Knowledge Discovery and Data Mining

Spring 2021

Chap 6. Mining Frequent Patterns, Association and Correlations: Basic Concepts and Methods

Jiawei Han, Micheline Kamber and Jian Pei, Data Mining: Concepts and Techniques, 3rd ed., The Morgan Kaufmann Series in Data Management Systems Morgan Kaufmann Publishers, July 2011. ISBN 978-0123814791

Tuong Le, PhD

Outline

What Is Frequent Pattern Analysis?

- Frequent pattern: a pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a data set
- Motivation: Finding inherent regularities in data
  - What products were often purchased together?— Beer and diapers?!
  - What are the subsequent purchases after buying a PC?
  - What kinds of DNA are sensitive to this new drug?
  - Can we automatically classify web documents?

First proposed by Agrawal et al. (1993) in the context of frequent itemsets and association rule mining

- Applications
  - Basket data analysis: The prototypical application was market basket analysis, that is, to mine the sets of items that are frequently bought together at a supermarket by analyzing the customer shopping carts (the so-called "market baskets").
  - Cross-marketing, catalog design, sale campaign analysis, Web log (click stream) analysis, and DNA sequence analysis.

Why Is Frequent Pattern Mining Important?

- Frequent pattern: An intrinsic and important property of datasets
- 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
  - Broad applications

Frequent Itemsets: Terminology

- **Itemsets**: Let $I = \{x_1, x_2, \ldots, x_m\}$ be a set of elements called *items*. A set $X \subseteq I$ is called an *itemset*. An itemset of cardinality (or size) k is called a k-itemset. Further, we denote by $I^{(k)}$ the set of all k-itemsets, that is, subsets of I with size k.

- **Tidsets**: Let $\mathcal{T} = \{t_1, t_2, \ldots, t_n\}$ be another set of elements called transaction identifiers of *tids*. A set $T \subseteq \mathcal{T}$ is called a *tidset*. Itemsets and tidsets are kept sorted in lexicographic order.

- **Transactions**: a transaction is a tuple of the form $\langle t, X \rangle$, where $t \in \mathcal{T}$ is a unique transaction identifier, and X is an itemset.

- **Database**: A binary database D is a binary relation on the set of tids and items, that is, $D \subseteq \mathcal{T} \times I$. We say that tid $t \in \mathcal{T}$ *contains* item $x \in I$ iff $(t, x) \in D$. In other words, $(t, x) \in D$ iff $x \in X$ in the tuple $\langle t, X \rangle$. We say that tid t contains itemset $X = \{x_1, x_2, \ldots, x_k\}$ iff $(t, x_i) \in D$ for all i=1,2,...,k.

Binary Database: Transaction and Vertical Format

D	A	B	C	D	E
1	1	1	0	1	1
2	0	1	1	0	1
3	1	1	0	1	1
4	1	1	1	0	1
5	1	1	1	1	1
6	0	1	1	1	0

Binary Database

t	$i(t)$
1	ABDE
2	BCE
3	ABDE
4	ABCE
5	ABCDE
6	BCD

Transaction Database

$t(x)$				
A	B	C	D	E
1	1	2	1	1
3	2	4	3	2
4	3	5	5	3
5	4	6	6	4
	5			5
	6			

Vertical Database

This dataset D has 5 items, $I = \{A, B, C, D, E\}$ and 6 tids $\mathcal{T} = \{1, 2, 3, 4, 5, 6\}$.

The first transaction is $\langle 1, \{A, B, D, E\} \rangle$ where we omit C since (1, C) $\notin D$. Henceforth, for convenience, we drop the set notation for itemsets and tidsets. Thus, we write $\langle 1, \{A, B, D, E\} \rangle$ as $\langle 1, ABDE \rangle$.

Support and Frequent Itemsets

The support of an itemset X in d database D, denoted $sup(X)$, is the number of transactions in D that contain X:

$$sup(X) = |\{t \mid \langle t, i(t)\rangle \in D \text{ and } X \subset i(t)\}| = |t(X)|$$

The relative support of X is the fraction of transactions that contain X:

$$rsup(X) = \frac{\sup(X)}{|D|}$$

It is an estimate of the joint probability of the items comprising X.

An itemset X is said to be frequent in D if $sup(X) \geq minSup$, where $minSup$ is a user defined minimum support threshold.

The set $\mathcal{F}$ to denotes the set of all frequent itemsets, and $\mathcal{F}^{(k)}$ denotes the set of frequent k-itemsets.

Frequent Itemsets

t	$i(t)$
1	$ABDE$
2	BCE
3	$ABDE$
4	$ABCE$
5	$ABCDE$
6	BCD

Transaction Database

sup	itemsets
6	B
5	E, BE
4	$A, C, D, AB, AE, BC, BD, ABE$
3	$AD, CE, DE, ABD, ADE, BCE, BDE, ABDE$

Frequent itemsets

The 19 frequent itemsets shown in the table comprise the set $\mathcal{F}$. The sets of all frequent k-itemsets are

$\mathcal{F}^{(k)} = \{A, B, C, D, E\}$, $\mathcal{F}^{(2)} = \{AB, AD, AE, BC, BD, BE, CE, DE\}$, $\mathcal{F}^{(3)} = \{ABD, ABE, ADE, BCE, BDE\}$, $\mathcal{F}^{(4)} = \{ABDE\}$

Association Rules

- An association rule is an expression:

$$X \xrightarrow{s,c} Y$$

where X and Y are itemsets and they are disjoint, that is, $X, Y \subseteq I$ and $X \cap Y = \emptyset$. Let the itemset $X \cup Y$ be denoted as XY.

