IFN509

Data Exploration and Mining

Week 6

Association Mining

Where are we?
Descriptive Data Mining

Week	Lecture	Tutorial	Practical	Assessment
Week 6	Descriptive Mining: Association Mining	Drop-in Session	Drop-in Session	Assessment 1 Report Due
Week 7	Descriptive Mining: Clustering	Assessment 1 marking	Assessment 1 marking	Assessment 2 Release

What Should You Do This Week?

- Review the lecture slides and reading materials.
- Attempt the exercise questions set in the tutorial
- Complete the python tasks
- Consult Lecturer/tutor if you have any questions related to the subject.
- Assessment Item 1
  - Due 18th April mid-night
  - Where should you be:
    - All Tasks are completed. Only tweaking of some tasks may be required.
    - A draft report is ready to be shared amongst the group members.
    - ChatGPT - Submitting a solution that was generated by an AI is no different to submitting a solution developed by another person or obtained from the internet. They are therefore liable for charges of academic misconduct including contract cheating, plagiarism and attempting to defeat the purpose of the assessment. We also reserve the right to conduct an authentication of learning where we suspect that an answer was not produced by the student (MOPP Section C 5.3.7)

Introduction: Association Mining

Association Analysis

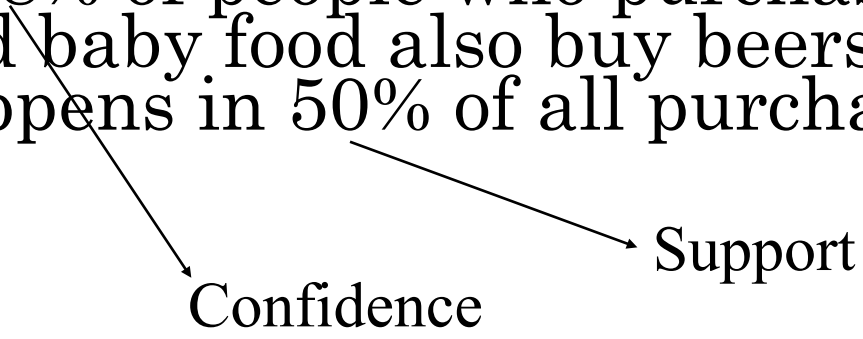
- Establishing links between variables according to a set of records in a data set.
- These links are often called **association**.
- Two specialisation:
  - Association discovery
  - Sequential pattern discovery

Why Is Frequent Pattern or Association Mining an Essential Task in Data Mining?

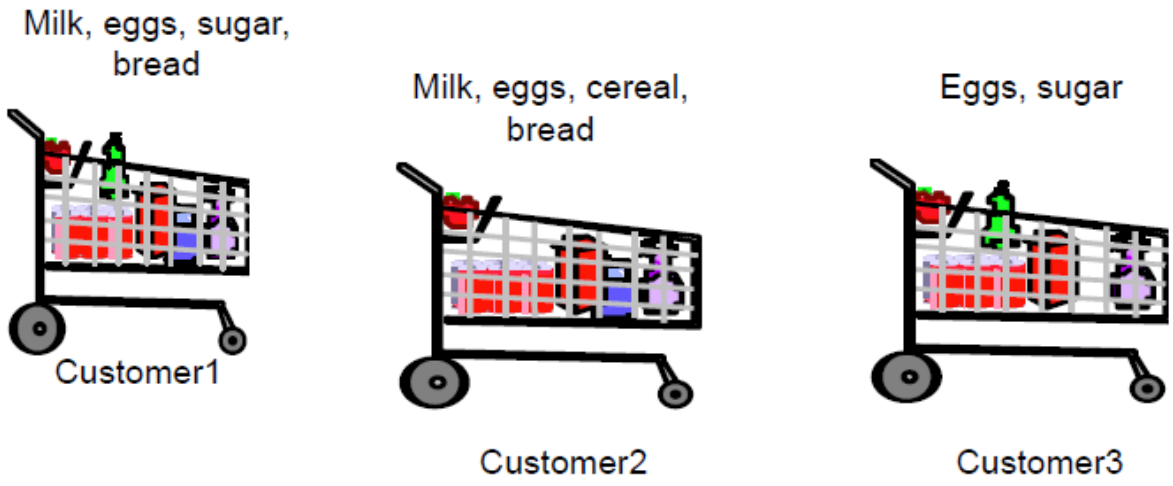
- Foundation for many essential data mining tasks
  - Association, correlation, causality
  - Sequential patterns, temporal or cyclic association
  - Associative classification, cluster analysis
- Broad applications
  - Market Basket data analysis, cross-marketing, catalog design, sale campaign analysis
  - Web log (click stream) analysis, DNA sequence analysis, etc.

