Introduction to Data Mining

Lecture7 Mining Frequent Patterns, Association and Correlations

Jun Huang

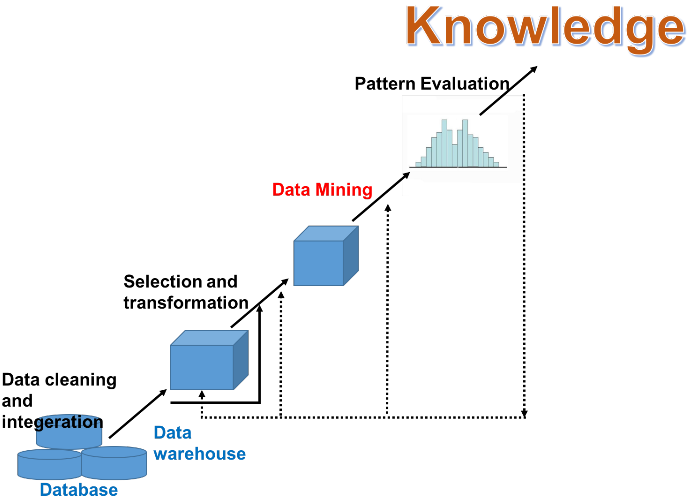
Anhui University of Technology

Spring 2018

huangjun_cs@163.com

KDD Process

Data Mining-Core of Knowledge discovery process



Mining Frequent Patterns, Association and Correlations

Introduction to Data Mining

Jun Huang

Mining Frequent Patterns

- Basic Concepts
- Boolean Association Rules
- FP-Tree
- Mining multilevel association rules
- Mining multidimensional association rules
- Sequential Patterns
- Summary

- Basic Concepts
- Mining single-dimensional Boolean association rules
- Mining multilevel association rules
- Mining multidimensional association rules
- Summary

Basic Concepts

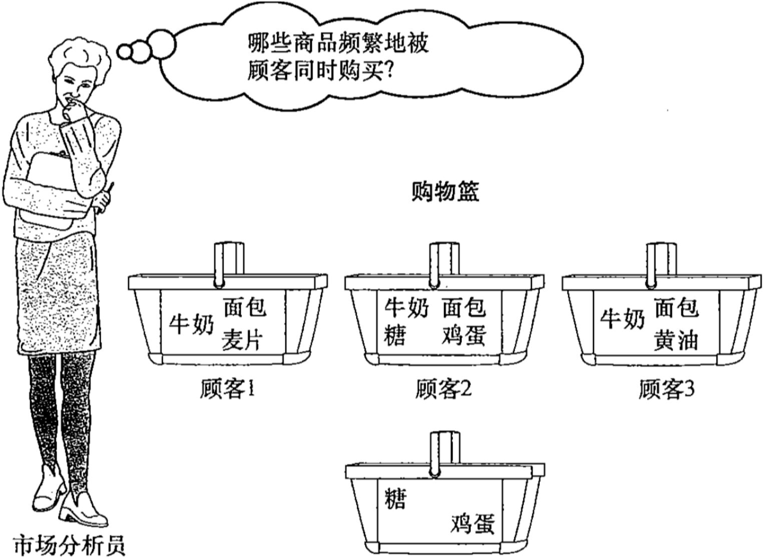
- **What are patterns?**
  - **Patterns**: A set of items, subsequences, or substructures that occur frequently together (or strongly correlated) in a data set
  - **Patterns** represent intrinsic and important properties of datasets
- **Pattern discovery**: Uncovering patterns from massive data sets
- **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?

Pattern Discovery: Why Is It Important?

Introduction to Data Mining
Jun Huang
Mining Frequent Patterns
Basic Concepts
Boolean Association Rules
FP-Tree
Mining multilevel association rules
Mining multidimensional association rules
Sequential Patterns
Summary

- Finding **inherent regularities** in a data set
- **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 subspace clustering
- **Broad applications**
  - Basket data analysis
  - Cross-marketing
  - Catalog design
  - Sale campaign analysis
  - Web log (click stream) analysis
  - DNA sequence analysis

Market Basket Analysis



What Is Association Rules Mining?

- **Association rules mining**
  - Finding frequent patterns, associations among sets of items or objects in transaction databases, relational databases, and other information repositories
- **Examples**
  - What products were often purchased together? —Beer and diapers?
  - What DNA segments often occur together in DNA sequences?
- **Where does the data come from?**
  - Supermarket transactions, membership cards, discount coupons, customer complaint calls

Basic Concepts

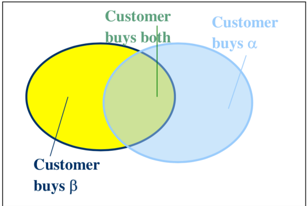
Transaction-ID	Items bought
10	A,B,D
20	A,C,D
30	A,D,E
40	B,E,F
50	B,C,D,E,F

- **Item collection** $X = \{x_1, ..., x_m\}$, e.g., {A,B,...,F}
- **Itemset**: a set of items, k-itemset
- **Transaction** $T \subseteq X$, each T associates a unique Tid and items bought by a customer
- **Rule** form $\alpha \geq \beta, \alpha \subset X, \beta \subset X, \alpha \cap \beta = \varnothing$

Mining Frequent Patterns
Basic Concepts
Boolean Association Rules
FP-Tree
Mining multilevel association rules
Mining multidimensional association rules
Sequential Patterns
Summary

Basic Concepts

- **Support**, s, probability that a transaction contains α and β
  - support ($\alpha => \beta$) $= P(\alpha \cap \beta)$
- Frequent itemset, occurrence greater than a min_support
- Frequent itemset mining, find all the rules $\alpha \geq \beta$ satisfying min_support
- Let supmin $= 50\%$,
- frequent Itemsets A:3, B:3, D:4, E:3, AD:3
- support (A) $= 3/5 = 60\%$, support (AD) $= 3/5 = 60\%$



Basic Concepts

- **Support**, s, conditional probability that a transaction having α also contains β
- **Confidence** $(\alpha => \beta) = P(\beta|\alpha) = \frac{P(\alpha \cap \beta)}{P(\alpha)}$
- Measure of rule interestingness
- Rules satisfy **min_support** and **min_confidence** are strong
- Let supmin = 50%, confmin = 50%,
- frequent itemsets A:3, B:3, D:4, E:3, AD:3
- Association rules: $\alpha \Rightarrow \beta$ (support, confidence)
  - A => D (60%, 100%)
  - D => A (60%, 75%)

There Are Too Many Frequent Patterns

- A long pattern contains a **combinatorial number of sub-patterns**
- How many frequent itemsets does the following TDB1 contain?
  - **TDB1**: $T_1 : \{a_1, ..., a_{50}\}; T_2 : \{a_1, ..., a_{100}\}$
  - 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: $2^{100} - 1$

Introduction to Data Mining
Jun Huang

Mining Frequent Patterns
Basic Concepts
Boolean Association Rules
FP-Tree
Mining multilevel association rules
Mining multidimensional association rules
Sequential Patterns
Summary

