Data Warehousing

Lecture 8 Frequent Itemset Mining and Association Rule Mining

CITS3401
CITS5504

Zeyi Wen

Computer Science and Software Engineering

School of Maths, Physics and Computing

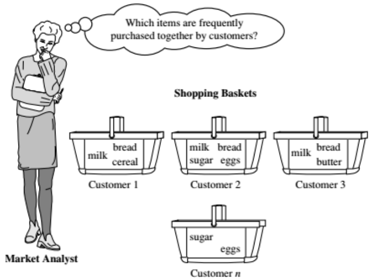
Acknowledgement: The lecture slides are based on online sources.

Lecture Outline

- Basic Concepts
- Closed Patterns and Max-Patterns
- Frequent Pattern Mining: Apriori Algorithm
- Association Rule Mining
- Challenges and Efficiency Improvement for Frequent Pattern Mining
- Other Advanced Topics in Frequent Pattern Mining

What is Pattern Analysis

- Pattern: a set of items, subsequences, substructures that occurs frequently together (or strongly correlated) in a data set
- Frequent pattern first proposed in the context of frequent itemsets and association rule mining
- Motivation examples:
  - What products were often purchased together?
  - What are the subsequent purchases after buying an iPad?
  - What code segments likely contain copy-and-paste bugs?
  - What word sequences likely form phrases in this corpus?



Why is Pattern Mining Important

THE UNIVERSITY OF WESTERN AUSTRALIA

- Frequent pattern: An intrinsic and important property of datasets.
- Uncovering patterns from massive data sets
- Foundation for many essential data mining tasks
  – Association, correlation, and causality analysis
  – Mining sequential, structural (e.g. sub-graph) patterns
  – Pattern analysis in spatiotemporal, multimedia, time-series, and stream data
  – Classification: discriminative pattern-based analysis
  – Cluster analysis: pattern-based sub-space clustering
- Broad applications
  – Basket data analysis, cross-marketing, catalog design, sale campaign analysis, Web log (click through rate) analysis, and DNA sequence analysis.

Basic Concepts: Frequent Patterns

- itemset: A set of one or more items
- k-itemset $X = \{x_1, \ldots, x_k\}$
  - 2-itemset, e.g. $X = \{Beer, Diaper\}$
- (absolute) support (count) of X: Frequency or occurrence of an itemset X
- (relative) support, is the fraction of transactions that contains X (i.e. the probability that a transaction contains X)

Tid	Items Bought
t1	Beer, Nuts, Diaper
t2	Beer, Coffee, Diaper
t3	Beer, Diaper, Eggs
t4	Nuts, Eggs, Milk
t5	Nuts, Coffee, Diaper, Eggs, Milk

Supports of Itermsets

THE UNIVERSITY OF WESTERN AUSTRALIA

- *(absolute) support (count)* of X, sup{X}: Frequency or the number of occurrences of an itemset X
  - Ex. sup{Beer} = 3
  - Ex. sup{Diaper} = 4
  - Ex. sup{Beer, Diaper} = 3
  - Ex. sup{Beer, Eggs} = 1

Tid	Items Bought
t1	Beer, Nuts, Diaper
t2	Beer, Coffee, Diaper
t3	Beer, Diaper, Eggs
t4	Nuts, Eggs, Milk
t5	Nuts, Coffee, Diaper, Eggs, Milk

- *(relative) support*, *s{X}:* The fraction of transactions that contains X (i.e. the probability that a transaction contains X)
  - Ex. s{Beer} = 3/5 = 60%
  - Ex. s{Diaper} = 4/5 = 80%
  - Ex. s{Beer, Eggs} = 1/5 = 20%

Basic Concepts: Frequent Patterns

- itemset: A set of one or more items
- k-itemset $X = \{x_1, \ldots, x_k\}$
  - 2-itemset, e.g. $X = \{Beer, Diaper\}$
- (absolute) support (count) of X: Frequency or occurrence of an itemset X
- (relative) support, is the fraction of transactions that contains X (i.e. the probability that a transaction contains X)
- An itemset X is frequent if X's support is no less than a $minsup$ threshold

Tid	Items Bought
t1	Beer, Nuts, Diaper
t2	Beer, Coffee, Diaper
t3	Beer, Diaper, Eggs
t4	Nuts, Eggs, Milk
t5	Nuts, Coffee, Diaper, Eggs, Milk

- **items:** Beer, Nuts, Diaper, Coffee, Eggs, Milk
- **Let** $minsup = \mathbf{50\%}$
- **Freq. 1-itemsets:**
  - Beer:3(60%); Nuts:3(60%); Diaper:4(80%); Eggs:3(60%)
- **Freq. 2-itemsets:**
  - {Beer, Diaper}:3(60%)

Basic Concepts: Association Rules

Find all the rules $X \implies Y$ with minimum support and confidence

- support, *s*, probability that a transaction contains $X \cup Y$

$$\mathbf{support(X \Longrightarrow Y) = P(X \cup \mathit{Y})}$$

- confidence, *c,* conditional probability that a transaction having X also contains Y

$$\mathbf{confidence(X \Longrightarrow \mathit{Y}) = \mathit{P(Y|X)} = \frac{\mathit{support(X \cup Y)}}{\mathit{support(X)}} = \frac{\mathit{support\_count(X \cup Y)}}{\mathit{support\_count(X)}}}$$