- The support of the rule is the number of transactions in which both X and Y co-occur as subsets.

$$s = \sup(X \to Y) = |t(XY)| = \sup(XY)$$

Association Rules

- The relative support of the rule is defined as the fraction of transactions where X and Y co-occur, and it provides an estimate of the joint probability of X and Y:

$$rsup(X \to Y) = \frac{\sup(XY)}{|D|} = P(X \wedge Y)$$

- The confidence of a rule is the conditional probability that a transaction contains Y given that it contains X:

$$c = conf(X \to Y) = P(Y|X) = \frac{P(X \wedge Y)}{P(X)} = \frac{\sup(XY)}{\sup(X)}$$

Computational Complexity of Frequent Itemset Mining

- How many itemsets are potentially to be generated in the worst case?
  - The number of frequent itemsets to be generated is senstive to the minsup threshold
  - When minsup is low, there exist potentially an exponential number of frequent itemsets
  - The worst case: M^N where M: # distinct items, and N: max length of transactions
- The worst case complexty vs. the expected probability
  - Ex. Suppose Walmart has 10^4 kinds of products
    - The chance to pick up one product 10^{-4}
    - The chance to pick up a particular set of 10 products: ~10^{-40}
    - What is the chance this particular set of 10 products to be frequent 10^3 times in 10^9 transactions?

Outline

1. Basic Concepts
2. **Frequent Itemset Mining Methods: Apriori, FPGrowth, Eclat & dEclat.**
3. Generating Association Rules
4. Pattern Evaluation Methods
5. Summary

The Downward Closure Property and Scalable Mining Methods

- The downward closure property of frequent patterns
  - Any subset of a frequent itemset must be frequent
  - If {beer, diaper, nuts} is frequent, so is {beer, diaper}
  - i.e., every transaction having {beer, diaper, nuts} also contains {beer, diaper}
- Scalable mining methods: Three major approaches
  - Apriori (Agrawal & Srikant@VLDB'94)
  - Frequent pattern growth (Fpgrowth — 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 superset should not be generated/tested! (Agrawal & Srikant @VLDB'94, Mannila, et al. @ KDD' 94)
- General idea:
  - Initially, scan *DB* once to get frequent 1-itemset
  - Generate length (k+1) candidate itemsets from length k frequent itemsets
  - Test the candidates against *DB*
  - Terminate when no frequent or candidate set can be generated

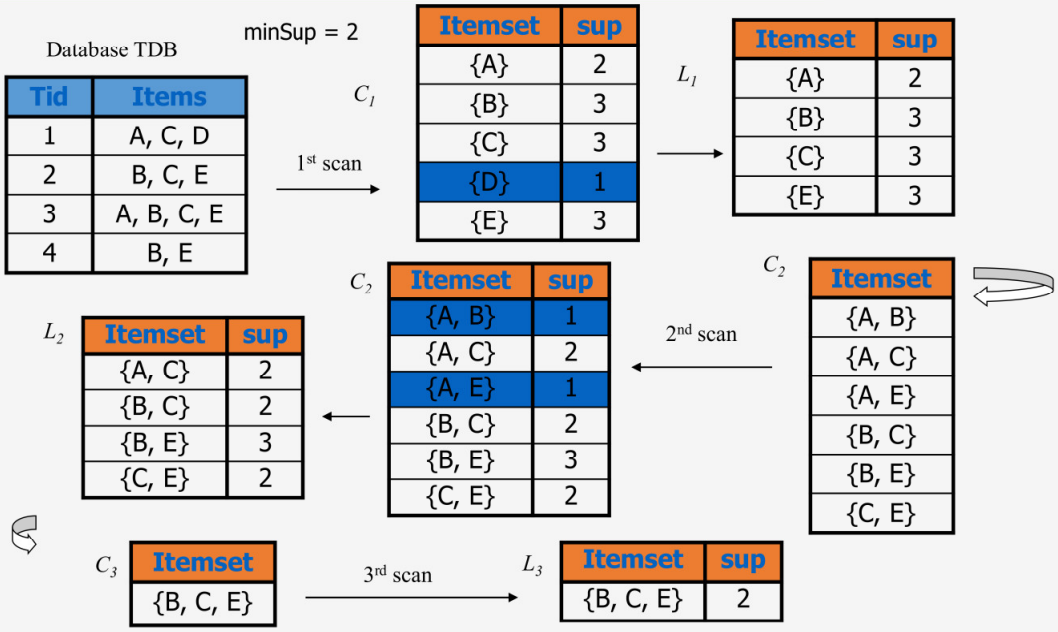
The Apriori Algorithm (Pseudo-Code)

```
Apriori (D, I, minsup):
1  F ← ∅
2  C(1) ← {∅} // Initial prefix tree with single items
3
4  foreach i ∈ I do Add i as child of ∅ in C(1) with sup(i) ← 0
5
6  k ← 1 // k denotes the level
7  while C(k) ≠ ∅ do
8      ComputeSupport (C(k), D)
9      foreach leaf X ∈ C(k) do
10         if sup(X) ≥ minsup then F ← F ∪ {(X, sup(X))}
11
12         else remove X from C(k)
13
14     C(k+1) ← ExtendPrefixTree (C(k))
15     k ← k + 1
16 return F(k)
```