- 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 TDB1: $T_1 : \{a_1, ..., a_{50}\}$; $T_2 : \{a_1, ..., a_{100}\}$
  - Suppose minsup $= 1$. How many closed patterns does TDB1 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: "$\{a_2, ..., a_{40}\} : 2$", "$\{a_5, a_{51}\} : 1$"

Introduction to Data Mining
Jun Huang
Mining Frequent Patterns
Basic Concepts
Boolean Association Rules
FP-Tree
Mining multilevel association rules
Mining multidimensional association rules
Sequential Patterns
Summary

- Solution 2: **Max-patterns**: A pattern 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 TDB1: $T_1 : \{a_1, ..., a_{50}\}; T_2 : \{a_1, ..., a100\}$
  - Suppose minsup $= 1$. How many max-patterns does TDB1 contain?
    - One: P : "$\{a_1, ..., a_{100}\} : 1$"
- **Max-pattern is a lossy compression!**
  - We only know one pattern is frequent, e.g., $\{a_1, ..., a_{40}\}$
  - 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**

Mining Frequent Patterns
Basic Concepts
Boolean Association Rules
FP-Tree
Mining multilevel association rules
Mining multidimensional association rules
Sequential Patterns
Summary

Association Rule Mining

- **Boolean** vs. **quantitative** associations (based on the types of valued handled)
  - **Boolean association rules**, only concern presence or absence of items, buys(x,"SQLServer") and buys(x,"DMBook") $\Rightarrow$ buys(x,"DBMiner")[0.2%,60%]
  - **Quantitative association rules**, concern quantitative attributes, age(x,"30$\cdots$39") and income(x,"42$\cdots$48K") $\Rightarrow$ buys(x,"HD TV") [1%, 75%]
- **Single level** vs. **multiple-level** analysis (based on the levels of abstraction involved)
  - age(x,"30$\cdots$39") $\Rightarrow$ buys(x,"laptop computer")
  - age(x,"30$\cdots$39") $\Rightarrow$ buys(x,"computer")
- **Single dimension** vs. **multiple dimensional** associations (based on dimensions involved)
  - buys(X, "milk") => buys(X, "bread")
  - age(X,"19-25") and occupation(X,"student") => buys(X, "coke")

Introduction to Data Mining

Jun Huang

Mining Frequent Patterns
Basic Concepts
Boolean Association Rules
FP-Tree
Mining multilevel association rules
Mining multidimensional association rules
Sequential Patterns
Summary

- Basic Concepts
- Mining single-dimensional Boolean association rules
- Mining multilevel association rules
- Mining multidimensional association rules
- Summary

Handling Exponential Complexity

- Given n transactions and m different items:
  - Number of possible association rules: $O(2^m)$
  - Computation complexity: $O(nm2^m)$
- Apriori Principle
  - Collect single item counts, find large items
  - Find candidate pairs, count them $=>$ large pairs of items
  - Find candidate triplets, count them $=>$ large triplets of items, And so on...
  - **Guiding Principle**: **Every subset of a frequent itemset has to be frequent**
    - Used for pruning many candidates

Apriori: A Candidate Generation and Test Approach

- Apriori uses prior knowledge of frequent itemsets
- Iterative approach, level-wise search
- The Apriori property (downward closure property, anti-monotone) of frequent patterns
  - **Any subset of a frequent itemset must be frequent**
  - **If any itemset is infrequent, its superset should not be generated/tested**
  - If {beer, diaper, nuts} is frequent, so is {beer, diaper}, every transaction having beer, diaper, nuts also contains beer, diaper
  - If {beer, diaper} is infrequent, {beer, diaper, nut} cannot be frequent at all

Apriori: A Candidate Generation and Test Approach

Introduction to Data Mining

Jun Huang

Mining Frequent Patterns
- Basic Concepts
- Boolean Association Rules
- FP-Tree
- Mining multilevel association rules
- Mining multidimensional association rules
- Sequential Patterns
- Summary

- **Apriori Method**:
  1. Initially, scan DB once to get frequent 1-itemset
  2. Generate length $(k+1)$ candidate itemsets from length k frequent itemsets
  3. Test the candidates against DB
  4. Terminate when no frequent or candidate set can be generated

Example of Apriori Algorithm

minsup = 2

Database TDB

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

1^{st} 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}

2^{nd} 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

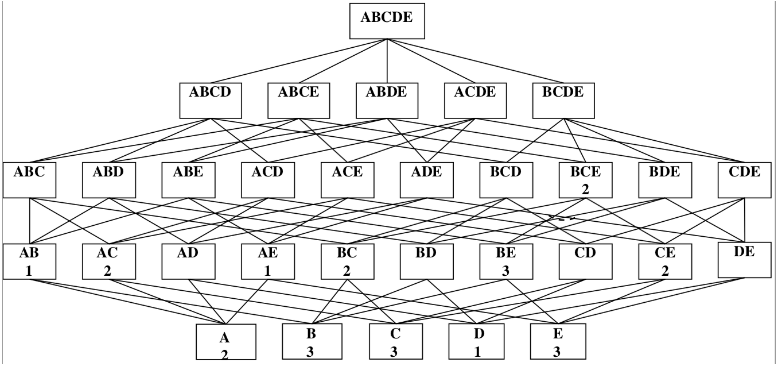
Itemset
{B, C, E}

3^{rd} scan

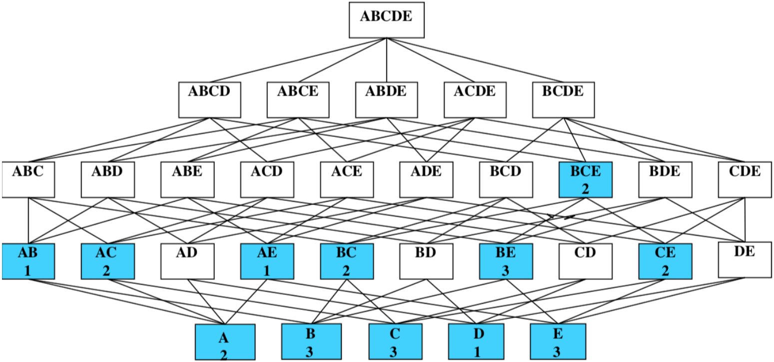
F_3

Itemset	sup
{B, C, E}	2

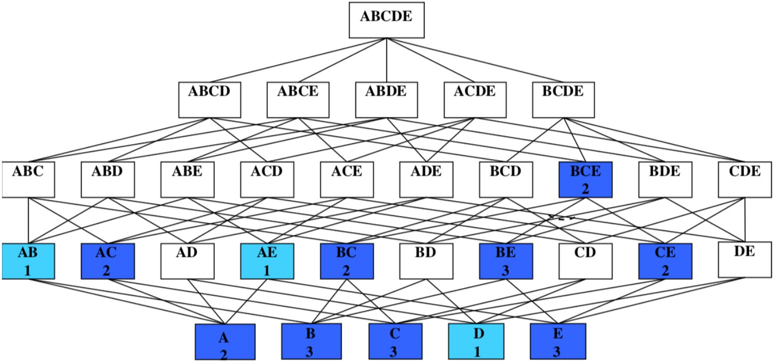
Example of Apriori Algorithm



Example of Apriori Algorithm



Example of Apriori Algorithm



Mining Frequent Patterns
Basic Concepts
Boolean Association Rules
FP-Tree
Mining multilevel association rules
Mining multidimensional association rules
Sequential Patterns
Summary

Apriori Algorithm
Pseudo-code

```
Ck: Candidate itemset of size k
Lk: frequent itemset of size k
Input: Database D, min_support
Output: frequent itemsets L
L1 = {frequent single items from D};
for (k = 2; Lk − 1! = ∅; k + +) do
   Ck = candidates generated from Lk−1;
   for each transaction t ∈ D do
      increment the count of all candidates in Ck which are
      contained in t
   end
   Lk = candidates in Ck with min_support
end
return L = ∪k Lk;
```

How to Generate Candidates?