Association Mining

- Often referred to as '**Market-basket analysis**'
- Finding patterns in transactional data
  - Find items that imply the presence of other items in the same transaction
    - Which items are generally purchased together in one transaction ?
    - Milk + Cereal, Chips +Salsa
    - Web pages A and B → an online order
- Example: 98% of people who purchase diapers and baby food also buy beers. This usually happens in 50% of all purchases.



Example: market basket analysis



KNOWLEDGE

Which products are bought together?

What are the subsequent purchases after buying some products?

…

DECISIONS

Layout of the store

Promotions

Targeted advertising

Example Rule: Barbie® ⇒ Candy

Rules are used in many marketing decisions such as:

- Put them closer together in the store.
- Put them far apart in the store.
- Package candy bars with the dolls.
- Package Barbie + candy + poorly selling item.
- Raise the price on one, lower it on the other.
- Barbie accessories for proofs of purchase.
- Do not advertise candy and Barbie together.
- Offer candies in the shape of a Barbie Doll.

Association Rules

Support, Confidence & Lift

Definitions

- Item: *attribute=value* pair or simply *value*
  - usually attributes are converted to binary *flags* for each value, e.g. **product="A"** is written as **"A"**
- Itemset : a subset of possible items
  - Example: {A,B,E} (order and frequency unimportant)
- Transaction: (TID, itemset)
  - TID is transaction identification (ID).

Transactions Example

TID	Products
1	MILK, BREAD, EGGS
2	BREAD, SUGAR
3	BREAD, CEREAL
4	MILK, BREAD, SUGAR
5	MILK, CEREAL
6	BREAD, CEREAL
7	MILK, CEREAL
8	MILK, BREAD, CEREAL, EGGS
9	MILK, BREAD, CEREAL

ITEMS:

A = milk
B= bread
C= cereal
D= sugar
E= eggs

TID	Products
1	A, B, E
2	B, D
3	B, C
4	A, B, D
5	A, C
6	B, C
7	A, C
8	A, B, C, E
9	A, B, C

(Variable = value) is referred as an **Item**.
Instance is referred as a **Transaction**.

Association Rule

- **An association rule:** *Itemset1* => *Itemset2*
  - In a set of transactions, *Itemset1* associates to *Itemset2.*
  - *Itemsets 1 or 2* may be a single item or sets of items.
  - *Itemsets 1 and 2* should be **disjoint**
    - **The same item should not appear in both itemsets.**
  - *Itemset2* cannot be empty.

  Examples
    - A,B => E,C
    - A => B,C
    - ~~A,B => B,C~~
    - ~~A, B => {}~~

Association Mining

- Usually, the underlying data is sparse.
- Can be applied if **no class is specified** and **any** kind of **structure** is considered "**interesting**"
- Typical Rule form: Body ==> Head.
  - Any number of items in the rule body and head.
- Hence: far **more association rules in numbers**

Measures of interestingness of the rule – Support and Confidence

- Association rules should be
  - non-trivial (and possibly unexpected)
  - actionable
  - easily explainable
- Not every rule is interesting
  - **Support**
    - indicates the frequency of the pattern.
  - **Confidence**
    - denotes the strength of the association.
- A minimum support and confidence is necessary if an association is going to be of some business value.

Support

- The support for A=>B is the probability of both A and B appearing together.
  - Measures how often items occur together, as a percentage of the total transactions.

$$\frac{number\ of\ transactions\ containing\ A\ and\ B}{the\ total\ number\ of\ transactions}$$

Confidence

The confidence of A=>B is the conditional probability of B appearing given that A exists, that is:

- $P(B \mid A) = P(A \cup B) / P(A)$
- Measures how much an item is dependent on another.

$$\frac{number\ of\ transaction\ supporting\ the\ rule}{number\ of\ transactions\ supporting\ the\ rule\ body}$$

