

Association Rules: Past, Present & Future

Ramakrishnan Srikant

www.almaden.ibm.com/cs/people/srikant/

R. Srikant

Talk Outline

- Association Rules
 - Motivation & Definition
 - Most Popular Computation Approach
- Other Computation Approaches
- Extensions
- Interest Measures
- Future Directions

Motivation

- Many organizations have amassed massive data (running into several gigabytes and more)
 - Retailing: Sears, Safeway, K-Mart, Proctor & Gamble
 - Finance: American Express, Citicorp
 - Insurance: Prudential
 - Transportation: United Airlines
 - Hospitals
- Potential goldmine of valuable business information

Association Rules

- Given:
 - a database of transactions
 - each transaction is a set of items
- *Example*: 30% of transactions that contain beer also contain diapers; 5% of transactions contain these items
 - 30% : *confidence* of the rule
 - 5% : *support* of the rule
- Find all association rules that satisfy user-specified minimum support and minimum confidence constraints.
- We are interested in *finding* all rules rather than *verifying* if a rule holds.

Association Rules (cont.)

- Problem introduced in SIGMOD '93 paper, “Mining association rules between sets of items in large databases” by R. Agrawal, T. Imielinski, and A. Swami.
- Search on “association rules”:
 - 3369 papers in Citeseer.
 - 29,300 hits in Google.

Application Examples

- Market Basket Analysis
 - “ * \Rightarrow Maintenance Agreement ”
What the store should do to boost Maintenance Agreement sales?
 - “Home Electronics \Rightarrow * ”
What other products should the store stock up on if the store has a sale on Home Electronics?
- HIC Australia “success story” (Nearhos et al., VLDB '96)
 - associations between medical payment codes
 - saved \$500,000 per year per state
- Reducing telecommunications order failures.
 - set of orders (transactions)
 - each order has around 3.5 sub-parts (USOCs), plus failure code (RMA) if automated processing failed
 - find sets of USOCs which lead to failure
 - only 2.5% of orders fail, 25 different failure codes

Problem Decomposition

1. Find all sets of items that have minimum support (*frequent itemsets*).
2. Use the frequent itemsets to generate the desired rules.

Problem Decomposition – Example

Transaction ID	Items Bought
1	Shoes, Shirt, Jacket
2	Shoes, Jacket
3	Shoes, Jeans
4	Shirt, Sweatshirt

For Minimum Support = 50% = 2 transactions,
and Minimum Confidence = 50%:

Frequent Itemset	Support
{Shoes}	75%
{Shirt}	50%
{Jacket}	50%
{Shoes, Jacket}	50%

For the rule Shoes \Rightarrow Jacket :

- Support = Support({Shoes, Jacket}) = 50%
- Confidence = $\frac{\text{Support}(\{\text{Shoes, Jacket}\})}{\text{Support}(\{\text{Shoes}\})} = \frac{50}{75} = 66.6\%$.

Jacket \Rightarrow Shoes has 50% support and 100% confidence.

Problem Statement

- $\mathcal{I} = \{i_1, i_2, \dots, i_m\}$: a set of literals, called items.
- Transaction T : a set of items such that $T \subseteq \mathcal{I}$.
- Database \mathcal{D} : a set of transactions.
- A transaction T *contains* X , a set of some items in \mathcal{I} , if $X \subseteq T$.
- An *association rule* is an implication of the form $X \Rightarrow Y$, where $X, Y \subset \mathcal{I}$
- The rule $X \Rightarrow Y$ holds in the transaction set \mathcal{D} with *confidence* c if $c\%$ of transactions in \mathcal{D} that contain X also contain Y .
- The rule $X \Rightarrow Y$ has *support* s in the transaction set \mathcal{D} if $s\%$ of transactions in \mathcal{D} contain $X \cup Y$.

Find all rules that have support and confidence greater than user-specified minimum support and minimum confidence.

The Apriori Algorithm

- L_k : Set of frequent itemsets of size k (those with minimum support).
- C_k : Set of candidate itemsets of size k (potentially frequent itemsets)

$L_1 = \{\text{frequent items}\};$

for ($k = 1; L_k \neq \emptyset; k++$) **do**

begin

$C_{k+1} =$ New candidates generated from L_k ;

foreach transaction t in the database **do**

 Increment the count of all candidates in C_{k+1} that
 are contained in t .

