

6. Mining Frequent Patterns, Associations and Correlations

Knowledge Discovery in Databases

Dominik Probst, Dominik.probst@fau.de

Chair of Computer Science 6 (Data Management), Friedrich-Alexander-University Erlangen-Nürnberg

Summer semester 2024

Outline

1. **Basic Concepts**
2. **Scalable frequent-itemset-mining methods**
3. **Generating association rules from frequent itemsets**
4. **Which patterns are interesting? Pattern-evaluation methods**
5. **Summary**

Basic Concepts

What is Frequent-pattern Analysis?

- **Frequent pattern:**
 - A pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a dataset.
- **Motivation: Finding inherent regularities in data:**
 - What products are often purchased together? Beer and diapers?!
 - What are the subsequent purchases after buying a PC?
 - Who bought this has often also bought . . ."
 - What kinds of DNA are sensitive to this new drug?
 - Can we automatically classify Web documents?
- **Applications:**
 - Basket-data analysis, cross-marketing, catalog design, sale-campaign analysis, Web-log (click-stream) analysis, and DNA-sequence analysis.

Why is Frequent-pattern Mining Important?

- **A frequent pattern is an intrinsic and important property of a dataset.**
- **Foundation for many essential data-mining tasks:**
 - Association, correlation, and causality analysis.
 - Sequential, structural (e.g., sub-graph) patterns.
 - Pattern analysis in spatiotemporal, multimedia, time-series, and stream data.
 - Classification: discriminative, frequent-pattern analysis.
 - Cluster analysis: frequent-pattern-based clustering.
 - Data warehousing: iceberg cube and cube gradient.
 - Semantic data compression: fascicles (Jagadish, Madar, and Ng, VLDB'99).
 - Broad applications.

Some Real World Examples

Frequently bought together




+

Total price: \$78.13


Add both to Cart

This item: Bounty Quick-Size Paper Towels, White, 16 Family Rolls = 40 Regular Rolls...
\$43¹⁵ (\$2.31/100 Sheets)


Charmin Ultra Soft Cushiony Touch Toilet Paper, 24 Family Mega Rolls = 123 Regular Rolls
\$34⁹⁸ (\$0.47/100 Sheets)

Some Real World Examples

Frequently bought together








This item: Bounty Quick-Size Paper Towels, White, 16 Family Rolls = 40 Regular Rolls...
\$43¹⁵ (\$2.31/100 Sheets)



Charmin Ultra Soft Toilet Paper, Mega Roll...
\$34⁹⁸ (\$0.34/Sheet)


Total price: \$78.13

Frequently Bought Together (20 items)

 <p>The Legend of Zelda: Tears of the Kingdom - Nintendo Switch... ★★★★★ (1,717) \$69.99</p> <p>Add to Cart</p>	 <p>PowerA - Controller Charging Base for Nintendo Switch (Joy-... ★★★★★ (12) \$29.99</p> <p>Add to Cart</p>	 <p>The Legend of Zelda: Breath of the Wild - Nintendo Switch ★★★★★ (20,144) \$55.99 \$59.99</p> <p>Add to Cart</p>	 <p>PowerA - Joy-Con Comfort Grip for Nintendo Switch - Black ★★★★★ (1,016) \$14.99</p> <p>Add to Cart</p>	 <p>Nintendo Wii U Console ★★★ \$349.99</p> <p>Add to Cart</p>
--	---	--	---	---

Some Real World Examples

Frequently bought together




This item: Bounty Quick-Size Paper Towels, White, 16 Family Rolls = 40 Regular Rolls...

\$43¹⁵ (\$2.31/100 Sheets)

Frequently Bought Together (20 items)

Total price: \$78.13




The Legend of Zelda: Tears of the Kingdom - Nintendo Switch...

★★★★★ (1,717)

\$69.99

[Add to Cart](#)




PowerA - Controller Charging Base for Nintendo Switch (Joy-...)

★★★★★ (12)

\$29.99

[Add to Cart](#)

KUNDEN KAUFEN AUCH:



Lagernd


CUSTOMERS' CHOICE

be quiet! Dark Rock Pro 4 Tower Kühler

€ 81,60*

Inkl. 19% USt + Versandkosten

über 70.380 verkauft



Lagernd


CUSTOMERS' CHOICE

1TB Samsung 980 Pro M.2 PCIe 4.0 3D NAND TLC (MZ-V8P1T0BW)

€ 84,40*

Inkl. 19% USt + Versandkosten

über 37.790 verkauft



Lagernd


Artnr: 74929

2TB Samsung 980 Pro M.2 PCIe 4.0 x4 3D NAND TLC (MZ-V8P2T0BW)

€ 140,50*

Inkl. 19% USt + Versandkosten

über 12.700 verkauft



Lagernd

CUSTOMERS' CHOICE

Gigabyte X670E Aorus Master AMD X670E SoAM5 Dual Channel DDR

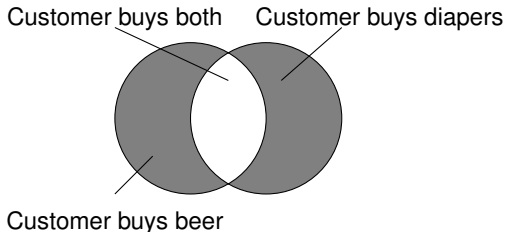
€ 464,51*

Inkl. 19% USt + Versandkosten

über 640 verkauft

An Theoretical Example (I)

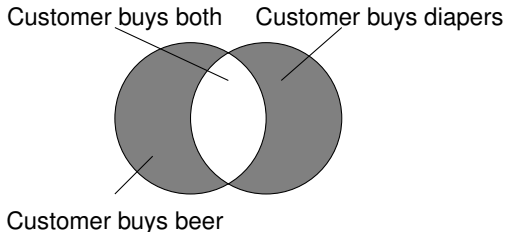
TID	Items bought
10	Beer, Nuts, Diapers
20	Beer, Coffee, Diapers
30	Beer, Diapers, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diapers, Eggs, Milk



- **Itemset:**
 - A set of one or more items.
 - k -itemset $X = \{x_1, x_2, \dots, x_k\}$.
- **(Absolute) Support, or support count of X :**
 - Frequency or occurrence of X .
- **(Relative) Support s :**
 - The fraction of the transactions that contain X .
 - I.e. the **probability** that a transaction contains X .
- **An itemset X is frequent, if X 's support is no less than a `min_sup` threshold.**

An Theoretical Example (II)

TID	Items bought
10	Beer, Nuts, Diapers
20	Beer, Coffee, Diapers
30	Beer, Diapers, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diapers, Eggs, Milk



- Find all the rules $X \implies Y$ with minimum support and confidence.
 - **Support** s : probability that a transaction contains $X \cup Y$.
 - **Confidence** c : conditional probability that a transaction having X also contains Y .
- **Example:**
 - $\text{min_sup} = 50\%$ and $\text{min_conf} = 50\%$.
 - Frequent itemsets:
 - Beer: 3, Nuts: 3, Diapers: 4, Eggs: 3, $\{\text{Beer, Diapers}\}$: 3.
 - **Association rules:**
 - Beer \implies Diapers (60%, 100%).
 - Diapers \implies Beer (60%, 75%).

