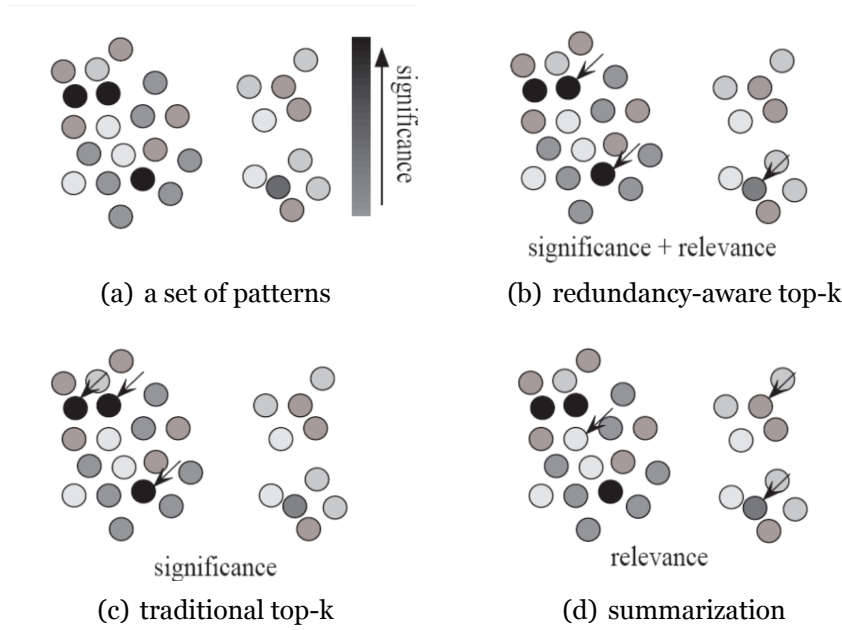


Figure 3: Desired patterns: high significance & low redundancy

Use **MMS (Maximal Marginal Significance)** for measuring the combined significance of a pattern set.<sup>4</sup>

## 4.6 Mining Colossal Patterns

### 4.6.1 Pattern-Fusion

**Pattern fusion strategy:** fuse small patterns together in one step to generate new pattern candidates of significant sizes.

Subpatterns  $\alpha_1$  to  $\alpha_k$  cluster tightly around the colossal pattern  $\alpha$  by sharing a similar support. Such subpatterns are **core patterns** of  $\alpha$ . A colossal pattern can be generated by merging a set of core patterns.

<sup>3</sup>Method for efficient, direct mining of compressed frequent patterns: Xin et al., VLDB'05.

<sup>4</sup>Xin et al., Extracting Redundancy-Aware Top-K Patterns, KDD'06.

#### 4.6.2 Robustness of Colossal Patterns

**Definition.** For a frequent pattern  $\alpha$ , a subpattern  $\beta$  is a  $\tau$ -core pattern of  $\alpha$  if  $\beta$  shares a similar support set with  $\alpha$ , i.e.,

$$\frac{|D_\alpha|}{|D_\beta|} \geq \tau, 0 < \tau \leq 1,$$

where  $\tau$  is called the **core ratio**.

**Definition.**  $(d, \tau)$ -robustness<sup>5</sup>: a pattern  $\alpha$  is  $(d, \tau)$ -robust if  $d$  is the maximum number of items that can be removed from  $\alpha$  for the resulting pattern to remain a  $\tau$ -core pattern of  $\alpha$ :

$$d = \max_{\beta} \{|\alpha| - |\beta| \mid \beta \subseteq \alpha, \text{ and } \beta \text{ is a } \tau\text{-core pattern of } \alpha\}$$

For a pattern  $\alpha$  let  $C_\alpha$  be the set of all its core patterns for a specified  $\tau$ :

$$C_\alpha = \{\beta \mid \beta \subseteq \alpha, \frac{|D_\alpha|}{|D_\beta|} \geq \tau\}$$

**Theorem.** For a  $(d, \tau)$ -robust pattern  $\alpha$ :

$$|C_\alpha| \geq 2^d$$

**Robustness of Colossal Patterns:** a colossal pattern tends to have much more core patterns than small patterns. Such core patterns can be clustered together to form «dense balls» based on pattern distance defined by

$$Dist(\alpha, \beta) = 1 - \frac{|D_\alpha \cap D_\beta|}{|D_\alpha \cup D_\beta|}$$