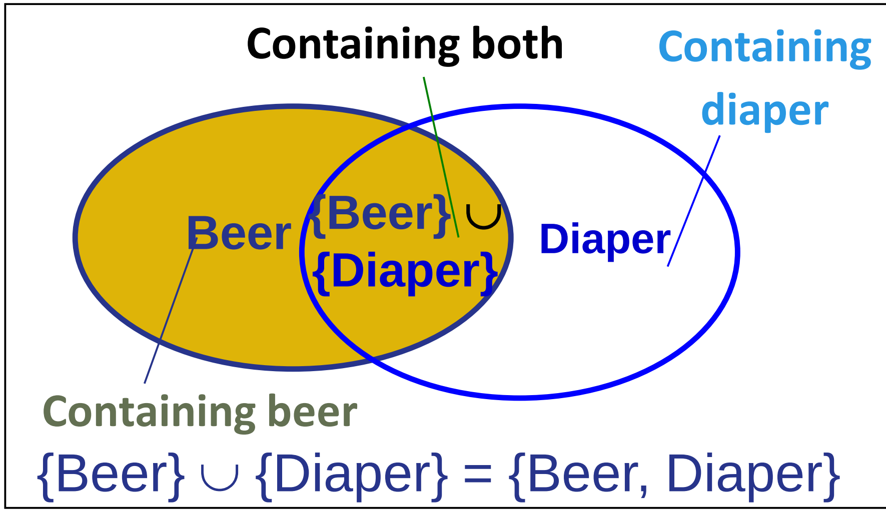
Basic Concepts: Association Rules

- **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

Let $minsup = 50\%, minconf = 50\%$

- ***Frequent itemsets:*** $\{Beer\}: 3, \{Nuts\}: 3, \{Diaper\}: 4, \{Eggs\}: 3$, $\{Beer, Diaper\}: 3$
- **Association rules: (many more…!)**
  - $Beer \implies Diaper \ \ (60\%, 100\%)$
  - $Diaper \implies Beer \ \ (60\%, 75\%)$

Tid	Items Bought
t1	Beer, Nuts, Diaper
t2	Beer, Coffee, Diaper
t3	Beer, Diaper, Eggs
t4	Nuts, Eggs, Milk
t5	Nuts, Coffee, Diaper, Eggs, Milk



Note: $X \cup Y$ is the union of two itemsets.
The set contains both X and Y.

Lecture Outline

- Basic Concepts
- Closed Patterns and Max-Patterns
- Frequent Pattern Mining: Apriori Algorithm
- Association Rule Mining
- Challenges and Efficiency Improvement for Frequent Pattern Mining
- Other Advanced Topics in Frequent Pattern Mining

There Are Too Many Frequent Patterns!

- **A long pattern contains a combinatorial number of sub-patterns**
- **How many frequent itemsets does the following TDB_1 contain?**
  - $TDB_{1:}$ T_1: {a_1, …, a_{50}}; T_2: {a_1, …, a_{100}}
  - Assuming (absolute) *minsup* = 1
  - Let's have a try

1-itemsets: {a_1}: 2, {a_2}: 2, …, {a_{50}}: 2, {a_{51}}: 1, …, {a_{100}}: 1,

2-itemsets: {a_1, a_2}: 2, …, {a_1, a_{50}}: 2, {a_1, a_{51}}: 1 …, …, {a_{99}, a_{100}}: 1,

…, …, …, …

99-itemsets: {a_1, a_2, …, a_{99}}: 1, …, {a_2, a_3, …, a_{100}}: 1

100-itemset: {a_1, a_2, …, a_{100}}: 1

- **The total number of frequent itemsets:**

$$\binom{100}{1} + \binom{100}{2} + \binom{100}{3} + \cdots + \binom{100}{100} = 2^{100} - 1$$

A too huge set for any one to compute or store!

Closed Patterns and Max-Patterns

- A long pattern contains a combinatorial number of sub-patterns, e.g. $\{a_1, \dots, a_{100}\}$ contains a large number of sub-patterns:
  - $\binom{100}{1} + \binom{100}{2} + \cdots + \binom{100}{100} = 2^{100} - 1 \approx 1.27 * 10^{30}$
  - Solution: *Mine closed patterns and max-patterns instead*
- A pattern (itemset) X is closed if X is *frequent* and there exists *no super-pattern* $Y \supset X$, *with the same support* as X.
- A pattern (itemset) X is a max-pattern if X is frequent and there exists no frequent super-pattern $Y \supset X$.
- Closed pattern is a lossless compression of freq. patterns
  - Reducing the # of patterns and rules

Expressing Patterns in Closed Patterns

- **Solution 1: Closed patterns:** A pattern (itemset) X is closed if X is *frequent* and there exists *no super-pattern* $Y \supset X$, *with the same support* as X.
  - Let Transaction DB TDB_1: T_1: $\{a_1, \ldots, a_{50}\}$; T_2: $\{a_1, \ldots, a_{100}\}$
  - Suppose *minsup* = 1. How many closed patterns does TDB_1 contain?
    - Two: P_1: "$\{a_1, \ldots, a_{50}\}$: 2"; P_2: "$\{a_1, \ldots, a_{100}\}$: 1"
- **Closed pattern is a lossless compression of frequent patterns**
  - Reduces the # of patterns but does not lose the support information!
  - You will still be able to say: "$\{a_2, \ldots, a_{40}\}$: 2", "$\{a_5, a_{51}\}$: 1"

Expressing Patterns in Max-Patterns

- **Solution 2: Max-patterns:** A pattern (itemset) X is a max-pattern if X is frequent and there exists no frequent super-pattern $Y \supset X$.
- **Difference from close-patterns?**
  - Do not care the real support of the sub-patterns of a max-pattern
  - Let Transaction DB TDB_1: T_1: $\{a_1, \ldots, a_{50}\}$; T_2: $\{a_1, \ldots, a_{100}\}$
  - Suppose *minsup* = 1. How many max-patterns does TDB_1 contain?
    - One: P: "$\{a_1, \ldots, a_{100}\}$: 1"
- **Max-pattern is a lossy compression!**
  - We only know $\{a_1, \ldots, a_{40}\}$ is frequent
  - But we do not know the real support of $\{a_1, \ldots, a_{40}\}$, …, any more!
- **Thus in many applications, mining close-patterns is more desirable than mining max-patterns**

Example: Closed Pattern and Maximal Freq. Pattern

- Suppose we have only two transactions and $\min\_\text{sup} = 1$:

TID	Items
t1	$\{a_1, a_2, \ldots, a_{100}\}$
t2	$\{a_1, a_2, \ldots, a_{50}\}$

- Then closed frequent itemsets (i.e. closed patterns) are:
  - $C = \{\{a_1, a_2, \ldots, a_{100}\}: 1, \{a_1, a_2, \ldots, a_{50}\}: 2\}$
- The maximal frequent itemset (i.e. max-patterns) is:
  - $M = \{\{a_1, a_2, \ldots, a_{100}\}: 1\}$

Closed Patterns

- An itemset X is closed if X is *frequent* and there exists *no super-pattern* $Y \supset X$, *with the same support* as X.