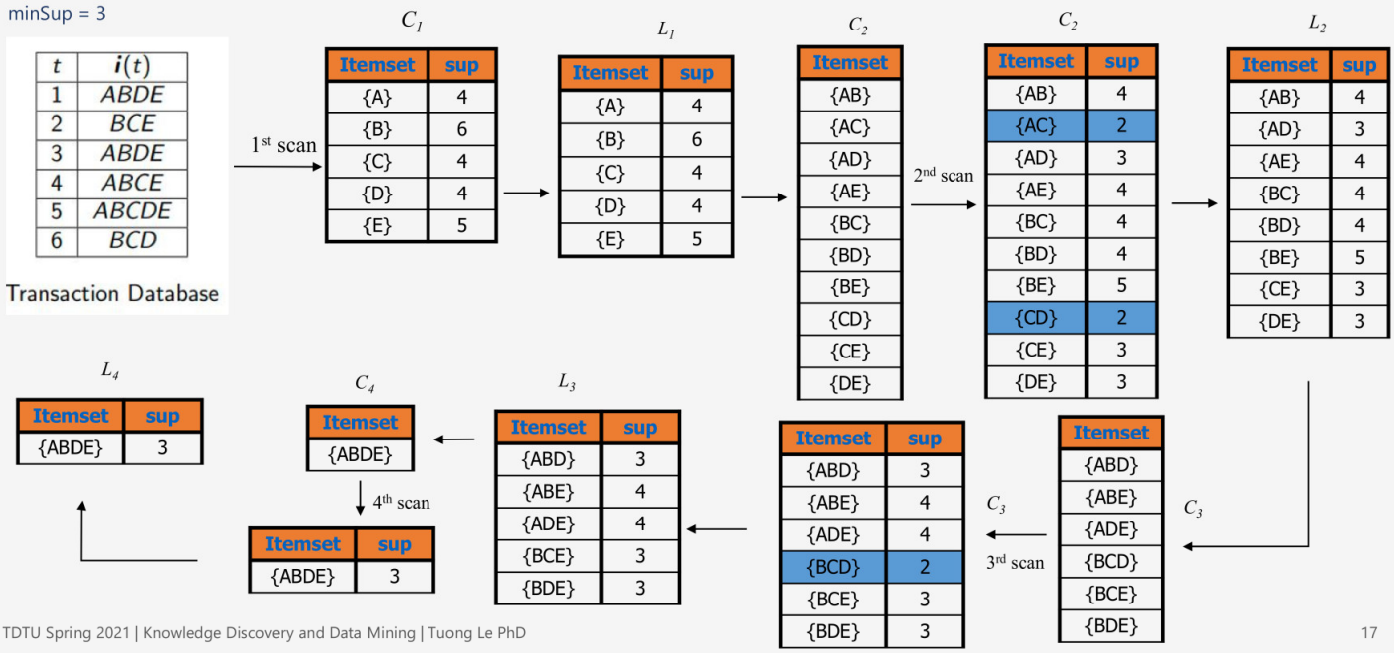
```
ComputeSupport (C(k), D):
1  foreach ⟨t, i(t)⟩ ∈ D do
2      foreach k-subset X ⊆ i(t) do
3          if X ∈ C(k) then sup(X) ← sup(X) + 1
4
```

```
ExtendPrefixTree (C(k)):
1  foreach leaf Xa ∈ C(k) do
2      foreach leaf Xb ∈ sibling(Xa), such that b > a do
3          Xab ← Xa ∪ Xb
           // prune candidate if there are any infrequent
              subsets
4          if Xj ∈ C(k), for all Xj ⊂ Xab, such that |Xj| = |Xab| − 1 then
5              Add Xab as child of Xa with sup(Xab) ← 0
6
7      if no extensions from Xa then
8          remove Xa, and all ancestors of Xa with no extensions, from
           C(k)
```


The Apriori Algorithm: An Example



Exercise 1. Find Frequent itemsets with Apriori Algorithm



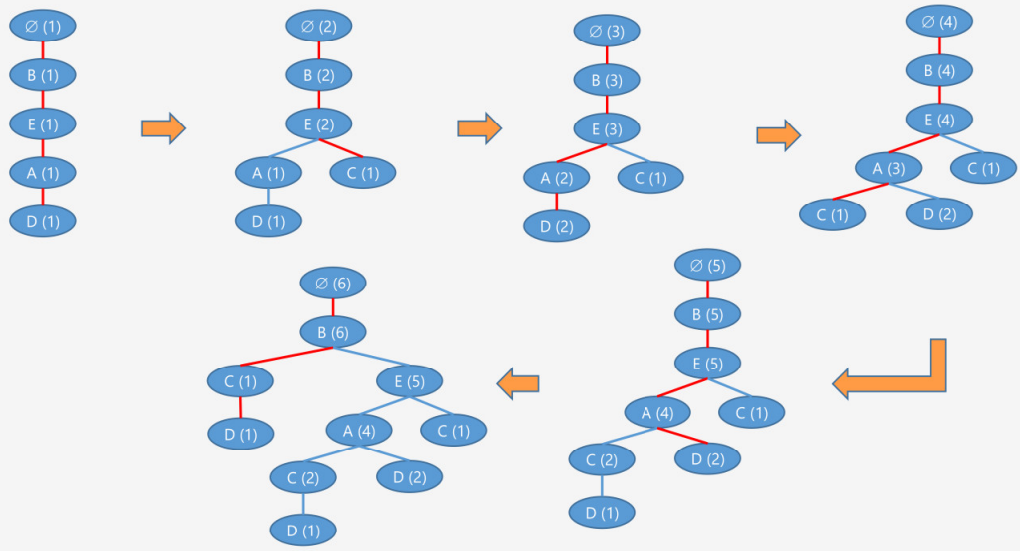
Pattern-Growth Approach: Mining Frequent Patterns Without Candidate Generation