- How to generate candidates?
  - **Step 1**: self-joining L_k
  - **Step 2**: pruning
- Example
  - $L_3 = \{abc, abd, acd, ace, bcd\}$
  - **Self-joining**: $L_3 \bowtie L_3$
    - abc and $abd \rightarrow abcd$, acd and $ace \rightarrow acde$
  - **Pruning**:
    - acde is pruned because ade is not in L_3
  - $C_4 = \{abcd\}$

How to Generate Candidates?

Suppose the items in L_{k-1} are listed in order
Step 1: self-joining L_{k-1}
for each itemset $l_1 \in L_{k-1}$
  for each itemset $l_2 \in L_{k-1}$
    if $(l_1[1] = l_2[1])$ and $(l_1[2] = l_2[2])$ and $\cdots$and
      $(l_1[k-2] = l_2[k-2])$ **then**
      $c = l_1$ join l_2
      pruning (c)
  end
end
Step 2: pruning
forall $(k-1)$-subsets s of c **do**
  if $(s$ is not in $L_{k-1})$ then delete c

How to Count Supports of Candidates?

Mining Frequent Patterns
Basic Concepts
Boolean Association Rules
FP-Tree
Mining multilevel association rules
Mining multidimensional association rules
Sequential Patterns
Summary

- **Why counting supports of candidates a problem?**
- The total number of candidates can be very huge
- One transaction may contain many candidates

- **Method:**
- Candidate itemsets are stored in a hash-tree
- Leaf node of hash-tree contains a list of itemsets and counts
- Interior node contains a hash table

Exercise

- A database has 9 transactions. Let min_sup $= 20\%$. Please present all the candidates and frequent itemsets at each iteration and frequent itemsets at each iteration.

TID	List of items_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

Introduction to Data Mining

Jun Huang

Mining Frequent Patterns
Basic Concepts
Boolean Association Rules
FP-Tree
Mining multilevel association rules
Mining multidimensional association rules
Sequential Patterns
Summary

Challenges of Frequent Pattern Mining

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

- **Improving Apriori**
- Reduce passes of transaction database scans
- Shrink number of candidates
- Facilitate support counting of candidates

Apriori: Improvements and Alternatives

- **Reduce passes of transaction database scans**
  - Partitioning (e.g., Savasere, et al., 1995)
  - Dynamic itemset counting (Brin, et al., 1997)
- **Shrink the number of candidates**
  - Hashing (e.g., DHP: Park, et al., 1995)
  - Pruning by support lower bounding (e.g., Bayardo 1998)
  - Sampling (e.g., Toivonen, 1996)
- **Exploring special data structures**
  - Tree projection (Agarwal, et al., 2001)
  - H-miner (Pei, et al., 2001)
  - Hypecube decomposition (e.g., LCM: Uno, et al., 2004)

Patition: Scan Database Only Twice

- A. Savasere, E. Omiecinski, and S. Navathe. **An efficient algorithm for mining association in large databases**. In VLDB' 95.
- Partitioning technique
  - Partition the data into N small partitions
  - **Phase 1**: find local frequent itemsets on each data partition. Record all local frequent itemsets.
  - **Phase 2**: Integrate all local frequent itemsets, scan database, find global frequent itemsets.
- **Correctness**: Any itemset that is potentially frequent in DB must be frequent in at least one of the partitions

Patition: Scan Database Only Twice

- Each partition can be fit into memory
- Scan database only **twice**! Reduce I/O cost!
- Execution time scales linearly
- Good for very large-scale database
- Applicable to **parallel/distributed** computing systems
  - Each processor performs FIM on its local data
  - Central server aggregates local frequent itemsets, broadcast potential global itemsets
  - Each processor scans local data to count the frequency
  - Central server aggregates the counts, find the global itemsets

Reduce the Number of Candidates

DHP:Direct Hashing and Pruning

- J. Park, M. Chen, and P. Yu. **An effective hash-based algorithm for mining association rules**. In SIGMOD' 95
- Hash-based technique
  - When scanning transactions to generate frequent k–itemsets, L_k, generate all $(k+1)$-itemsets for each transaction
  - Hash all $(k+1)$-itemsets into buckets, increase bucket count
  - If a $(k+1)$-itemset bucket count is below min_sup, it must be removed from $(k+1)$ candidate itemsets, C_{k+1}
- **Correctness**:A k-itemset whose corresponding hashing bucket count is below the threshold cannot be frequent

Reduce the Number of Candidates
DHP:Direct Hashing and Pruning

- Example: At the 1st scan of TDB, count 1-itemset, and Hash 2-itemsets in the transaction to its bucket
  - $\{ab, ad, ce\}$
  - $\{bd, be, de\}$
  - $\cdots$
- At the end of the first scan,
- if minsup $= 80$, remove ab, ad, ce, since count{ab, ad, ce} < 80

Itemsets	Count
{ab, ad, ce}	35
{bd, be, de}	298
……	…
{yz, qs, wt}	58

Hash Table

Mining Frequent Patterns
Basic Concepts
Boolean Association Rules
FP-Tree
Mining multilevel association rules
Mining multidimensional association rules
Sequential Patterns
Summary

Bottleneck of Frequent-pattern Mining

- Multiple database scans are costly
- Mining long patterns needs many passes of scanning and generates lots of candidates
- To find frequent itemset $i_1, i_2, ..., i_{100}$
  - \# of scans: 100
  - \# of Candidates: $(100^1) + (100^2) + ... + (100^{100}) = 2^{100} - 1 \approx 1.27 * 10^{30}$
- Bottleneck: candidate generation and test
- Can we avoid candidate generation?