Basic Concepts: Association Rules

- **Implication of the form $A \implies B$:**
 - where $A \neq \emptyset$, $B \neq \emptyset$ and $A \cap B = \emptyset$.
- **Strong rule:**
 - Satisfies both min_sup and min_conf

$$\begin{aligned}\text{support}(A \implies B) &= P(A \cup B), \\ \text{confidence}(A \implies B) &= P(B|A) \\ &= \frac{\text{support}(A \cup B)}{\text{support}(A)}.\end{aligned}$$

- I.e. confidence of rule can be easily derived from the support counts of A and $A \cup B$.
- **Association-rule mining:**
 - Find all frequent itemsets.
 - Generate strong association rules from the frequent itemsets.

Closed Itemsets and Max-itemsets (I)

- **A long itemset contains a combinatorial number of sub-itemsets.**

- E.g. $\{a_1, a_2, \dots, a_{100}\}$ contains

$$\binom{100}{1} + \binom{100}{2} + \dots + \binom{100}{100} = 2^{100} - 1 \approx 1.27 \cdot 10^{30} \text{ sub-itemsets!}$$

- **Solution:**
 - Mine closed itemsets and max-itemsets instead.
- **An itemset X is closed, if X is frequent and there exists no super-itemset $X \subset Y$ with the same support as X .** (Pasquier et al., ICDT'99)
- **An itemset X is a max-itemset, if X is frequent and there exists no frequent super-itemset $X \subset Y$.** (Bayardo, SIGMOD'98)
- **Closed itemset is a lossless "compression" of frequent itemsets.**
 - Reducing the number of itemsets (and rules).

Closed Itemsets and Max-itemsets (II)

- **Example:**
 - $DB = \{\langle a_1, a_2, \dots, a_{100} \rangle, \langle a_1, a_2, \dots, a_{50} \rangle\}$.
 - I.e. just two transactions.
 - $\min_sup = 1$.
- **What are the closed itemsets?**
 - $\langle a_1, a_2, \dots, a_{100} \rangle : 1$,
 - $\langle a_1, a_2, \dots, a_{50} \rangle : 2$,
 - Number behind the colon: support_count.
- **What are the max-itemsets?**
 - $\langle a_1, a_2, \dots, a_{100} \rangle : 1$.
- **What is the set of all frequent itemsets?**

Scalable frequent-itemset-mining methods

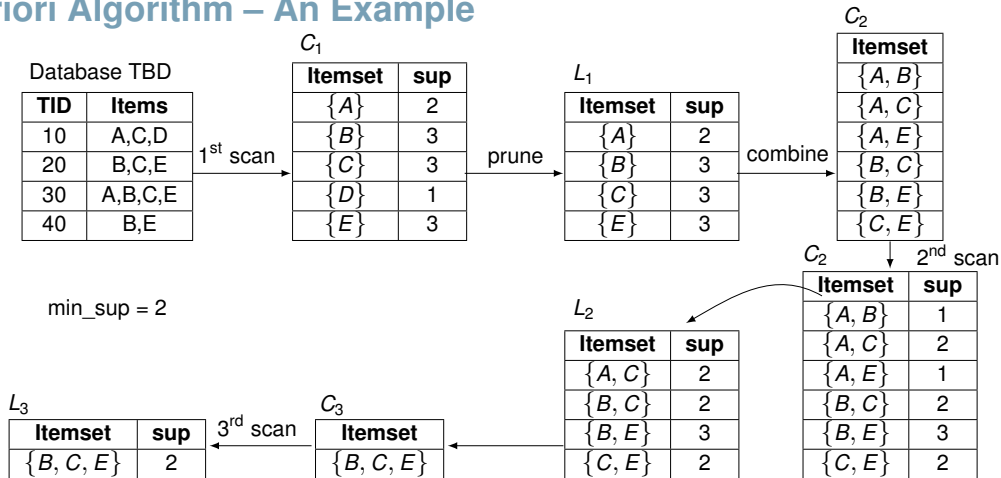
The Downward-closure Property and Scalable Mining Methods

- **The downward-closure property of frequent patterns:**
 - **Any subset of a frequent itemset must also be frequent.**
 - If $\{\text{Beer, Diapers, Nuts}\}$ is frequent, so is $\{\text{Beer, Diapers}\}$.
 - I.e. every transaction having $\{\text{Beer, Diapers, Nuts}\}$ also contains $\{\text{Beer, Diapers}\}$.
- **Scalable mining methods: three major approaches.**
 - Apriori (Agrawal & Srikant, VLDB'94).
 - Frequent-pattern growth (FP-growth) (Han, Pei & Yin, SIGMOD'00).
 - Vertical-data-format approach (CHARM) (Zaki & Hsiao, SDM'02).

Apriori: A Candidate Generation & Test Approach

- **Apriori pruning principle:**
 - If there is any itemset which is infrequent, its supersets should not be generated/tested!
(Agrawal & Srikant, VLDB'94; Mannila et al., KDD'94)
- **Method:**
 - Initially, scan DB once to get frequent 1-itemsets.
 - Generate length- $(k + 1)$ candidate itemsets from length- k frequent itemsets.
 - Test the candidates against DB, discard those that are infrequent.
 - Terminate when no further candidate or frequent itemset can be generated.