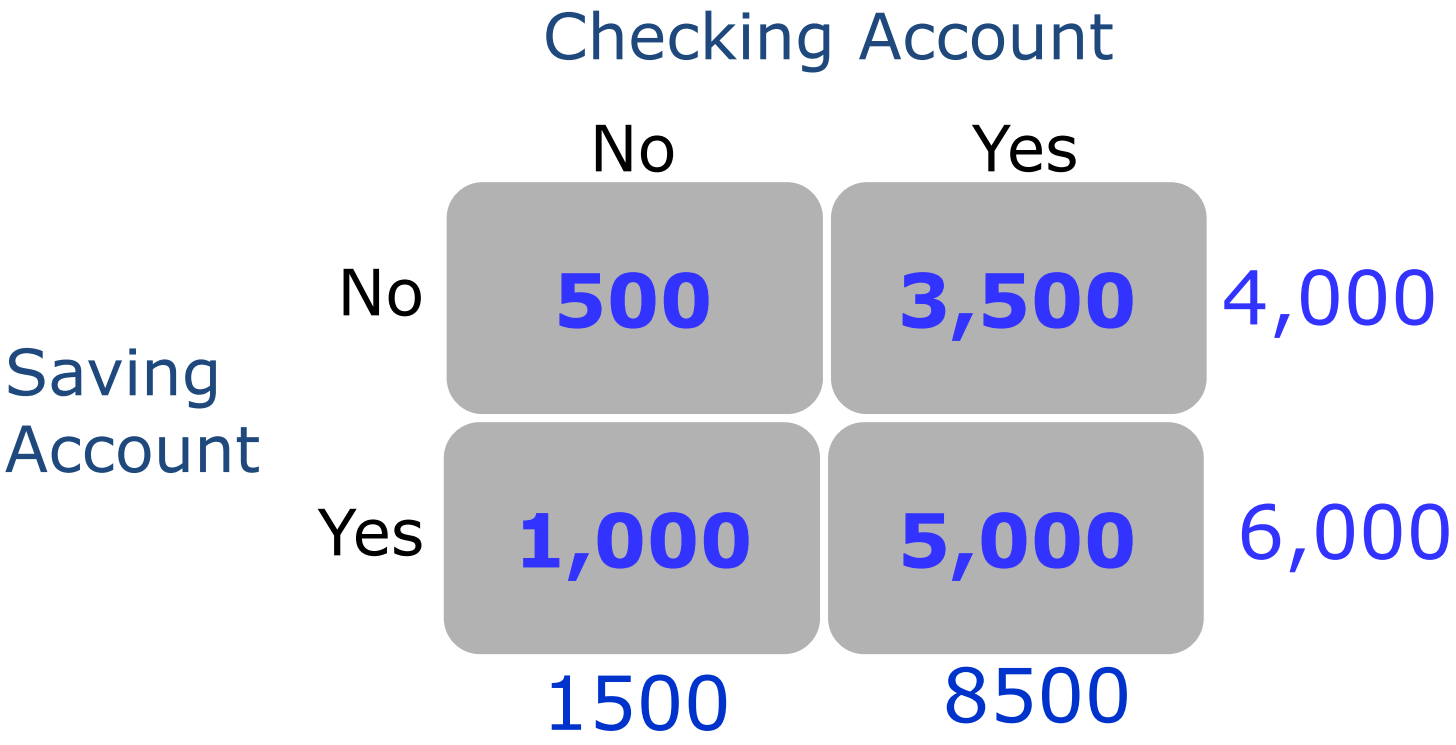
$$\frac{number\ of\ transactions\ containing\ A\ and\ B}{number\ of\ transactions\ containing\ A\ only}$$

Association Rules: Example

Rule	Support	Confidence
A ⇒ D	2/5	2/3
C ⇒ A	2/5	2/4
A ⇒ C	2/5	2/3
B & C ⇒ D	1/5	1/3

✓Aim: Determine the strength of all association rules among a set of items

Another Example



Total Customers: 10,000

Association Rule

	Checking Account: No	Checking Account: Yes
Saving Account: No	500	3,500
Saving Account: Yes	1,000	5,000

- SVG $\Rightarrow$ CK

  Support(SVG $\Rightarrow$ CK) = 50%
  (5000/10000)

  Confidence(SVG $\Rightarrow$ CK) = 83%
  (5000/6000)

- CK $\Rightarrow$ SVG

  Support (CK $\Rightarrow$ SVG) = 50%
  (5000/10000)

  Confidence (CK $\Rightarrow$ SVG) = 59%
  (5000/8500)

Implication

Saving Account \ Checking Account	No	Yes
No	500	3,500
Yes	1,000	5,000

- SVG ⇒ CK (A strong rule?)

  Support(SVG ⇒ CK) = 50%

  Confidence(SVG ⇒ CK) = 83%

- Those without a saving account, having check account (~SVG ⇒ CK)

  Support is 35% (3500/10000)

  Confidence is 87.5% (3500/4000)

- Check and savings are negatively correlated as well.