Construct FP-Tree from a Transaction Database

Introduction to Data Mining

Jun Huang

Mining Frequent Patterns
- Basic Concepts
- Boolean Association Rules
- FP-Tree
- Mining multilevel association rules
- Mining multidimensional association rules
- Sequential Patterns
- Summary

1. **Scan DB once**, find frequent 1-itemset (single item pattern)
2. Sort frequent items in frequency descending order L
3. Create the root of the tree, labeled with "null"
4. **Scan DB again**, sort each transaction in L order, a branch is created for each transaction
   - Increment the count of each node along a common prefix by 1
   - Create nodes for the items following the prefix
5. Build a header table, connect each item point in the tree

Construct FP-Tree from a Transaction Database

Introduction to Data Mining

Jun Huang

Mining Frequent Patterns
Basic Concepts
Boolean Association Rules
FP-Tree
Mining multilevel association rules
Mining multidimensional association rules
Sequential Patterns
Summary

TID	Items in the Transaction	Ordered, frequent itemlist
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

- Let min_support $= 3$
- 1-itemset: $f:4, a:3, c:4, b:3, m:3, p:3$
- $L = f \to c \to a \to b \to m \to p$

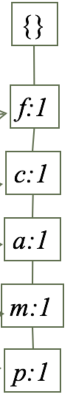
Mining Frequent Patterns
- Basic Concepts
- Boolean Association Rules
- FP-Tree
- Mining multilevel association rules
- Mining multidimensional association rules
- Sequential Patterns
- Summary

Construct FP-Tree from a Transaction Database

After inserting the 1st frequent Itemlist: *"f, c, a, m, p"*

Header Table

Item	Frequency	header
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	

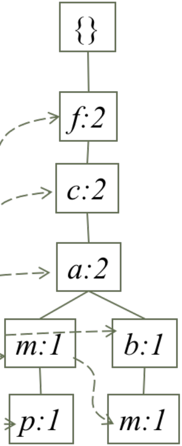


Construct FP-Tree from a Transaction Database

After inserting the 2nd frequent itemlist "*f, c, a, b, m*"

Header Table

Item	Frequency	header
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	



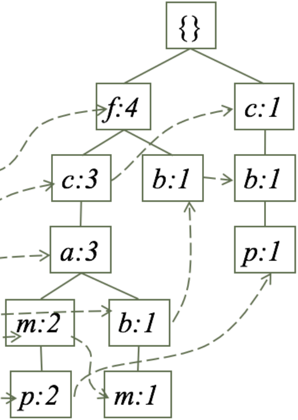
Mining Frequent Patterns
Basic Concepts
Boolean Association Rules
FP-Tree
Mining multilevel association rules
Mining multidimensional association rules
Sequential Patterns
Summary

Construct FP-Tree from a Transaction Database

After inserting all the frequent itemlists

Header Table

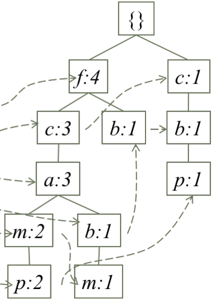
Item	Frequency	header
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	



Construct FP-Tree from a Transaction Database

min_support = 3

Item	Frequency	Header
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	



Conditional database of each pattern

Item	*Conditional database*
c	***f:3***
a	***fc:3***
b	***fca:1, f:1, c:1***
m	***fca:2, fcab:1***
p	***fcam:2, cb:1***

Benefits of the FP-tree Structure

- **Completeness**
- Preserve complete information for frequent pattern mining
- Never break a long pattern of any transaction

- **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

Mining Frequent Patterns With FP-trees

procedure **FP_growth**(Tree, α)

if Tree contains a single path P then

  for each combination (denoted as β) of the nodes in the path P

    generate pattern $\beta \cup \alpha$ with support_count $=$ minimum support count of nodes in β

else

  for each α_i in the header of Tree {

    generate pattern $\beta = \alpha_i \cup \alpha$ with support_count $=$ α_i.support_count

    construct β's conditional pattern base and then β's conditional FP_tree Tree$_\beta$

    if Tree$_\beta$ **then**

      call **FP_growth**(Tree$_\beta$, β)}

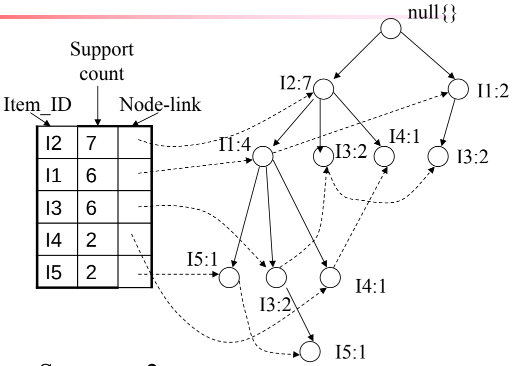
Exercise

- **A database has 9 transactions. Let min_sup = 20%. Please present all the candidates and frequent itemsets at each iteration and frequent itemsets at each iteration.**

TID	List of items_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

Solution

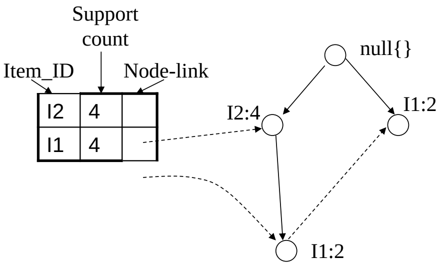
TID	List of items_IDs
T100	I1,I2,I5
T200	I2,I4
T300	I2,I3
T400	I1,I2,I4
T500	I1,I3
T600	I2,I3
T700	I1,I3
T800	I1,I2,I3,I5
T900	I1,I2,I3



Support =2

Solution

item	*conditional pattern base*	*conditional FP-tree*	*frequent patterns generated*
I5	{{I2,I1: 1}, {I2,I1,I3: 1}}	⟨I2: 2, I1: 2⟩	{I2,I5: 2}, {I1,I5: 2}, {I2,I1,I5: 2}
I4	{{I2,I1: 1}, {I2: 1}}	⟨I2: 2⟩	{I2,I4: 2}
I3	{{I2,I1: 2}, {I2: 2}, {I1: 2}}	⟨I2: 4, I1: 2⟩, ⟨I1: 2⟩	{I2,I3: 4}, {I1,I3: 4}, {I2,I1,I3: 2}
I1	{{I2: 4}}	⟨I2: 4⟩	{I2,I1: 4}

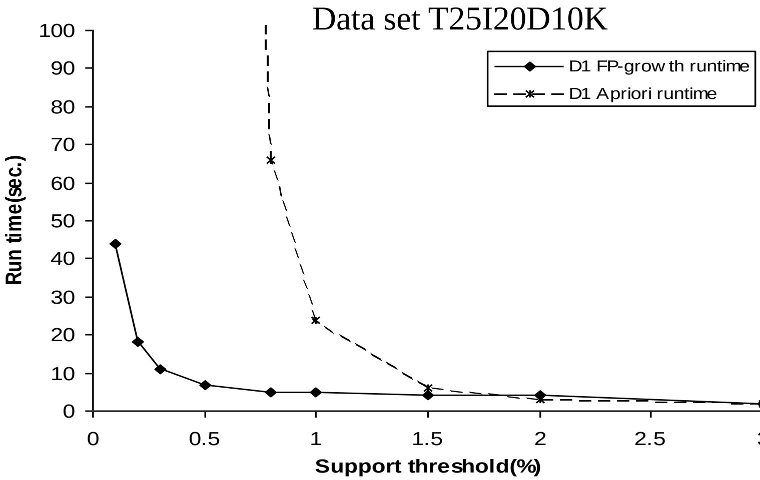


FP Tree vs. Apriori: Scalability With the Support Threshold

Introduction to Data Mining

Jun Huang

Mining Frequent Patterns
- Basic Concepts
- Boolean Association Rules
- FP-Tree
- Mining multilevel association rules
- Mining multidimensional association rules
- Sequential Patterns
- Summary



Why Is FP-Growth the Winner?