$L_{k+1} =$ Candidates in C_{k+1} with minimum support.

end

Answer = $\bigcup_k L_k$;

- R. Agrawal and R. Srikant, “Fast Algorithms for Mining Association Rules”, VLDB '94
- H. Mannila, H. Toivonen and A.I. Verkamo, “Efficient Algorithms for Discovering Association Rules”, KDD '94

Apriori – Example

Dataset \mathcal{D}

TID	Items
10	1 3 4
20	2 3 5
30	1 2 3 5
40	2 5

Minimum Support = 50% = 2 trans.

Scan \mathcal{D} →

Itemset	Sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

Itemset	Sup.
{1}	2
{2}	3
{3}	3
{5}	3

Scan \mathcal{D} →

Itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

Itemset	Sup.
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

Itemset	Sup.
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

Scan \mathcal{D} →

Itemset
{2 3 5}

Itemset	Sup.
{2 3 5}	2

Itemset	Sup.
{2 3 5}	2

Apriori Candidate Generation

Monotonicity Property: All subsets of a frequent itemset are frequent.

Given L_k , generate C_{k+1} in two steps:

$$L_3 = \begin{array}{|l} \{1\ 2\ 3\} \\ \{1\ 2\ 4\} \\ \{1\ 3\ 4\} \\ \{1\ 3\ 5\} \\ \{2\ 3\ 4\} \end{array}$$

1. *Join Step* : Join L_k with L_k , with the join condition that the first $k - 1$ items should be the same **and** $l^1[k] < l^2[k]$.

Now, $C_4 = \{ \{1\ 2\ 3\ 4\}, \{1\ 3\ 4\ 5\} \}$.

2. *Prune Step* : Delete all candidates which have a non-frequent subset.

Now, $C_4 = \{ \{1\ 2\ 3\ 4\} \}$.

On-The-Fly vs. Apriori Candidate Generation

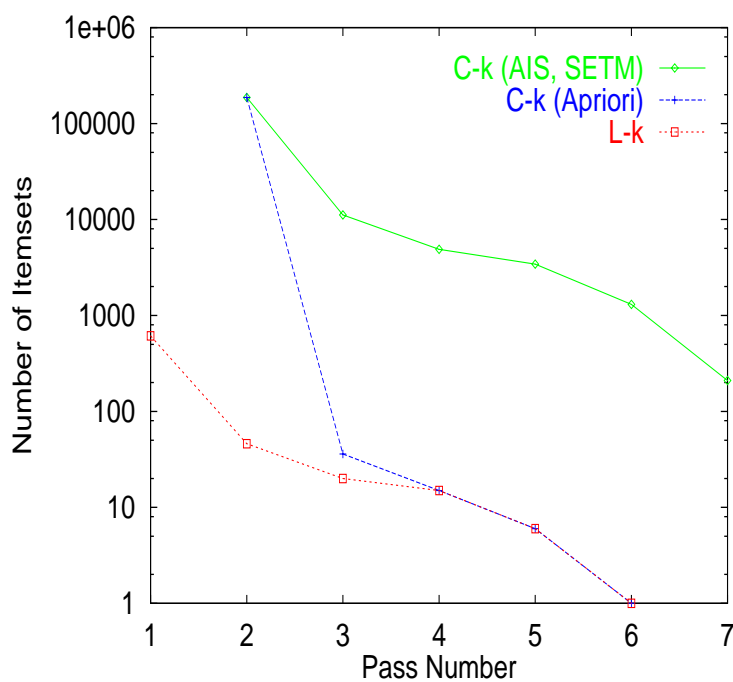
$$L_3 = \begin{array}{|l} \{1\ 2\ 3\} \\ \{1\ 2\ 4\} \\ \{1\ 3\ 4\} \\ \{1\ 3\ 5\} \\ \{2\ 3\ 4\} \end{array}$$

Transaction T : $\{1\ 2\ 3\ 4\ 5\}$

$\{1\ 2\ 3\}$ contained in T .

