Association rule mining

Philippe Fournier-Viger
http://www.philippe-Fournier-viger.com

R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. Research Report RJ 9839, IBM Almaden Research Center, San Jose, California, June 1994.

Source code and datasets available in the SPMF library

Association rule mining

- A **data analysis** task proposed by Agrawal & Srikant (1994)
- **Goal**: find interesting associations between values in a dataset.
- E.g. discover the products that people like to purchase together frequently

- In this video, I will explain "*association rule mining*" and the relationship with "itemset mining"

ITEMSET MINING (BRIEF REVIEW)

Frequent itemset mining (频繁项集挖掘)

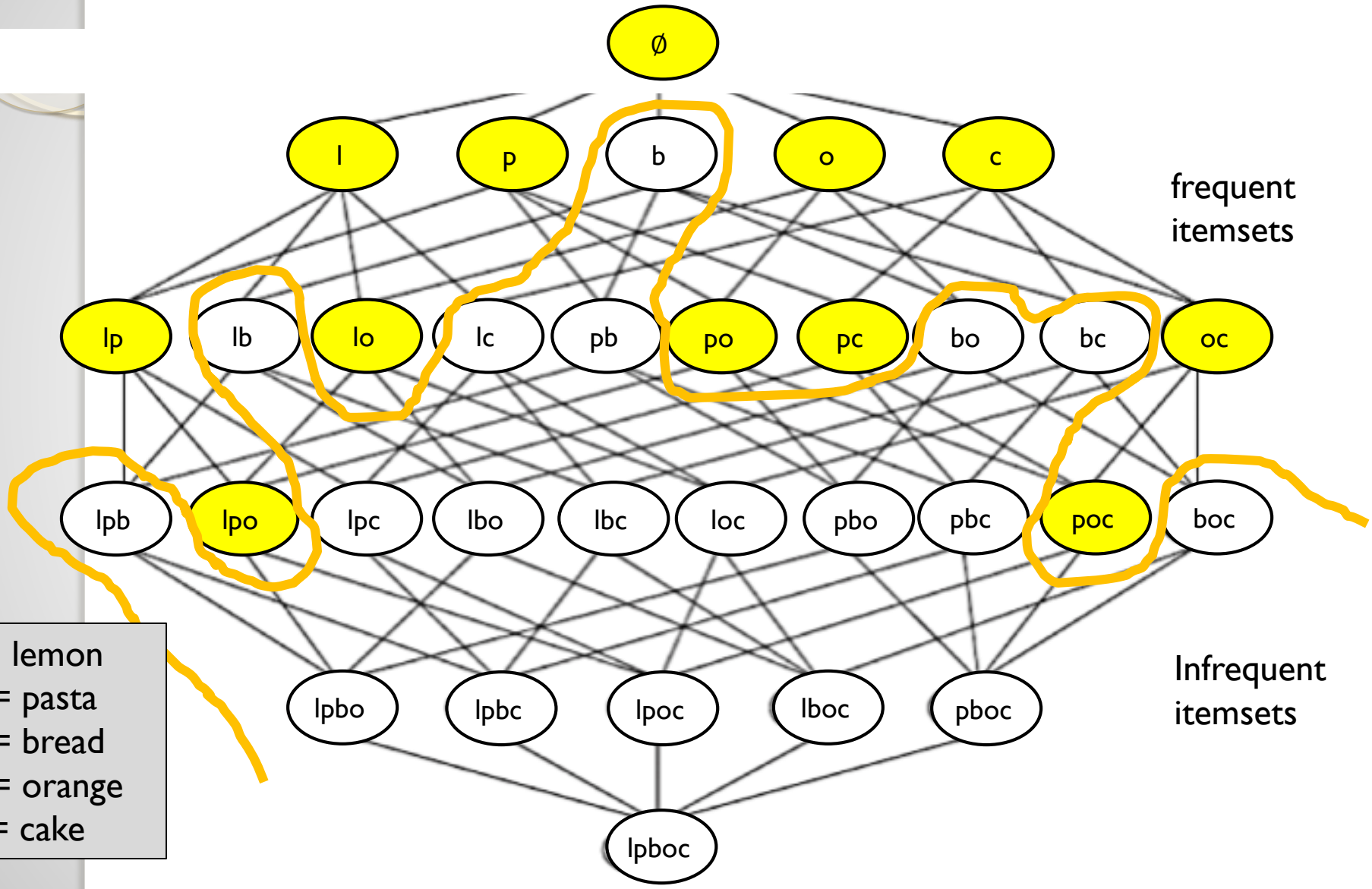
A transaction database:

Transaction	Items appearing in the transaction
T1	{pasta, lemon, bread, orange}
T2	{pasta, lemon}
T3	{pasta, orange, cake}
T4	{pasta, lemon, orange cake}

For *minsup* = 2, the frequent itemsets are:

{lemon}, {pasta}, {orange}, {cake}, {lemon, pasta}, {lemon, orange}, {pasta, orange}, {pasta, cake}, {orange, cake}, {lemon, pasta, orange}

minsup =2


Ø
l
p
b
o
c
lp
lb
lo
lc
pb
po
pc
bo
bc
oc
lpb
lpo
lpc
lbo
lbc
loc
pbo
pbc
poc
boc
lpbo
lpbc
lpoc
lboc
pboc
lpboc
frequent itemsets
Infrequent itemsets
l = lemon
p = pasta
b = bread
0 = orange
c = cake

Property 2: Let there be an itemset Y.
If there exists an itemset $X \subset Y$ such that X is infrequent, then Y is infrequent.