Apriori Algorithm – An Example



Apriori Algorithm (Pseudo Code)

C_k : candidate itemsets of size k

L_k : frequent itemsets of size k

$L_1 = \{\text{frequent items}\};$

for ($k = 1; L_k \neq \emptyset; k++$) **do begin**

C_{k+1} = candidates generated from L_k ;

for each transaction t in database **do**

 increment the count of all candidates in C_{k+1} that are contained in t ;

L_{k+1} = candidates in C_{k+1} with min_sup;

end;

return $\bigcup_k L_k$;

Implementation of Apriori (I)

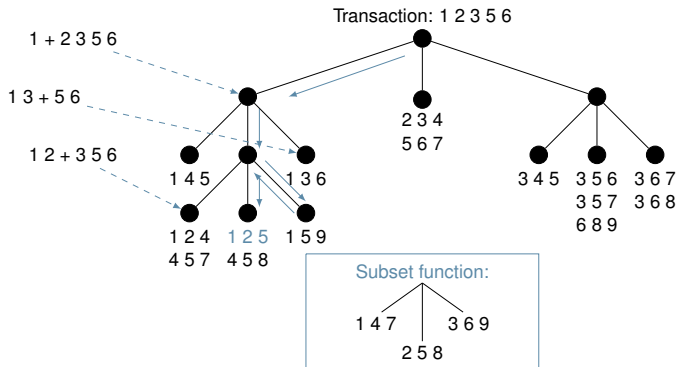
- **How to generate candidates?**
 - Step 1: self-joining L_k (or joining L_k with L_1).
 - Step 2: pruning.
- **Example of candidate generation:**
 - $L_3 = \{abc, abd, acd, ace, bcd\}$.
 - Self-joining: $L_3 \bowtie L_3$:
 - $abcd$ from abc and abd .
 - $acde$ from acd and ace .
 - **Pruning:**
 - $acde$ is removed because ade is not in L_3 .
 - $C_4 = \{abcd\}$.

Implementation of Apriori (II)

- **Why is counting supports of candidates a problem?**
 - The total number of candidates can be huge.
 - One transaction may contain many candidates.
- **Method:**
 - Candidate itemsets are stored in a **hash tree**.
 - Leaf node of hash tree contains a list of itemsets and counts.
 - Interior node contains a hash table.
 - Subset function: finds all the candidates contained in a transaction.

Counting Supports of Candidates using Hash Tree

15 candidate itemsets: 145, 124, 457, 125, 458, 159, 136, 234, 567, 345, 356, 357, 689, 367, 368.



Candidate Generation: An SQL Implementation

- **SQL implementation of candidate generation.**

- Suppose the items in L_{k-1} are listed in order.

1. Self-joining L_{k-1} .

```
INSERT INTO  $C_k$ 
```

```
(SELECT p.item1, p.item2, ..., p.item $k-1$ , q.item $k-1$   
FROM  $L_{k-1}p, L_{k-1}q$   
WHERE p.item1 = q.item1, ..., p.item $k-2$  = q.item $k-2$ ,  
      p.item $k-1$  < q.item $k-1$ );
```

2. Pruning.

```
forall itemsets  $c$  in  $C_k$  do
```

```
    forall  $(k-1)$ -subsets  $s$  of  $c$  do
```

```
        if ( $s$  is not in  $L_{k-1}$ ) then DELETE  $c$  FROM  $C_k$ ;
```

- **Use object-relational extensions like UDFs, BLOBs, and table functions for efficient implementation.** (Sarawagi, Thomas & Agrawal, SIGMOD'98)

Further Improvement of the Apriori Method

- **Major computational challenges.**
 - Multiple scans of transaction database.
 - Huge number of candidates.
 - Support counting for candidates is laborious.
- **Improving Apriori: general ideas.**
 - Reduce passes of transaction-database scans.
 - Shrink number of candidates.
 - Facilitate support counting of candidates.

Hashing: Reduce the Number of Candidates

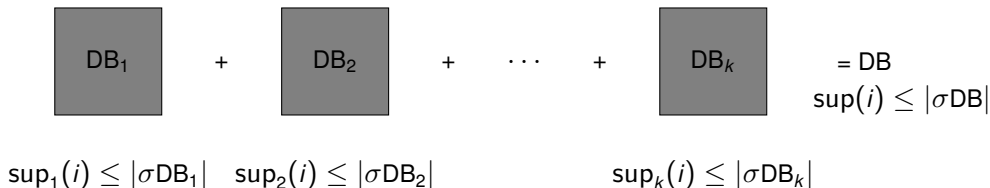
- A k -itemset whose corresponding hashing-bucket count is below the threshold cannot be frequent.
 - Candidates: a, b, c, d, e .
 - While scanning DB for frequent 1-itemsets, create hash entries for 2-itemsets:
 - $\{ab, ad, ae\}$
 - $\{bd, be, de\}$
 - ...
 - Frequent 1-itemset: a, b, d, e .
 - ab is not a candidate 2-itemset, if the sum of count of $\{ab, ad, ae\}$ is below support threshold.
 - (Park, Chen & Yu, SIGMOD'95)

Hash table:

count	itemsets
35	$\{ab, ad, ae\}$
88	$\{bd, be, de\}$
\vdots	\vdots
102	$\{yz, qs, wt\}$

Partition: Scan Database Only Twice

- **Any itemset that is potentially frequent in DB must be frequent in at least one of the partitions of DB.**
 - Scan 1: partition database and find local frequent patterns:
 - $\min_sup_i = \min_sup[\%] \cdot |\sigma DB_i|$.
 - Scan 2: consolidate global frequent patterns.
(Savasere, Omiecinski & Navathe, VLDB'95)



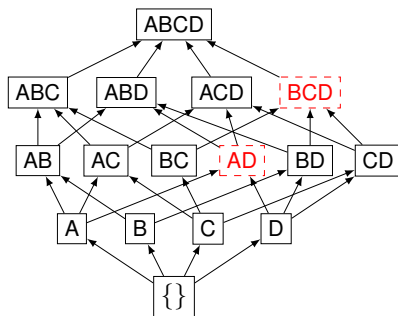
Sampling for Frequent Patterns

- **Select a sample of original database, mine frequent patterns within sample using Apriori.**
- **Scan database once to verify frequent itemsets found in sample, only **borders** of closure of frequent patterns are checked.**
 - Example: check *abcd* instead of *ab*, *ac*, . . . , etc.
- **Scan database again to find missed frequent patterns.**
(Toivonen, VLDB'96)

Dynamic Itemset Counting: Reduce Number of Scans (I)

- **Adding candidate itemsets at different points during a scan.**
 - DB partitioned into blocks marked by **start points**.
 - New candidate itemsets can be added at any start point during a scan.
 - E.g. if A and B are already found to be frequent, AB are also counted from that starting point on.
 - Uses the count-so-far as the lower bound of the actual count.
 - If count-so-far passes minimum support, itemset is added to frequent-itemset collection.
 - Can then be used to generate even longer candidates.