- **Divide-and-conquer:**
  - Decompose both the mining task and DB according to the frequent patterns obtained so far
  - Focus searching on smaller databases
- **Other factors**
  - No candidate generation, no candidate test
  - Compressed database: FP-tree structure
  - Two scans of entire database
  - Basic ops—counting local freq items and building sub FP-tree, no pattern search and matching

Scaling FP-growth by Item-Based Data Projection

Introduction to Data Mining

Jun Huang

Mining Frequent Patterns

Basic Concepts

Boolean Association Rules

FP-Tree

Mining multilevel association rules

Mining multidimensional association rules

Sequential Patterns

Summary

- What if FP-tree cannot fit in memory? —Do not construct FP-tree
  - "Project" the database based on frequent single items
  - Construct & mine FP-tree for each projected DB
- **Parallel projection** vs. **partition projection**
  - Parallel projection: Project the DB on each frequent item
    - Space costly, all partitions can be processed in parallel
  - Partition projection: Partition the DB in order
    - Passing the unprocessed parts to subsequent partitions

Mining Frequent Patterns

Basic Concepts

Boolean Association Rules

FP-Tree

Mining multilevel association rules

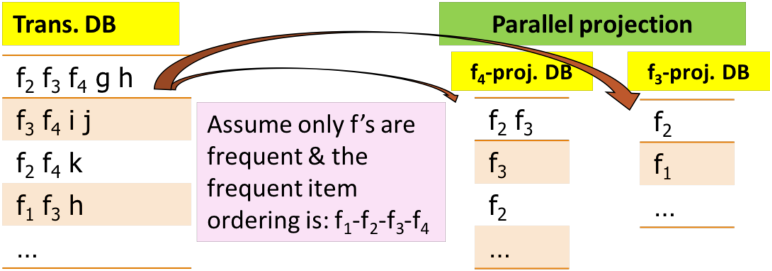
Mining multidimensional association rules

Sequential Patterns

Summary

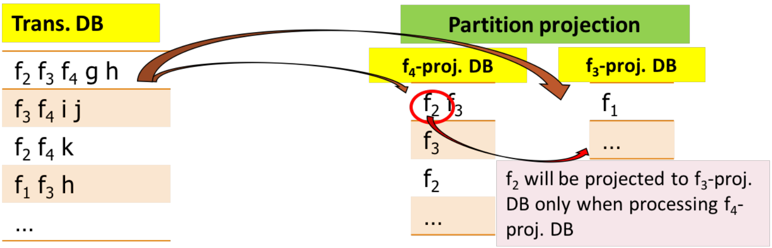
Scaling FP-growth by Item-Based Data Projection

- **Parallel projection**: Project the DB on each frequent item
  - Space costly, all partitions can be processed in parallel



Scaling FP-growth by Item-Based Data Projection

- **Partition projection**: Partition the DB in order
  - Passing the unprocessed parts to subsequent partitions



Introduction to Data Mining

Jun Huang

Mining Frequent Patterns
Basic Concepts
Boolean Association Rules
FP-Tree
Mining multilevel association rules
Mining multidimensional association rules
Sequential Patterns
Summary

A transaction DB in Horizontal Data Format

Tid	Itemset
10	a, c, d, e
20	a, b, e
30	b, c, e

The transaction DB in Vertical Data Format

Item	TidList
a	10, 20
b	20, 30
c	10, 30
d	10
e	10, 20, 30

Introduction to Data Mining

Jun Huang

Mining Frequent Patterns
- Basic Concepts
- Boolean Association Rules
- FP-Tree
- Mining multilevel association rules
- Mining multidimensional association rules
- Sequential Patterns
- Summary

- ECLAT: A depth-first search algorithm using set intersection [Zaki et al. KDD’ 97]
- **Tid-List**: List of transaction-ids containing an itemset
- **Vertical format**: $t(e) = \{T_{10}, T_{20}, T_{30}\}$; $t(a) = \{T_{10}, T_{20}\}$; $t(ae) = \{T_{10}, T_{20}\}$
- **Properties of Tid-Lists**
  - $t(X) = t(Y)$: X and Y always happen together (e.g., $t(ac) = t(d)$)
  - $t(X) \subset t(Y)$: transaction having X always has Y (e.g., $t(ac) \subset t(ce)$)
- Deriving frequent patterns based on vertical intersections
- Using **diffset** to accelerate mining
  - Only keep track of differences of tids
  - $t(e) = \{T_{10}, T_{20}, T_{30}\}, t(ce) = \{T_{10}, T_{30}\} \rightarrow Diffset(ce, e) = \{T_{20}\}$

Mining Frequent Patterns, Association and Correlations

- Basic Concepts
- Mining single-dimensional Boolean association rules
- Mining multilevel association rules
- Mining multidimensional association rules
- Summary

Mining multilevel association rules

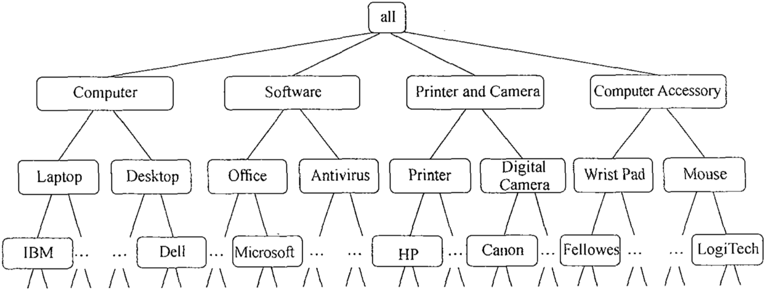
- Association rules at high concept levels may represent common sense knowledge
- Hard to find association rules at low concept level
- Items at the lower level usually have lower support, less than min_support threshold
- Mining association rules at multiple levels of abstraction
- Example: sales in AllElectronics store computer sector

Introduction to Data Mining

Jun Huang

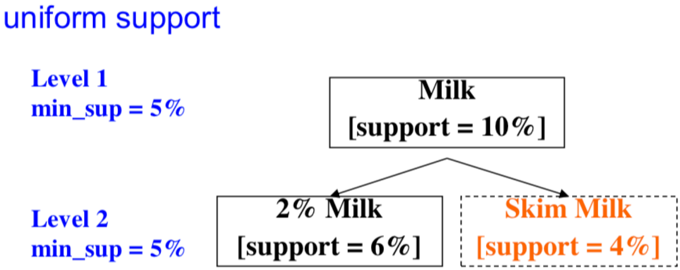
Mining Frequent Patterns
Basic Concepts
Boolean Association Rules
FP-Tree
Mining multilevel association rules
Mining multidimensional association rules
Sequential Patterns
Summary