Example:

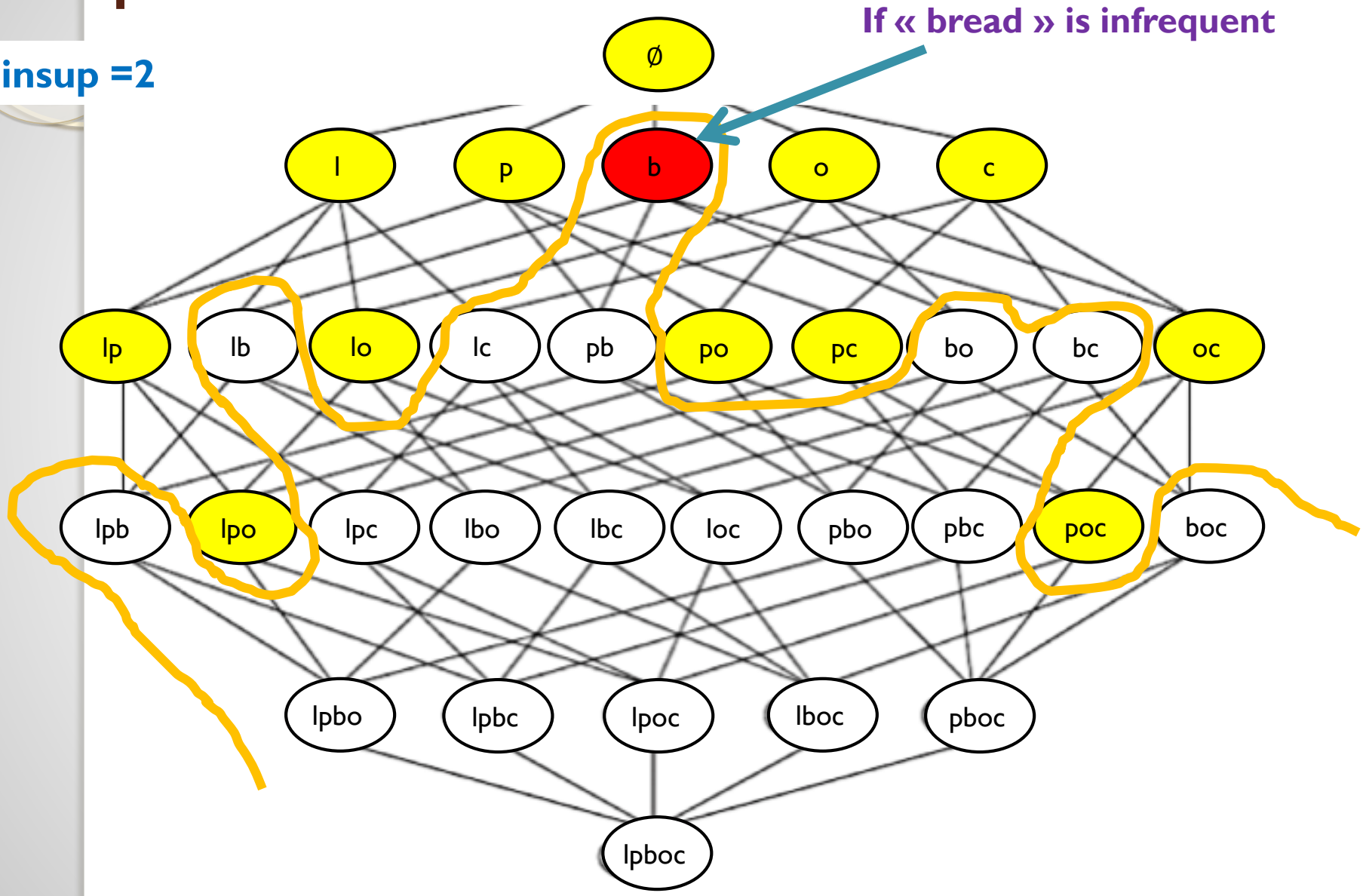
- Consider **{bread, lemon}.**
- If we know that **{bread}** is infrequent, then we can infer that **{bread, lemon}** is also infrequent.

Transaction	Items appearing in the transaction
T1	{pasta, lemon, bread, orange}
T2	{pasta, lemon}
T3	{pasta, orange, cake}
T4	{pasta, lemon, orange, cake}

This property is useful to reduce the search space.

Example:

minsup =2

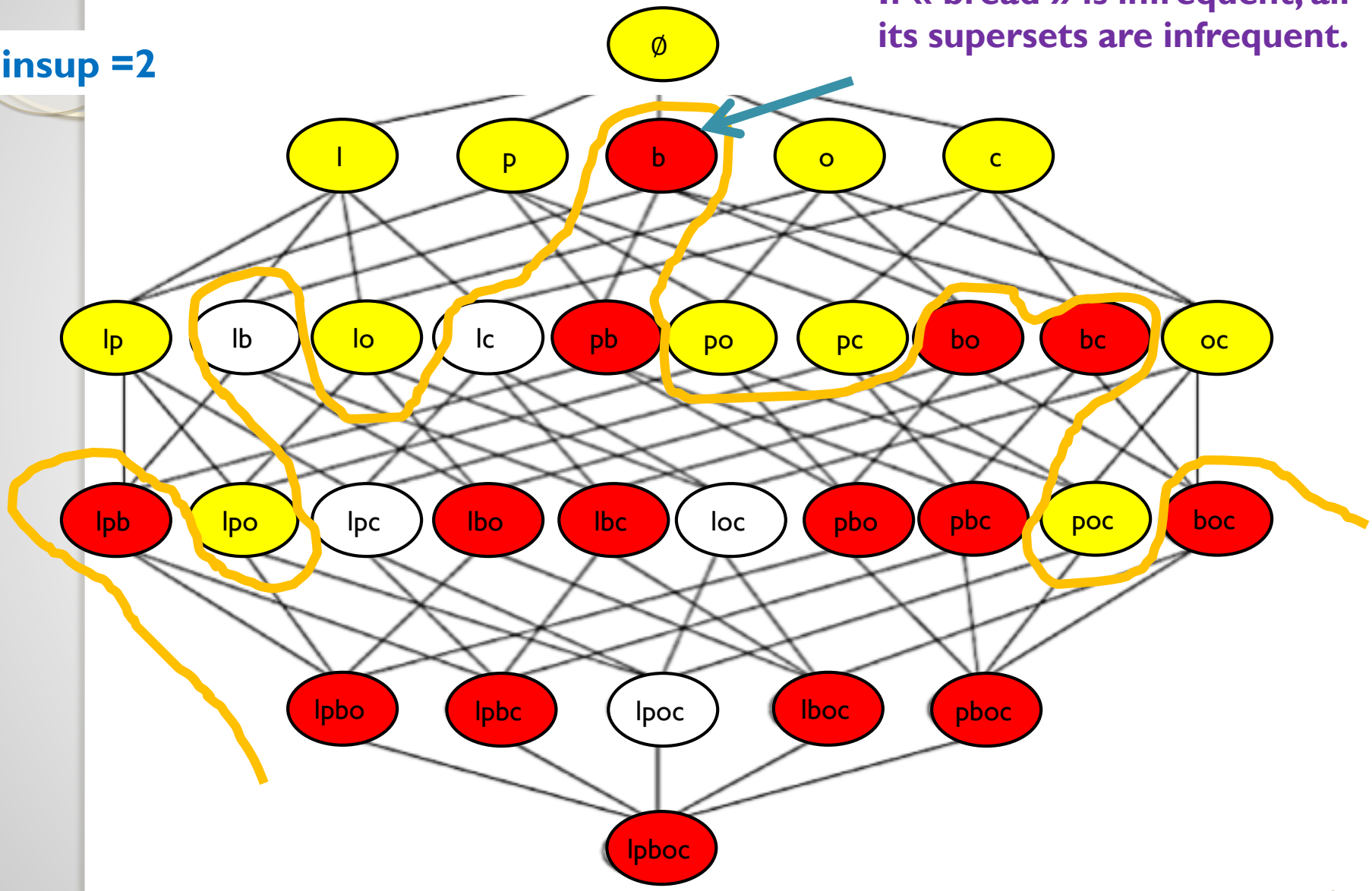


This property is useful to reduce the search space.

Example:

minsup =2

If « bread » is infrequent, all its supersets are infrequent.



ASSOCIATION RULE MINING
(关联规则挖掘)

Introduction

- Finding frequent patterns in a database allows to find useful information.
- But it has some limitations→

Introduction

A **transactional database *D***

Transaction	Items in the transaction
T1	{pasta, lemon, bread, orange}
T2	{pasta, lemon}
T3	{pasta, orange, cake}
T4	{pasta, lemon, orange, cake}

If minsup = 2, then {pasta, cake} is frequent.

Can we conclude that people who buy pasta will also buy cakes?

Association rule

An **association rule** is a rule of the form $X \rightarrow Y$ where

- X *and* Y are itemsets,
- and $X \cap Y = \emptyset$.

e.g. {orange, cake} → {pasta}
{lemon,orange} → {pasta}
{pasta} → {bread}
…

Support

The **support of a rule** $X \rightarrow Y$ is calculated as $sup(X \rightarrow Y) = sup(XUY)/ |D|$
where |D| is the number of transactions.

Transaction	Items in the transaction
T1	{**pasta, lemon**, bread, **orange**}
T2	{pasta, lemon}
T3	{pasta, orange, cake}
T4	{**pasta, lemon**, **orange**, cake}

e.g. {lemon, orange} → {pasta}
has a support of 0.5
i.e. two out of four transactions.

Confidence

The **confidence of a rule** $X \rightarrow Y$ is calculated as $conf(X \rightarrow Y) = sup(XUY)/\ sup(X)|$.

Transaction	Items in the transaction
T1	{**pasta**, **lemon**, bread, **orange**}
T2	{pasta, lemon}
T3	{pasta, orange, cake}
T4	{**pasta**, **lemon**, **orange**, cake}

{lemon, orange} → {pasta} has a confidence of 1.0 (100%)

Confidence

The **confidence of a rule** $X \rightarrow Y$ is calculated as $conf(X \rightarrow Y) = sup(XUY)/\ sup(X)|$.

Transaction	Items in the transaction
T1	{**pasta, lemon**, bread, orange}
T2	{**pasta, lemon**}
T3	{**pasta**, orange, cake}
T4	{**pasta, lemon**, orange, cake}

{pasta} → {lemon} has a confidence of 0.75
{lemon} → {pasta} has a confidence of 1.0

Association rule mining

Input:

- A transaction database (set of transactions)
- A parameter *minsup* $(0 \geq minsup \geq 1)$
- A parameter *minconf* $(0 \geq minconf \geq 1)$

Output: each association rule $X \rightarrow Y$ such that:

- $\text{sup}(X \rightarrow Y) \geq minsup$ and
- $\text{conf}(X \rightarrow Y) \geq minconf$

{pasta} ⟶ {lemon} has a confidence of 0.75
{lemon} ⟶ {pasta} has a confidence of 1.0

Example

minsup = 0.5 minconf = 0.75

Transaction	Items in the transaction
T1	{**pasta**, **lemon**, bread, **orange**}
T2	{pasta, lemon}
T3	{pasta, orange, cake}
T4	{**pasta**, **lemon**, **orange**, cake}

- lemon → pasta support: 3 confidence: 1
- pasta → lemon support: 3 confidence: 0,75
- orange → pasta support: 3 confidence: 1
- pasta → orange support: 3 confidence: 0,75
- cake → pasta support: 2 confidence: 1
- cake → orange support: 2 confidence: 1
- lemon orange → pasta support: 2 confidence: 1
- orange cake → pasta support: 2 confidence: 1
- pasta cake → orange support: 2 confidence: 1
- cake → pasta orange support: 2 confidence: 1

Why using the support and confidence?

