slideshare
a Scribd company
Upload Login Signup
Search

Apriori and Eclat algorithm in Association Rule Mining

Apr 30, 2014 • Download as PPT, PDF • 33 likes • 44,366 views

Wan Aezwani Wab
ALAM
Follow

Health & Medicine Business

Fall 2004, CIS, Temple University

CIS527: Data Warehousing, Filtering, and Mining

Lecture 4

- Tutorial: Connecting SQL Server to Matlab using Database Matlab Toolbox
- Association Rule MIning

Lecture slides taken/modified from:

– Jiawei Han (http://www-sal.cs.uiuc.edu/~hanj/DM_Book.html)

– Vipin Kumar (http://www-users.cs.umn.edu/~kumar/csci5980/index.html)

Recommended

Apriori Algorithm
International School of En
224.5K views • 14 slides

Association rule mining.pptx
maha797959
608 views • 74 slides

Règles d'association
Hassine Hammami
2.8K views • 26 slides

Lect6 Association rule & Apriori algorithm
hktripathy
5.1K views • 41 slides

Data mining - Associativité
Mohamed Heny SELMI
3.5K views • 19 slides

Assosiate rule mining
Tilani Gunawardena PhD(
2.7K views • 30 slides

Sequential Pattern Mining and GSP
Hamidreza Mahdavipanah
3.3K views • 29 slides

Apriori algorithm
nouraalkhatib
32.1K views • 14 slides

Save slide

Motivation: Association Rule Mining

- Given a set of transactions, find rules that will predict the items in the transaction

Apriori and Eclat algorithm in Association Rule Mining 3 of 44 Download Now

Market-Basket transactions

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example of Association Rules

{Diaper} → {Beer},
{Milk, Bread} → {Eggs,Coke},
{Beer, Bread} → {Milk},

Implication means co-occurrence, not causality!

Applications: Association Rule Mining

- * ⇒ Maintenance Agreement
  - What the store should do to boost Maintenance Agreement sales
- Home Electronics ⇒ *
  - What other products should the store stocks up?
- Attached mailing in direct marketing
- Detecting “ping-ponging” of patients
- Marketing and Sales Promotion
- Supermarket shelf management

Definition: Frequent Itemset

- **Itemset**
  - A collection of one or more items
    - Example: {Milk, Bread, Diaper}
  - k-itemset
    - An itemset that contains k items
- **Support count (σ)**
  - Frequency of occurrence of an itemset
  - E.g. σ({Milk, Bread,Diaper}) = 2
- **Support**
  - Fraction of transactions that contain an itemset
  - E.g. s({Milk, Bread, Diaper}) = 2/5
- **Frequent Itemset**
  - An itemset whose support is greater than or equal to a *minsup* threshold

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Definition: Association Rule

- **Association Rule**
  - An implication expression of the form $X \rightarrow Y$, where X and Y are itemsets
  - Example: {Milk, Diaper} → {Beer}
- **Rule Evaluation Metrics**
  - Support (s)
    - Fraction of transactions that contain both X and Y
  - Confidence (c)
    - Measures how often items in Y appear in transactions that contain X

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example:

$$\{\text{Milk, Diaper}\} \Rightarrow \text{Beer}$$

$$s = \frac{\sigma(\text{Milk, Diaper, Beer})}{|\text{T}|} = \frac{2}{5} = 0.4$$

$$c = \frac{\sigma(\text{Milk, Diaper, Beer})}{\sigma(\text{Milk, Diaper})} = \frac{2}{3} = 0.67$$