- An itemset is closed if none of its **immediate supersets** has the **same support** as the itemset.

TID	Items
t1	{A,B}
t2	{B,C,D}
t3	{A,B,C,D}
t4	{A,B,D}
t5	{A,B,C,D}

$$minsup = 2$$

Itemset	Support
{A}	4
{B}	5
{C}	3
{D}	4
{A,B}	4
{A,C}	2
{A,D}	3
{B,C}	3
{B,D}	4
{C,D}	3
{A,B,C}	2
{A,B,D}	3
{A,C,D}	2
{B,C,D}	3
{A,B,C,D}	2

Max-Pattern

- Difference from closed patterns
  - Do not care for the real support of the sub-patterns of a max-pattern
- Max-pattern: frequent patterns without proper frequent super pattern
  - BCDE, ACD are max-patterns
  - BCD is not a max-pattern

$$minsup = 2$$

TID	Items
t1	{A,B,C,D,E}
t2	{B,C,D,E}
t3	{A,C,D,F}

- An itemset X is a max-pattern if X is frequent and there exists no frequent super-pattern $Y \supset X$.

Max-Pattern Question

THE UNIVERSITY OF WESTERN AUSTRALIA

- An itemset X is a max-pattern if X is frequent and there exists no frequent super-pattern $Y \supset X$.
- Which is a max-pattern?
- Is there always only one max-pattern for any data sets?

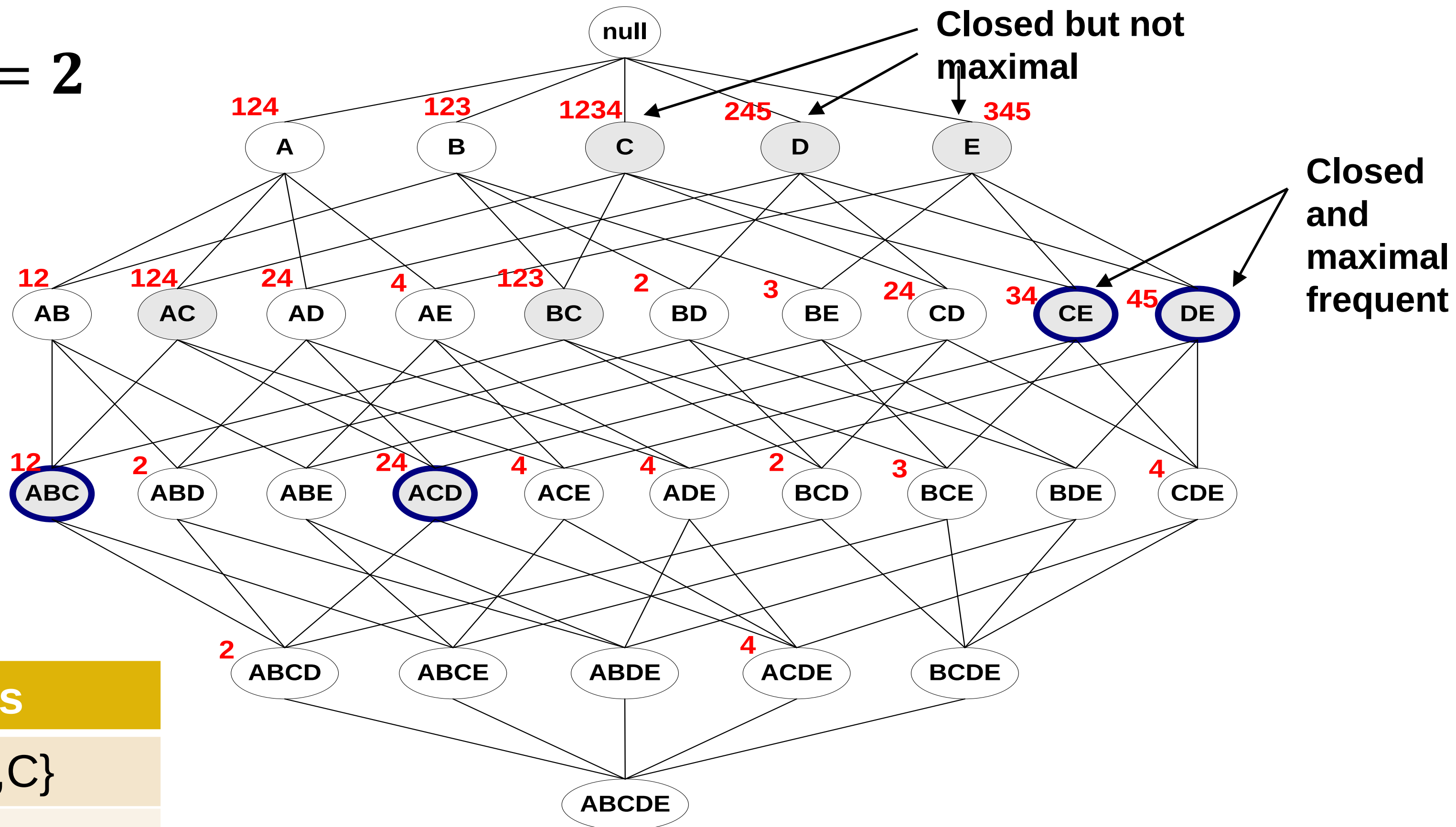
TID	Items
t1	{A,B}
t2	{B,C,D}
t3	{A,B,C,D}
t4	{A,B,D}
t5	{A,B,C,D}

$$minsup = 2$$

Itemset	Support
{A}	4
{B}	5
{C}	3
{D}	4
{A,B}	4
{A,C}	2
{A,D}	3
{B,C}	3
{B,D}	4
{C,D}	3
{A,B,C}	2
{A,B,D}	3
{A,C,D}	2
{B,C,D}	3
{A,B,C,D}	2