**Theorem.** For two patterns  $\beta_1, \beta_2 \in C_\alpha$

$$Dist(\beta_1, \beta_2) \leq r(\tau), \text{ where } r(\tau) = 1 - \frac{1}{2/\tau - 1}$$

#### 4.6.3 The Pattern-Fusion Algorithm

- Initialization (Creating initial pool): Use an existing algorithm to mine all frequent patterns up to a small size, e.g., 3
- Iteration (Iterative Pattern Fusion):
  - At each iteration,  $K$  seed patterns are randomly picked from the current pattern pool
  - For each seed pattern thus picked, we find all the patterns within a bounding ball centered at the seed pattern
  - All these patterns found are fused together to generate a set of super-patterns
  - All the super-patterns thus generated form a new pool for the next iteration
- Termination: when the current pool contains no more than  $K$  patterns at the beginning of an iteration

---

<sup>5</sup>Robustness - прочность

## 4.7 Recommended Readings

- R. Srikant and R. Agrawal, «Mining generalized association rules», VLDB'95
- Y. Aumann and Y. Lindell, «A Statistical Theory for Quantitative Association Rules», KDD'99
- D. Xin, J. Han, X. Yan and H. Cheng, «On Compressing Frequent Patterns», Knowledge and Data Engineering, 60(1): 5-29, 2007
- D. Xin, H. Cheng, X. Yan, and J. Han, «Extracting Redundancy-Aware Top-K Patterns», KDD'o6
- F. Zhu, X. Yan, J. Han, P. S. Yu, and H. Cheng, «Mining Colossal Frequent Patterns by Core Pattern Fusion», ICDE'o7
- J. Han, H. Cheng, D. Xin, and X. Yan, «Frequent Pattern Mining: Current Status and Future Directions», Data Mining and Knowledge Discovery, 15(1): 55-86, 2007

## 5 Constraint-Based Pattern Mining

### 5.1 Meta-Rule Guided Mining

In general, (meta) rules can be in the form of

$$P_1 \wedge P_2 \wedge \dots \wedge P_l \Rightarrow Q_1 \wedge Q_2 \wedge \dots \wedge Q_r$$

Method to find meta-rules:

- Find frequent ( $l + r$ ) predicates (based on min-support)
- Push constraints deeply when possible into the mining process
- Also, push min\_conf, min\_correlation, and other measures as early as possible (measures acting as constraints)

### 5.2 Kinds of Constraints

- Pattern space pruning constraints
  - Anti-monotonic: If constraint  $c$  is violated, its further mining can be terminated
  - Monotonic: If  $c$  is satisfied, no need to check  $c$  again
  - Succinct<sup>6</sup>: if the constraint  $c$  can be enforced by directly manipulating the data
  - Convertible:  $c$  can be converted to monotonic or anti-monotonic if items can be properly ordered in processing
- Data space pruning constraints
  - Data succinct: Data space can be pruned at the initial pattern mining process
  - Data anti-monotonic: If a transaction  $t$  does not satisfy  $c$ , then  $t$  can be pruned to reduce data processing effort

Anti-monotonic constraints have more pruning power than monotonic constraints.

---

<sup>6</sup>Succinct - краткий

### 5.2.1 Pattern space pruning constraints

Constraint  $c$  is **anti-monotone**: if an itemset  $S$  violates constraint  $c$ , so does any of its superset. That is, mining on itemset  $S$  can be terminated. For example, constraint  $\sup(S) \geq \sigma$  is anti-monotone.

A constraint  $c$  is **monotone**: if an itemset  $S$  satisfies the constraint  $c$ , so does any of its superset. That is, we do not need to check  $c$  in subsequent mining. For example, constraints  $\sum(S.price) \geq v$  or  $\min(S.price) \leq v$  are monotone.

### 5.2.2 Data space pruning constraints

A constraint  $c$  is **data anti-monotone**: if a data entry  $t$  cannot satisfy a pattern  $p$  under constraint  $c$ ,  $t$  cannot satisfy  $p$ 's superset either. That's why, data entry  $t$  can be pruned.