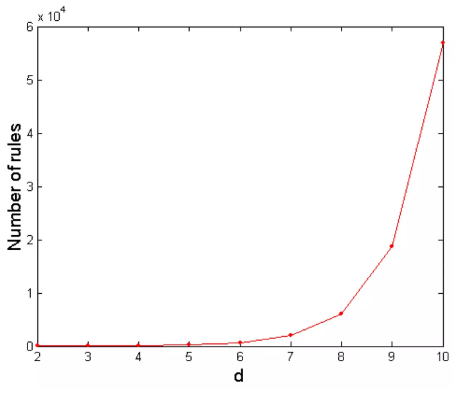
Association Rule Mining Task

- Given a set of transactions T, the goal of association rule mining is to find all rules having
  - support ≥ *minsup* threshold
  - confidence ≥ *minconf* threshold
- **Brute-force approach**:
  - List all possible association rules
  - Compute the support and confidence for each rule
  - Prune rules that fail the *minsup* and *minconf* thresholds

  ⇒ Computationally prohibitive!

Computational Complexity

- Given d unique items:
  - Total number of itemsets = 2^d
  - Total number of possible association rules:



$$R = \sum_{k=1}^{d-1}\left[\binom{d}{k} \times \sum_{j=1}^{d-k}\binom{d-k}{j}\right]$$

$$= 3^d - 2^{d+1} + 1$$

If d=6, R = 602 rules

Mining Association Rules: Decoupling

TID	Items
1	Bread, Milk
2	Bread, Diaper, Beer, Eggs
3	Milk, Diaper, Beer, Coke
4	Bread, Milk, Diaper, Beer
5	Bread, Milk, Diaper, Coke

Example of Rules:

{Milk,Diaper} → {Beer} (s=0.4, c=0.67)
{Milk,Beer} → {Diaper} (s=0.4, c=1.0)
{Diaper,Beer} → {Milk} (s=0.4, c=0.67)
{Beer} → {Milk,Diaper} (s=0.4, c=0.67)
{Diaper} → {Milk,Beer} (s=0.4, c=0.5)
{Milk} → {Diaper,Beer} (s=0.4, c=0.5)

Observations:

- All the above rules are binary partitions of the same itemset:
  {Milk, Diaper, Beer}
- Rules originating from the same itemset have identical support but can have different confidence
- Thus, we may **decouple** the support and confidence requirements

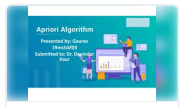
More Related Content

What's hot (20)

Apriori algorithm
Ashis Kumar Ch… • 6.2K views

Eclat algorithm in association rule mining
Deepa Jeya • 22.6K views

Les algorithmes de génération des règles d …
Hajer Trabelsi • 13.5K views



Apriori algorithm
Gaurav Aggarwal • 1.6K views

Association rule mining
Utkarsh Sharma • 340 views

Les algorithmes de génération des règles d …
Hajer Trabelsi • 1.5K views

Viewers also liked (7)

Fp growth algorithm
Pradip Kumar • 15.7K views

Fp growth
Farah M. Altufaili • 13.6K views

Techniques du data mining
Donia Hammami • 9.4K views

Data mining fp growth
Shihab Rahman • 65.3K views

Lecture13 - Association Rules
Albert Orriols-P… • 42.5K views

The comparative study of apriori and FP-growth …
deepti92pawar • 58.5K views

Similar to Apriori and Eclat algorithm in Association Rule Mining (20)

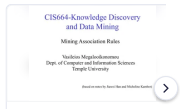
DM -Unit 2-PPT.ppt
raju980973 • 4 views

Rules of data mining
Sulman Ahmed • 683 views

Rules of data mining
Sulman Ahmed • 1.3K views

Chapter 01 Introduction DM.pptx
ssuser957b41 • 2 views

AssociationRule.pdf
WailaBaba • 28 views



Associations.ppt
QuynS90023 • 9 views

Slideshows for you (20)

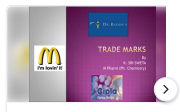
Human values unit 2 notes
Sonam kapil • 169.1K views

Format Of Synopsis
Abdul Rahman … • 324.5K views

research-methodology-ppt
sheetal321 • 317K views

SYNOPSIS WRITING
Prof. Mridul Pan… • 370.9K views

Top 20 Entrpreneurs in India
Sandra4Smiley • 78.9K views



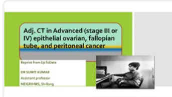
Trade marks
Malla Reddy Co… • 46.7K views