Maximal vs. Closed Frequent Itemsets

$minsup = 2$



TID	Items
t1	{A,B,C}
t2	{A,B,C,D}
t3	{B,C,E}
t4	{A,C,D,E}
t5	{D,E}

\# Closed = 9

\# Maximal = 4

Property of Closed Patterns

THE UNIVERSITY OF WESTERN AUSTRALIA

- Closed Patterns are Lossless: the support for any frequent itemset can be deduced from the closed frequent itemsets.
- The itemsets in black can be derived from the closed itemsets in red.

$$minsup = 2$$

Itemset	Support
{A}	4
{B}	5
{C}	3
{D}	4
{A,B}	4
{A,C}	2
{A,D}	3
{B,C}	3
{B,D}	4
{C,D}	3
{A,B,C}	2
{A,B,D}	3
{A,C,D}	2
{B,C,D}	3
{A,B,C,D}	2

Property of Max-pattern

- Max-pattern is a lossy compression. We only know all its subsets are frequent but not the real support.

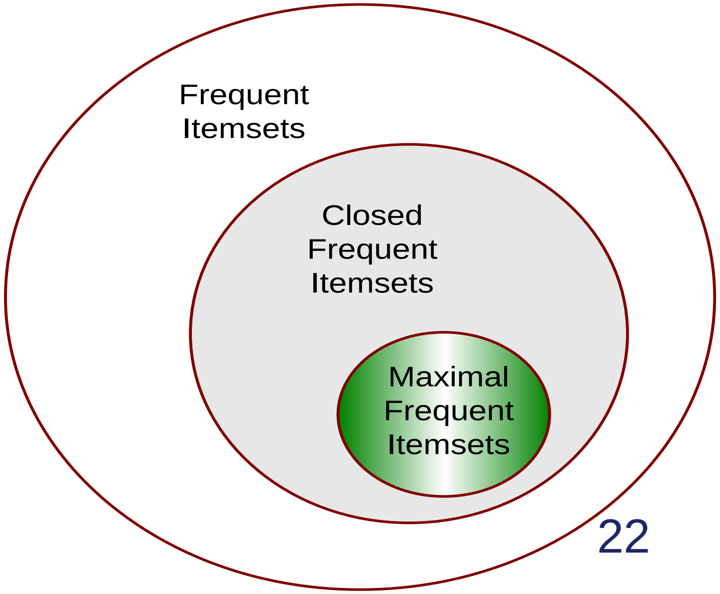
- Max-pattern: frequent patterns without proper frequent super pattern
  - BCDE, ACD are max-patterns
  - BCD is not a max-pattern

$$minsup = 2$$

TID	Items
t1	{A,B,C,D,E}
t2	{B,C,D,E}
t3	{A,C,D,F}

Max vs. Closed Patterns

- Closed Patterns are <u>Lossless</u>: the support for any frequent itemset can be deduced from the closed frequent itemsets.
- Max-pattern is a lossy compression. We only know all its subsets are frequent but not the real support.
- Thus in many applications, mining closed-patterns is more desirable than mining max-patterns.



Lecture Outline

- Basic Concepts
- Closed Patterns and Max-Patterns
- Frequent Pattern Mining: Apriori Algorithm
- Association Rule Mining
- Challenges and Efficiency Improvement for Frequent Pattern Mining
- Other Advanced Topics in Frequent Pattern Mining

How to mine frequent itemsets?

The Downward Closure Property

- Observation: From $TDB_{1:}$ T_1: $\{a_1, …, a_{50}\}$; T_2: $\{a_1, …, a_{100}\}$
  - We get a frequent itemset: $\{a_1, …, a_{50}\}$
  - Also, its subsets are all frequent: $\{a_1\}$, $\{a_2\}$, …, $\{a_{50}\}$, $\{a_1, a_2\}$, …, $\{a_1, …, a_{49}\}$, …
  - There must be some hidden relationships among frequent patterns!
- The downward closure (also called "Apriori") property of frequent patterns
  - If **{beer, diaper, nuts}** is frequent, so is **{beer, diaper}**
  - Every transaction containing {beer, diaper, nuts} also contains {beer, diaper}
  - Apriori: Any subset of a frequent itemset must be frequent
- **Efficient mining methodology**
  - If any subset of an itemset S is infrequent, then there is no chance for S to be frequent—why do we even have to consider S!?

A sharp knife for pruning!

Key Observation (monotonicity)

- Any subset of a frequent itemset must also be frequent: Downward closure property (also called Apriori propery)
  - If {beer, diaper, nuts} is frequent, so is {beer, diaper}
- Efficient mining methodology: Apriori pruning principle
  - Any superset of an infrequent itemset must also be infrequent.
  - If any subset of an itemset S is infrequent, then there is no chance for S to be frequent—we don't need to consider S!