Improvement or Lift

- High confidence rules are not necessarily useful
  - what if confidence of {A ⇒ C} is less than Support(C)?
  - Note that Confidence (SVG ⇒ CK) = 83% is less than Support (CK) = 85%.

- Lift gives the predictive power of a rule compared to just random chance:

$$improvement = \frac{\Pr(result \mid condition)}{\Pr(result)} = \frac{confidence(rule)}{support(result)}$$

- In our example, the lift provided by the SVG ⇒ CK rule is:

  0.97 (= 83%/85%).

Lift

- **Lift measures the strength of an effect.**
  - interpreted as a general measure of associations between the two item sets.
    - Lift > 1 indicate positive correlation
    - Lift < 1 indicate negative correlation
    - Lift = 1 indicate zero correlation
      - Two itemsets are independent

Interesting Rules

It is worth noting:

- The only rules of interest are with very high or very low lift (why?)
- Items that appear on most (or least) transactions are of interest (why?)
- Similar items should be combined to reduce the number of total items (why?)

Association Rule Mining

Frequent and Candidate Itemsets
Apriori algorithm
(Generate and Test)

Frequent Itemset

- Frequent itemset I is the itemset with minimum support count

  $\sup(I) >= minsup$

- Apriori Property: Any subset of frequent itemset must be frequent.
- Q: Why is it so?
- A: Example: Suppose {A,B} is frequent. Since each occurrence of A,B includes both A and B, then both A and B must also be frequent
- Similar argument for larger itemsets
- Almost all association rule algorithms are based on this subset property

Definitions

- 1-itemset = items with 1 item …
  {A},{B}, {C},…
- 2-itemset = itemsets with 2 items …
  {A,B},{B,C}, {C,D},…
- k-itemset = itemsets with k items
- candidate itemset = itemset that may have support > minimum support threshold
- frequent itemset = candidate itemset with a support higher than a certain threshold (minimum support threshold)
  - Sometimes frequent itemsets are also called as large itemsets.

The APriori algorithm, Agrawal et al (1993)

- "Generate and Test" or "Candidate Generation"
- To find associations rules, a simple two step approach is used:
  - Step 1 - discover all frequent items that have support above the minimum support required
  - Step 2 - Use the set of frequent items to generate the association rules that have high enough confidence level

Apriori Algorithm: Generating Frequent and Candidate Itemsets

- start with all 1-itemsets
- go through data and count their support and find all “frequent” 1-itemsets
- combine them to form “candidate” 2-itemsets
- go through data and count their support and find all “frequent” 2-itemsets
- combine them to form “candidate” 3-itemsets …

Apriori Algorithm: Generating Association Rules

- Only strong association rules are generated.
  - Frequent itemsets satisfying minimum support threshold are considered.
  - Strong rules are those that satisfy minimum confidence threshold.