**Succinctness**: if the constraint  $c$  can be enforced by directly manipulating the data.

**Convertible constraints**: convert tough<sup>7</sup> constraints into (anti-)monotone by proper ordering of items in transactions. For example, ordering items in value-descending order makes the constraint  $\text{avg}(S.profit) > 20$  anti-monotone *if the patterns grow in the right order*.

## 5.3 Recommended Readings

- R. Srikant, Q. Vu, and R. Agrawal, «Mining association rules with item constraints», KDD'97
- R. Ng, L.V.S. Lakshmanan, J. Han & A. Pang, Exploratory mining and pruning optimizations of constrained association rules», SIGMOD'98
- G. Grahne, L. Lakshmanan, and X. Wang, «Efficient mining of constrained correlated sets», ICDE'00
- J. Pei, J. Han, and L. V. S. Lakshmanan, «Mining Frequent Itemsets with Convertible Constraints», ICDE'01
- J. Pei, J. Han, and W. Wang, «Mining Sequential Patterns with Constraints in Large Databases», CIKM'02
- F. Bonchi, F. Giannotti, A. Mazzanti, and D. Pedreschi, «ExAnte: Anticipated Data Reduction in Constrained Pattern Mining», PKDD'03
- F. Zhu, X. Yan, J. Han, and P. S. Yu, «gPrune: A Constraint Pushing Framework for Graph Pattern Mining», PAKDD'07

## 6 Sequential Pattern Mining

### 6.1 Sequential Pattern

Sequence  $\rightarrow$  Element  $\rightarrow$  Item or Event (items within an element are unordered)

---

<sup>7</sup>Tough - жесткий

The Apriori property still holds: if a subsequence  $s_1$  is infrequent, none of  $s_1$ 's super-sequences can be frequent.

Algorithms:

- Generalized Sequential Patterns: **GSP**
- Vertical format-based mining: **SPADE**
- Pattern-growth methods: **PrefixSpan**
- Mining closed sequential patterns: **CloSpan**

## 6.2 GSP: Apriori-Based Sequential Pattern Mining

---

### Algorithm 4 GSP

---

```

k = 1
repeat
  find length=k frequent sequences
  Apriori: remove candidates with sup < min_sup
  length=k frequent sequences  $\Rightarrow$  length=(k+1) candidate sequences
  k = k + 1
until no frequent sequences or candidates

```

---

## 6.3 SPADE: Sequential Pattern Mining in Vertical Data Format

**SPADE** = Sequential **P**attern **D**iscovery using **E**quivalent Class

SID	Sequence
1	<a( <u>abc</u> )(a <u>c</u> )d(cf)>
2	<(ad)c(bc)(ae)>
3	<(ef)( <u>ab</u> )(df) <u>cb</u> >
4	<eg(af)cbc>

*min\_sup = 2*

Figure 4: A sequence database

SID	EID	Items
1	1	a
1	2	abc
1	3	ac
1	4	d
1	5	cf
2	1	ad
2	2	c
2	3	bc
2	4	ae
3	1	ef
3	2	ab
3	3	df
3	4	c
3	5	b
4	1	e
4	2	g
4	3	af
4	4	c
4	5	b
4	6	c

a		b		...
SID	EID	SID	EID	...
1	1	1	2	
1	2	2	3	
1	3	3	2	
2	1	3	5	
2	4	4	5	
3	2			
4	3			

ab			ba		
SID	EID (a)	EID(b)	SID	EID (b)	EID(a)
1	1	2	1	2	3
2	1	3	2	3	4
3	2	5			
4	3	5			

aba					...
SID	EID (a)	EID(b)	EID(a)		
1	1	2	3		
2	1	3	4		

Figure 5: SPADE algorithm

## 6.4 PrefixSpan: A Pattern-Growth Approach

**PrefixSpan** = Prefix-projected Sequential pattern mining

SID	Sequence	Prefix	Suffix (Projection)
10	<a(abc)(ac)d(cf)>	<a>	<(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>	<aa>	<(_bc)(ac)d(cf)>
30	<(ef)(ab)(df)cb>	<ab>	<(_c)(ac)d(cf)>
40	<eg(af)cbc>		

Figure 6: SPADE algorithm

PrefixSpan Mining: Prefix Projections

- Step 1: Find length-1 sequential patterns: <a>, <b>, etc.
- Step 2: Divide search space and mine each projected DB: <a>-projected DB, <b>-projected DB, etc.