- Bottlenecks of the Apriori approach
  - Breadth-first (i.e., level-wise) search
  - Candidate generation and test
    - Often generates a huge number of candidates
- The FPGrowth Approach (J. Han, J. Pei, and Y. 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., project DB on abc: DB|abc
  - "d" is a local frequent item in DB|abc → abcd is a frequent pattern

Frequent Pattern Tree

The FP-tree is a prefix compressed representation of D. For most compression items are sorted in descending order of support.

Transactions
BEAD
BEC
BEAD
BEAC
BEACD
BCD



The Frequent Pattern Growth Mining Method

❑ Idea: Frequent pattern growth

- Recursively grow frequent patterns by pattern and database partition

❑ Method

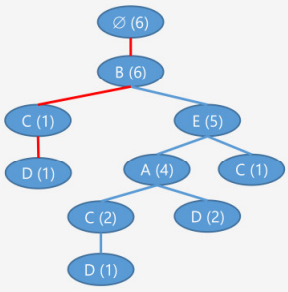
- For each frequent item, construct its conditional pattern-base, and then its conditional FP-tree
- Repeat the process on each newly created conditional FP-tree
- Until the resulting FP-tree is empty, or it contains only one path—single path will generate all the combinations of its sub-paths, each of which is a frequent pattern

FPGrowth Algorithm

```
// Initial Call:  R ← FP-tree(D), P ← ∅, F ← ∅
FPGrowth (R, P, F, minsup):
1  Remove infrequent items from R
2  if IsPath(R) then // insert subsets of R into F
3      foreach Y ⊆ R do
4          X ← P ∪ Y
5          sup(X) ← min_{x∈Y}{cnt(x)}
6          F ← F ∪ {(X, sup(X))}
7  else   // process projected FP-trees for each frequent item i
8      foreach i ∈ R in increasing order of sup(i) do
9          X ← P ∪ {i}
10         sup(X) ← sup(i) // sum of cnt(i) for all nodes labeled i
11
12         F ← F ∪ {(X, sup(X))}
13         R_X ← ∅ // projected FP-tree for X
14
15         foreach path ∈ PathFromRoot(i) do
16             cnt(i) ← count of i in path
17             Insert path, excluding i, into FP-tree R_X with count cnt(i)
18         if R_X ≠ ∅ then FPGrowth (R_X, X, F, minsup)
```

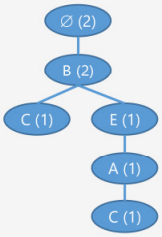

Projected Frequent Pattern Tree for D

FP-tree

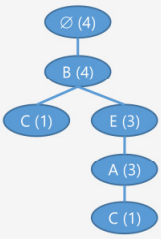


Add BC, cnt=1

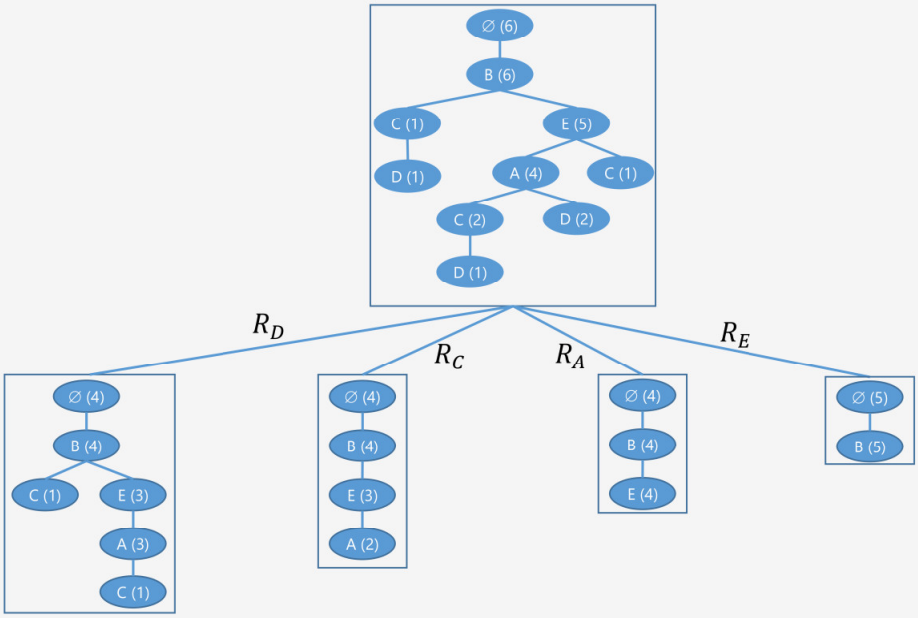
Add BEAC, cnt=1



Add BEA, cnt=2



FPGrowth Algorithm: Frequent Pattern Tree Projection



Advantages of the Pattern Growth Approach

- Divide-and-conquer:
  - Decompose both the mining task and DB according to the frequent patterns obtained so far
  - Lead 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 freq items and building sub FP-tree, no pattern search and matching
- A good open-source implementation and refinement of FPGrowth
  - FPGrowth+ (Grahne and J. Zhu, FIMI'03)

Tidset Intersection Approach: Eclat Algorithm

The support counting step can be improved significantly if we can index the database in such a way that it allows fast frequency computations.