- The **support** allows to:
  - find patterns that are less likely to be random.
  - reduce the number of patterns,
  - make the algorithms more efficient.
- The **confidence** allows to:
  - measure the strength of associations
  - obtain an estimation of the conditional probability P(**Y** | X **)**.
  - **Warning**: a strong association does not mean that there is causality!

How to find the association rules? Naïve approach

1. Create **all** association rules.
2. Calculate their confidence and support by scanning the database.
3. Keep only the valid rules.

This approach is inefficient. For d items, there are:

possible rules.

$$\sum_{k=1}^{d-1}\left[\binom{d}{k}\times\sum_{j=1}^{d-k}\binom{d-k}{j}\right]=3^d-2^d+1$$

For d =6, this means 602 rules!
For d =100, this means 10^{47} rules!

Observation 1

Transaction	Items in the transaction
T1	{**pasta**, **lemon**, bread, **orange**}
T2	{pasta, lemon}
T3	{pasta, orange, cake}
T4	{**pasta**, **lemon**, **orange**, cake}

- lemon → pasta support: 3 confidence: 1
- pasta → lemon support: 3 confidence: 0,75
- orange → pasta support: 3 confidence: 1
- pasta → orange support: 3 confidence: 0,75

Observation 1. All the rules containing the same items can be viewed as having been derived from a same frequent itemset.
e.g. {pasta, lemon}

Observation 2

Transaction	Items in the transaction
T1	{**pasta**, **lemon**, bread, **orange**}
T2	{pasta, lemon}
T3	{pasta, orange, cake}
T4	{**pasta**, **lemon**, **orange**, cake}

- lemon → pasta support: **3** confidence: **1**
- pasta → lemon support: **3** confidence: **0,75**
- orange → pasta support: **3** confidence: **1**
- pasta → orange support: **3** confidence: **0,75**

Observation 2. All the rules containing the same items have the same support, but may not have the same confidence.
e.g. {pasta, lemon}

Observation 3

Transaction	Items in the transaction
T1	{**pasta**, **lemon**, bread, **orange**}
T2	{pasta, lemon}
T3	{pasta, orange, cake}
T4	{**pasta**, **lemon**, **orange**, cake}

- lemon → pasta support: 3 confidence: 1
- pasta → lemon support: 3 confidence: 0,75
- orange → pasta support: 3 confidence: 1
- pasta → orange support: 3 confidence: 0,75

Observation 3. If an itemset is infrequent, all rules derived from that itemset can be ignored. **e.g.** If minsup = 4…, rules derived from {pasta, lemon} can be ignored, since its support is 3.

How to find association rules eficiently?

Aggrawal & Srikant (1993).

Two steps:

1. Discover the **frequent itemsets.**
2. Use the frequent itemsets to generate **association rules** having a confidence greater or equal to *minconf*.

Step 1 is the most difficult.

Thus, most studies are on improving the efficiency of Step 1.

Generating rules

- Each **frequent itemset** Y of size k can produce 2^k-2 rules.
- A rule can be created by dividing an itemset Y in two non empty subsets to obtain a rule $X \rightarrow Y - X$.
- Then, the confidence of the rule must be calculated.

Generating rules

Example: using the itemset X={a, b, c}, we can generate:

- $\{a, b\} \rightarrow \{c\}$
- $\{a, c\} \rightarrow \{b\}$
- $\{b, c\} \rightarrow \{a\}$
- $\{a\} \rightarrow \{b, c\}$
- $\{b\} \rightarrow \{a, c\}$
- $\{c\} \rightarrow \{a, b\}$

Calculating the confidence

Example: using the itemset X={a, b, c}, we can generate:

- $\{a, b\} \rightarrow \{c\}$
- $\{a, c\} \rightarrow \{b\}$
- $\{b, c\} \rightarrow \{a\}$
- $\{a\} \rightarrow \{b, c\}$
- $\{b\} \rightarrow \{a, c\}$
- $\{c\} \rightarrow \{a, b\}$

How can we calculate the confidence of rules derived from X?

- We must know the support of all subsets of **X**.
- We know it already, since if **X** is a frequent itemset, then all its subsets are frequent!

Calculating the confidence

The result of a frequent itemset mining program looks like this:

{pasta}	support = 4
{lemon}	support = 3
{orange}	support = 3
{cake}	support = 2
{pasta, lemon}	support: 3
{pasta, orange}	support: 3
{pasta, cake}	support: 2
{lemon, orange}	support: 2
{orange, cake}	support: 2
{pasta, lemon, orange}	support: 2
{pasta, orange, cake}	support: 2

How can we quickly search for the support of a set?

Calculating the confidence

Solution 1:

- Itemsets are grouped by size,
- Itemsets having the same size are sorted by some total order $\succ$.
- binary search…