Dynamic Itemset Counting: Reduce Number of Scans (II)

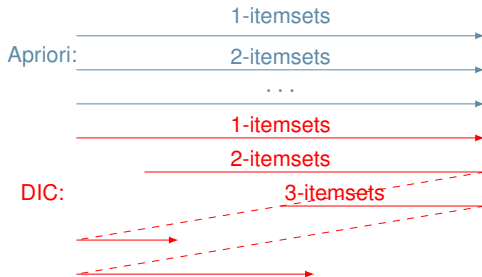


Itemset lattice

(Brin, Motwani, Ullman & Tsur, SIGMOD'97)

- Once both A and D are determined frequent, the counting of AD begins.
- Once length-2 subsets of BCD are determined frequent, the counting of BCD begins.

Transactions



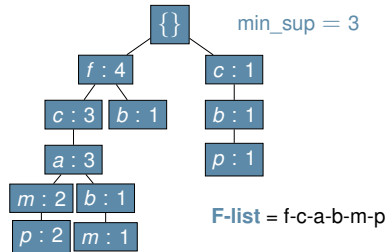
FP-growth: Mining Frequent Patterns without Candidate Generation

- **Bottlenecks of the Apriori approach.**
 - Breadth-first (i.e., level-wise) search.
 - Candidate generation and test.
 - Often generates a huge number of candidates.
- **The FP-growth Approach.** (Han, Pei & Yin, SIGMOD'00)
 - Depth-first search.
 - Avoid explicit candidate generation.
- **Major philosophy: Grow long patterns from short ones using local frequent items only.**
 - abc is a frequent pattern.
 - Get all transactions having abc , i.e. restrict DB on abc : $DB|_{abc}$.
 - d is a local frequent item in $DB|_{abc} \implies abcd$ is a frequent pattern.

Construct FP-tree from a Transaction Database

TID	Items bought	(ordered) frequent items
100	$\{f, a, c, d, g, i, m, p\}$	$\{f, c, a, m, p\}$
200	$\{a, b, c, f, l, m\}$	$\{f, c, a, b, m\}$
300	$\{b, f, h, j, o, w\}$	$\{f, b\}$
400	$\{b, c, k, s, p\}$	$\{c, b, p\}$
500	$\{a, f, c, e, l, p, m, n\}$	$\{f, c, a, m, p\}$

1. Scan DB once, find frequent 1-itemsets (single-item patterns).
2. Sort frequent items in frequency-descending order, creating the **f-list**.
3. Scan DB again, construct **FP-tree**.

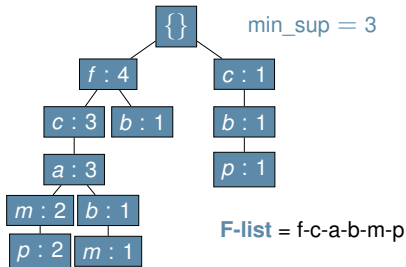


Partition Itemsets and Databases

- **Frequent itemsets can be partitioned into subsets according to f-list.**
 - F-list = f-c-a-b-m-p.
 - Patterns containing p.
 - The least-frequent item (at the end of the f-list, suffix).
 - Patterns having m but not p.
 - ⋮
 - Patterns having c but not a nor b, m, p.
 - Pattern f.
- **This processing order guarantees completeness and non-redundancy.**

Find Itemsets having Item p from p 's Conditional Pattern Base

- Starting at the frequent-item header table in the FP-tree.
- Traverse the FP-tree by following the link of frequent item p .
- Accumulate all transformed **prefix paths** of item p to form p 's **conditional pattern base**.



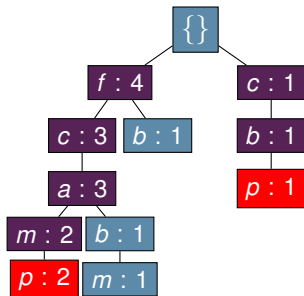
Header table:

item	Frequency
f	4
c	4
a	3
b	3
m	3
p	3

Conditional pattern bases:

item	pattern base
c	f:3
a	fc:3
b	fca:1, f:1, c:1
m	fca:2, fcab:1
p	fcam:2, cb:1

p 's Conditional Pattern Base

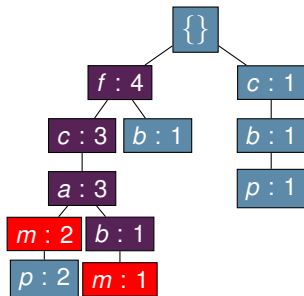


Header table:

item	Frequency
f	4
c	4
a	3
b	3
m	3
p	3

Hence, p 's conditional pattern base is
fcam:2, cb:1
both below min_sup.

m 's Conditional Pattern Base

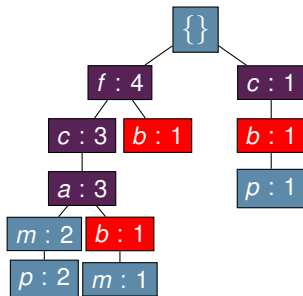


Header table:

item	Frequency
f	4
c	4
a	3
b	3
m	3
p	3

Hence, m 's conditional pattern base is
fca:2, fcab:1
both below min_{sup}.

b 's Conditional Pattern Base

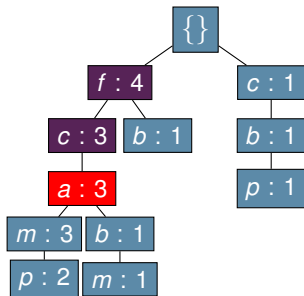


Header table:

item	Frequency
f	4
c	4
a	3
b	3
m	3
p	3

Hence, b 's conditional pattern base is
fca:1, f:1, c:1
all below min_sup.

a 's Conditional Pattern Base



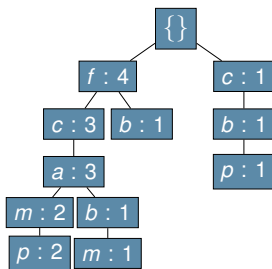
Header table:

item	Frequency
f	4
c	4
a	3
b	3
m	3
p	3

Hence, a 's conditional pattern base is
 $fc:3$
 has min_sup.