\Rightarrow Generate $\{1\ 2\ 3\ 4\}$ and $\{1\ 2\ 3\ 5\}$.

Similarly, generate $\{1\ 2\ 4\ 5\}$, $\{1\ 3\ 4\ 5\}$ and $\{2\ 3\ 4\ 5\}$.



Finding Candidates Contained in a Transaction

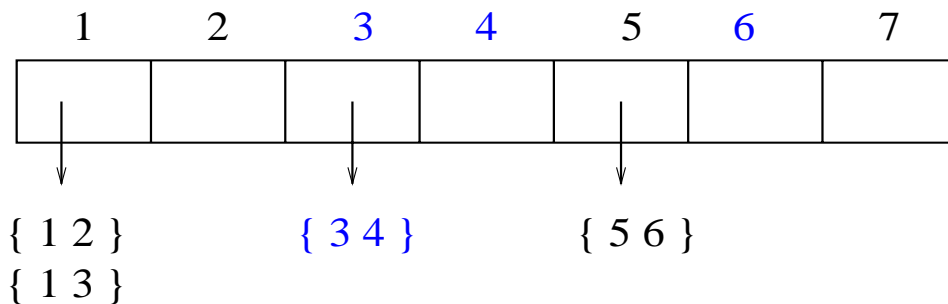
Given

- a transaction T and
- a set of candidates C_k ,

find all members of C_k which are contained in T .

$C_2 : \{ \{1, 2\}, \{1, 3\}, \{3, 4\}, \{5, 6\} \}$

$T : \{3, 4, 6\}$

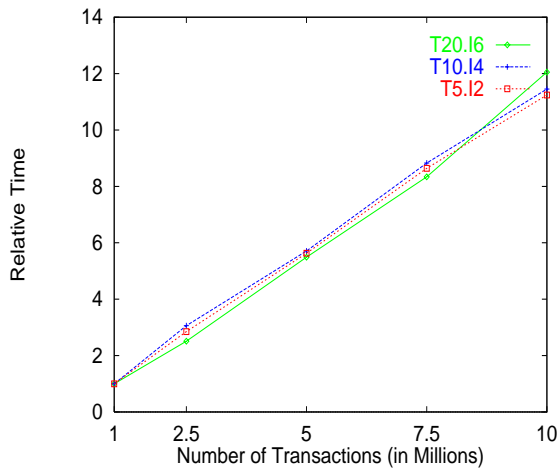


Only check itemsets in buckets corresponding to 3, 4, and 6, i.e., $\{3,4\}$.

- avg. number of items in trans. \ll total number of items
- generalized into a *hash-tree*

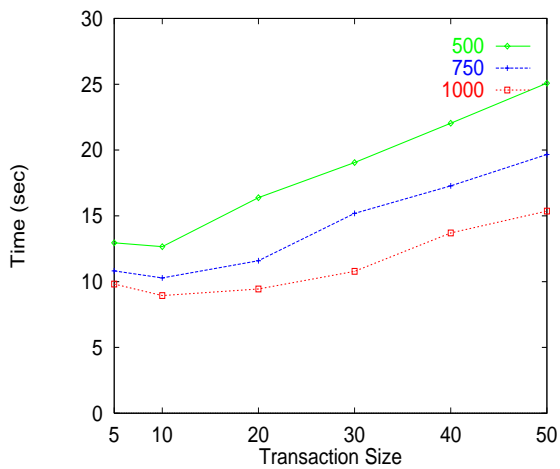
Scaleup

Number of transactions scale-up:



- 0.75% support
- similar results at other levels

Transaction size scale-up:



- Database size kept constant

Apriori vs. Earlier Algorithms

- Apriori 3 to 10 times faster than AIS.
- Apriori 3 to more than 100 times faster than SETM.
- Performance gap increases with problem size.
- Apriori scales linearly with the number of transactions.
- Apriori also has excellent scale-up properties with respect to the transaction size and the number of items in the database.