## 6.5 CloSpan: Mining Closed Sequential Patterns

**Definition.** A closed sequential pattern  $\alpha$ : there exists no superpattern  $\beta$  such that  $\beta$  and  $\alpha$  have the same support:



$$CS = \{\alpha \mid \alpha \in FS \text{ and } \nexists \beta \in FS, \text{ such that } \alpha \subseteq \beta \text{ and } sup(\alpha) = sup(\beta)\}$$

CloSpan is based on this property: if  $s \supset s_1$  then  $s$  is closed only if two project DBs have the same size. So redundant search space can be pruned using **Backward Subpattern** and **Backward Superpattern** pruning.

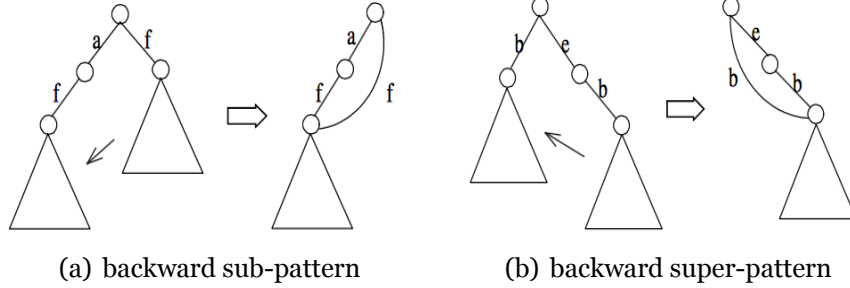


Figure 7: CloSpan pruning algorithm

## 6.6 Constraint-Based Sequential-Pattern Mining

- **Anti-monotonic:** If  $S$  violates  $c$ , the super-sequences of  $S$  also violate  $c$
- **Monotonic:** If  $S$  satisfies  $c$ , the super-sequences of  $S$  also do so
- **Data anti-monotonic:** If a sequence  $s_1$  with respect to  $S$  violates  $c_3$ ,  $s_1$  can be removed
- **Succinct:** Enforce constraint  $c$  by explicitly manipulating data
- **Convertible:** Projection based on the sorted value not in sequence order

### 6.6.1 Timing-Based Constraints

- **Order constraint:** Some items must happen before the other. Anti-monotonic: constraint-violating sub-patterns pruned
- **Min-gap/max-gap constraint:** Confines two elements in a pattern. Succinct: enforced directly during pattern growth
- **Max-span constraint:** Maximum allowed time difference between the 1st and the last elements in the pattern. Succinct: enforced directly when the 1st element is determined
- **Window size constraint:** Time window allows a group of consecutive elements of a data-sequence to be merged and treated as a single element as long as their timestamps are within the user-specified window-size.

## 6.7 Recommended Readings

- M. N. Garofalakis, R. Rastogi, K. Shim: Mining Sequential Patterns with Regular Expression Constraints. IEEE Trans. Knowl. Data Eng. 14(3), 2002

- H. Mannila, H. Toivonen, and A. I. Verkamo, "Discovery of frequent episodes in event sequences", Data Mining and Knowledge Discovery, 1997
- J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu, "Mining Sequential Patterns by Pattern-Growth: The PrefixSpan Approach", IEEE TKDE, 16(10), 2004
- J. Pei, J. Han, and W. Wang, "Constraint-based sequential pattern mining: the pattern-growth methods", J. Int. Inf. Sys., 28(2), 2007
- R. Srikant and R. Agrawal, "Mining sequential patterns: Generalizations and performance improvements", EDBT'96
- X. Yan, J. Han, and R. Afshar, "CloSpan: Mining Closed Sequential Patterns in Large Datasets", SDM'03
- M. Zaki, "SPADE: An Efficient Algorithm for Mining Frequent Sequences", Machine Learning, 2001