Recently uploaded (20)

Part I: Not all DCIS is "Risky"
bkling • 21 views

Disinfection.pptx
Dr. Rakesh Pras... • 12 views

Physical and Chemical Properties of Dental ...
DrShivaSai • 37 views



advanced stage ovary tumor.pptx
Dr. Sumit KUMAR • 19 views

Let's Talk About It: Ovarian Cancer (Self-Compassion:...
bkling • 24 views

Personal Healthcare
Bassem Matta • 20 views

Related Books

Free with a 30 day trial from Everand

View All

Ebook
Deep Learning with Python,...
Francois Chollet
0 / 5

Ebook
Mastering Python for Finance
James Ma Weiming
5 / 5

Ebook
Machine Trading: Deploying...
Ernest P. Chan
4 / 5

Ebook
NumPy Beginner's Guide
Ivan Idris
5 / 5

Ebook
Information Systems: BCS...
Ian Huke
0 / 5

Ebook
Machine Learning with R - Third...
Brett Lantz
0 / 5

Ebook
Data Lake Development...
Pasupuleti Pradeep
0 / 5

Ebook
Machine Learning with R
Brett Lantz
4 / 5

Related Audiobooks

Free with a 30 day trial from Everand

View All

Audiobook
ALGORITHMIC TRADING: A...
Michael Bateman
5 / 5

Audiobook
Kubernetes: The Updated Guide ...
Steve D. Pountol
3 / 5

Audiobook
Mastering Ethereum
Andreas M. Antonop...
4.5 / 5

Audiobook
Deep Learning: Guide to Machi...
David Feldspar
4.5 / 5

Audiobook
From PMO to VMO: Managing...
Sanjiv Augustine
5 / 5

Audiobook
Data Visualization Guide: Clear...
Alex Campbell
5 / 5

Audiobook
Accelerated DevOps with AI,...
Stephen Fleming
3 / 5

Audiobook
Data Science from Scratch
Steven Cooper
0 / 5

Apriori and Eclat algorithm in Association Rule Mining

1. **Fall 2004, CIS,** Temple University CIS527: Data Warehousing, Filtering, and Mining Lecture 4 • Tutorial: Connecting SQL Server to Matlab using Database Matlab Toolbox • Association Rule MIning Lecture slides taken/modified from: – Jiawei Han (http://www-sal.cs.uiuc.edu/~hanj/DM_Book.html) – Vipin Kumar (http://www-users.cs.umn.edu/~kumar/csci5980/index.html)
2. **Motivation: Association Rule** Mining • Given a set of transactions, find rules that will predict the occurrence of an item based on the occurrences of other items in the transaction Market-Basket transactions TID Items 1 Bread, Milk 2 Bread, Diaper, Beer, Eggs 3 Milk, Diaper, Beer, Coke 4 Bread, Milk, Diaper, Beer 5 Bread, Milk, Diaper, Coke Example of Association Rules {Diaper} → {Beer}, {Milk, Bread} → {Eggs,Coke}, {Beer, Bread} → {Milk}, Implication means co-occurrence, not causality!
3. **Applications: Association Rule** Mining • * ⇒ Maintenance Agreement – What the store should do to boost Maintenance Agreement sales • Home Electronics ⇒ * – What other products should the store stocks up? • Attached mailing in direct marketing • Detecting "ping-ponging" of patients • Marketing and Sales Promotion • Supermarket shelf management
4. **Definition: Frequent Itemset •** Itemset – A collection of one or more items • Example: {Milk, Bread, Diaper} – k-itemset • An itemset that contains k items • Support count (σ) – Frequency of occurrence of an itemset – E.g. σ({Milk, Bread,Diaper}) = 2 • Support – Fraction of transactions that contain an itemset – E.g. s({Milk, Bread, Diaper}) = 2/5 • Frequent Itemset – An itemset whose support is greater than or equal to a minsup threshold TID Items 1 Bread, Milk 2 Bread, Diaper, Beer, Eggs 3 Milk, Diaper, Beer, Coke 4 Bread, Milk, Diaper, Beer 5 Bread, Milk, Diaper, Coke