From Conditional Pattern Bases to Conditional FP-trees

- For each conditional pattern base:
 - Accumulate the count for each item in the base.
 - Construct the conditional FP-tree for the frequent items of the pattern base.



Header table:

item	Frequency
f	4
c	4
a	3
b	3
m	3
p	3

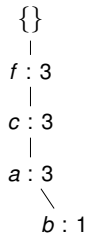
All frequent patterns related to *m*:

m, *fm*, *cm*, *am*, *fcam*, *fam*, *cam*, *fcam*

m's conditional pattern base:

fca:2, *fcab*:1

m's conditional FP-tree:



Recursion: Mining each Conditional FP-tree

***m*'s conditional FP-tree:**



***am*'s conditional pattern base: fc:3**

***am*'s conditional FP-tree:**



***cm*'s conditional FP-tree:**



***cm*'s conditional pattern base: f:3**

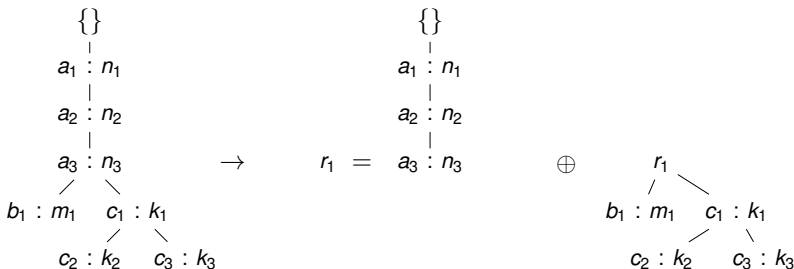
***cam*'s conditional FP-tree:**



***cam*'s conditional pattern base: f:3**

A Special Case: Single Prefix Path in FP-tree (I)

- Suppose a (conditional) FP-tree T has a shared single prefix-path P .
- Mining can be decomposed into two parts.
 - Reduction of the single prefix path into one node.
 - Concatenation of the mining results of the two parts.



A Special Case: Single Prefix Path in FP-tree (II)

- **Completeness.**
 - Preserve complete information for frequent-pattern mining.
 - Never break a long pattern of any transaction.
- **Compactness.**
 - Reduce irrelevant info - infrequent items are removed.
 - Items in frequency-descending order.
 - The more frequently occurring, the more likely to be shared.
 - Never larger than the original database.
 - Not counting node links and the count fields.

The FP-growth Mining Method

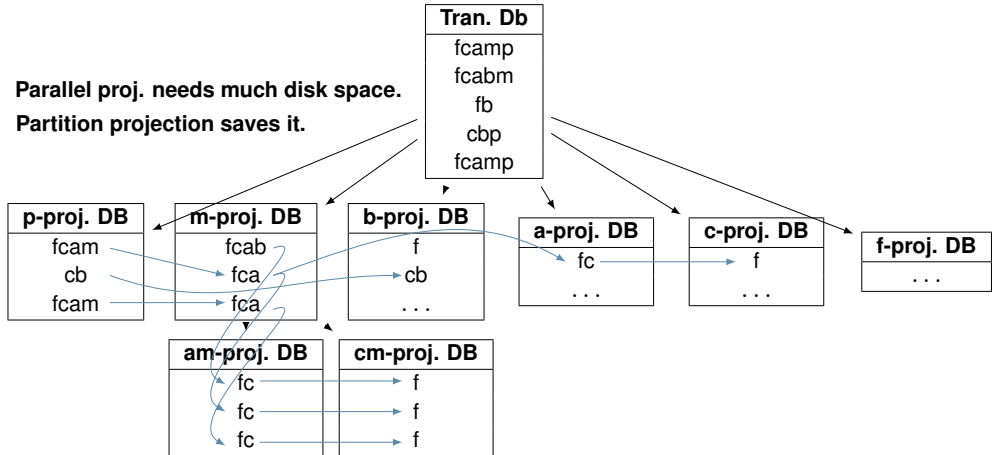
- **Idea: FP-growth.**
 - Recursively grow frequent patterns by pattern and database partition.
- **Method:**
 - For each frequent item, construct its conditional pattern base, and then its conditional FP-tree.
 - Repeat the process on each newly created conditional FP-tree.
 - Until the resulting FP-tree is empty, or it contains only one path.
 - Single path will generate all the combinations of its sub-paths, each of which is a frequent pattern.

Scaling FP-growth by Database Projection

- **What if FP-tree does not fit in memory?**
 - DB projection.
- **First partition database into a set of projected DBs.**
- **Then construct and mine FP-tree for each projected DB.**
- **Parallel-projection vs. partition-projection techniques:**
 - **Parallel projection:**
 - Project the DB in parallel for each frequent item.
 - Parallel projection is space costly.
 - All the partitions can be processed in parallel.
 - **Partition projection:**
 - Partition the DB based on the ordered frequent items.
 - Passing the unprocessed parts to the subsequent partitions.

Partition-based Projection

Parallel proj. needs much disk space.
Partition projection saves it.



Advantages of the FP-growth Approach

- **Divide-and-conquer:**
 - Decompose both the mining task and DB according to the frequent patterns obtained so far.
 - This leads to focused search of smaller databases.
- **Other factors:**
 - No candidate generation, no candidate test.
 - Compressed database: FP-tree structure.
 - No repeated scan of entire database.
 - Basic ops: counting local frequent items and building sub FP-tree, no pattern search and matching.