Talk Outline

- Association Rules
- Other Computation Approaches
 - Sampling
 - Transaction IDs
 - Maximal Associations
 - Data Projection
 - Constraints
- Extensions of the Concept
- Interest Measures
- Future Directions

Sampling

- Use sampling to reduce the number of passes
- Approach:
 1. Run algorithm on a sample \mathcal{D}_S of the dataset \mathcal{D} .
 2. Find support in \mathcal{D} for
 - “Expected Frequent”: all itemsets that were frequent in \mathcal{D}_S , plus
 - “Negative Border”: those itemsets that were not frequent in \mathcal{D}_S , but all of whose subsets were frequent in \mathcal{D}_S .
 3. If any of the itemsets in the negative border are frequent in \mathcal{D} , need mop-up pass for potentially frequent extensions of those itemsets.
- Typically run on \mathcal{D}_S at a lower support to minimize size and number of mop-up passes.

Sampling (cont.)

- Pro: reduces the number of passes.
- Con: have to count significantly (or substantially) more candidates.
- H. Toivonen, “Sampling Large Databases for Association Rules”, VLDB '96.
- S. Brin, R. Motwani, J.D. Ullman and S. Tsur, “Dynamic Itemset Counting and Implication Rules for Market Basket Data”, SIGMOD '97.

Transaction IDs

- For each itemset, keep list of transaction ids that support the itemset.
- Find support of “abc” by merging lists of “ab” and “ac”.
- Generate itemsets in depth-first manner (a, ab, ac, abc, ...) to minimize disk I/O.
- Pro: substantially faster for longer associations
- Con: substantially slower for shorter associations
- M.J. Zaki et al., “New Algorithms for Fast Discovery of Association Rules”, KDD '97.

Maximal Associations

- R.J. Bayardo, “Efficiently Mining Long Patterns from Databases”, SIGMOD '98.
 - Finds only the maximal patterns.
 - Scales roughly linearly in the number of maximal patterns!
- R. Bayardo, R. Agrawal and D. Gunopulos, “Constraint-Based Rule Mining in Large, Dense Series Databases”, ICDE '99.
 - Fixed RHS
 - Finds all rules whose confidence is significantly higher than any of their simplifications.
 - Prune based on confidence.

Data Projection

- Generate associations in depth-first manner.
- Project database down the tree.
- Example: To count all itemsets starting with $\{a, b\}$, with possible extensions $\{d, f, g\}$:
 - Select transactions that contain a and b .
 - Project only d, f and g from these transactions.
- Pro: substantially faster for longer associations.
- Cons: database cannot be much larger than memory, not much speedup if most patterns are short.
- R.C. Agarwal et al., “Depth First Generation of Long Patterns”, KDD 2000.
- J. Han, J. Pei and Y. Yin, “Mining Frequent Patterns without Candidate Generation”, SIGMOD 2000.

Item Constraints

- Users are often interested in a subset of rules.
- Can express constraints as boolean expressions over (the presence of) items.
 - (Shirts AND Shoes) OR (Outerwear AND NOT Hiking Boots)
- R. Srikant, Q. Vu and R. Agrawal, “Mining Association Rules with Item Constraints”, KDD '97.

Approach Overview

1. Find L , the set of all *frequent itemsets* (those with minimum support) satisfying the constraint \mathcal{B} .
2. Count the support of subsets of the frequent itemsets in L .
3. Generate rules from the frequent itemsets in L .
 - $\text{support}(AB \Rightarrow CD) = \text{support}(ABCD)$
 - $\text{confidence}(AB \Rightarrow CD) = \frac{\text{support}(ABCD)}{\text{support}(AB)}$

Can we push the constraint?

For any frequent itemset with k items that satisfies \mathcal{B}

- there is a subset with $k - 1$ items that satisfies \mathcal{B} , **unless**
- the itemset corresponds to a disjunct in \mathcal{B} with exactly k non-negated terms.

Example:

- let $\mathcal{B} = (1 \wedge 2) \vee (4 \wedge \neg 5)$
- $\{1\ 2\ 4\}$ has a subset $\{1\ 2\}$ that satisfies \mathcal{B}
- $\{1\ 2\}$ does not have any subset that satisfies \mathcal{B} , but corresponds to the disjunct " $1 \wedge 2$ "

Constraint Transformation

1. Generate a set of *selected items* \mathcal{S} such that any itemset that satisfies the constraint \mathcal{B} will contain at least one selected item.
2. Generate only candidates that contain selected items.
3. Discard frequent itemsets that do not satisfy \mathcal{B} .

