Mining Association Rules
——Constraint-based Association Mining——

徐华
清华大学 计算机系 智能技术与系统国家重点实验室
xuhua@tsinghua.edu.cn

1

Association and Correlations

- Association and Correlations
- Efficient and Scalable Frequent Itemset Mining Methods
- Mining Various Kinds of Association Rules
- From Association Mining to Correlation Analysis
- Constraint-based Association Mining

2

Constraint-based Mining

- Finding all the patterns in a database autonomously? — unrealistic!
  - The patterns could be too many but not focused!
- Data mining should be an interactive process
  - User directs what to be mined using a data mining query language (or a graphical user interface)
- Constraint-based mining
  - User flexibility: provides constraints on what to be mined
  - System optimization: explores such constraints for efficient mining—constraint-based mining

Constraints in Data Mining

- Knowledge type constraint
  - classification, association, etc.
- Data constraint — using SQL-like queries
  - find product pairs sold together in stores in Chicago in Dec.' 02
- Dimension/level constraint
  - in relevance to region, price, brand, customer category
- Rule (or pattern) constraint
  - small sales (price < $10) triggers big sales (sum > $200)
- Interestingness constraint
  - strong rules: min_support ≥ 3%, min_confidence ≥ 60%
- Constraint based mining makes mining effective and efficient

Metarule-Guided Mining of Asso. Rule

- Specify the syntactic form of rules they are interested in mining.
- Metarule can be used as constraints to help improve the efficiency of the mining process.
- Metarules are based on the analyst's experiment, expectations, or intuition regarding the data.

  metarule

  P1(X, W) ∧ P2(X, V) ⇒ buys(X, "educational software")

  matched rule:

  age(X, "30..39") ∧ income(x, "42..48K")

  ⇒ buys (X, "educational software")

  predicate variable and attribute variable

5

Metarule-Guided Mining of Asso. Rule

A rule template (inter-dimention association rule) :

$$P_1 \wedge P_2 \wedge \cdots \wedge P_l \Rightarrow Q_1 \wedge Q_2 \wedge \cdots \wedge Q_r$$

P_i and Q_i: instantiated predicates, predicate variables,

p = l + r: the number of predicates in metarule ,

- Find all the frequent p-predicate sets, Lp
- Have the support or count of the l-predicate subsets of Lp in order to compute the confidence of rules derived from Lp
- Data cube: p-D cuboid and l-D cuboid

6

Constraint pushing: Mining Guided by Additional Rule Constrains

- Hybrid-dimensional association rule mining
  - Constant initiation and aggregate functions
- One example
  - Find the sales of what cheap items that may promote the sales of what expensive items in the same category for Chicago customers in 2004
    - Sales (customer-name, item-name, TID)
    - lives-in (customer-name, region, city)
    - Item (item-name, group, price)
    - Transaction (TID, day, month, year)

Constraint pushing: Mining Guided by Additional Rule Constrains

(1) mine associations as

(2) lives in(C; ; “Chicago") ^ sales+(C; ?{I}; {S})) sales+(C; ?{J}; {T}) (metarule)

(3) from sales

(4) where S.year = 2004 and T.year = 2004 and I.group = J.group

(5) group by C, I.group (dimension level constraints)

(6) having sum(I.price) < 100 and min(J.price) >=500

(constraint pushing? Rule constraints)

(7) with support threshold = 1% (interestingness constraints)

(8) with confidence threshold = 50%

lives in(C; ; “Chicago") ^ sales(C; “Census_CD";)^

sales(C; “MS=Office";)=>sales(C; “MS=SQLServer";); [1:5%; 68%]

Constraint pushing: Mining Guided by Additional Rule Constrains

- How can we use rule constraints to prune the search space?
- what kind of rule constraints can be 'pushed' deep into the mining process and still ensure the completeness of the answer returned for a mining query?
- Categories of rule constraint
  - Anti-monotonic (反单调的)
  - Monotonic (单调的)
  - Succinct (简洁的)
  - Convertible (可转变的)
  - Inconvertible (不可转变的)

9

Anti-Monotonicity in Constraint Pushing

- Anti-monotonicity
  - *When an itemset S violates the constraint, so does any of its superset*
  - *sum(S.Price)* $\leq$ *v* is anti-monotone
  - *sum(S.Price)* $\geq$ *v* is not anti-monotone
- Example. C: range(S.profit) $\leq$ 15 is anti-monotone
  - Itemset *ab* violates C
  - So does every superset of *ab*