5. **Definition: Association Rule Example: Beer}Diaper,Milk{** ⇒ 4.0 5 2 |T|)BeerDiaper,,Milk(=== σ s 67.0 3 2)Diaper,Milk( )BeerDiaper,Milk,(=== σ σ c • Association Rule – An implication expression of the form X → Y, where X and Y are itemsets – Example: {Milk, Diaper} → {Beer} • Rule Evaluation Metrics – Support (s) • Fraction of transactions that contain both X and Y – Confidence (c) • Measures how often items in Y appear in transactions that contain X TID Items 1 Bread, Milk 2 Bread, Diaper, Beer, Eggs 3 Milk, Diaper, Beer, Coke 4 Bread, Milk, Diaper, Beer 5 Bread, Milk, Diaper, Coke
6. **Association Rule Mining** Task • Given a set of transactions T, the goal of association rule mining is to find all rules having – support ≥ minsup threshold – confidence ≥ minconf threshold • Brute-force approach: – List all possible association rules – Compute the support and confidence for each rule – Prune rules that fail the minsup and minconf thresholds ⇒ Computationally prohibitive!
7. **Computational Complexity • Given** d unique items: – Total number of itemsets = 2d – Total number of possible association rules: 123 1 1 1 1 +−= ⊠ ⊠ ⊠ ⊠ ⊠ ⊠ ⊠ ⊠ ⊠ ⊠ ⊠ ⊠ ⊠ – ×⊠ ⊠ ⊠ ⊠ ⊠ ⊠ = + − = − = ∑ ∑ dd d k kd j j kd k d R If d=6, R = 602 rules
8. **Mining Association Rules:** Decoupling Example of Rules: {Milk,Diaper} → {Beer} (s=0.4, c=0.67) {Milk,Beer} → {Diaper} (s=0.4, c=1.0) {Diaper,Beer} → {Milk} (s=0.4, c=0.67) {Beer} → {Milk,Diaper} (s=0.4, c=0.67) {Diaper} → {Milk,Beer} (s=0.4, c=0.5) {Milk} → {Diaper,Beer} (s=0.4, c=0.5) TID Items 1 Bread, Milk 2 Bread, Diaper, Beer, Eggs 3 Milk, Diaper, Beer, Coke 4 Bread, Milk, Diaper, Beer 5 Bread, Milk, Diaper, Coke Observations: • All the above rules are binary partitions of the same itemset: {Milk, Diaper, Beer} • Rules originating from the same itemset have identical support but can have different confidence • Thus, we may decouple the support and confidence requirements
9. **Mining Association Rules •** Two-step approach: 1. Frequent Itemset Generation – Generate all itemsets whose support ≥ minsup 1. Rule Generation – Generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset • Frequent itemset generation is still computationally expensive
10. **Frequent Itemset Generation •** Brute-force approach: – Each itemset in the lattice is a candidate frequent itemset – Count the support of each candidate by scanning the database – Match each transaction against every candidate – Complexity ~ O(NMw) => Expensive since M = 2d !!!
11. **Frequent Itemset Generation** Strategies • Reduce the number of candidates (M) – Complete search: M=2d – Use pruning techniques to reduce M • Reduce the number of transactions (N) – Reduce size of N as the size of itemset increases – Use a subsample of N transactions • Reduce the number of comparisons (NM) – Use efficient data structures to store the candidates or transactions – No need to match every candidate against every transaction
12. **Reducing Number of** Candidates: Apriori • Apriori principle: – If an itemset is frequent, then all of its subsets must also be frequent • Apriori principle holds due to the following property of the support measure: – Support of an itemset never exceeds the support of its subsets – This is known as the anti-monotone property of support)()()(:, YsXsYXYX ≥⇒⊆∀