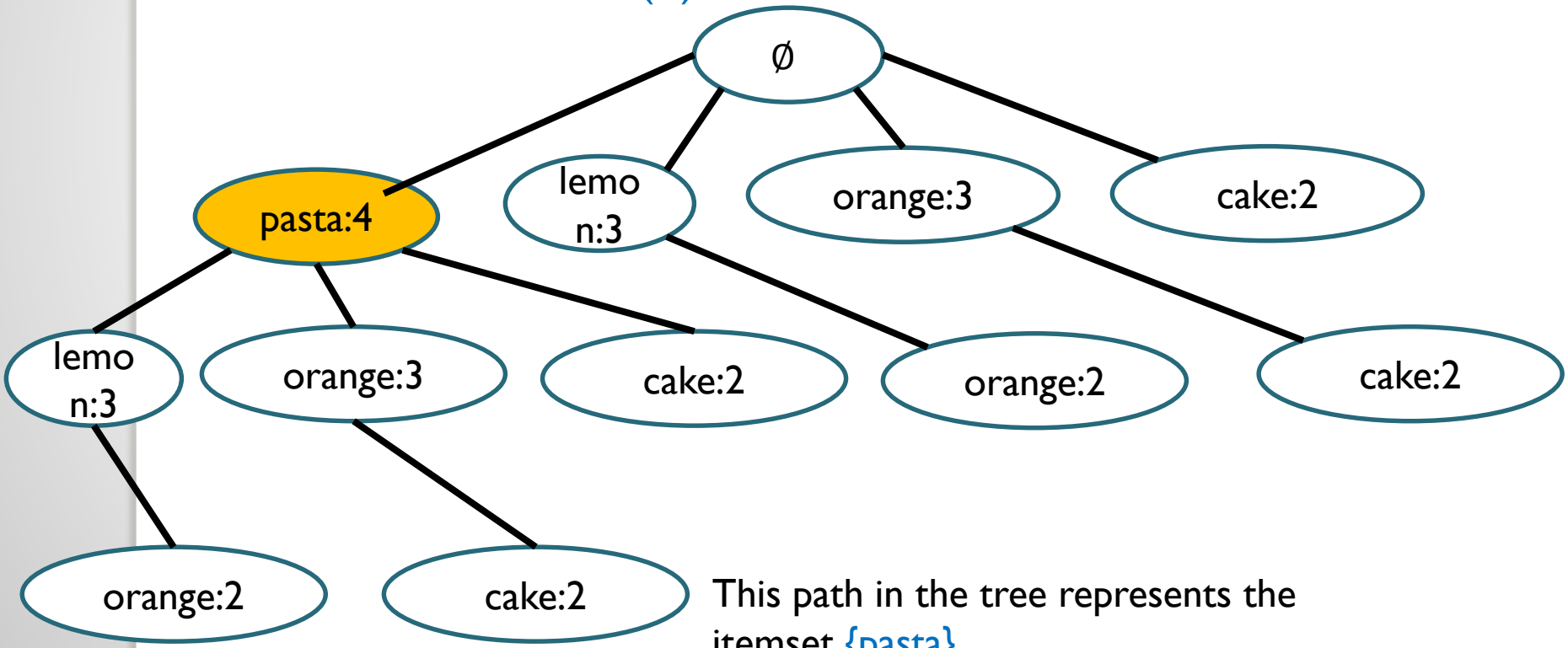
{pasta, lemon}	support: 3
{pasta, orange}	support: 3
{pasta, cake}	support: 2
{lemon, orange}	support: 2
{orange, cake}	support: 2

pasta $\succ$ lemon $\succ$ bread $\succ$ orange $\succ$ cake

Calculating the confidence

Solution 2:

- itemsets are stored in a « trie » to search for itemsets in O(1) time

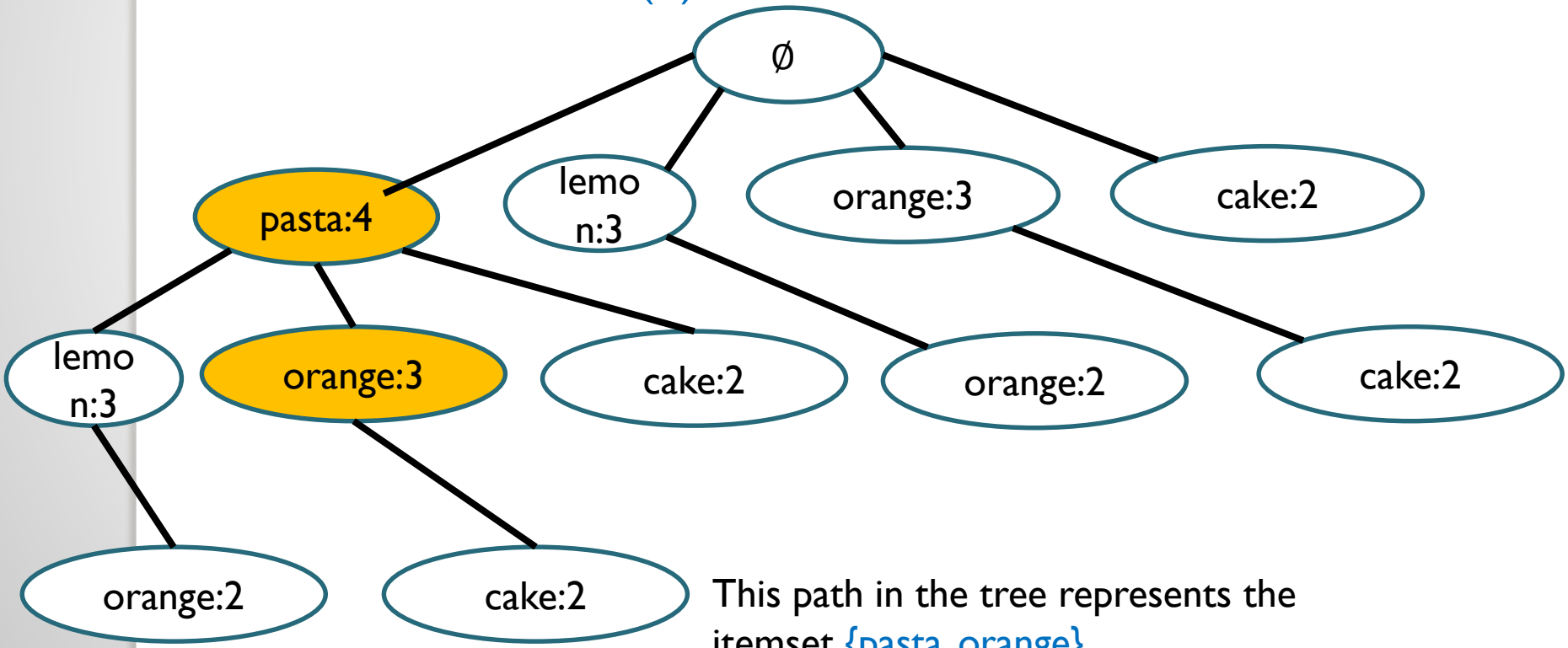


This path in the tree represents the itemset {pasta}
which has a support of 4

Calculating the confidence

Solution 2:

- itemsets are stored in a « trie » to search for itemsets in O(1) time

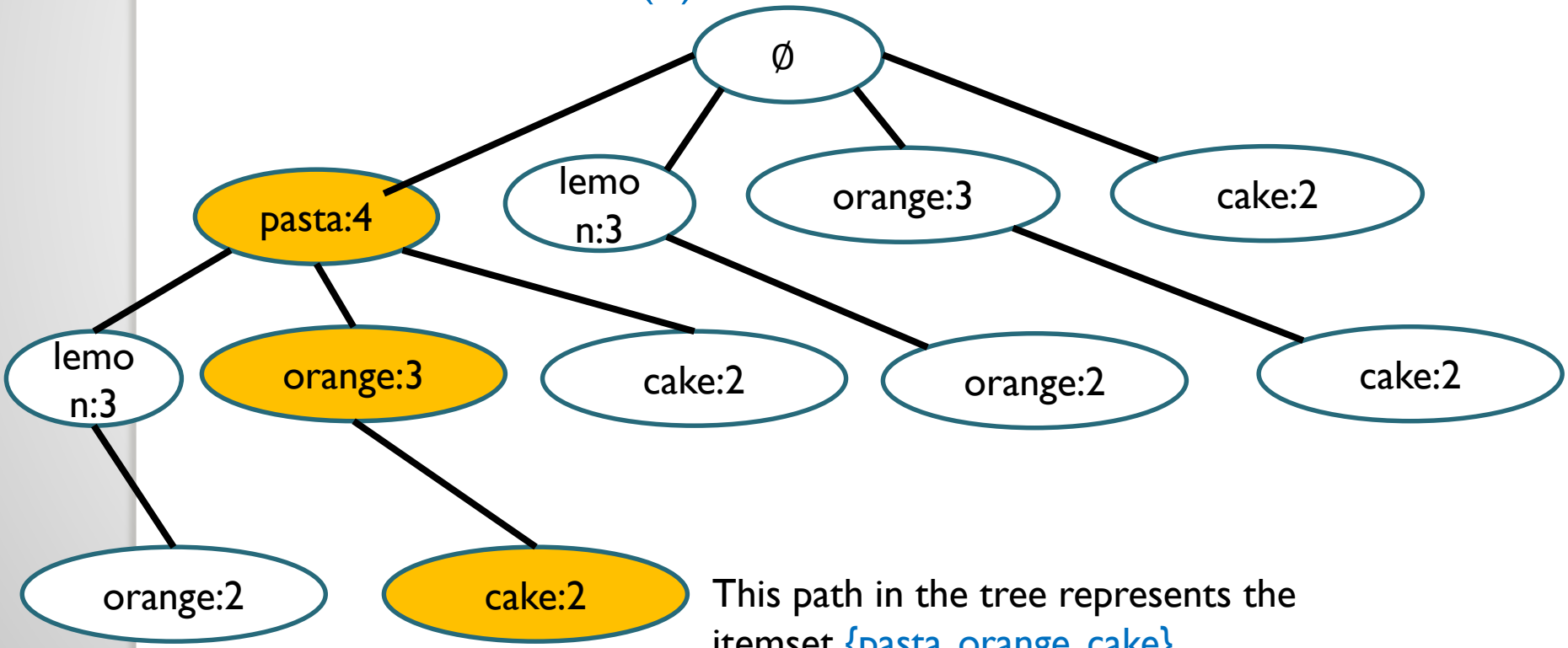


This path in the tree represents the itemset {pasta, orange} which has a support of 3

Calculating the confidence

Solution 2:

- itemsets are stored in a « trie » to search for itemsets in O(1) time



This path in the tree represents the itemset {pasta, orange, cake} which has a support of 2

Reducing the search space

Can we reduce the search space using the *confidence* measure?

- Confidence is not an anti-monotone measure.
- However, the following relationship between two rules can be proved:

Theorem: If a rule $X \rightarrow Y - X$ does not satisfy the confidence threshold, any rule $X' \rightarrow Y - X'$ such that $X' \subseteq X$ will also not satisfy the confidence threshold.

Proof

Let there be two rules
$X \rightarrow Y - X$ and
$X' \rightarrow Y - X'$ such that $X' \subseteq X$.