Example


all
Computer
Software
Printer and Camera
Computer Accessory
Laptop
Desktop
Office
Antivirus
Printer
Digital Camera
Wrist Pad
Mouse
IBM
Dell
Microsoft
HP
Canon
Fellowes
LogiTech

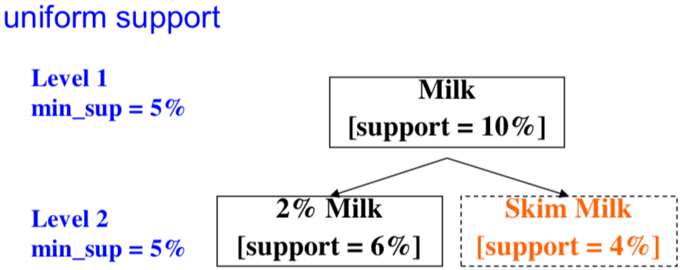
Mining multilevel association rules

- **Uniform support**
  - Top-down, level-wise
  - Use uniform minimum support for each level
  - Perform Apriori at each level
  - Optimization: if an ancestor is infrequent, the search on the descendants can be avoided



Mining Frequent Patterns
Basic Concepts
Boolean Association Rules
FP-Tree
Mining multilevel association rules
Mining multidimensional association rules
Sequential Patterns
Summary

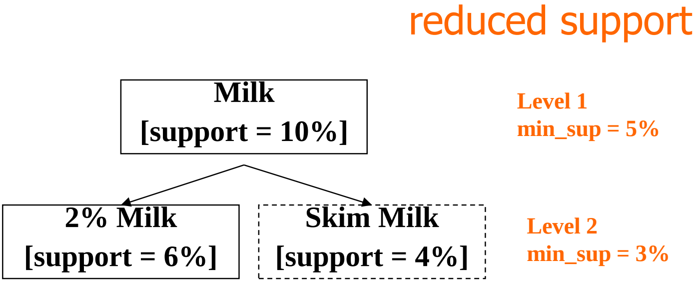
Mining multilevel association rules



- **Drawbacks**
  - Miss interesting associations with too high threshold
  - Generate too many uninteresting rules with too low threshold

Mining multilevel association rules

- **Reduced support**
  - Top-down, level-wise
  - Each concept level has its own minimum support threshold
  - The lower level, the smaller threshold
  - Perform Apriori at each level

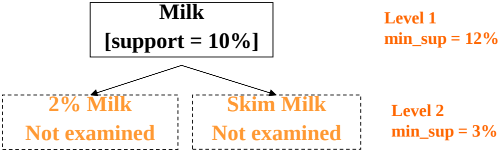


Mining Frequent Patterns
Basic Concepts
Boolean Association Rules
FP-Tree
Mining multilevel association rules
Mining multidimensional association rules
Sequential Patterns
Summary

Mining multilevel association rules

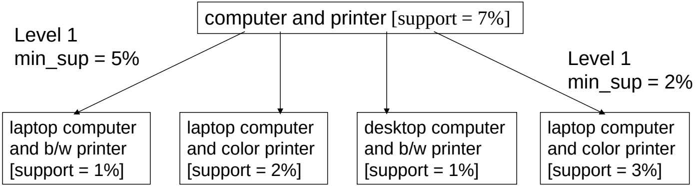
- **Reduced support**
  - Optimization – level-cross filtering by single item
  - An item at the ith concept level is examined iff its parent concept at the $(i-1)$th level is frequent
  - If a concept is infrequent, its descendents are pruned from the database
  - Drawbacks
    - Miss associations at low level items which are frequent based on a reduced min_support, but whose ancestors do not satisfy min_support

reduced support



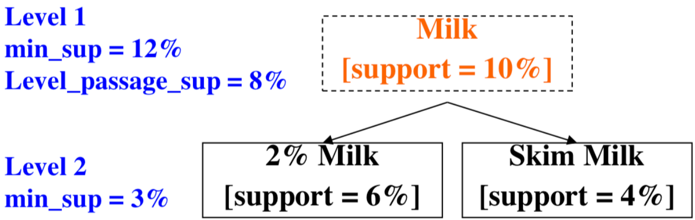
Mining multilevel association rules

- **Reduced support**
  - Optimization – level-cross filtering by k-itemset
    - Only the children of frequent k-itemsets are examined
    - Drawback: many valuable patterns may be filtered out



Mining multilevel association rules

- **Reduced support**
  - Optimization – Controled level-cross filtering by single item
    - next level min sup < level passage threshold < min sup
    - Allow the children of items that do not satisfy the min_sup to be examined if they satisfy the level passage threshold



Multi-level Association: Redundancy Filtering

- Some rules may be redundant due to "ancestor" relationships between items
- Example
  - buys(X,"Laptop computer")=> buys(X,"HP printer") [support = 8%, confidence = 70%]
  - buys(X,"IBM laptop computer")=> buys(X,"HP printer") [support = 2%, confidence = 72%]
- We say the first rule is an ancestor of the second rule
- A rule is redundant if its support is close to the "expected" value, based on the rule's ancestor

Mining Frequent Patterns, Association and Correlations

- Basic Concepts
- Mining single-dimensional Boolean association rules
- Mining multilevel association rules
- Mining multidimensional association rules
- Summary

Mining multidimensional association rules

Introduction to Data Mining

Jun Huang

Mining Frequent Patterns
- Basic Concepts
- Boolean Association Rules
- FP-Tree
- Mining multilevel association rules
- Mining multidimensional association rules
- Sequential Patterns
- Summary

- Single-dimensional rules:
  buys(X, "milk") => buys(X, "bread")
- Multi-dimensional rules: $\geq$2 dimensions or predicates
  - Inter-dimension assoc. rules (no repeated predicates)
    age(X,"19-25") and occupation(X,"student") => buys(X, "coke")
  - hybrid-dimension assoc. rules (repeated predicates)
    age(X,"19-25") and **buys**(X, "popcorn") => **buys**(X, "coke")
- **Categorical Attributes**: finite number of possible values, no ordering among values
- **Quantitative Attributes**: numeric, implicit ordering among values —discretization, clustering approaches

Mining Quantitative Associations

Techniques can be used to **categorize numerical attributes**

- **Static discretization** based on predefined concept hierarchies
- **Dynamic discretization** based on data distribution
- **Clustering**: Distance-based association
  - one dimensional clustering then association

Multidimensional Association Rules and Static Discretization of Quantitative Attributes

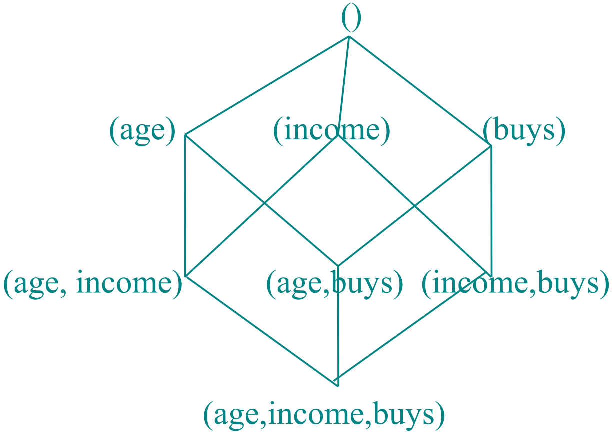
Introduction to Data Mining

Jun Huang

Mining Frequent Patterns
- Basic Concepts
- Boolean Association Rules
- FP-Tree
- Mining multilevel association rules
- Mining multidimensional association rules
- Sequential Patterns
- Summary