Apriori: A Candidate Generation & Test Approach

- Outline of Apriori
  - level-wise, candidate generation and testing
- Method:
  1. Initially, scan the database once to get frequent 1-itemset; k=1
  2. Repeat
     a) Generate length (k+1) candidate itemsets from length k frequent itemsets
     b) Test the candidates against the database to find frequent (k+1) itemsets
     c) Set k=k+1
  3. Terminate when no frequent or candidate set can be generated
  4. Return all the frequent itemsets

The Apriori Algorithm (Pseudo-Code)

C_k : candidate k-itemsets
F_k : frequent k-itemsets

$k = 1$;
$F_1 =$ {frequent items}; //frequent 1-itemset

while (F_k ! $= \emptyset$) do{ //when F_k is not empty
 /** candidates generation **/
 $C_{k+1} =$ {candidates generated from F_k};

 /** F_{k+1} = candidates in C_{k+1} with minsup **/
 Derive F_{k+1} by counting candidates in C_{k+1} w.r.t. DB at $minsup$;
 $k = k + 1$;
}
return $\cup_k L_k$;

The Apriori Algorithm—An Example

THE UNIVERSITY OF WESTERN AUSTRALIA

minsup = 2

Tid	Items
10	A, C, D
20	B, C, E
30	A, B, C, E
40	B, E

1st scan → C_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{D}	1
{E}	3

→ F_1

Itemset	sup
{A}	2
{B}	3
{C}	3
{E}	3

C_2

Itemset
{A, B}
{A, C}
{A, E}
{B, C}
{B, E}
{C, E}

2nd scan → C_2

Itemset	sup
{A, B}	1
{A, C}	2
{A, E}	1
{B, C}	2
{B, E}	3
{C, E}	2

→ F_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

C_3

Itemset
{B, C, E}

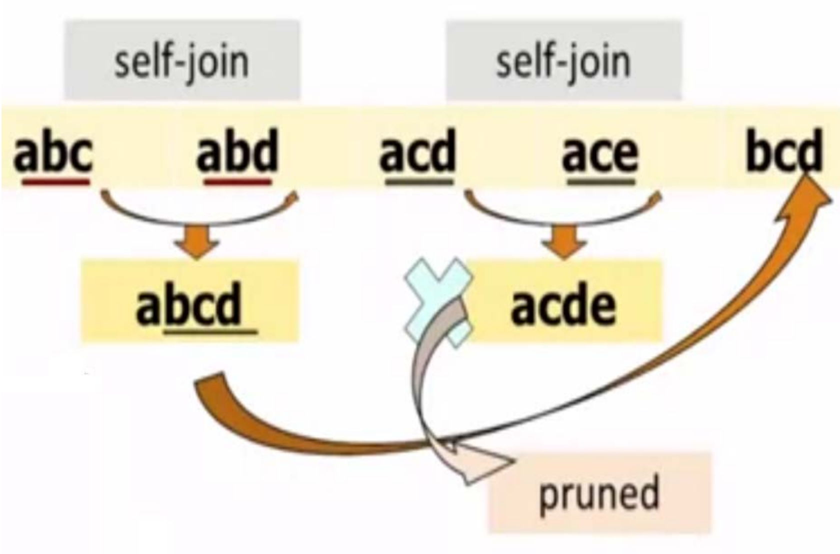
3rd scan → F_3

Itemset	sup
{B, C, E}	2

Self-join: members of F_{k-1} are joinable if their first (k-2) items are in common

Apriori Implementation Trick

- **How to generate candidates?**
  - **Step 1**: self-joining F_k
  - **Step 2:** pruning
- **Example of Candidate-generation**
  - F_3={*abc, abd, acd, ace, bcd*}
  - Self-joining: $F_3 * F_3$
    - *abcd* from *abc* and *abd*
    - *acde* from *acd* and *ace*
  - Pruning:
    - *acde* is removed because *ade* is not in F_3
  - C_4 = {*abcd*}



Any (*k*-1)-itemset that is not frequent cannot be a subset of a frequent *k*-itemset

Another Example (minsup=2)

Transactional Data for an *AllElectronics* Branch

TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Scan D for count of each candidate →

C_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Compare candidate support count with minimum support count →

L_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Generate C_2 candidates from L_1 →

C_2

Itemset
{I1, I2}
{I1, I3}
{I1, I4}
{I1, I5}
{I2, I3}
{I2, I4}
{I2, I5}
{I3, I4}
{I3, I5}
{I4, I5}

Scan D for count of each candidate →

C_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I4}	1
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2
{I3, I4}	0
{I3, I5}	1
{I4, I5}	0

Compare candidate support count with minimum support count →

L_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

$$\{I1, I2\} \Rightarrow I5, \quad confidence = 2/4 = 50\%$$
$$\{I1, I5\} \Rightarrow I2, \quad confidence = 2/2 = 100\%$$
$$\{I2, I5\} \Rightarrow I1, \quad confidence = 2/2 = 100\%$$
$$I1 \Rightarrow \{I2, I5\}, \quad confidence = 2/6 = 33\%$$
$$I2 \Rightarrow \{I1, I5\}, \quad confidence = 2/7 = 29\%$$
$$I5 \Rightarrow \{I1, I2\}, \quad confidence = 2/2 = 100\%$$

Generate C_3 candidates from L_2 →

C_3

Itemset
{I1, I2, I3}
{I1, I2, I5}

Scan D for count of each candidate →

C_3

Itemset	Sup. count
{I1, I2, I3}	2
{I1, I2, I5}	2

Compare candidate support count with minimum support count →

L_3

Itemset	Sup. count
{I1, I2, I3}	2
{I1, I2, I5}	2

Self-Join and Pruning

(a) Join: $C_3 = L_2 \bowtie L_2 = \{\{I1, I2\}, \{I1, I3\}, \{I1, I5\}, \{I2, I3\}, \{I2, I4\}, \{I2, I5\}\}$
$\bowtie \{\{I1, I2\}, \{I1, I3\}, \{I1, I5\}, \{I2, I3\}, \{I2, I4\}, \{I2, I5\}\}$
$= \{\{I1, I2, I3\}, \{I1, I2, I5\}, \{I1, I3, I5\}, \{I2, I3, I4\}, \{I2, I3, I5\}, \{I2, I4, I5\}\}$.