The confidence of these rules are

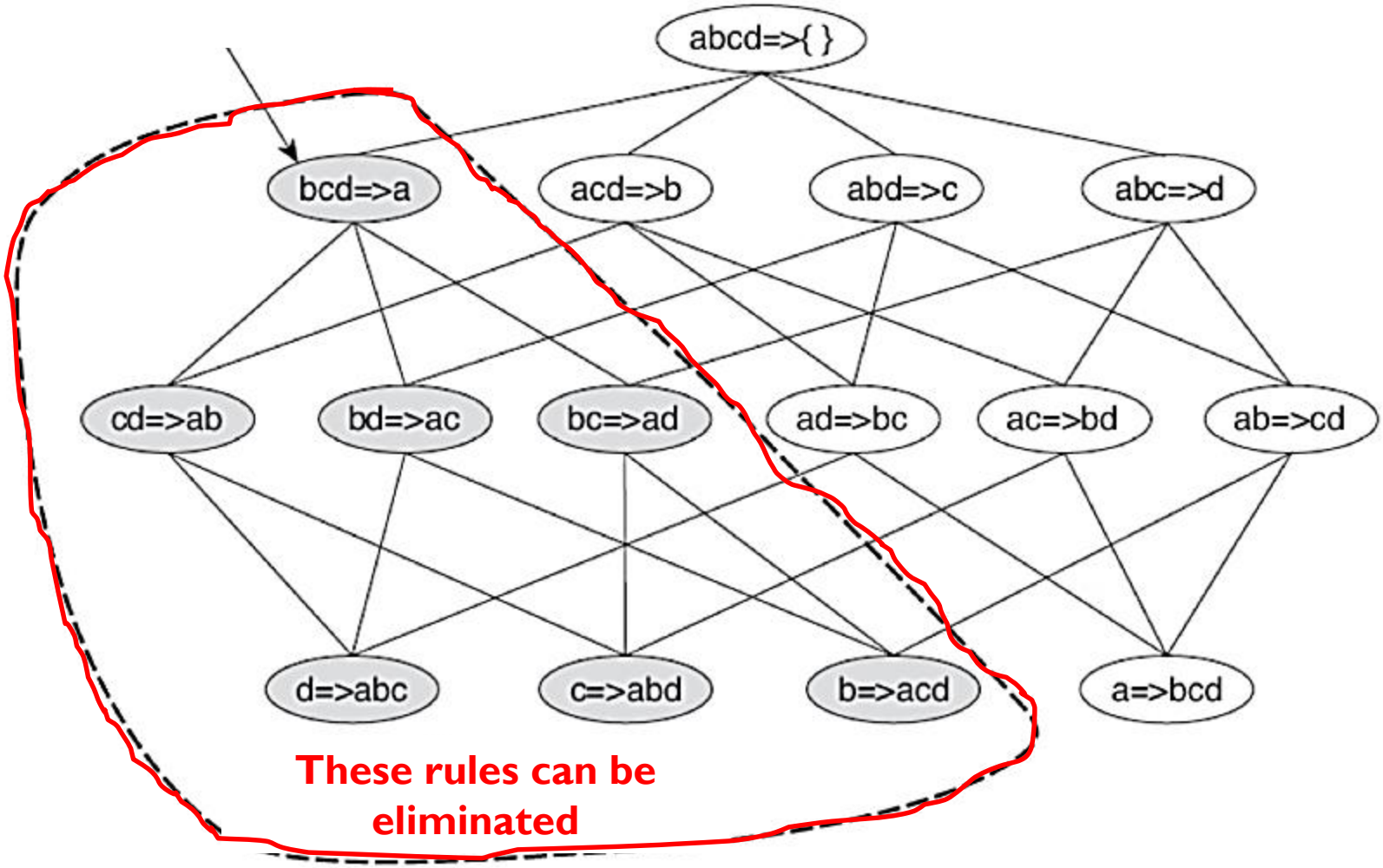
- $\text{conf}(X \rightarrow Y - X) = \dfrac{\sup(XUY)}{\sup(X)}$
- $\text{conf}(X' \rightarrow Y - X') = \dfrac{\sup(XUY)}{\sup(X')}$

Since $X' \subseteq X$, it follows that $\sup(X') \geq \sup(X)$.

Thus: $\text{conf}(X' \rightarrow Y - X') \leq \text{conf}(X \rightarrow Y - X)$
and the theorem holds.

Illustration

Low-confidence rule



Generating rules

Algorithm 8.6: Algorithm AssociationRules

```
AssociationRules (F, minconf):
1  foreach Z ∈ F, such that |Z| ≥ 2 do
2      A ← {X | X ⊂ Z, X ≠ ∅}
3      while A ≠ ∅ do
4          X ← le plus grand itemset dans A
5          A ← A \ X//  enlever X de A
6          c ← sup(Z)/sup(X)
7          if c ≥ minconf then
8              print X ⟶ Y, sup(Z), c
9          else
10             A ← A \ {W | W ⊂ X} // remove all subsets of X from A
```

where *F* is the set of all frequent itemsets

A is the set of all proper non empty subsets of *F*

EVALUATING ASSOCIATIONS

Evaluating associations

- A large amount of patterns can be discovered
- **How to find the most interesting patterns?**
- Interestingness measures:
  - **objective measures**: statistical reasons for selecting patterns
  - **subjectives**: discover surprising or interesting patterns (**e.g.** *diaper → beer* is more surprising than *mouse →keyboard*).
- It is more difficult to consider subjective measures in the search for patterns.

Objective measure

- Independent from any domain
- **e.g.** *support* and *confidence*
- Several objective measures can be calculated using a contingency table.
  e.g. a table with 2 binary attributes

	B	$\bar{B}$	
A	f_{11}	f_{10}	f_{1+}
$\bar{A}$	f_{01}	f_{00}	f_{0+}
	f_{+1}	f_{+0}	N

Limitations of the support and confidence

If we use the *minsup* threshold,

- we will find less results,
- it will be faster,
- but we may eliminate some rare patterns that are interesting.

Another problem

- Consider: {tea} → {coffee}
  support : 15% confidence : 75 %
- This seems like an interesting pattern…

	$Coffee$	$\overline{Coffee}$	
Tea	150	50	200
$\overline{Tea}$	650	150	800
	800	200	1000

$$conf(X \rightarrow Y) = \frac{\sup(X \cup Y)}{\sup(X)}$$

- However, 80 % of the people drink coffee no matter if they drink tea or not.
- In fact, the probability of drinking coffee for tea drinkers is lower than for non tea drinkers (80 instead of 75 %)!
- This problem occurs because the confidence measure does not consider the support of the left side of rules.