- Discretized prior to mining using **concept hierarchy**
- Numeric values are **replaced by ranges**
- In relational database, finding all frequent k-predicate sets will require k or $k+1$ table scans
- **Data cube** is well suited for mining
  - The cells of a n-dimensional: cuboid correspond to the dimensions
  - Mining from data cubes can be much faster

Multidimensional Association Rules and Static Discretization of Quantitative Attributes

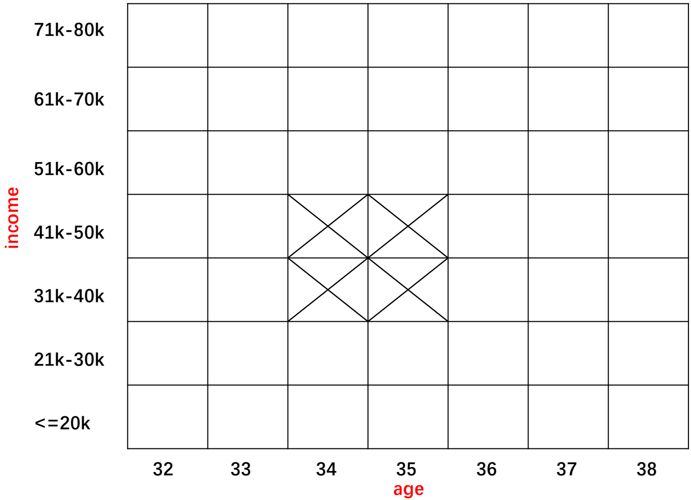


Quantitative Association Rules

- Numeric attributes are dynamically discretized
  - Such that the confidence or compactness of the rules mined is maximized
- 2-D quantitative association rules:
  A_{quan1} and $A_{quan2} => A_{cat}$
- Association rule clustering system (ARCS)
  - Binning: 2-D grid, manageable size
  - Finding frequent predicate sets: scan the database, count the support for each grid cell
  - Clustering the rules: cluster adjacent cells to form a rule

Quantitative Association Rules

Example: age and income => buy HD TV



Introduction to Data Mining

Jun Huang

Mining Frequent Patterns

- Basic Concepts
- Boolean Association Rules
- FP-Tree
- Mining multilevel association rules
- Mining multidimensional association rules
- Sequential Patterns
- Summary

Quantitative Association Rules

- **Example**:
  - age(X,"34") and income(X,"31-40K") => buys(X,"HD TV")
  - age(X,"35") and income(X,"31-40K") => buys(X,"HD TV")
  - age(X,"34") and income(X,"41-50K") => buys(X,"HD TV")
  - age(X,"35") and income(X,"41-50K") => buys(X,"HD TV")
- => age(X,"34-35") and income(X,"31-50K") => buys(X,"HD TV")

Interestingness Measure: Correlations (Lift)

Introduction to Data Mining

Jun Huang

Mining Frequent Patterns

Basic Concepts

Boolean Association Rules

FP-Tree

Mining multilevel association rules

Mining multidimensional association rules

Sequential Patterns

Summary

- play basketball $=>$ eat cereal [40%, 66.7%] is misleading
- The overall percentage of students eating cereal is 75% $>$ 66.7%.
- Measure of dependent/correlated events:

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

- $lift(A, B) = 1$: A and B are independent
- $lift(A, B) > 1$: A and B are positive correlated
- $lift(A, B) < 1$: A and B are negative correlated

Interestingness Measure: Correlations (Lift)

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

- $lift(A, B) = \frac{2000/5000}{(3000/5000)*(3750/5000)} = 0.89$
- $lift(A, \bar{B}) = \frac{1000/5000}{(3000/5000)*(1250/5000)} = 1.33$
- $A \Rightarrow B$ [**support, confidence, correlation**]

Sequence Databases & Sequential Patterns

Introduction to Data Mining

Jun Huang

Mining Frequent Patterns

Basic Concepts
Boolean Association Rules
FP-Tree
Mining multilevel association rules
Mining multidimensional association rules
Sequential Patterns
Summary

- Sequential pattern mining has broad applications
  - Customer shopping sequences
  - Purchase a laptop first, then a digital camera, and then a smartphone, within 6 months
  - Medical treatments, natural disasters (e.g., earthquakes), science & engineering processes, stocks and markets, ...
  - Weblog click streams, calling patterns, $\cdots$
  - Software engineering: Program execution sequences, $\cdots$
  - Biological sequences: DNA, protein, $\cdots$
- Transaction DB, sequence DB vs. time-series DB
- Gapped vs. non-gapped sequential patterns
  - Shopping sequences, clicking streams vs. biological sequences

Sequential Pattern and Sequential Pattern Mining

- Sequential pattern mining: Given a set of sequences, find the complete set of frequent subsequences (i.e., satisfying the min_sup threshold)

A *sequence database*

SID	Sequence
10	<a(abc)(ac)d(cf)>
20	<(ad)c(bc)(ae)>
30	<(ef)(ab)(df)cb>
40	<eg(af)cbc>

- A sequence: $<(ef)(ab)(df)cb>$
- An element may contain a set of items (also called events)
- Items within an element are unordered and we list them alphabetically
- $<a(bc)dc>$ is a subsequence of $<a(abc)(ac)d(cf)>$
- Given support threshold min_sup $=2$, <(ab)c> is a sequential pattern

Introduction to Data Mining

Jun Huang

Mining Frequent Patterns
Basic Concepts
Boolean Association Rules
FP-Tree
Mining multilevel association rules
Mining multidimensional association rules
Sequential Patterns
Summary

- Algorithm requirement: Efficient, scalable, finding complete set, incorporating various kinds of user-specific constraints
- The Apriori property still holds: If a subsequence s1 is infrequent, none of s1's super-sequences can be frequent
- Representative algorithms
  - GSP (Generalized Sequential Patterns): Srikant & Agrawal @ EDBT'96)
  - Vertical format-based mining: SPADE (Zaki@Machine Leanining'00)
  - Pattern-growth methods: PrefixSpan (Pei, et al. @TKDE'04)
- Mining closed sequential patterns: CloSpan (Yan, et al. @SDM'03)
- Constraint-based sequential pattern mining (to be covered in the constraint mining section)

Mining Frequent Patterns
Basic Concepts
Boolean Association Rules
FP-Tree
Mining multilevel association rules
Mining multidimensional association rules
Sequential Patterns
Summary

