Data Warehousing

Lecture 5 Frequent Itemset Mining and Association Rule Mining

CITS3401
CITS5504

Dr. Sirui Li

Computer Science and Software Engineering

School of Physics Mathematics and Computing

Acknowledgement: The lecture slides are based on online sources.

Lecture Outline

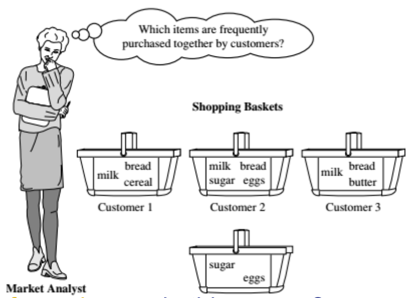
- Basic Concepts
- Closed Patterns and Max-Patterns
- Frequent Pattern Mining: Apriori Algorithm
- Association Rule Mining
- Challenges and Efficiency Improvement for Frequent Pattern Mining

Data Warehouse Usage

- **Three kinds of data warehouse applications**
  - Information processing
    - supports querying, basic statistical analysis, and reporting using crosstabs, tables, charts and graphs
  - Analytical processing
    - multidimensional analysis of data warehouse data
    - supports basic OLAP operations, slice-dice, drilling, pivoting
  - Data mining
    - knowledge discovery from hidden patterns
    - supports associations, constructing analytical models, performing classification and prediction, and presenting the mining results using visualization tools.
- **Differences among the three tasks**

What is Pattern Analysis

- Frequent 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 word sequences likely form phrases in this corpus?



Why is Pattern Mining Important

- 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

- *(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%)

Your Turn!

Tid	Items Bought
t1	a, b, c
t2	a, b, c, d
t3	b, c, e
t4	a, c, d, e
t5	d, e

Assume sup{X} represents absolute support for itemset X; s{X} represents relative support for itemset X.

sup{b, c}=	s{b, c}=
sup{a, b, c}=	s{a, b, c}=
sup{b, c, d}=	s{b, c, d}=
sup{a, b, c, d}=	s{a, b, c, d}=
sup{a, b, c, d, e}=	s{a, b, c, d, e}=

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

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!

Expressing Patterns in Compressed Form: Closed Patterns

How to handle such a challenge?

- **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, …, a_{50}}; T_2: {a_1, …, a_{100}}
  - Suppose *minsup* = 1. How many closed patterns does TDB_1 contain?
    - Two: P_1: "{a_1, …, a_{50}}: 2"; P_2: "{a_1, …, 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, …, a_{40}}: 2", "{a_5, a_{51}}: 1"

Expressing Patterns in Compressed Form: 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, …, a_{50}}; T_2: {a_1, …, a_{100}}
  - Suppose *minsup* = 1. How many max-patterns does TDB$_1$ contain?
    - One: P: "{a_1, …, a_{100}}: 1"
- **Max-pattern is a lossy compression!**
  - We only know {a_1, …, a_{40}} is frequent
  - But we do not know the real support of {a_1, …, a_{40}}, …, any more!
- **Thus in many applications, mining close-patterns is more desirable than mining max-patterns**

Your Turn!

Tid	Items Bought
t1	a, b, c
t2	a, b, c, d
t3	b, c, e
t4	a, c, d, e
t5	d, e

Is {b, c} closed? Is {a, b} closed?

sup{b, c}=

sup{a, b, c}=

sup{b, c, d}=

sup{a, b, c, d}=

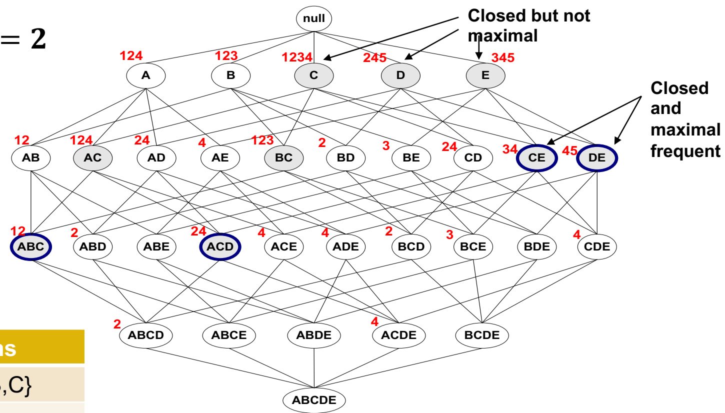
sup{a, b, c, d, e}=

sup{a, b}=

sup{a, b, c}=

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}

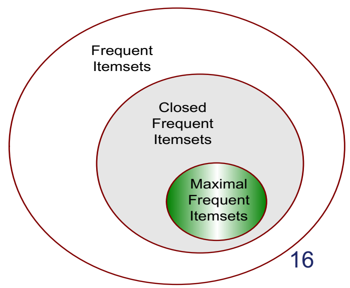
\# Closed = 9

\# Maximal = 4

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.