(b) Prune using the Apriori property: All nonempty subsets of a frequent itemset must also be frequent. Do any of the candidates have a subset that is not frequent?

- The 2-item subsets of {I1, I2, I3} are {I1, I2}, {I1, I3}, and {I2, I3}. All 2-item subsets of {I1, I2, I3} are members of L_2. Therefore, keep {I1, I2, I3} in C_3.
- The 2-item subsets of {I1, I2, I5} are {I1, I2}, {I1, I5}, and {I2, I5}. All 2-item subsets of {I1, I2, I5} are members of L_2. Therefore, keep {I1, I2, I5} in C_3.
- The 2-item subsets of {I1, I3, I5} are {I1, I3}, {I1, I5}, and {I3, I5}. {I3, I5} is not a member of L_2, and so it is not frequent. Therefore, remove {I1, I3, I5} from C_3.
- The 2-item subsets of {I2, I3, I4} are {I2, I3}, {I2, I4}, and {I3, I4}. {I3, I4} is not a member of L_2, and so it is not frequent. Therefore, remove {I2, I3, I4} from C_3.
- The 2-item subsets of {I2, I3, I5} are {I2, I3}, {I2, I5}, and {I3, I5}. {I3, I5} is not a member of L_2, and so it is not frequent. Therefore, remove {I2, I3, I5} from C_3.
- The 2-item subsets of {I2, I4, I5} are {I2, I4}, {I2, I5}, and {I4, I5}. {I4, I5} is not a member of L_2, and so it is not frequent. Therefore, remove {I2, I4, I5} from C_3.

(c) Therefore, $C_3 = \{\{I1, I2, I3\}, \{I1, I2, I5\}\}$ after pruning.

Lecture Outline

- Basic Concepts
- Closed Patterns and Max-Patterns
- Frequent Pattern Mining: Apriori Algorithm
- Association Rule Mining
- Challenges and Efficiency Improvement for Frequent Pattern Mining
- Other Advanced Topics in Frequent Pattern Mining

Mining Association Rules

- For each frequent itemset F, generate all nonempty subsets of F.
- For every nonempty subset s of F, output the rule "$s \Rightarrow (F - s)$" if $\frac{support\_count(F)}{support\_count(s)} \geq \min\_conf$
- Example
  - Frequent itemset $F = \{I1, I2, I5\}$
  - Nonempty subset $\{I1, I2\}, \{I2, I5\}, \{I1, I5\}, \{I1\}, \{I2\}, \{I5\}$

TID	*List of item_IDs*
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

$$\{I1, I2\} \Rightarrow I5, \quad confidence = 2/4 = 50\%$$
$$\{I1, I5\} \Rightarrow I2, \quad confidence = 2/2 = 100\%$$
$$\{I2, I5\} \Rightarrow I1, \quad confidence = 2/2 = 100\%$$
$$I1 \Rightarrow \{I2, I5\}, \quad confidence = 2/6 = 33\%$$
$$I2 \Rightarrow \{I1, I5\}, \quad confidence = 2/7 = 29\%$$
$$I5 \Rightarrow \{I1, I2\}, \quad confidence = 2/2 = 100\%$$

Strong Rules are not necessarily interesting

- Let $game$ refer to the transactions containing computer games, and $video$ refer to those containing videos.
- Of the 10,000 transactions analysed,
  - 6,000 of the customer transactions included computer games,
  - 7,500 included videos, and
  - 4,000 included both computer games and videos.
- $minsup = 30\%$ and $minconf = 60\%$

$$buys(X, \text{“computer games”}) \Rightarrow buys(X, \text{“videos”})$$

$$[support = 40\%, confidence = 66\%].$$

But p(videos) = 75%

Association to Correlation Analysis

$$A \Rightarrow B \,[support,\, confidence,\, correlation].$$

- **Lift**
  - Assesses the degree to which the occurrence of one “lifts” the occurrence of the other.
  - Computed by:

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

  - If $lift < 1, then\ occurence\ of\ A\ is\ negatively\ correlated\ with\ B$;
  - If $lift > 1, then\ occurence\ of\ A\ is\ positively\ correlated\ with\ B$;
  - If $lift = 1, then\ occurence\ of\ A\ is\ independent\ of\ B$;

$$P(\{game, video\})/(P(\{game\}) \times P(\{video\})) = 0.40/(0.60 \times 0.75) = 0.89$$

Lecture Outline

- Basic Concepts
- Closed Patterns and Max-Patterns
- Frequent Pattern Mining: Apriori Algorithm
- Association Rule Mining
- Challenges and Efficiency Improvement for Frequent Pattern Mining
- Other Advanced Topics in Frequent Pattern Mining

Challenges of Frequent Pattern Mining

- **Challenges**
  - Multiple scans of transaction database
  - Huge number of candidates
  - Tedious workload of support counting for candidates

Scan D for count of each candidate →

C_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Compare candidate support count with minimum support count →

L_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Generate C_2 candidates from L_1 →

C_2

Itemset
{I1, I2}
{I1, I3}
{I1, I4}
{I1, I5}
{I2, I3}
{I2, I4}
{I2, I5}
{I3, I4}
{I3, I5}
{I4, I5}

Scan D for count of each candidate →

C_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I4}	1
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2
{I3, I4}	0
{I3, I5}	1
{I4, I5}	0

Compare candidate support count with minimum support count →

L_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

Challenges of Frequent Pattern Mining

- **Challenges**
  - Multiple scans of transaction database
  - Huge number of candidates
  - Tedious workload of support counting for candidates
- **Improving Apriori: general ideas**
  - Reduce passes of transaction database scans
  - Shrink number of candidates
  - Facilitate support counting of candidates

Scan D for count of each candidate →

C_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Compare candidate support count with minimum support count →

L_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

Generate C_2 candidates from L_1 →

C_2

Itemset
{I1, I2}
{I1, I3}
{I1, I4}
{I1, I5}
{I2, I3}
{I2, I4}
{I2, I5}
{I3, I4}
{I3, I5}
{I4, I5}