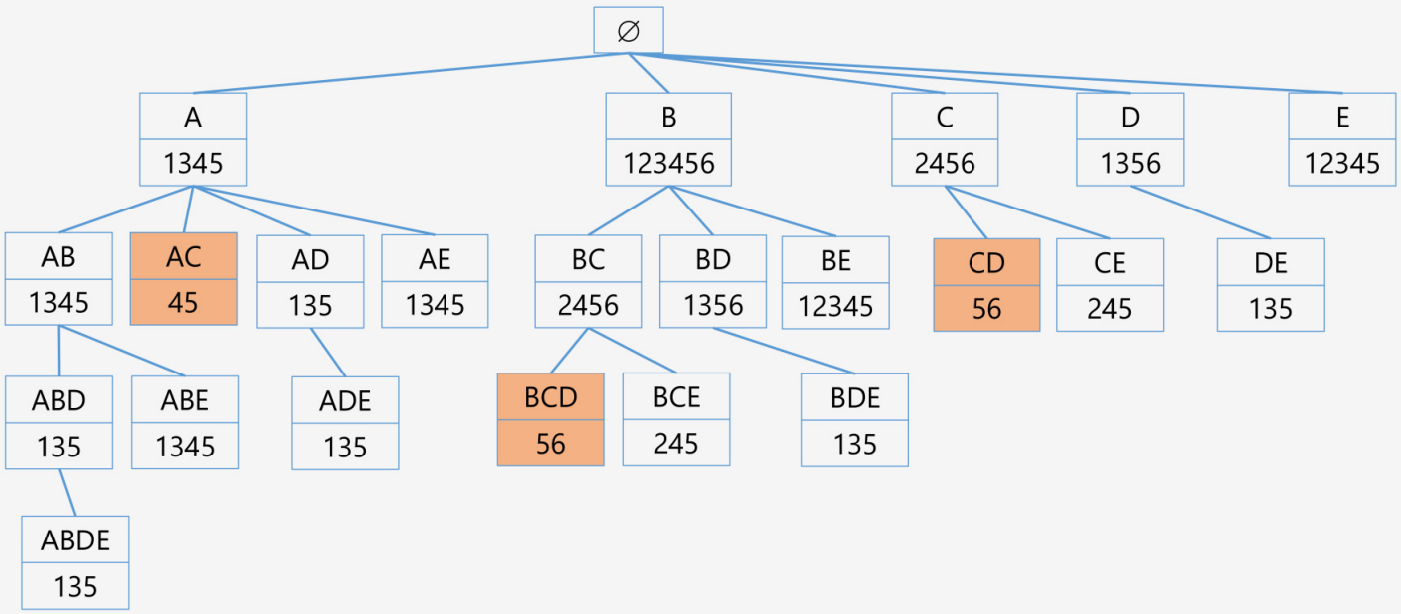
The Eclat algorithm leverages the tidsets directly for support copuatation. The basic idea is that the support of a candidate itemset can be computed by intersecting the tidsets of suitably chosen subsets. In general, given $\boldsymbol{t}(X)$ and $\boldsymbol{t}(Y)$ for any two frequent itemsets X and Y, we have

$$t(XY) = t(X) \cap t(Y)$$

The support of candidate XY is sumply the cardinality of $\boldsymbol{t}(XY)$, that is, $sup(XY) = |t(XY)|$.

Eclat intersects the tidsets only if the frequent itemsets share a common prefix, and it traverses the prefix search tree in a DFS-like manner, processing a group of itemsets that have the same prefix, also called a prefix equivalence class.

Eclat Algorithm: Tidlist Intersections



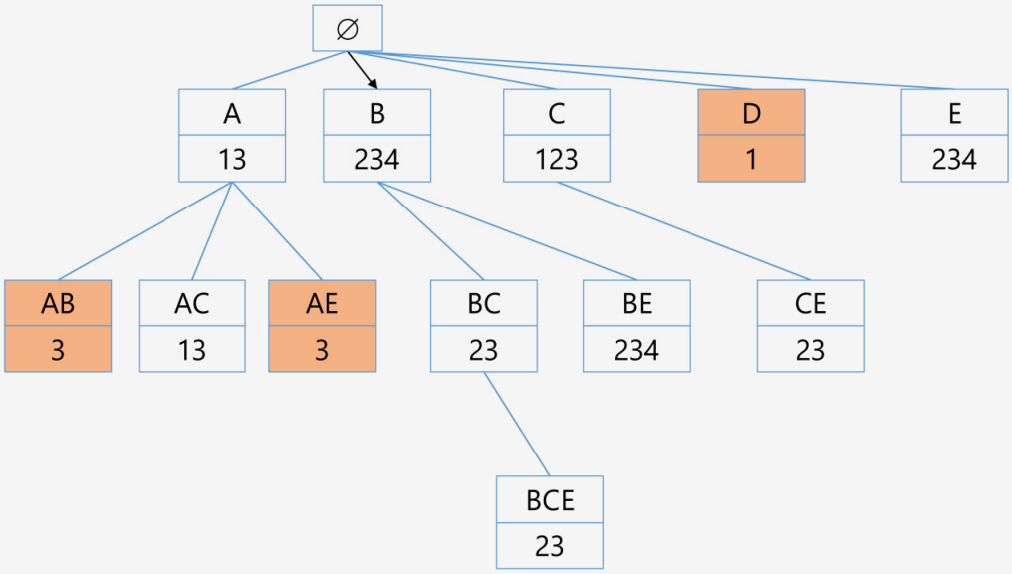
Eclat Algorithm

```
// Initial Call:  F ← ∅, P ← {⟨i, t(i)⟩ | i ∈ I, |t(i)| ≥ minsup}
Eclat (P, minsup, F):
1  foreach ⟨X_a, t(X_a)⟩ ∈ P do
2      F ← F ∪ {(X_a, sup(X_a))}
3      P_a ← ∅
4      foreach ⟨X_b, t(X_b)⟩ ∈ P, with X_b > X_a do
5          X_ab = X_a ∪ X_b
6          t(X_ab) = t(X_a) ∩ t(X_b)
7          if sup(X_ab) ≥ minsup then
8              P_a ← P_a ∪ {⟨X_ab, t(X_ab)⟩}
9      if P_a ≠ ∅ then Eclat (P_a, minsup, F)
10
```