TDB (min_sup=2)

TID	Transaction
10	a, b, c, d, f
20	b, c, d, f, g, h
30	a, c, d, e, f
40	c, e, f, g

Item	Profit
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

10

Monotonicity for Constraint Pushing

- Monotonicity
  - *When an itemset S satisfies the constraint, so does any of its superset*
  - *sum(S.Price)* ≥ *v* is monotone
  - *min(S.Price)* ≤ *v* is monotone
- Example. C: range(S.profit) ≥ 15
  - Itemset *ab* satisfies C
  - So does every superset of *ab*

TDB (min_sup=2)

TID	Transaction
10	a, b, c, d, f
20	b, c, d, f, g, h
30	a, c, d, e, f
40	c, e, f, g

Item	Profit
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

Succinctness

- Succinctness:
  - Given A_1, the set of items satisfying a succinctness constraint C, then any set S satisfying C is based on A_1 , i.e., S contains a subset belonging to A_1
  - Idea: Without looking at the transaction database, whether an itemset S satisfies constraint C can be determined based on the selection of items
  - min(S.Price) ≤ v is succinct
  - sum(S.Price) ≥ v is not succinct
- Optimization: If C is succinct, C is pre-counting pushable.

The Apriori Algorithm — Example

Sup$_{min}$=2

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

itemset
{2 3 5}

Scan D

L_3

itemset	sup
{2 3 5}	2

Naïve Algorithm: Apriori + Constraint

Sup$_{min}$=2

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
~~{5}~~	~~3~~

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

itemset	sup
{1 3}	2
~~{2 3}~~	~~2~~
~~{2 5}~~	~~3~~
~~{3 5}~~	~~2~~

C_3

itemset
{2 3 5}

Scan D

L_3

itemset	sup
~~{2 3 5}~~	~~2~~

Constraint:
Sum{S.price} < 5

The Constrained Apriori Algorithm: Push an Anti-monotone Constraint Deep

$Sup_{min}=2$

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
~~{5}~~	~~3~~

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
~~{5}~~	~~3~~

C_2

itemset
{1 2}
{1 3}
~~{1 5}~~
{2 3}
~~{2 5}~~
~~{3 5}~~

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
~~{1 5}~~	~~1~~
~~{2 3}~~	~~2~~
~~{2 5}~~	~~3~~
~~{3 5}~~	~~2~~

L_2

itemset	sup
{1 3}	2
~~{2 3}~~	~~2~~
~~{2 5}~~	~~3~~
~~{3 5}~~	~~2~~

C_3

itemset
~~{2 3 5}~~

Scan D

L_3

itemset	sup
~~{2 3 5}~~	~~2~~

Constraint:
Sum{S.price} < 5

The Constrained Apriori Algorithm: Push a Succinct Constraint Deep

$Sup_{min}=2$

Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Scan D

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
~~{2 3}~~
~~{2 5}~~
~~{3 5}~~

not immediately to be used

Scan D

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
~~{2 3}~~	~~2~~
~~{2 5}~~	~~3~~
~~{3 5}~~	~~2~~

L_2

itemset	sup
{1 3}	2
~~{2 3}~~	~~2~~
~~{2 5}~~	~~3~~
~~{3 5}~~	~~2~~

C_3

itemset
~~{2 3 5}~~

Scan D

L_3

itemset	sup
~~{2 3 5}~~	~~2~~

Constraint:
min{S.price } <= 1

Converting "Tough" Constraints

- **Convert tough constraints into anti-monotone or monotone by properly ordering items**
- **Examine C: avg(*S*.profit) ≥ 25**
  - **Order items in value-descending order**
    - ***<a, f, g, d, b, h, c, e>***
  - **If an itemset *afb* violates C**
    - **So does *afbh, afb****
    - **It becomes anti-monotone!**

TDB (min_sup=2)

TID	Transaction
10	a, b, c, d, f
20	b, c, d, f, g, h
30	a, c, d, e, f
40	c, e, f, g

Item	Profit
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

Strongly Convertible Constraints

- **avg(X) ≥ 25 is convertible anti-monotone w.r.t. item value descending order R: *<a, f, g, d, b, h, c, e>***
  - **If an itemset *af* violates a constraint C, so does every itemset with *af* as prefix, such as *afd***
- **avg(X) ≥ 25 is convertible monotone w.r.t. item value ascending order R^{-1}: *<e, c, h, b, d, g, f, a>***
  - **If an itemset *d* satisfies a constraint *C*, so does itemsets *df* and *dfa*, which having *d* as a prefix**
- **Thus, avg(X) ≥ 25 is strongly convertible**

Item	Profit
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

Can Apriori Handle Convertible Constraint?