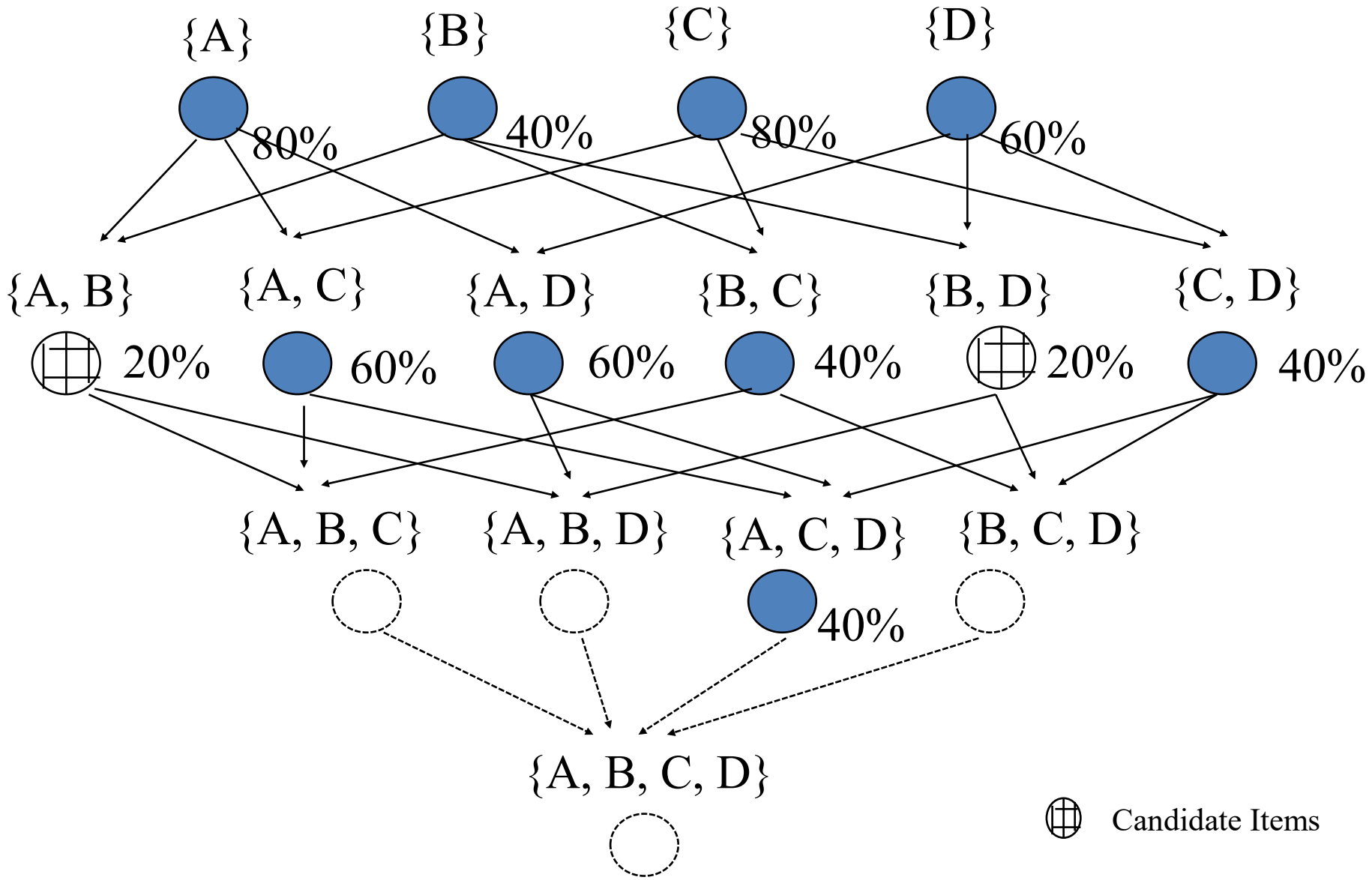
For each frequent itemset, **f**, generate all non-empty subsets of **f**
For every non-empty subset **s** of **f** **do**
    if support(**f**)/support(**s**) ≥ min_confidence **then**
        output rule **s** ==> **(f-s)**
end

Generating Association Rules: An example

Assume that the minimum support and confidence threshold is at 40%.

- **Candidate itemsets** are itemsets satisfying the condition that any subset of the candidate itemset must also have minimum support.
- **Frequent itemsets** are itemsets satisfying the condition that the itemset must have minimum support.

Transaction Id	Items Bought
001	{A,C,D}
002	{A,B,C,D}
003	{B,C}
004	{A, D}
005	{A,C}



Minimum Support &
Confidence Threshold = 40%

Association Rules: Example (cont)

- First generate all nonempty subsets of {A, C, D} and use each of it on the LHS and remaining symbols on the RHS.
  - For example, subsets of {A, C, D} are A, C, D, AC, AD, CD.

- The possible rules therefore are
  A => CD, C => AD, D => AC,
  AC => D, AD => C, CD => A, and some more…
- Confidence of each rules is calculated.
- Rules satisfying minimum confidence threshold are included.

Association Rules: Example (cont)

- Example association rules for this data set are:

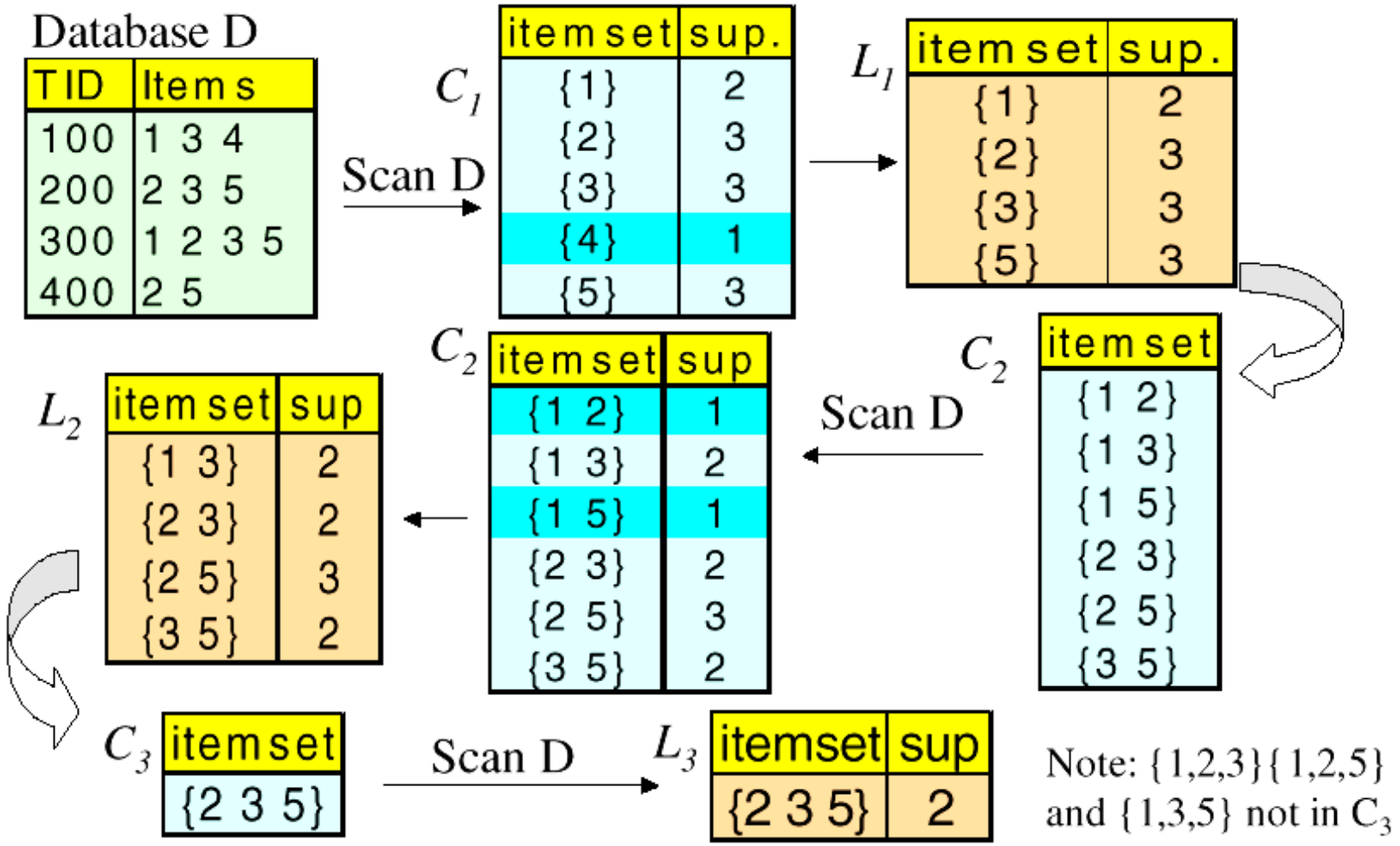
  A,C => D, A,D => C, C,D => A,

  A => C,D, C => A,D, D => A,C

  A => C, A => D, C => A, C => D, D => A, D => C,

  B => C, C => B

Another Example – Generating Frequent and Candidate Itemsets



C_1, C_2,C_3: Candidate Itemsets
L_1, L_2, L_3: Frequent Itemsets

Minimum Support: 50%

Bottleneck of Frequent-pattern Mining

- Apriori scans an entire transaction database for every round of support counting
  - Multiple database scans are costly
- Mining long patterns needs many passes of scanning and generates lots of candidates
  - To find frequent itemset $i_1 i_2 \ldots i_{100}$
    - # of scans: 100
    - # of Candidates: $\binom{1}{100} + \binom{2}{100} + \ldots + \binom{100}{100} = 2^{100}-1 = 1.27*10^{30}$!
  - Huge number of candidates generated
    - Many candidates might have low support, or do not even exist in the database
  - Tedious workload of support counting for candidates
- Bottleneck: candidate-generation-and-test
- Can we avoid candidate generation?

Association Rule Mining

FP-Tree Mining Algorithm
Prefix-tree

Mining Frequent Patterns Without Candidate Generation

- Grow long patterns from short ones using local frequent items
  - "abc" is a frequent pattern
  - Get all transactions having "abc": DB|abc
  - "d" is a local frequent item in DB|abc → abcd is a frequent pattern