The *lift*

- $lift(X \rightarrow Y) = \frac{conf(X \cup Y)}{sup(Y)} = \frac{sup(X \cup Y)}{sup(X) \times sup(Y)}$
- if = 1, X and Y are independent
- if > 1, X and Y are positively correlated
- if < 1, X and Y are negatively correlated

Example:

lift({tea}→{coffee}) = 0.9375,

This indicates a slightly negative correlation

Limitations of the lift

Example:

	p	$\bar{p}$	
q	880	50	930
$\bar{q}$	50	20	70
	930	70	1000

	r	$\bar{r}$	
s	20	50	70
$\bar{s}$	50	880	930
	70	930	1000

$lift(\{p\} \rightarrow \{q\})$ =1.02 even if they appear together in 88 % of the transactions

$lift(\{r\} \rightarrow \{s\})$ = 4.08 even if they rarely appear together.

In this case, using the confidence provides better results:

$conf(\{p\} \rightarrow \{q\})$ = 94.6 % $conf(\{r\} \rightarrow \{s\})$ = 28.6 %

Many other measures…

Tan and al. (2004)

#	Measure	Definition
1	ϕ-coefficient	$\frac{P(A,B)-P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
2	Goodman-Kruskal's (λ)	$\frac{\sum_j \max_k P(A_j,B_k)+\sum_k \max_j P(A_j,B_k)-\max_j P(A_j)-max_k P(B_k)}{2-\max_j P(A_j)-\max_k P(B_k)}$
3	Odds ratio (α)	$\frac{P(A,B)P(\overline{A},\overline{B})}{P(A,\overline{B})P(\overline{A},B)}$
4	Yule's Q	$\frac{P(A,B)P(\overline{AB})-P(A,\overline{B})P(\overline{A},B)}{P(A,B)P(\overline{AB})+P(A,\overline{B})P(\overline{A},B)} = \frac{\alpha-1}{\alpha+1}$
5	Yule's Y	$\frac{\sqrt{P(A,B)P(\overline{AB})}-\sqrt{P(A,\overline{B})P(\overline{A},B)}}{\sqrt{P(A,B)P(\overline{AB})}+\sqrt{P(A,\overline{B})P(\overline{A},B)}} = \frac{\sqrt{\alpha}-1}{\sqrt{\alpha}+1}$
6	Kappa (κ)	$\frac{P(A,B)+P(\overline{A},\overline{B})-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A)P(B)-P(\overline{A})P(\overline{B})}$
7	Mutual Information (M)	$\frac{\sum_i \sum_j P(A_i,B_j)\log\frac{P(A_i,B_j)}{P(A_i)P(B_j)}}{\min(-\sum_i P(A_i)\log P(A_i), -\sum_j P(B_j)\log P(B_j))}$
8	J-Measure (J)	$\max\big(P(A,B)\log(\frac{P(B\|A)}{P(B)}) + P(A\overline{B})\log(\frac{P(\overline{B}\|A)}{P(\overline{B})}),$ $P(A,B)\log(\frac{P(A\|B)}{P(A)}) + P(\overline{A}B)\log(\frac{P(\overline{A}\|B)}{P(\overline{A})})\big)$
9	Gini index (G)	$\max\big(P(A)[P(B\|A)^2+P(\overline{B}\|A)^2]+P(\overline{A})[P(B\|\overline{A})^2+P(\overline{B}\|\overline{A})^2] - P(B)^2-P(\overline{B})^2,$ $P(B)[P(A\|B)^2+P(\overline{A}\|B)^2]+P(\overline{B})[P(A\|\overline{B})^2+P(\overline{A}\|\overline{B})^2] - P(A)^2-P(\overline{A})^2\big)$
10	Support (s)	$P(A,B)$
11	Confidence (c)	$\max(P(B\|A), P(A\|B))$
12	Laplace (L)	$\max\big(\frac{NP(A,B)+1}{NP(A)+2}, \frac{NP(A,B)+1}{NP(\overline{B})+2}\big)$
13	Conviction (V)	$\max\big(\frac{P(A)P(\overline{B})}{P(A\overline{B})}, \frac{P(B)P(\overline{A})}{P(B\overline{A})}\big)$
14	Interest (I)	$\frac{P(A,B)}{P(A)P(B)}$
15	cosine (IS)	$\frac{P(A,B)}{\sqrt{P(A)P(B)}}$
16	Piatetsky-Shapiro's (PS)	$P(A,B)-P(A)P(B)$
17	Certainty factor (F)	$\max\big(\frac{P(B\|A)-P(B)}{1-P(B)}, \frac{P(A\|B)-P(A)}{1-P(A)}\big)$
18	Added Value (AV)	$\max(P(B\|A)-P(B), P(A\|B)-P(A))$
19	Collective strength (S)	$\frac{P(A,B)+P(\overline{AB})}{P(A)P(B)+P(\overline{A})P(\overline{B})} \times \frac{1-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A,B)-P(\overline{AB})}$
20	Jaccard (ζ)	$\frac{P(A,B)}{P(A)+P(B)-P(A,B)}$
21	Klosgen (K)	$\sqrt{P(A,B)}\max(P(B\|A)-P(B), P(A\|B)-P(A))$

Many measures

They have different properties.

e.g. symetrical, anti-monotonic, etc.

References

Some content from these sources:

- Data Mining: The Textbook by Aggarwal (2015)
- Data Mining and Analysis Fundamental Concepts and Algorithms by Zaki & Meira (2014)
- Han and Kamber (2011), Data Mining: Concepts and Techniques, 3rd edition, Morgan Kaufmann Publishers,
- Tan, Steinbach & Kumar (2006), Introduction to Data Mining, Pearson education, ISBN-10: 0321321367.