ECLAT: Mining by Exploring Vertical Data Format

- **Vertical format:** $t(AB) = \{T_{11}, T_{25}, \dots\}$
 - Tid-list: list of transaction ids containing an itemset.
- **Deriving frequent itemsets based on vertical intersections.**
 - $t(X) = t(Y)$: X and Y always happen together.
 - $t(X) \implies t(Y)$: transaction having X always has Y .
- **Using diffset to accelerate mining.**
 - Only keep track of differences of tids.
 - $t(X) = \{T_1, T_2, T_3\}, t(XY) = \{T_1, T_3\}$.
 - Diffset $(XY, X) = \{T_2\}$.
- **ECLAT** (Zaki et al., KDD'97)
- **Mining closed itemsets using vertical format: CHARM** (Zaki & Hsiao, SDM'02)

Mining Closed Itemsets: CLOSET (I)

- **F-list: list of all frequent items in support-ascending order.**
 - F-list: d-a-f-e-c.
- **Divide search space.**
 - Itemsets having d.
 - Itemsets having d but not a, etc.
- **Find closed itemsets recursively.**
 - Every transaction having d also has $cfa \implies cfad$ is a closed itemset.
 - (Pei, Han & Mao, DMKD'00)

TID	Items
10	a,c,d,e,f
20	a,b,e
30	c,e,f
40	a,c,d,f
50	c,e,f

Mining Closed Itemsets: CLOSET (II)

- **Itemset merging:**
 - If Y appears in each occurrence of X , then Y is merged with X .
- **Sub-itemset pruning:**
 - If $X \subset Y$ and $\text{sup}(X) = \text{sup}(Y)$, X and all of X 's descendants in the set enumeration tree can be pruned.
- **Item skipping:**
 - If a local frequent item has the same support in several header tables at different levels, one can prune it from the header table at higher levels.
- **Efficient subset checking.**

MaxMiner: Mining Max-itemsets

- **1st scan: find frequent items.**
 - A, B, C, D, E
- **2nd scan: find support for:**
 - AB, AC, AD, AE, **ABCDE**
 - BC, BD, BE, **BCDE**
 - CD, CE, **CDE**, DE
- **Potential max-itemsets: ABCDE, BCDE, CDE.**
- **Since BCDE is a max-itemset, no need to check BCD, BDE, CDE in later scan.** (Bayardo, SIGMOD'98)

TID	Items
10	A,B,C,D,E
20	B,C,D,E
30	A,C,D,F

Generating association rules from frequent itemsets

Generating Association Rules from Frequent Itemsets

- **Once frequent itemsets from transactions in database D found:**
 - Generate strong association rules from them,
Where "strong" = satisfying both minimum support and minimum confidence.

$$\text{confidence}(A \implies B) = P(B|A) = \frac{\text{support}(A \implies B)}{\text{support}(A)}.$$

- **For each frequent itemset l :**
 - Generate all **nonempty subsets** of l .
- **For every s in l :**
 - Output the rule $s \implies (l - s)$, if
 - min_sup is satisfied, because only frequent itemsets used.

Which patterns are interesting? Pattern-evaluation methods

Interestingness Measure: Correlation (Lift) (I)

- **(play) basketball \implies (eat) cereal (40%, 66.7%) misleading:**
 - The overall % of students eating cereal is 75% $>$ 66.7%.
- **basketball \implies no cereal (20%, 33.3%) more accurate:**
 - Although with lower support and confidence.
- **Reason: negative correlation.**
 - Choice of one item decreases likelihood of choosing the other.
- **Measure of dependent/correlated events: lift.**
 - value 1: independence; value $<$ 1: negatively correlated.

Interestingness Measure: Correlation (Lift) (II)

- **Values:**

	basketball	no basketball	sum (row)
cereal	2000	1750	3750
no cereal	1000	250	1250
sum (col.)	3000	2000	5000

- **Computation:**

$$\text{lift}(A, B) = \frac{P(A \cup B)}{P(A)P(B)}$$

$$\text{lift}(B, C) = \frac{2000/5000}{3000/5000 \cdot 3750/5000} = 0.89,$$

$$\text{lift}(B, \neg C) = \frac{1000/5000}{3000/5000 \cdot 1250/5000} = 1.33.$$

Are Lift and χ^2 Good Measures of Correlation? (I)

- Support and confidence are not good to indicate correlation.
- Over 20 interestingness measures have been proposed. (Tan, Kumar & Sritastava, KDD'02)
- Which are good ones?

symbol	name	range	formula
ψ	ψ -coefficient	$[-1, 1]$	$\frac{P(A,B) - P(A)P(B)}{\sqrt{P(A)P(B)(1-P(A))(1-P(B))}}$
Q	Yule's Q	$[-1, 1]$	$\frac{P(A,B)P(\neg A, \neg B) - P(A, \neg B)P(\neg A, B)}{P(A,B)P(\neg A, \neg B) + P(A, \neg B)P(\neg A, B)}$
Y	Yule's Y	$[-1, 1]$	$\frac{\sqrt{P(A,B)P(\neg A, \neg B)} - \sqrt{P(A, \neg B)P(\neg A, B)}}{\sqrt{P(A,B)P(\neg A, \neg B)} + \sqrt{P(A, \neg B)P(\neg A, B)}}$
k	Cohen's k	$[-1, 1]$	$\frac{P(A,B) + P(\neg A, \neg B) - P(A)P(B) - P(\neg A)P(\neg B)}{1 - P(A)P(B) - P(\neg A)P(\neg B)}$
PS	Patetsky-Shapiro's	$[-0.25, 0.25]$	$P(A, B) - P(A)P(B)$
F	Certainty factor	$[-1, 1]$	$\max(\frac{P(B A) - P(B)}{1 - P(B)}, \frac{P(A B) - P(A)}{1 - P(A)})$
AV	Added Value	$[-0.5, 1]$	$\max(P(B A) - P(B), P(A B) - P(A))$
K	Klogsen's Q	$[-0.33, 0.38]$	$\sqrt{P(A, B) \max(P(B A) - P(B), P(A B) - P(A))}$
g	Goodman-kruskal's	$[0, 1]$	$\frac{\sum_i \max_k P(A_i, B_k) + \sum_k \max_j P(A_j, B_k) - \max_j P(A_j) - \max_k P(B_k)}{2 - \max_j P(A_j) - \max_k P(B_k)}$
M	Mutual information	$[0, 1]$	$\frac{\sum_i \sum_j P(A_i, B_j) \log \frac{P(A_i, B_j)}{P(A_i)P(B_j)}}{\min(-\sum_i P(A_i) \log P(A_i) \log P(A_i), -\sum_j P(B_j) \log P(B_j) \log P(B_j))}$