Construct FP-tree From A Transaction Database

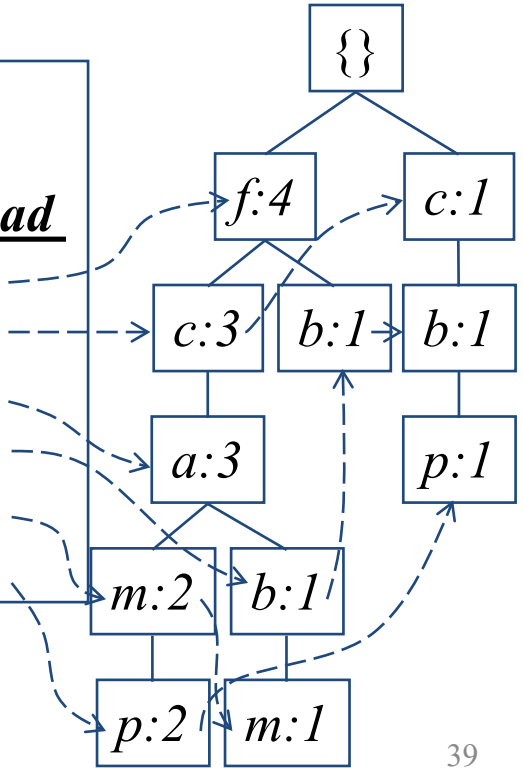
TID	Items bought	(ordered) frequent items
100	{f, a, c, d, g, i, m, p}	{f, c, a, m, p}
200	{a, b, c, f, l, m, o}	{f, c, a, b, m}
300	{b, f, h, j, o, w}	{f, b}
400	{b, c, k, s, p}	{c, b, p}
500	{a, f, c, e, l, p, m, n}	{f, c, a, m, p}

min_support = 3

1. Scan DB once, find frequent 1-itemset (single item pattern)
2. Sort frequent items in frequency descending order, f-list
3. Create a Header Table with f-list items that have support =>min_support
4. Scan DB again, construct FP-tree

Header Table

Item	frequency	head
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	



Benefits of the FP-tree Structure

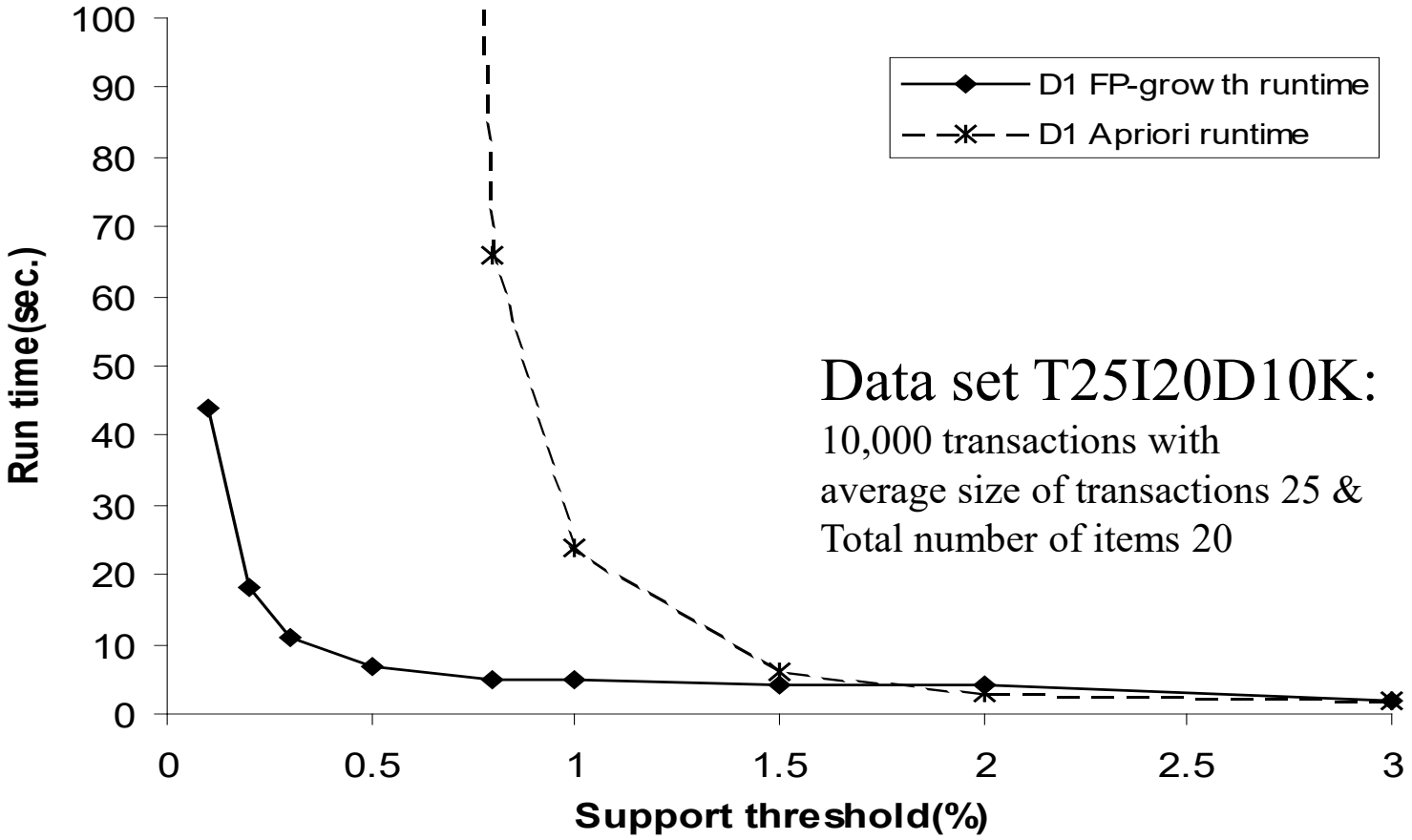
- Completeness
  - Preserve complete information for frequent pattern mining
- Compactness
  - Reduce irrelevant info—infrequent items are gone
  - Items in frequency descending order: the more frequently occurring, the more likely to be shared
  - Never be larger than the original database (not count node-links and the *count* field)