13. **Found to be Infrequent Illustrating** Apriori Principle Pruned supersets
14. **Illustrating Apriori Principle Item** Count Bread 4 Coke 2 Milk 4 Beer 3 Diaper 4 Eggs 1 Itemset Count {Bread,Milk} 3 {Bread,Beer} 2 {Bread,Diaper} 3 {Milk,Beer} 2 {Milk,Diaper} 3 {Beer,Diaper} 3 Itemset Count {Bread,Milk,Diaper} 3 Items (1-itemsets) Pairs (2-itemsets) (No need to generate candidates involving Coke or Eggs) Triplets (3-itemsets) Minimum Support = 3 If every subset is considered, 6 C1 + 6 C2 + 6 C3 = 41 With support-based pruning, 6 + 6 + 1 = 13
15. **Apriori Algorithm • Method: –** Let k=1 – Generate frequent itemsets of length 1 – Repeat until no new frequent itemsets are identified • Generate length (k+1) candidate itemsets from length k frequent itemsets • Prune candidate itemsets containing subsets of length k that are infrequent • Count the support of each candidate by scanning the DB • Eliminate candidates that are infrequent, leaving only those that are frequent
16. **Apriori: Reducing Number** of Comparisons • Candidate counting: – Scan the database of transactions to determine the support of each candidate itemset – To reduce the number of comparisons, store the candidates in a hash structure • Instead of matching each transaction against every candidate, match it against candidates contained in the hashed buckets
17. **Apriori: Implementation Using** Hash Tree 1,4,7 2,5,8 3,6,9 Hash function Suppose you have 15 candidate itemsets of length 3: {1 4 5}, {1 2 4}, {4 5 7}, {1 2 5}, {4 5 8}, {1 5 9}, {1 3 6}, {2 3 4}, {5 6 7}, {3 4 5}, {3 5 6}, {3 5 7}, {6 8 9}, {3 6 7}, {3 6 8} You need: • Hash function • Max leaf size: max number of itemsets stored in a leaf node (if number of candidate itemsets exceeds max leaf size, split the node) 1 5 9 1 4 5 1 3 6 3 4 5 3 6 7 3 6 8 3 5 6 3 5 7 6 8 9 2 3 4 5 6 7 1 2 4 4 5 7 1 2 5 4 5 8 1 5 91 5 9 1 4 51 4 5 1 3 61 3 6 3 4 53 4 5 3 6 7 3 6 8 3 6 73 6 7 3 6 83 6 8 3 5 6 3 5 7 6 8 9 3 5 6 3 5 7 3 5 63 5 6 3 5 73 5 7 6 8 96 8 9 2 3 4 5 6 7 2 3 42 3 4 5 6 75 6 7 1 2 4 4 5 7 1 2 41 2 4 4 5 74 5 7 1 2 5 4 5 8 1 2 51 2 5 4 5 84 5 8
18. **Apriori: Implementation Using** Hash Tree 1 5 9 1 4 5 1 3 6 3 4 5 3 6 7 3 6 8 3 5 6 3 5 7 6 8 9 2 3 4 5 6 7 1 2 4 4 5 7 1 2 5 4 5 8 1 2 3 5 6 3 5 61 2 + 5 61 3 + 61 5 + 3 5 62 + 5 63 + 1 + 2 3 5 6 transaction Match transaction against 11 out of 15 candidates
19. **Apriori: Alternative Search** Methods • Traversal of Itemset Lattice – General-to-specific vs Specific-to-general
20. **• Traversal of** Itemset Lattice – Breadth-first vs Depth-first Apriori: Alternative Search Methods
21. **Bottlenecks of Apriori •** Candidate generation can result in huge candidate sets: – 104 frequent 1-itemset will generate 107 candidate 2- itemsets – To discover a frequent pattern of size 100, e.g., {a1, a2, …, a100}, one needs to generate 2100 ~ 1030 candidates. • Multiple scans of database: – Needs (n +1) scans, n is the length of the longest pattern
22. **ECLAT: Another Method** for Frequent Itemset Generation • ECLAT: for each item, store a list of transaction ids (tids); vertical data layout TID-list
23. **ECLAT: Another Method** for Frequent Itemset Generation • Determine support of any k-itemset by intersecting tid- lists of two of its (k-1) subsets. • 3 traversal approaches: – top-down, bottom-up and hybrid • Advantage: very fast support counting • Disadvantage: intermediate tid-lists may become too large for memory A 1 4 5 6 7 8 9 B 1 2 5 7 8 10 ∧ → AB 1 5 7 8