- A convertible, not monotone nor anti-monotone nor succinct constraint cannot be pushed deep into an Apriori mining algorithm
  - Within the level wise framework, no direct pruning based on the constraint can be made
  - Itemset df violates constraint C: avg(X)>=25
  - Since adf satisfies C, Apriori needs df to assemble adf, df cannot be pruned
- But it can be pushed into frequent-pattern growth framework!

Item	Value
a	40
b	0
c	-20
d	10
e	-30
f	30
g	20
h	-10

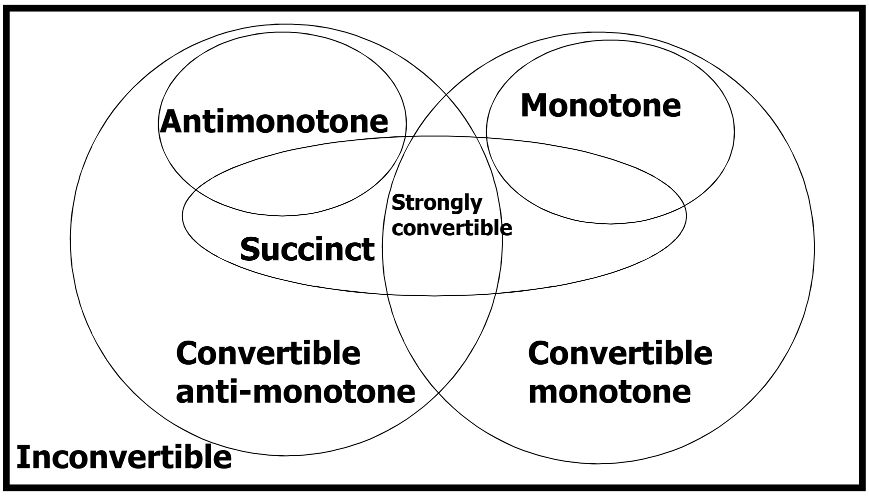
What Constraints Are Convertible?

Constraint	Convertible anti-monotone	Convertible monotone	Strongly convertible
avg(S) ≤ , ≥ v	Yes	Yes	Yes
median(S) ≤ , ≥ v	Yes	Yes	Yes
sum(S) ≤ v (items could be of any value, v ≥ 0)	Yes	No	No
sum(S) ≤ v (items could be of any value, v ≤ 0)	No	Yes	No
sum(S) ≥ v (items could be of any value, v ≥ 0)	No	Yes	No
sum(S) ≥ v (items could be of any value, v ≤ 0)	Yes	No	No
……			

Constraint-Based Mining—A General Picture

Constraint	Antimonotone	Monotone	Succinct
$v \in S$	no	yes	yes
$S \supseteq V$	no	yes	yes
$S \subseteq V$	yes	no	yes
$\min(S) \leq v$	no	yes	yes
$\min(S) \geq v$	yes	no	yes
$\max(S) \leq v$	yes	no	yes
$\max(S) \geq v$	no	yes	yes
$count(S) \leq v$	yes	no	weakly
$count(S) \geq v$	no	yes	weakly
$sum(S) \leq v$ ($a \in S, a \geq 0$)	yes	no	no
$sum(S) \geq v$ ($a \in S, a \geq 0$)	no	yes	no
$range(S) \leq v$	yes	no	no
$range(S) \geq v$	no	yes	no
$avg(S)\ \theta\ v, \theta \in \{=, \leq, \geq\}$	convertible	convertible	no
$support(S) \geq \xi$	yes	no	no
$support(S) \leq \xi$	no	yes	no

A Classification of Constraints



Summary

- Concept of Association rule mining
- Association rule categories
- Apriori association rule mining
- FP-tree growth association rule mining
- Mining various kinds of association rules
- Constraint based association rule mining

23

Thanks !

24