Exercise 2. Find Frequent itemsets with Eclat Algorithm

minSup = 2

Tid	Items
1	A, C, D
2	B, C, E
3	A, B, C, E
4	B, E



Diffsets: Difference of Tidsets

The Eclat algorithm can be significantly improved if we can shrink the size of the intermediate tidsets. This can be achieved by keeping track the differences in the tidsets as opposed to the full tidsets.

Let $X_a = \{x_1, \ldots, x_{k-1}, x_a\}$ and $X_b = \{x_1, \ldots, x_{k-1}, x_b\}$, so that $X_{ab} = X_a \cup X_b = \{x_1, \ldots, x_{k-1}, x_a, x_b\}$.

The diffset of X_{ab} is the set of tids that contain the prefix X_a, but not the item X_b

$$\mathrm{d}(X_{ab}) = \mathrm{t}(X_a) \setminus \mathrm{t}(X_{ab}) = \mathrm{t}(X_a) \setminus \mathrm{t}(X_b)$$

We can obtain an expression for $\mathrm{d}(X_{ab})$ in terms of $\mathrm{d}(X_a)$ and $\mathrm{d}(X_b)$ as follows:

$$\mathrm{d}(X_{ab}) = \mathrm{d}(X_b) \setminus \mathrm{d}(X_a)$$

which means that we can replace all intersection operations in Eclat with diffset operations.

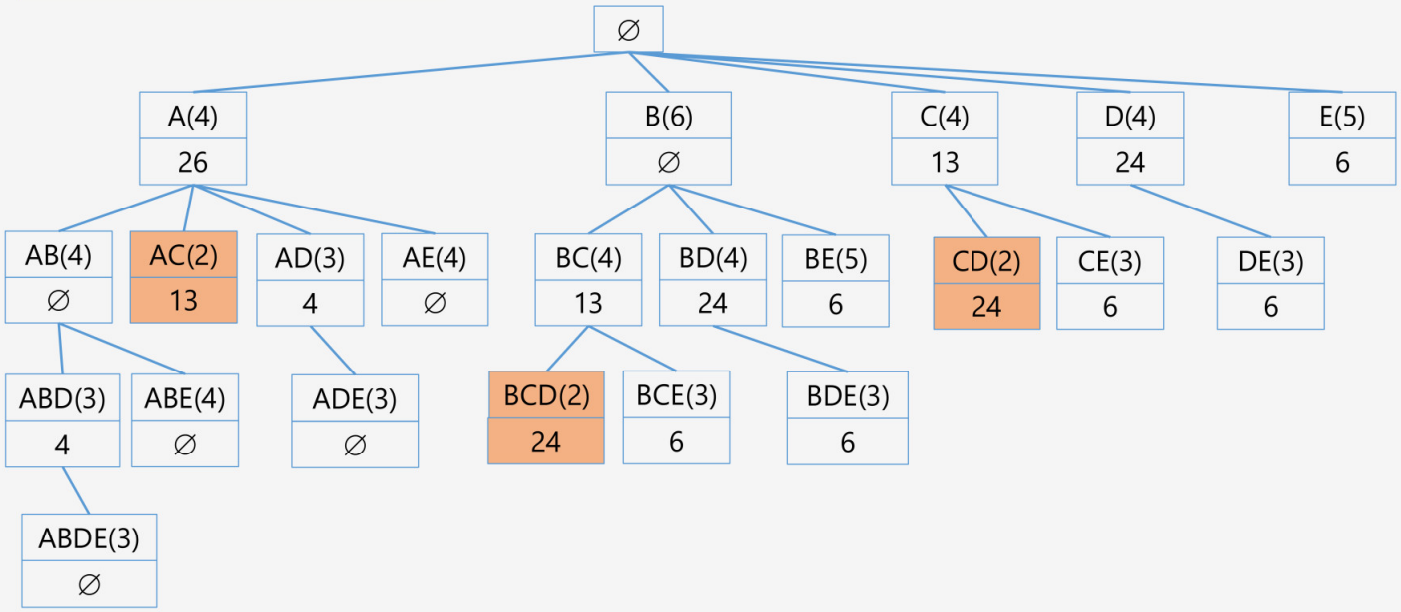
dEclat Algorithm

// Initial Call: $\mathcal{F} \leftarrow \emptyset$,
$P \leftarrow \{\langle i, \boldsymbol{d}(i), sup(i)\rangle \mid i \in \mathcal{I}, \boldsymbol{d}(i) = \mathcal{T} \setminus \boldsymbol{t}(i), sup(i) \geq minsup\}$

dEclat (P, $minsup$, $\mathcal{F}$):