Example:

- let the set of items be $\{1, 2, 3, 4, 5\}$
- if $\mathcal{B} = (1 \wedge 2) \vee 3$
 - \mathcal{S} could be $\{1, 3\}$, $\{2, 3\}$ or $\{1, 2, 3, 4, 5\}$
- if $\mathcal{B} = (1 \wedge 2) \vee \neg 3$
 - \mathcal{S} could be $\{1, 2, 4, 5\}$

Choosing Selected Items

- Assume that \mathcal{B} is in DNF (without loss of generality)
 - $\mathcal{B}: D_1 \vee D_2 \vee \dots \vee D_m$
 - $D_i: \alpha_{i1} \wedge \alpha_{i2} \wedge \dots \wedge \alpha_{in_i}$
 - α_{ij} : either $\langle item \rangle$ or NOT $\langle item \rangle$
- Choose one element from each disjunct in \mathcal{B}
 - if $\langle item \rangle$, add the item to \mathcal{S} .
 - if NOT $\langle item \rangle$, add all the other items (in the dataset) to \mathcal{S} .
- Heuristic: minimize the sum of the supports of the elements in \mathcal{S} .

What Constraints can be Pushed?

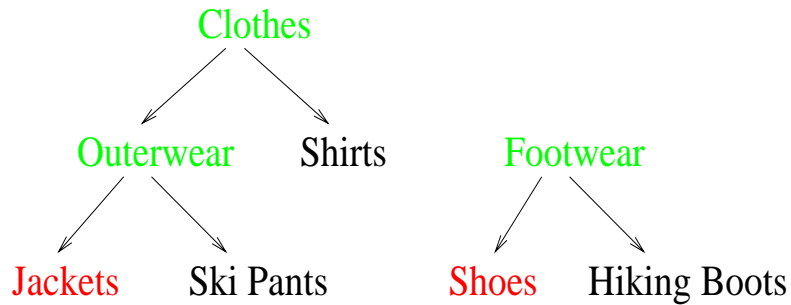
- J. Pei and J. Han, “Can We Push More Constraints into Frequent Pattern Mining?”, KDD 2000.
- Classify constraints into succinct, anti-monotone, monotone, convertible and inconvertible.
- The first four classes can be pushed (to varying degrees).
- M. Garofalakis, R. Rastogi and K. Shim, “SPIRIT: Sequential Pattern Mining with Regular Expression Constraints”, VLDB '99.

Talk Outline

- Association Rules
- Other Computation Approaches
- Extensions
 - Taxonomies
 - Quantitative Association Rules
 - Sequential Patterns
- Interest Measures
- Future Directions

Generalized Association Rules

- Given a taxonomy \mathcal{T} :

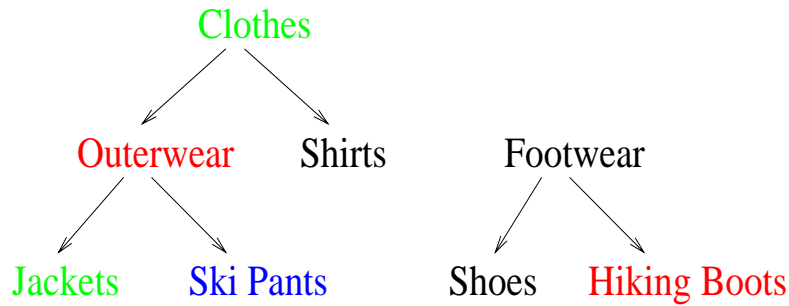


- Find associations between items at any level of the taxonomy
- A transaction {Jacket, Shoes} supports the rules
 - Jacket \Rightarrow Shoes,
 - Outerwear \Rightarrow Footwear,
 - Clothes \Rightarrow Shoes, etc.

Motivation:

- Rules at lower levels may not have minimum support.
- Replace many specialized rules with one general rule.
- Use taxonomy information to identify interesting rules.