FP-Growth: Summary and Discussion

- Divide-and-conquer:
  - decompose both the mining task and DB according to the frequent patterns obtained so far
  - 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
    - only 2 passes over the data-set

➢ Disadvantages of FP-Growth
  - FP-Tree may not fit in memory!!
  - FP-Tree is expensive to build

FP-Growth vs. Apriori: Scalability With the Support Threshold



Sequential Pattern Discovery

Sequential Pattern Discovery

- Detects patterns between transactions such that the presence of one set of items is followed by another set of items in a database of transactions
- The concept of support factor is important
  - Indicating the relative occurrence of the detected sequential patterns within the overall transactions
  - the number of customers supporting the sequence/the total number of customers.
- Basic Apriori property still applies.
  - Any subset of frequent itemset must be frequent.
  - If a sequence S is not frequent then none of the super-sequences of S is frequent

    E.g, <hb> is infrequent → so do <hab> and <(ah)b>

Sequential Pattern Discovery: Example

Customer	*Trscation Time*	*Items Bought*
B. Adams	21/03/00, 5:27 pm	Beer
B. Adams	22/03/00, 10:34 an	Brandy
J. Brown	20/03/00, 10:13 an	Juice, Coke
J. Brown	20/03/00, 11:47 an	Beer
J. Brown	29/03/00, 9:22 am	Wine, Cider
J. Mitchell	21/03/00, 3:19 pm	Beer, Gin, Cider
B. Moore	01/03/00, 2:32 pm	Beer
B. Moore	06/03/00, 6:17 pm	Wine, Cider
B. Moore	23/03/00, 5:03 pm	Brandy
F. Zappa	07/03/00, 11:02 pn	Brandy

Sequential Pattern Discovery: Transaction Database

Sequential Pattern Discovery: Example

B. Adams	(Beer) (Brandy)
J. Brown	(Juice, Coke) (Beer) (Wine, Cider)
J. Mitchell	(Beer, Gin, Cider)
B. Moore	(Beer)(Wine, Cider)(Brandy)
F. Zappa	(Brandy)

Sequential Pattern Discovery: Customer Sequence

Supporting Customers	***Resulting Patterns***
B. Adams, B. Moore	(Beer) (Brandy)
J. Brown, B. Moore	(Beer) (Wine, Cider)

Sequential Pattern Discovery:
Support => 40 %

Association Mining: Summary

- Important task in data mining
- **Association Discovery**
  - Looks for links between records (items) in a data set (the same event).
  - Example: When a customer rents property for more than 2 years and is more than 25 years old, in 40% of cases, the customer will buy a property. Association happens in 35% of all customers who rent properties.
  - Apriori (itemsets, candidate generation & test)
  - Projection-based (FP-growth)
- **Sequential Pattern Discovery**
  - Looks for temporal links between purchases, rather than relationships between items in a single transaction.
  - Application: Used to understand long-term customer buying behavior.
  - Example: Within three months of buying property, new homeowners will purchase goods such as cookers, freezers, and washing machines.
  - GSP (sequences, candidate generation & test)
  - Projection-based (PrefixSpan)

References

- Data Mining techniques and concepts by Han J and Kamber M, 2011.
- Introduction to Data Mining by Tan, Steinbach and Kumar