We have closed but not max patterns, but all max patterns are closed 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

How to mine frequent itemsets?

The Downward Closure Property of Frequent Patterns

Observation:

– Suppose we have only two transactions and $\min\_\sup = 1$

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

From TDB$_1$: T$_1$: $\{a_1, \ldots, a_{50}\}$; T$_2$: $\{a_1, \ldots, a_{100}\}$

We get a frequent itemset: $\{a_1, \ldots, a_{50}\}$

Also, its subsets are all frequent: $\{a_1\}, \{a_2\}, \ldots, \{a_{50}\}, \{a_1, a_2\}, \ldots, \{a_1, \ldots, a_{49}\}, \ldots$

There must be some hidden relationships among frequent patterns!

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!

A sharp knife for pruning!

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

for $(k = 2;\ F_{k-1}\ != \emptyset;\ \boldsymbol{k}++)$ do{
  /** candidates generation **/
  C_k = {candidates generated from F_{k-1}};

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

The Apriori Algorithm—An Example

minsup = 2

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

C_1 (1st scan)

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}

C_2 (2nd scan)

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}

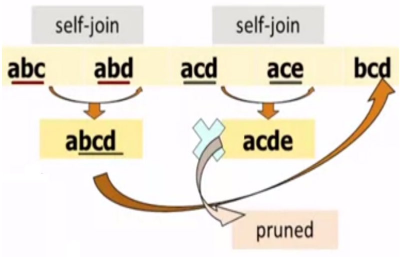
3rd scan → F_3

Itemset	sup
{B, C, E}	2

Self-join: members of $F_{k\text{-}1}$ 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

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

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

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

supports

100

Basic Concepts: Association Rules

Find all the rules $X \implies Y$ with minimum support and confidence, T is a set of transactions $T = \{t_1, \ldots, t_n\}$, X and Y are items in each transaction, $X, Y \in t_i$.

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

$$\textbf{support}(\mathbf{X} \Longrightarrow \mathbf{Y}) = \mathbf{P}(T_X \cap T_Y) = \mathbf{P}(\{X\} \cup \{Y\})$$

where $T_X \subseteq T$ is the subset transactions that contain item X; and $T_Y \subseteq T$ is the subset transactions that contain item Y

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

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

Note: $X \cup Y$ is the union of two items. The set $\{X \cup Y\}$ contains **both** X and Y. The set of transactions containing $\{X \cup Y\}$ is the intersection of the transactions containing {X} and the transactions containing {Y}.

Basic Concepts: Association Rules

THE UNIVERSITY OF WESTERN AUSTRALIA

- **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 $minsup = 50\%, minconf = 50\%$

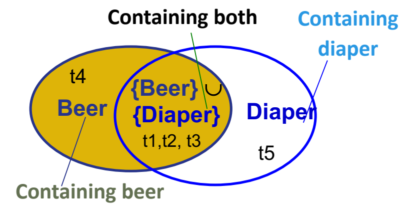
Frequent itemsets:

$\{Beer\}: \mathbf{4}, \{Nuts\}: 3, \{Diaper\}: 4,$
$\{Eggs\}: 3, \{Beer, Diaper\}: 3$

Association rules: (many more…!)

$Beer \implies Diaper \ (60\%, 75\%)$
$Diaper \implies Beer \ (60\%, 75\%)$

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



{Beer} ∪ {Diaper} = {Beer, Diaper}

Another Example (minsup=2)

THE UNIVERSITY OF WESTERN AUSTRALIA

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

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

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

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

Mining Association Rules

THE UNIVERSITY OF WESTERN AUSTRALIA

- 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 $\dfrac{support\_count(F)}{support\_count(s)} \geq \min\_conf$

- Example
  - Frequent itemset $F = \{I1, I2, I5\}$
  - Nonempty subset (proper 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

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

Strong Rules are not necessarily interesting

THE UNIVERSITY OF WESTERN AUSTRALIA

- Association rules that satisfy both a minimum support threshold (min_sup) and a minimum confidence threshold (min_conf) are called **strong**.
- 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.
- $minsup = 30\% \text{ and } minconf = 60\%$

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

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

But p(videos) = 75%

Computer games and videos are negatively associated.

$p = \frac{7500}{10{,}000}$

support

compute →

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

$\frac{4000}{10{,}000}$

confidence = $\frac{4000}{6000}$

$\frac{4000}{P(\text{computer games})}$

$\frac{4000}{6000}$

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

$\frac{0.4}{60{,}000}$

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

THE UNIVERSITY OF WESTERN AUSTRALIA

- **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

THE UNIVERSITY OF WESTERN AUSTRALIA

- 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]

THE UNIVERSITY OF WESTERN AUSTRALIA

- 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.

Summary

THE UNIVERSITY OF WESTERN AUSTRALIA

- Frequent patterns
- Closed Patterns and Max-Patterns
- Apriori algorithm for mining frequent patterns
- Association Rule Mining
- (Aside) Improving the efficiency of Apriori: Transaction Reduction, Sampling

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