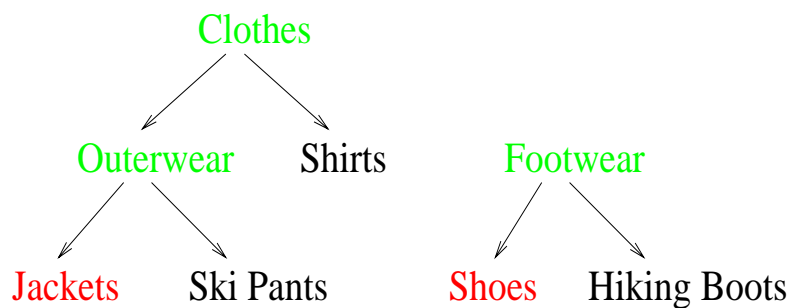
Generalized Association Rules (cont.)



- “Outerwear \Rightarrow Hiking Boots” may be a valid rule, even if
 - “Jackets \Rightarrow Hiking Boots” doesn’t have min. support .
 - “Clothes \Rightarrow Hiking Boots” doesn’t have min. confidence.
- $\text{support}(\text{“Outerwear} \Rightarrow \text{Hiking Boots”})$ is not equal to $\text{support}(\text{“Jackets} \Rightarrow \text{Hiking Boots”}) + \text{support}(\text{“Ski Pants} \Rightarrow \text{Hiking Boots”})$.

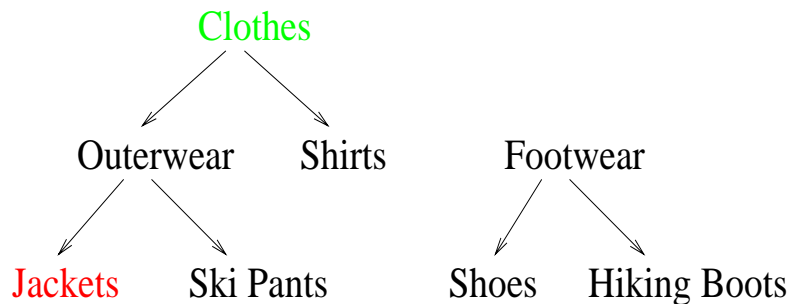
Approach

- Add all ancestors of each item in a transaction to the transaction.
 - *Example:* A transaction {Jackets, Shoes} is replaced with {Jackets, Outerwear, Clothes, Shoes, Footwear}.
- Run the algorithm for mining association rules over these “extended transactions”.



Enhancements

- Pre-compute ancestors.
- Only add relevant ancestors to transactions.
 - If “Jacket” is present in a transaction T , but “Clothes” is not in any of the candidates, don't add “Clothes” to T .
 - Combine with pre-computing ancestors.
- Prune candidate itemsets that contain an item and its ancestor.
 - $\text{support}(\{\text{Jacket}, \text{Clothes}\})$ equals $\text{support}(\{\text{Jacket}\})$.
 - Pruning C_2 is sufficient.



- R. Srikant and R. Agrawal, “Mining Generalized Association Rules”, VLDB '95.
- J. Han and Y. Fu, “Discovery of Multiple-Level Association Rules from Large Databases”, VLDB '95.

Talk Outline

- Association Rules
- Other Computation Approaches
- Extensions
 - Taxonomies
 - Quantitative Association Rules
 - Sequential Patterns
- Interest Measures
- Future Directions

Quantitative Associations

- Given a relational table:
 - a set of records
 - each record has categorical & quantitative attributes
- Example: “30% of married people between age 45 and 60 have at least 2 cars; 5% of records have these properties”
- Find all association rules that satisfy user-specified minimum support and minimum confidence constraints.
- Can we map this problem to boolean associations?