GSP: Apriori-Based Sequential Pattern Mining

- Initial candidates: All 8-singleton sequences
- $<a>, <b>, <c>, <d>, <e>, <f>, <g>, <h>$
- Scan DB once, count support for each candidate
- Generate length-2 candidate sequences
- Repeat (for each level (i.e., length-k))
  - Scan DB to find length-k frequent sequences
  - Generate length-$(k+1)$ candidate sequences from length-k frequent sequences using Apriori
  - set $k = k + 1$
- Until no frequent sequence or no candidate can be found

GSP: Apriori-Based Sequential Pattern Mining

Algorithm GSP($\mathcal{S}$)

$C_1 \leftarrow$ init-pass ($\mathcal{S}$)
$F_1 \leftarrow \{< \{f\} > | f \in C_1, f.count/n \geq min\_sup\}$
for $(k = 2; F_{k-1} \neq \varnothing; k++)$ **do**
    $C_k \leftarrow$ candidate-gen-SPM(F_{k-1})
    for each data sequence $s \in \mathcal{S}$ **do**
        for each candidate $c \in C_k$ **do**
            if c is contained in s **then**
                $c.count++;$
            end
        end
        $F_k \leftarrow \{c \in C_k | c.count/n \geq min\_sup\}$
    end
    return $F \leftarrow \cup_k F_k$
end

GSP: Apriori-Based Sequential Pattern Mining

Function candidate-gen-SPM(F_{k-1})

1. **Joint step**. Candidate sequences are generated by joining F_{k-1} with F_{k-1}. A sequence s_1 joins with s_2 if the subsequence obtained by dropping the first item of s_1 is the same as the subsequence obtained by dropping the last item of s_2. The candidate sequenc generated by joining s_1 with s_2 is the sequence s_1 extended with the last item in s_2. There are two cases:
   - the added item forms a separate element if it was a separate element in s_2, and is appended at the end of s_1 in the merged sequence
   - the added item is part of the last element of s_1 in the merged sequence

   When joining F_1 with F_1, we need to add the item in s_2 both as part of an itemset and as a separate element. That is, joining $<\{x\}>$ with $<\{y\}>$ gives us both $<\{x, y\}>$ and $<\{x\}, \{y\}>$. Note that x and y in $\{x, y\}$ are ordered.
2. **Prune step**. A candidate sequence is pruend if any one of its $(k-1)$-subsequences is infrequent (without minimum support)

GSP: Apriori-Based Sequential Pattern Mining

min_sup = 2

Cand.	sup
<a>	3
<b>	5
<c>	4
<d>	3
<e>	3
<f>	2
~~<g>~~	1
~~<h>~~	1

	<a>	<b>	<c>	<d>	<e>	<f>
<a>	<aa>	<ab>	<ac>	<ad>	<ae>	<af>
<b>	<ba>	<bb>	<bc>	<bd>	<be>	<bf>
<c>	<ca>	<cb>	<cc>	<cd>	<ce>	<cf>
<d>	<da>	<db>	<dc>	<dd>	<de>	<df>
<e>	<ea>	<eb>	<ec>	<ed>	<ee>	<ef>
<f>	<fa>	<fb>	<fc>	<fd>	<fe>	<ff>

	<a>	<b>	<c>	<d>	<e>	<f>
<a>		<(ab)>	<(ac)>	<(ad)>	<(ae)>	<(af)>
<b>			<(bc)>	<(bd)>	<(be)>	<(bf)>
<c>				<(cd)>	<(ce)>	<(cf)>
<d>					<(de)>	<(df)>
<e>						<(ef)>
<f>						

GSP: Apriori-Based Sequential Pattern Mining

SID	Sequence
10	<(bd)cb(ac)>
20	<(bf)(ce)b(fg)>
30	<(ah)(bf)abf>
40	<(be)(ce)d>
50	<a(bd)bcb(ade)>

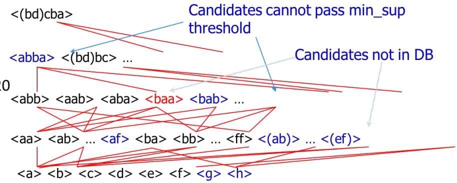
5th scan: 1 cand. 1 length-5 seq. pat.

4th scan: 8 cand. 7 length-4 seq. pat.

3rd scan: 46 cand. 20 length-3 seq. pat. 20 cand. not in DB at all

2nd scan: 51 cand. 19 length-2 seq. pat. 10 cand. not in DB at all

1st scan: 8 cand. 6 length-1 seq. pat.



Summary

- Frequent pattern mining —an important task in data mining
- Scalable frequent pattern mining methods
  - Apriori (Candidate generation & test)
  - Partition, DIC, DHP, etc.
  - Projection-based (FP-growth)
- Mining a variety of rules and interesting pattern

Readings

- R. Agrawal, T. Imielinski, and A. Swami, “Mining association rules between sets of items in large databases”, in Proc. of SIGMOD'93
- R. J. Bayardo, “Efficiently mining long patterns from databases”, in Proc. of SIGMOD'98
- N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal, “Discovering frequent closed itemsets for association rules”, in Proc. of ICDT'99
- J. Han, H. Cheng, D. Xin, and X. Yan, “Frequent Pattern Mining: Current Status and Future Directions”, Data Mining and Knowledge Discovery, 15(1): 55-86, 2007
- R. Agrawal and R. Srikant, “Fast algorithms for mining association rules”, VLDB'94
- A. Savasere, E. Omiecinski, and S. Navathe, “An efficient algorithm for mining association rules in large databases”, VLDB'95
- J. S. Park, M. S. Chen, and P. S. Yu, “An effective hash-based algorithm for mining association rules”, SIGMOD'95

Readings

- S. Sarawagi, S. Thomas, and R. Agrawal, "Integrating association rule mining with relational database systems: Alternatives and implications" , SIGMOD'98
- M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li, "Parallel algorithm for discovery of association rules" , Data Mining and Knowledge Discovery, 1997
- J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation" , SIGMOD' 00
- M. J. Zaki and Hsiao, "CHARM: An Efficient Algorithm for Closed Itemset Mining" , SDM'02
- J. Wang, J. Han, and J. Pei, "CLOSET+: Searching for the Best Strategies for Mining Frequent Closed Itemsets" , KDD'03
- C. C. Aggarwal, M.A., Bhuiyan, M. A. Hasan, "Frequent Pattern Mining Algorithms: A Survey" , in Aggarwal and Han (eds.): Frequent Pattern Mining, Springer, 2014

Readings

- C. C. Aggarwal and P. S. Yu. A New Framework for Itemset Generation. PODS' 98
- S. Brin, R. Motwani, and C. Silverstein. Beyond market basket: Generalizing association rules to correlations. SIGMOD'97
- M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A. I. Verkamo. Finding interesting rules from large sets of discovered association rules. CIKM'94
- E. Omiecinski. Alternative Interest Measures for Mining Associations. TKDE' 03
- P.-N. Tan, V. Kumar, and J. Srivastava. Selecting the Right Interestingness Measure for Association Patterns. KDD'02
- T. Wu, Y. Chen and J. Han, Re-Examination of Interestingness Measures in Pattern Mining: A Unified Framework, Data Mining and Knowledge Discovery, 21(3):371-397, 2010