Scan D for count of each candidate →

C_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I4}	1
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2
{I3, I4}	0
{I3, I5}	1
{I4, I5}	0

Compare candidate support count with minimum support count →

L_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

Apriori: Improvements and Alternatives

- **Reduce passes of transaction database scans**
  - Partitioning (e.g. Savasere, et al., 1995)
  - Dynamic itemset counting (DIC) (Brin, et al.,1997)
- **Shrink the number of candidates**
  - Hash-based technique (e.g., DHP: Park, et al., 1995)
  - Transaction reduction (e.g., Bayardo 1998)
  - Sampling (e.g., Toivonen, 1996)

Transaction Reduction

- Any transaction that does not contain any frequent k-itemsets cannot contain any frequent (k+1)-itemsets and such a transaction may be marked or removed.

- Frequent items F_1 are {A}, {B}, {D}, {M}, {T}. We are not able to use these to eliminate any transactions since all transactions have at least one of the items in F_1.
- The frequent 2-itemsets C_2 are $\{A, B\}$ and $\{B, M\}$. How can we reduce transactions using these?

TID	Items bought
001	B, M, T, Y
002	B, M
003	T, S, P
004	A, B, C, D
005	A, B
006	T, Y, E
007	A, B, M
008	B, C, D, T, P
009	D, T, S
010	A, B, M

Sampling [Toivonen, 1995]

- A random sample (usually large enough to fit in the main memory) may be obtained from the overall set of transactions and the sample is searched for frequent itemsets. These frequent itemsets are called sample frequent itemsets.

- Not guaranteed to be accurate but we sacrifice accuracy for efficiency. A lower support threshold may be used for the sample to ensure not missing any frequent datasets.

- Sample size is small such that the search for frequent itemsets for the sample can be done in main memory.

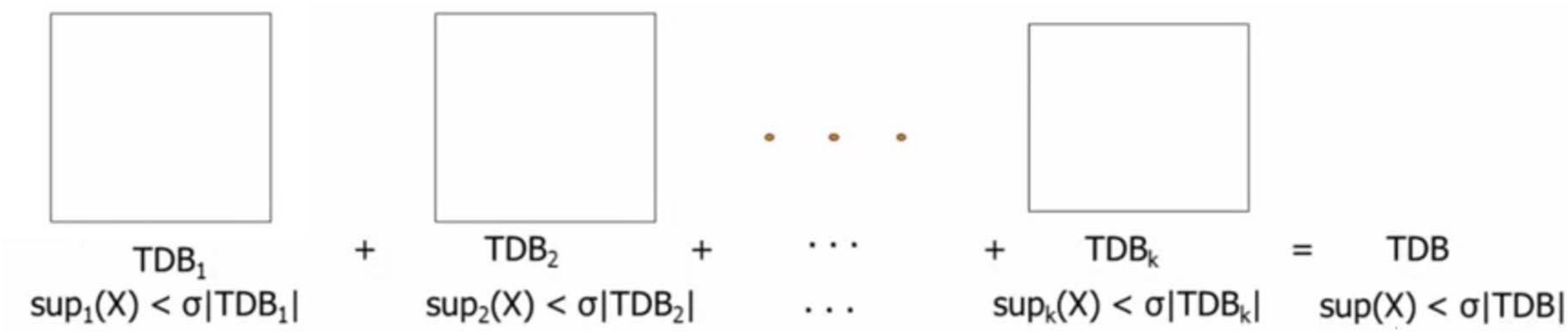
Lecture Outline

- Basic Concepts
- Closed Patterns and Max-Patterns
- Frequent Pattern Mining: Apriori Algorithm
- Association Rule Mining
- Challenges and Efficiency Improvement for Frequent Pattern Mining
- Other Advanced Topics in Frequent Pattern Mining

*** The rest of the slides in this lecture are optional material and not examinable**

Partitioning : Scan Database Only Twice

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



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

Direct Hashing & Pruning (DHP)

- When generating L1, the algorithm also generates all the 2-itemsets for each transaction, hashes them to a hash table and keeps a count.

TID	Items
100	Bread, Cheese, Eggs, Juice
200	Bread, Cheese, Juice
300	Bread, Milk, Yogurt
400	Bread, Juice, Milk
500	Cheese, Juice, Milk

Bit vector	Bucket number	Count	Pairs	C_2
1	0	3	(C, J) (B, Y) (M, Y)	(C, J)
0	1	1	(C, M)	
0	2	1	(E, J)	
0	3	0		
0	4	2	(B, C)	
1	5	3	(B, E) (J, M)	(J, M)
1	6	3	(B, J)	(B, J)
1	7	3	(C, E) (B, M)	(B, M)

100	(B, C) (B, E) (B, J) (C, E) (C, J) (E, J)
200	(B, C) (B, J) (C, J)
300	(B, M) (B, Y) (M, Y)
400	(B, J) (B, M) (J, M)
500	(C, J) (C, M) (J, M)

Park, Jong Soo, Ming-Syan Chen, and Philip S. Yu. "An effective hash-based algorithm for mining association rules." *Acm sigmod record* 24, no. 2 (1995): 175-186.