Mapping to Boolean Associations

RecID	Age	Married	#Cars
100	23	No	1
200	25	Yes	1
300	29	No	0
400	34	Yes	2
500	38	Yes	2

RecID	Age: 20..29	Age: 30..39	Married: Yes	Married: No	#Cars:	#Cars:
100	1	0	0	1	0	1
200	1	0	1	0	0	1
300	1	0	0	1	1	0
400	0	1	1	0	0	0
500	0	1	1	0	0	0

Rule: $\langle \text{Age: } 30..39 \rangle$ and $\langle \text{Married: Yes} \rangle \Rightarrow \langle \text{NumCars: } 2 \rangle$
 (40% support, 100% confidence)

Mapping Woes

- “MinSup”: small intervals \Rightarrow miss rules because of low support
- “MinConf”: large intervals \Rightarrow miss rules because of low confidence

RecID	Age	Married	NumCars
100	23	No	1
200	25	Yes	1
300	29	No	0
400	34	Yes	2
500	38	Yes	2

Rule	Support	Confidence
$\langle \text{NumCars: } 0 \rangle \Rightarrow \langle \text{Married: No} \rangle$	20%	100%
$\langle \text{NumCars: } 0..1 \rangle \Rightarrow \langle \text{Married: No} \rangle$	40%	66%

Mapping Woes: Solution

Solution:

- use small intervals, and
- combine adjacent intervals.

But...

- execution time high
- many similar rules

Note: Not meaningful to combine categorical attribute values unless a taxonomy is present.

Approach

- How do we reduce execution time?
 - *maxsupport* limit for combining adjacent intervals.
- Should we partition a quantitative attribute?
If so, how many partitions?
 - Partial completeness measure.
- How do we deal with similar rules?
 - “Greater-than-expected-value” interest measure.
- How do we compute the rules?
 - Extend algorithm for boolean associations.
- R. Srikant and R. Agrawal, “Mining Quantitative Association Rules in Large Relational Tables”, SIGMOD '96
- K. Yoda, T. Fukuda, Y. Morimoto, S. Morishita and T. Tokuyama, “Computing Optimized Rectilinear Regions for Association Rules”, KDD '97.

Talk Outline

- Association Rules
- Other Computation Approaches
- Extensions
 - Taxonomies
 - Quantitative Association Rules
 - Sequential Patterns
- Interest Measures
- Future Directions

Sequential Patterns

- Given:
 - A database of customer transactions
 - Each transaction is a set of items
- Example: 10% of customers bought “shirts” and “jackets” in one transaction, followed by “shoes” in another transaction.
 - 10% is called the *support* of the pattern
- Find all sequential patterns supported by more than a user-specified percentage of customers.
- Constraints
 - max/min time gap between elements
 - “sliding window” transactions
- Applications:
 - attached mailing
 - customer satisfaction
 - medical research
- R. Srikant and R. Agrawal, “Mining Sequential Patterns: Generalizations and Performance Improvements”, EDBT '96.

Talk Outline

- Association Rules
- Other Computation Approaches
- Extensions
- Interest Measures
- Future Directions

“Simple” Interest Measures

- Statistical Measures, e.g., p-value of independence test.
- “Lift”: ratio of support to expected support assuming independence.
- These measures are not based on what the other rules are.

Interest Measures based on Other Rules

- Clothes \Rightarrow Shoes
 - 8% support, 70% confidence
- Quarter of sales of Clothes are Jackets
- Jackets \Rightarrow Shoes
 - expect 2% support, 70% confidence
- Interesting rule if support/confidence is greater from “expected” value.
- User-specified “interest level”
- R. Srikant and R. Agrawal, “Mining Generalized Association Rules”, VLDB '95.

Interest Measures based on Other Rules (cont.)

- Jackets \Rightarrow Shoes [2% support, 70% confidence]
- Jackets and Shirts \Rightarrow Shoes [1.5% support, 71% confidence]
- Second rule not very useful.
- Can take one step further & consider only the direction of the correlation (positive, negative or neutral).
- Extension of a rule is interesting only if its direction is different.
- B. Liu, W. Hsu and Y. Ma, "Pruning and Summarizing the Discovered Associations", KDD '99.

Future Directions

- Faster computation (still an active area!)
- What are the interesting rules?
- Privacy
 - Can we find association rules while preserving privacy at the individual transaction level?
 - R. Agrawal and R. Srikant, “Privacy Preserving Data Mining”, SIGMOD 2000.
 - Y. Lindell and B. Pinkas, “Privacy Preserving Data Mining”, Crypto 2000.