24. **FP-growth: Another Method** for Frequent Itemset Generation • Use a compressed representation of the database using an FP-tree • Once an FP-tree has been constructed, it uses a recursive divide-and-conquer approach to mine the frequent itemsets
25. **FP-Tree Construction TID Items 1** {A,B} 2 {B,C,D} 3 {A,C,D,E} 4 {A,D,E} 5 {A,B,C} 6 {A,B,C,D} 7 {B,C} 8 {A,B,C} 9 {A,B,D} 10 {B,C,E} null A:1 B:1 null A:1 B:1 B:1 C:1 D:1 After reading TID=1: After reading TID=2:
26. **FP-Tree Construction null A:7 B:5 B:3 C:3 D:1 C:1 D:1 C:3 D:1 D:1 E:1 E:1 TID Items 1** {A,B} 2 {B,C,D} 3 {A,C,D,E} 4 {A,D,E} 5 {A,B,C} 6 {A,B,C,D} 7 {B,C} 8 {A,B,C} 9 {A,B,D} 10 {B,C,E} Pointers are used to assist frequent itemset generation D:1 E:1 Transaction Database Item Pointer A B C D E Header table
27. **FP-growth null A:7 B:5 B:3 C:3 D:1 C:1 D:1 C:3 D:1 E:1 D:1 E:1 Build conditional pattern base** for E: P = {(A:1,C:1,D:1), (A:1,D:1), (B:1,C:1)} Recursively apply FP- growth on P E:1 D:1
28. **FP-growth null A:2 B:1 C:1 C:1 D:1 D:1 E:1 E:1 Conditional Pattern** base for E: P = {(A:1,C:1,D:1,E:1), (A:1,D:1,E:1), (B:1,C:1,E:1)} Count for E is 3: {E} is frequent itemset Recursively apply FP- growth on P E:1 Conditional tree for E:
29. **FP-growth Conditional pattern base for** D within conditional base for E: P = {(A:1,C:1,D:1), (A:1,D:1)} Count for D is 2: {D,E} is frequent itemset Recursively apply FP- growth on P Conditional tree for D within conditional tree for E: null A:2 C:1 D:1 D:1
30. **FP-growth Conditional pattern base for** C within D within E: P = {(A:1,C:1)} Count for C is 1: {C,D,E} is NOT frequent itemset Conditional tree for C within D within E: null A:1 C:1
31. **FP-growth Count for A** is 2: {A,D,E} is frequent itemset Next step: Construct conditional tree C within conditional tree E Continue until exploring conditional tree for A (which has only node A) Conditional tree for A within D within E: null A:2
32. **Benefits of the** FP-tree Structure • Performance study shows – FP-growth is an order of magnitude faster than Apriori, and is also faster than tree-projection • Reasoning – No candidate generation, no candidate test – Use compact data structure – Eliminate repeated database scan – Basic operation is counting and FP-tree building 0 10 20 30 40 50 60 70 80 90 100 0 0.5 1 1.5 2 2.5 3 Support threshold(%) Runtime(sec.) D1 FP-grow th runtime D1 Apriori runtime
33. **Complexity of Association** Mining • Choice of minimum support threshold – lowering support threshold results in more frequent itemsets – this may increase number of candidates and max length of frequent itemsets • Dimensionality (number of items) of the data set – more space is needed to store support count of each item – if number of frequent items also increases, both computation and I/O costs may also increase • Size of database – since Apriori makes multiple passes, run time of algorithm may increase with number of transactions • Average transaction width – transaction width increases with denser data sets – This may increase max length of frequent itemsets and traversals of hash tree (number of subsets in a transaction increases with its width)