Are Lift and χ^2 Good Measures of Correlation? (II)

symbol	name	range	formula
J	J-Measure	$[0, 1]$	$\max(P(A, B) \log \frac{P(B A)}{P(B)} + P(\neg A, B) \log \frac{P(\neg A, B)}{P(\neg A)},$ $P(A, B) \log \frac{P(B A)}{P(A)} + P(\neg A, B) \log \frac{P(\neg A, B)}{P(\neg B)})$
G	Gini index	$[0, 1]$	$\max(P(A)[P(B A)^2 + P(\neg B A)^2] +$ $P(\neg A)[P(B \neg A)^2 + P(\neg B \neg A)^2]P(B)^2 - P(\neg B)^2,$ $P(B)[P(A B)^2 + P(\neg A B)^2] +$ $P(\neg B)[P(A \neg B)^2 + P(\neg A \neg B)^2] - P(A)^2 - P(\neg A)^2)$
s	Support	$[0, 1]$	$P(A, B)$
c	Confidence	$[0, 1]$	$\max(P(B A), P(A B))$
L	Laplace	$[0, 1]$	$\max(\frac{NP(A, B)+1}{NP(A)+2}, \frac{NP(A, B)+1}{NP(B)+2})$
\cos	Cosine	$[0, 1]$	$\frac{P(A, B)}{\sqrt{P(A)P(B)}}$
γ	coherence(Jaccard)	$[0, 1]$	$\frac{P(A, B)}{P(A)+P(B)-P(A, B)}$
α	all_confidence	$[0, 1]$	$\frac{P(A, B)}{\max(P(A), P(B))}$
o	Odds ratio	$[0, \infty)$	$\frac{P(A, B)P(\neg A, \neg B)}{P(\neg A, B)P(A, \neg B)}$
V	Conviction	$[0.5, \infty)$	$\max(\frac{P(A)P(\neg B)}{P(A, \neg B)}, \frac{P(B)P(\neg A)}{P(B, \neg A)})$
λ	Lift	$[0, \infty)$	$\frac{P(A, B)}{P(A)P(B)}$
S	Collective strength	$[0, \infty)$	$\frac{P(A, B)+P(\neg A, \neg B)}{P(A)P(B)+P(\neg A)P(\neg B)} \cdot \frac{1-P(A)P(B)-P(\neg A)P(\neg B)}{1-P(A, B)-P(\neg A, \neg B)}$
χ^2	χ^2	$[0, \infty)$	$\sum_i \frac{(P(A_i) - E_i)^2}{E_i}$

Null-invariant Measures (I)

- **Null-transaction:**
 - A transaction that does not contain any of the itemsets being examined.
 - Can outweigh the number of individual itemsets.
- **A measure is null-invariant,**
 - if its value is free from the influence of null-transactions.
 - Lift and χ^2 are not null-invariant.

Null-invariant Measures (II)

Symbol	Measure	Range	O1	O2	O3	O3'	O4
φ	φ -coefficient	$[-1, 1]$	Y	N	Y	Y	N
λ	Goodman-Kruskal's	$[0, 1]$	Y	N	N*	Y	N
α	Odds ratio	$[0, \infty)$	Y	Y	Y*	Y	N
Q	Yule's Q	$[-1, 1]$	Y	Y	Y	Y	N
Y	Yule's Y	$[-1, 1]$	Y	Y	Y	Y	N
κ	Cohen's	$[-1, 1]$	Y	N	N	Y	N
M	Mutual information	$[0, 1]$	N**	N	N*	Y	N
J	J -Measure	$[0, 1]$	N**	N	N	N	N
G	Gini index	$[0, 1]$	N**	N	N*	Y	N
s	Support	$[0, 1]$	Y	N	N	N	N
c	Confidence	$[0, 1]$	N**	N	N	N	Y
L	Laplace	$[0, 1]$	N**	N	N	Y	N
V	Conviction	$[0.5, \infty)$	N**	N	N	Y	N
I	Interest	$[0, \infty)$	Y	N	N	N	N
\cos	Cosine	$[0, 1]$	Y	N	N	N	Y
PS	Piatetsky-Shapiro's	$[-0.25, 0.25]$	Y	N	Y	Y	N
F	Certainty factor	$[-1, 1]$	N**	N	N	Y	N
AV	Added value	$[-0.5, 1]$	N**	N	N	N	N
S	Collective strength	$[0, \infty]$	Y	N	Y*	Y	N
θ	Jaccard	$[0, 1]$	Y	N	N	N	Y
K	Klosgen's	$[(\frac{2}{\sqrt{3}} - 1)^{\frac{1}{2}}[2 - \sqrt{3} - \frac{1}{\sqrt{3}}], \frac{2}{3\sqrt{3}}]$	N**	N	N	N	N

O1: Symmetry under variable permutation.

O2: Row and column scaling invariance.

O3: Antisymmetry under row or column permutation.

O3': Inversion invariance

O4: Null invariance.

Y*: Yes if measure is normalized.

N*: Symmetry under row or column permutation.

N:** No unless the measure is symmetrized by taking $\max(M(A, B), M(B, A))$.

Comparison of Interestingness Measures

- Null-(transaction) invariance is crucial for correlation analysis.
- 5 null-invariant measures:

	Milk	No milk	Sum (row)
Coffee	m,c	$\neg m,c$	c
No coffee	m, $\neg c$	$\neg m,\neg c$	$\neg c$
Sum (col)	m	$\neg m$	

Measure	Definition	Range	Null-invariant
Allconf(a, b)	$\frac{\sup(ab)}{\max(\sup(a)\sup(b))}$	[0, 1]	Y
Coherence(a, b)	$\frac{\sup(ab)}{\sup(a) + \sup(b) - \sup(ab)}$	[0, 1]	Y
Cosine(a, b)	$\frac{\sup(ab)}{\sqrt{\sup(a)\sup(b)}}$	[0, 1]	Y
Kulc(a, b)	$\frac{\sup(ab)}{2} \left(\frac{1}{\sup(a)} + \frac{1}{\sup(b)} \right)$	[0, 1]	Y
maxconf(a, b)	$\max\left(\frac{\sup(ab)}{\sup(a)}, \frac{\sup(ab)}{\sup(b)}\right)$	[0, 1]	Y

Data set	mc	$\neg mc$	m $\neg c$	$\neg m\neg c$	AllConf	Coherence	Cosine	Kulc	MaxConf
D1	10,000	1,000	1,000	100,000	0.91	0.83	0.91	0.91	0.91
D2	10,000	1,000	1,000	100	0.91	0.83	0.91	0.91	0.91
D3	100	1,000	1,000	100,000	0.09	0.05	0.09	0.09	0.09
D4	1,000	1,000	1,000	100,000	0.5	0.33	0.5	0.5	0.5
D5	1,000	100	10,000	100,000	0.09	0.09	0.29	0.5	0.91
D6	1,000	10	100,000	100,000	0.01	0.01	0.10	0.5	0.99

Analysis of DBLP Coauthor Relationships

- Recent DB conferences, removing balanced associations, low sup, etc.