1 **foreach** $\langle X_a, \boldsymbol{d}(X_a), sup(X_a)\rangle \in P$ **do**
2 $\mathcal{F} \leftarrow \mathcal{F} \cup \{(X_a, sup(X_a))\}$
3 $P_a \leftarrow \emptyset$
4 **foreach** $\langle X_b, \boldsymbol{d}(X_b), sup(X_b)\rangle \in P$, *with* $X_b > X_a$ **do**
5 $X_{ab} = X_a \cup X_b$
6 $\boldsymbol{d}(X_{ab}) = \boldsymbol{d}(X_b) \setminus \boldsymbol{d}(X_a)$
7 $sup(X_{ab}) = sup(X_a) - |\boldsymbol{d}(X_{ab})|$
8 **if** $sup(X_{ab}) \geq minsup$ **then**
9 $P_a \leftarrow P_a \cup \{\langle X_{ab}, \boldsymbol{d}(X_{ab}), sup(X_{ab})\rangle\}$
10 **if** $P_a \neq \emptyset$ **then** dEclat (P_a, $minsup$, $\mathcal{F}$)
11

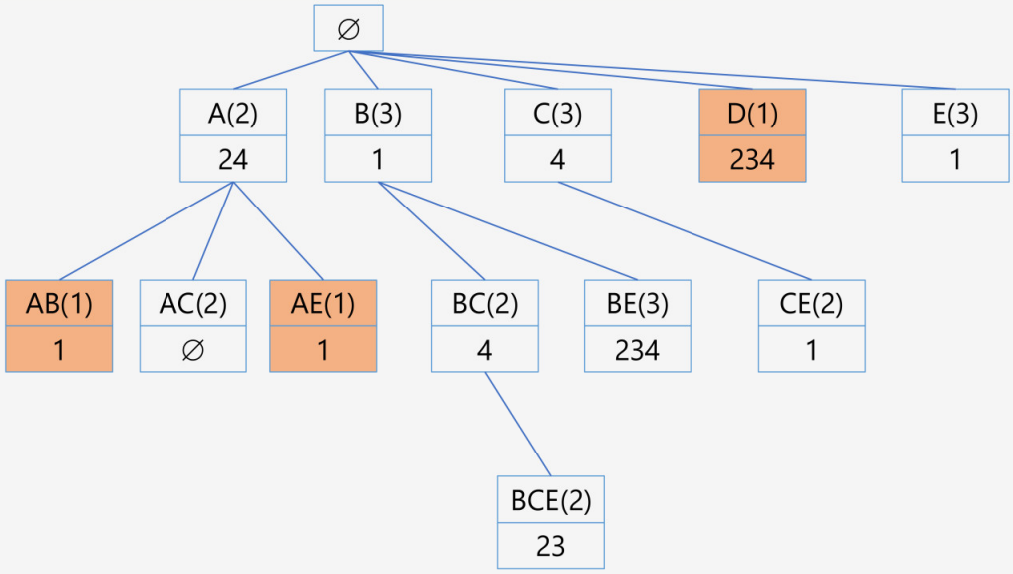
dEclat Algorithm: Diffsets



Exercise 3. Find Frequent itemsets with dEclat Algorithm

minSup = 2

Tid	Items
1	A, C, D
2	B, C, E
3	A, B, C, E
4	B, E



Outline

1. Basic Concepts

2. Frequent Itemset Mining Methods: Apriori, FPGrowth, Eclat & dEclat.

3. Generating Association Rules

4. Pattern Evaluation Methods

5. Summary

Generating Association Rules

Given a frequent itemset $Z \in \mathcal{F}$, we look at all proper subsets $X \subset Z$ to compute rules of the form

$$X \xrightarrow{s,c} Y, \text{ where } Y = Z \setminus X$$

where $Z \setminus X = Z - X$.

The rule must be frequent because

$$s = \sup(XY) = \sup(Z) \geq \text{minSup}$$

We compute the confidence as follows.

$$c = \frac{\sup(X \cup Y)}{\sup(X)} = \frac{\sup(Z)}{\sup(X)}$$

If $c \geq$ minConf, then the rule is a strong rule. On the other hand, if conf($X \rightarrow Y$)< c, then conf($W \rightarrow Z \setminus W$) < c for all subsets $W \subset X$, as $\sup(W) \geq \sup(X)$. We can thus avoid checking subsets of X.

