Association Rule Mining

- Association rule mining
  - Problem, Concept, Measures
- AR Mining Algorithm – Apriori
- Comments on Apriori
- Criticism to support and confidence

What is Association Rule Mining (ARM)?

- Association rule mining:
  - Finding frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories.
- Examples.
  - Rule form: "Body $\Rightarrow$ Head [support, confidence]".
  - major(x, "CS") ^ takes(x, "DB") $\Rightarrow$ grade(x, "A") [1%, 75%]
  - buys(x, "diapers") $\Rightarrow$ buys(x, "beers") [0.5%, 60%]

Transaction Database

Transaction	Items bought
100	Coke, Milk
200	Beer, Diapers, Nuts
300	Diapers, Chips, Ice-cream
400	Milk, Nuts, Beer, Diapers
500	Sprite, Biscuits
600	Milk
…	…

Idea of ARM — Frequent Pattern Analysis

- Frequent pattern: A pattern (a set of items, subsequences, substructures, etc.) that occurs frequently in a data set
- Motivation: Finding inherent regularities in data
  - What products were often purchased together?— Beer and diapers?!
  - What are the subsequent purchases after buying a PC?
  - What kinds of DNA are sensitive to this new drug?
  - Can we automatically categorize web documents?
- Applications
  - Basket data analysis, cross-marketing, catalog design, sale campaign analysis, Web log (click stream) analysis, and DNA sequence analysis

Why is Frequent Pattern important?

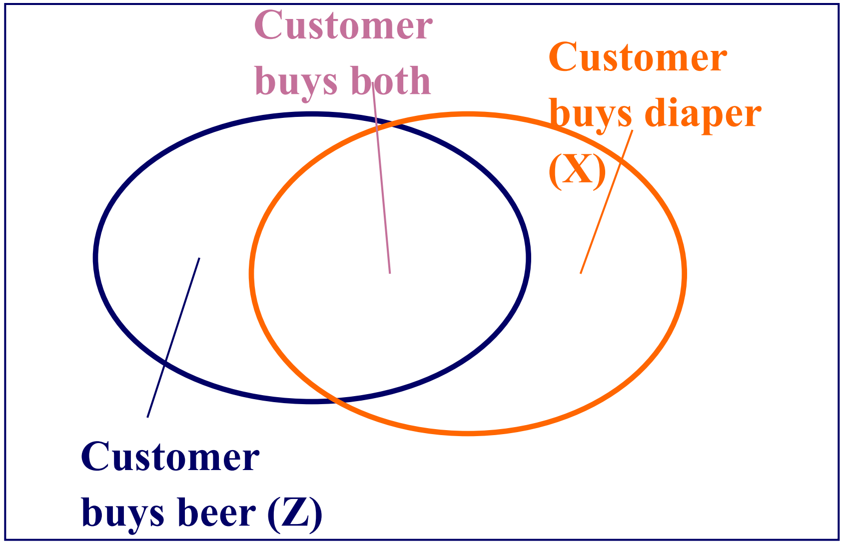
- Discloses an intrinsic and important property of data sets
- Forms the foundation for many essential data mining tasks
  - Association, correlation, and causality analysis
  - Sequential, structural (e.g., sub-graph) patterns
  - Pattern analysis in spatiotemporal, multimedia, time-series, and stream data
  - Classification: associative classification
  - Cluster analysis: frequent pattern-based clustering
  - Data warehousing: iceberg cube and cube-gradient
  - Broad applications

Basic Concept of Association Rule Mining

- Given: (1) database of transactions, (2) each transaction is a list of items (purchased by a customer in a visit)
- Find: all rules that correlate the presence of one set of items with that of another set of items
  - E.g., *98% of people who purchase tires and auto accessories also get automotive services done*
- Applications
  - $* \Rightarrow$ *Maintenance Agreement* (What the store should do to boost Maintenance Agreement sales)
  - *Home Electronics* $\Rightarrow *$ (What other products should the store stock up?)

wildcard: means anything (any item or itemset)

Rule Measures: Support and Confidence



- Find all the rules $X \Rightarrow Z$ with minimum confidence and support
  - support, *s*, probability that a transaction contains both X & Z
  - confidence, *c*, conditional probability that a transaction having X also contains Z

Transaction ID	Items Bought
1000	A,B,C
2000	A,C
3000	A,D
4000	B,E,F

Let minimum support 50%, and minimum confidence 50%, we have

- $A \Rightarrow C$ (50%, 66.6%)
- $C \Rightarrow A$ (50%, 100%)

Rule Measures: Support and Confidence (cont.)