ID	Author a	Author b	$\text{sup}(ab)$	$\text{sup}(a)$	$\text{sup}(b)$	Coherence	Cosine	Kulc
1	Hans-Peter Kriegel	Martin Ester	28	146	54	0.163 (2)	0.315 (7)	0.355 (9)
2	Michael Carey	Miron Livny	26	104	58	0.191 (1)	0.335 (4)	0.349 (10)
3	Hans-Peter Kriegel	Joerg Sander	24	146	36	0.152 (3)	0.331 (5)	0.416 (8)
4	Christos Faloutsos	Spiros Papadimitriou	20	162	26	0.119 (7)	0.308 (10)	0.446 (7)
5	Hans-Peter Kriegel	Martin Pfeifle	18	146	18	0.123 (6)	0.351 (2)	0.562 (2)
6	Hector Garcia-Molina	Wilburt Labio	16	144	18	0.110 (9)	0.314 (8)	0.500 (4)
7	Divyakant Agrawal	Wang Hsiung	16	120	16	0.133 (5)	0.365 (1)	0.567 (1)
8	Elke Rundensteiner	Murali Mani	16	104	20	0.148 (4)	0.351 (3)	0.477 (6)
9	Divyakant Agrawal	Oliver Po	12	120	12	0.100 (10)	0.316 (6)	0.550 (3)
10	Gerhard Weikum	Martin Theobald	12	106	14	0.111 (8)	0.312 (9)	0.485 (5)

Advisor-advisee relation: **coherence**: low, **cosine**: middle, **kulc**: high

Which Null-invariant Measure is Better?

- **Imbalance Ratio (IR):**

- Measure the imbalance of two itemsets A and B in rule implications

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

- **Kulczynski and IR together present a clear picture for all the three datasets D4 through D6.**

- D4 is balanced & neutral.
- D5 is imbalanced & neutral.
- D6 is very imbalanced & neutral.

Data	mc	$\neg mc$	$m \neg c$	$\neg m \neg c$	all_conf.	max_conf.	Kulc	Cosine	IR
D1	10,000	1,000	1,000	100,000	0.91	0.91	0.91	0.91	0.0
D2	10,000	1,000	1,000	100	0.91	0.91	0.91	0.91	0.0
D3	100	1,000	1,000	100,000	0.09	0.09	0.09	0.09	0.0
D4	1,000	1,000	1,000	100,000	0.5	0.5	0.5	0.5	0.0
D5	1,000	100	10,000	100,000	0.09	0.91	0.5	0.29	0.89
D6	1,000	10	100,000	100,000	0.01	0.99	0.5	0.10	0.99

Summary


Summary

- **Basic concepts:**
 - Association rules.
 - Support-confidence framework.
 - Closed and max-itemsets.
- **Scalable frequent-itemset-mining methods:**
 - Apriori:
 - Candidate generation & test.
 - Projection-based:
 - FP-growth, CLOSET+, ...
 - Vertical-format approach:
 - ECLAT, CHARM, ...
- **Association rules generated from frequent itemsets.**
- **Which patterns are interesting?**
 - Pattern-evaluation methods.

Any questions about this chapter?

Ask them now or ask them later in our forum:

StudOn Forum

 <https://www.studon.fau.de/frm5699567.html>

Appendix

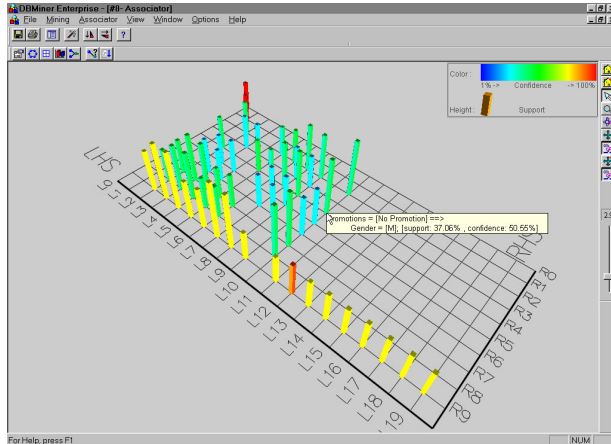
Further Improvements of Mining Methods

- **AFOPT** (Liu et al., KDD'03)
 - A "push-right" method for mining condensed frequent-pattern (CFP) tree.
- **Carpenter** (Pan et al., KDD'03)
 - Mine datasets with small rows but numerous columns.
 - Construct a row-enumeration tree for efficient mining.
- **FP-growth+** (Grahne & Zhu, FIMI'03)
 - Efficiently using prefix-trees in mining frequent itemsets.
- **TD-Close** (Liu et al., SDM'06)

Extension of Pattern-growth Mining Methodology

- **Mining closed frequent itemsets and max-patterns.**
 - CLOSET (DMKD'00), FPclose, and FPMax (Grahne & Zhu, FIMI'03)
- **Mining sequential patterns.**
 - PrefixSpan (ICDE'01), CloSpan (SDM'03), BIDE (ICDE'04)
- **Mining graph patterns.**
 - gSpan (ICDM'02), CloseGraph (KDD'03)
- **Constraint-based mining of frequent patterns.**
 - Convertible constraints (ICDE'01), gPrune (PAKDD'03)
- **Computing iceberg data cubes with complex measures.**
 - H-tree, H-cubing, and Star-cubing (SIGMOD'01, VLDB'03)
- **Pattern-growth-based clustering.**
 - MaPle (Pei et al., ICDM'03)
- **Pattern-growth-based classification.**
 - Mining frequent and discriminative patterns (Cheng et al., ICDE'07)

Visualization of Association Rules (I)



Visualization of Association Rules (II)