Association Rule Mining Algorithm

AssociationRules ($\mathcal{F}$, *minconf*):

1 **foreach** $Z \in \mathcal{F}$, *such that* $|Z| \geq 2$ **do**
2 $\mathcal{A} \leftarrow \{X \mid X \subset Z, X \neq \emptyset\}$
3 **while** $\mathcal{A} \neq \emptyset$ **do**
4 $X \leftarrow$ maximal element in $\mathcal{A}$
5 $\mathcal{A} \leftarrow \mathcal{A} \setminus X$ // remove X from $\mathcal{A}$
6
7 $c \leftarrow sup(Z)/sup(X)$
8 **if** $c \geq minconf$ **then**
9 print $X \longrightarrow Y$, $sup(Z)$, c
10 **else**
11 $\mathcal{A} \leftarrow \mathcal{A} \setminus \{W \mid W \subset X\}$
 // remove all subsets of X from $\mathcal{A}$
12

Outline

1. Basic Concepts

2. Frequent Itemset Mining Methods: Apriori, FPGrowth, Eclat & dEclat.

3. Generating Association Rules

4. Pattern Evaluation Methods

5. Summary

Interestingness Measure: Correlations (Lift)

- *play basketball* ⇒ *eat cereal* [40%, 66.7%] is misleading
  - The overall % of students eating cereal is 75% > 66.7%.
- *play basketball* ⇒ *not eat cereal* [20%, 33.3%] is more accurate, although with lower support and confidence
- Measure of dependent/correlated events: lift

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

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

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

	Basketball	Not basketball	Sum (row)
Cereal	2000	1750	3750
Not cereal	1000	250	1250
Sum(col.)	3000	2000	5000

Are *lift* and χ^2 Good Measures of Correlation?

- *"Buy walnuts ⇒ buy milk* [1%, 80%]" is misleading if 85% of customers buy milk
- Support and confidence are not good to indicate correlations
- Over 20 interestingness measures have been proposed (Tan, Kumar, Sritastava @KDD'02)

symbol	measure	range	formula
ϕ	ϕ-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	$\frac{P(A,B)P(\overline{A},\overline{B})-P(A,\overline{B})P(\overline{A},B)}{P(A,B)P(\overline{A},\overline{B})+P(A,\overline{B})P(\overline{A},B)}$
Y	Yule's Y	-1 . . . 1	$\frac{\sqrt{P(A,B)P(\overline{A},\overline{B})}-\sqrt{P(A,\overline{B})P(\overline{A},B)}}{\sqrt{P(A,B)P(\overline{A},\overline{B})}+\sqrt{P(A,\overline{B})P(\overline{A},B)}}$
k	Cohen's	-1 . . . 1	$\frac{P(A,B)+P(\overline{A},\overline{B})-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A)P(B)-P(\overline{A})P(\overline{B})}$
PS	Piatetsky-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	Klosgen'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	$\frac{\sum_j \max_k P(A_j,B_k)+\sum_k \max_j P(A_j,B_k)-\max_j P(A_j)-\max_k P(B_k)}{2-\max_j P(A_j)-max_k P(B_k)}$
M	Mutual Information	0 . . . 1	$\frac{\sum_i\sum_j P(A_i,B_j)\log\frac{P(A_i,B_j)}{P(A_i)P(B_j)}}{\min(-\sum_i P(A_i)\log P(A_i), -\sum_i P(B_i)\log P(B_i))}$
J	J-Measure	0 . . . 1	$\max(P(A,B)\log(\frac{P(B\|A)}{P(B)})+P(A\overline{B})\log(\frac{P(\overline{B}\|A)}{P(\overline{B})}))$
G	Gini index	0 . . . 1	$P(A,B)\log(\frac{P(A\|B)}{P(A)})+P(A\overline{B})\log(\frac{P(\overline{A}\|B)}{P(\overline{A})})$ $\max(P(A)[P(B\|A)^2+P(\overline{B}\|A)^2]+P(\overline{A})[P(B\|\overline{A})^2+P(\overline{B}\|\overline{A})^2]-P(B)^2-P(\overline{B})^2,$ $P(B)[P(A\|B)^2+P(\overline{A}\|B)^2]+P(\overline{B})[P(A\|\overline{B})^2+P(\overline{A}\|\overline{B})^2]-P(A)^2-P(\overline{A})^2)$
s	support	0 . . . 1	$P(A,B)$
c	confidence	0 . . . 1	$max(P(B\|A), P(A\|B))$
L	Laplace	0 . . . 1	$\max(\frac{NP(A,B)+1}{NP(A)+2}, \frac{NP(A,B)+1}{NP(B)+2})$
IS	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 . . . ∞	$\frac{P(A,B)P(\overline{A},\overline{B})}{P(\overline{A},B)P(A,\overline{B})}$
V	Conviction	0.5 . . . ∞	$\max(\frac{P(A)P(\overline{B})}{P(A\overline{B})}, \frac{P(B)P(\overline{A})}{P(B\overline{A})})$
λ	lift	0 . . . ∞	$\frac{P(A,B)}{P(A)P(B)}$
S	Collective strength	0 . . . ∞	$\frac{P(A,B)+P(\overline{A}\overline{B})}{P(A)P(B)+P(\overline{A})P(\overline{B})} \times \frac{1-P(A)P(B)-P(\overline{A})P(\overline{B})}{1-P(A,B)-P(\overline{A}\overline{B})}$
χ^2	χ^2	0 . . . ∞	$\sum_i \frac{(P(A_i)-E_i)^2}{E_i}$

Outline

1. Basic Concepts
2. Frequent Itemset Mining Methods: Apriori, FPGrowth, Eclat & dEclat.
3. Generating Association Rules
4. Pattern Evaluation Methods
5. **Summary**

Summary

- ❑ Basic concepts: **Frequent pattern**, **association rules**.
- ❑ Frequent pattern mining methods
  - **Apriori**: A Candidate Generation-and-Test Approach
  - **FPGrowth**: A Frequent Pattern-Growth Approach
  - **Eclat** & **dEclat**: Frequent Pattern Mining with Vertical Data Format
- ❑ **Generating Association Rules**
- ❑ Which patterns are interesting?
  - **Pattern evaluation** methods