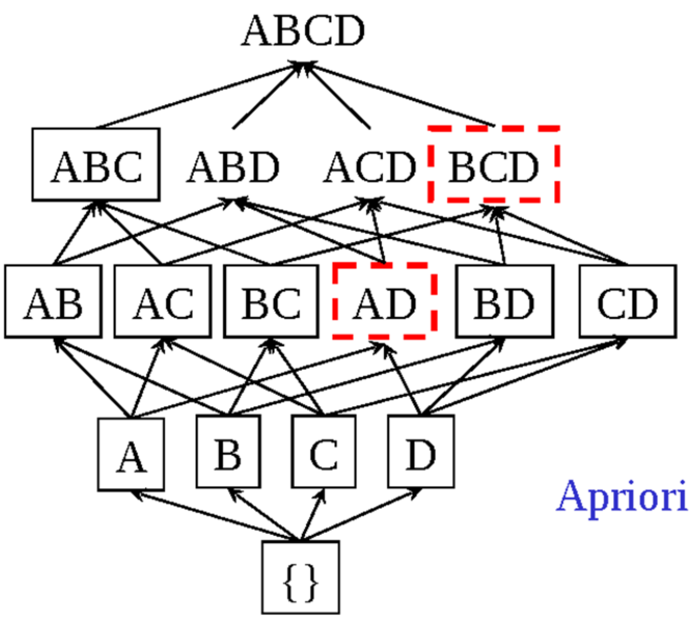
Hash Function Used

- For each pair, a numeric value is obtained by first representing B by 1, C by 2, E 3, J 4, M 5 and Y 6. Now each pair can be represented by a two digit number, for example (B, E) by 13 and (C, M) by 26.
- The two digits are then coded as modulo 8 number (dividing by 8 and using the remainder). This is the bucket address.
- A count of the number of pairs hashed is kept. Those addresses that have a count above the support value have the bit vector set to 1 otherwise 0.
- All pairs in rows that have zero bit are removed.

Dynamic Itemset Counting

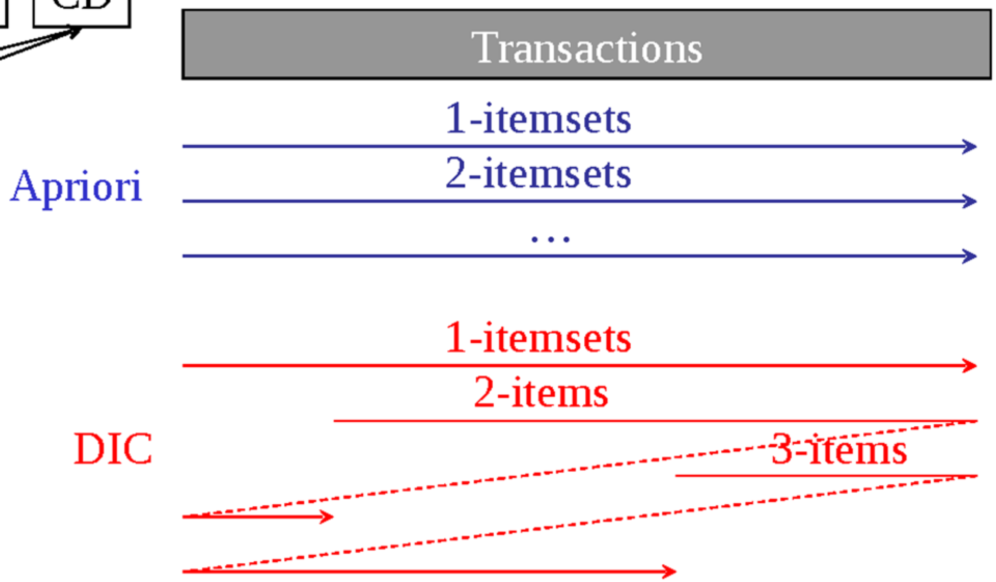
- ***Interrupt algorithm after every M transactions while scanning.***
- Itemsets which are already frequent are **combined** in pairs to generate higher order itemsets.
- The technique is dynamic in that, it starts estimating support for all the itemsets if all of their subsets are already found frequent.
- The resulting algorithm requires fewer database scans than Apriori.

DIC: Reduce Number of Scans



Itemset lattice

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



Scalable Mining Methods

- **Three major approaches**
  - Level-wise, join-based approach: Apriori (Agrawal & Srikant@VLDB'94)
  - Freq. pattern projection and growth (FPgrowth—Han, Pei & Yin @SIGMOD'00)
  - Vertical data format approach (Eclat—Zaki , Parthasarathy Ogihara, Li @KDD'97)

Customised Min-Supports for Different Kinds of Items

- We have used the same min-support threshold for all the items or item sets to be mined in each association mining
- In reality, some items (e.g., diamond, watch, …) are valuable but less frequent
- It is necessary to have customised min-support settings for different kinds of items
- One Method: Use group-based "individualised" min-support
  – E.g., {diamond, watch}: 0.05%; {bread, milk}: 5%; …
  – (aside) How to mine such rules efficiently?
    - Existing scalable mining algorithms can be easily extended to cover such cases

Rare Patterns vs. Negative Patterns

THE UNIVERSITY OF WESTERN AUSTRALIA

- **Rare patterns**
  - Very low support but interesting (e.g., buying Rolex watches)
  - How to mine them? Setting individualised, group-based min-support thresholds for different groups of items
- **Negative patterns**
  - Negatively correlated: Unlikely to happen together
  - Ex.: Since it is unlikely that the same customer buys both a Ford Expedition (an SUV car) and a Ford Fusion (a hybrid car), buying a Ford Expedition and buying a Ford Fusion are likely negatively correlated patterns
  - How to define negative patterns?

Summary

- Frequent patterns
- Closed patterns and Max-patterns
- Apriori algorithm for mining frequent patterns
- Association Rule Mining
- (Aside) Improving the efficiency of apriori: Partitioning, DHP, DIC

Reference

- Han et al.'s book
  - The lecture content is mainly based on Chapter 6.
  - Chapter 7 contains advanced techniques in pattern mining.
- Readings
  - The story of "Beer and Diaper".

Copyright Notice

Copyright Notice

Material used in this recording may have been reproduced and communicated to you by or on behalf of **The University of Western Australia** in accordance with section 113P of the *Copyright Act 1968*.

Unless stated otherwise, all teaching and learning materials provided to you by the University are protected under the Copyright Act and is for your personal use only. This material must not be shared or distributed without the permission of the University and the copyright owner/s.