34. **Compact Representation of** Frequent Itemsets • Some itemsets are redundant because they have identical support as their supersets • Number of frequent itemsets • Need a compact representation TID A1 A2 A3 A4 A5 A6 A7 A8 A9 A10 B1 B2 B3 B4 B5 B6 B7 B8 B9 B10 C1 C2 C3 C4 C5 C6 C7 C8 C9 C10 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 4 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 5 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 6 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 7 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 8 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 9 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 10 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 11 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 12 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 13 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 14 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 15 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 ∑ = ⊠ ⊠ ⊠ ⊠ ⊠ ⊠ ×= 10 1 10 3 k k
35. **Maximal Frequent Itemset Border Infrequent Itemsets Maximal Itemsets An** itemset is maximal frequent if none of its immediate supersets is frequent
36. **Closed Itemset • Problem** with maximal frequent itemsets: – Support of their subsets is not known – additional DB scans are needed • An itemset is closed if none of its immediate supersets has the same support as the itemset TID Items 1 {A,B} 2 {B,C,D} 3 {A,B,C,D} 4 {A,B,D} 5 {A,B,C,D} 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 Itemset Support {A,B,C} 2 {A,B,D} 3 {A,C,D} 2 {B,C,D} 2 {A,B,C,D} 2
37. **Maximal vs Closed** Frequent Itemsets null AB AC AD AE BC BD BE CD CE DE A B C D E ABC ABD ABE ACD ACE ADE BCD BCE BDE CDE ABCD ABCE ABDE ACDE BCDE ABCDE 124 123 1234 245 345 12 124 24 4 123 2 3 24 34 45 12 2 24 4 4 2 3 4 2 4 Minimum support = 2 # Closed = 9 # Maximal = 4 Closed and maximal Closed but not maximal TID Items 1 ABC 2 ABCD 3 BCE 4 ACDE 5 DE
38. **Maximal vs Closed** Itemsets
39. **Rule Generation • Given** a frequent itemset L, find all non-empty subsets f ⊂ L such that f → L – f satisfies the minimum confidence requirement – If {A,B,C,D} is a frequent itemset, candidate rules: ABC →D, ABD →C, ACD →B, BCD →A, A →BCD, B →ACD, C →ABD, D →ABC AB →CD, AC → BD, AD → BC, BC →AD, BD →AC, CD →AB, • If |L| = k, then there are 2k – 2 candidate association rules (ignoring L → ∅ and ∅ → L)
40. **Rule Generation • How** to efficiently generate rules from frequent itemsets? – In general, confidence does not have an anti- monotone property c(ABC →D) can be larger or smaller than c(AB →D) – But confidence of rules generated from the same itemset has an anti-monotone property – e.g., L = {A,B,C,D}: c(ABC → D) ≥ c(AB → CD) ≥ c(A → BCD) • Confidence is anti-monotone w.r.t. number of items on the RHS of the rule
41. **Rule Generation Lattice of** rules Pruned Rules Low Confidence Rule
42. **Presentation of Association** Rules (Table Form)
43. **Visualization of Association** Rule Using Plane Graph
44. **Visualization of Association** Rule Using Rule Graph

Download Now

About Support Terms Privacy Copyright Cookie Preferences Do not sell or share my personal information Everand

English

© 2024 SlideShare from Scribd