Support :

- Given the association rule $\mathrm{X} \Rightarrow \mathrm{Z}$, the support is the percentage of records consisting of X & Z together, i.e.

  Supp.= P(X & Z)

- indicates the ***statistical significance*** of the association rule.

Confidence :

- Given the association rule $\mathrm{X} \Rightarrow \mathrm{Z}$, the confidence is the percentage of records also having Z, within the group of records having X, i.e.

  Conf.= P(Z| X)

- The degree of correlation in the dataset between the itemset $\{X\}$ and the itemset $\{Z\}$.
- is a measure of the ***rule's strength***.

Variants of Association Rule Mining

- Boolean vs. quantitative associations (Based on the types of values handled)
  - buys(x, "SQLServer") ^ buys(x, "DMBook") $\Rightarrow$ buys(x, "DBMiner") [0.2%, 60%]
  - age(x, "30..39") ^ income(x, "42..48K") $\Rightarrow$ buys(x, "PC") [1%, 75%]
- Single dimension vs. multiple dimensional associations (see examples above)
- Single level vs. multiple-level analysis
  - What brands of beers are associated with what brands of diapers?
- Various extensions
  - Correlation, causality analysis
  - Inter-transaction association rule mining
  - Sequential association rule mining
  - Constraints enforced
    - E.g., small sales (sum < 100) trigger big buys (sum > 1,000)?

Association Rule Mining

- Association rule mining
  - Problem, Concept, Measures
- AR Mining Algorithm – Apriori
- Comments on Apriori
- Criticism to support and confidence

Mining Association Rules—An Example

Transaction ID	Items Bought
2000	A,B,C
1000	A,C
4000	A,D
5000	B,E,F

Min. support 50%
Min. confidence 50%

Frequent Itemset	Support
{A}	75%
{B}	50%
{C}	50%
{A,C}	50%

For rule $A \Rightarrow C$:

support = support($\{A, C\}$) = 50%

confidence = support($\{A, C\}$)/support($\{A\}$) = 66.6%

The Apriori principle:

Any subset of a frequent itemset must be frequent!!!

Mining Association Rules: A Key Step – Mining Frequent Itemsets

- Two key steps in AR mining: (i) Frequent Itemset Mining and (ii) Rule Generation
- 1st key step: Find the *frequent itemsets,* i.e. the sets of items that have minimum support
  - Apply the Apriori principle: A subset of a frequent itemset must also be a frequent itemset
    - i.e., if {*A, B*} is a frequent itemset, both {*A*} and {*B*} should be a frequent itemset
  - Iteratively find the frequent itemsets with cardinality from 1 to *k* (*k*-itemset)
- 2nd key step: Use the frequent itemsets found in previous step to generate association rules

Mining Frequent Itemsets: The Apriori Algorithm

- Pseudo-code:

  C_k: Candidate itemset of size k
  L_k : frequent itemset of size k

```
L_1 = {frequent items};
for (k = 1; L_k != ∅; k++) do begin
    C_{k+1} = candidates generated from L_k;
    for each transaction t in 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 min_support
    end
return ∪_k L_k;
```

- Two important steps:
  - Join Step: C_k is generated by joining $L_{k-1}$with itself
  - Prune Step: Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset

The Apriori Algorithm — An Example

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

How to Generate Candidates?

- Suppose the items in L_{k-1} are listed in an order (ordered list: e.g. {B D A E} being ordered as {A B D E})
- Step 1: self-joining L_{k-1}

  insert into $\boldsymbol{C_k}$

  select $\boldsymbol{p.item_1, p.item_2, \ldots, p.item_{k-1}, q.item_{k-1}}$

  from $\boldsymbol{L_{k-1}\ p, L_{k-1}\ q}$

  where $\boldsymbol{p.item_1=q.item_1, \ldots, p.item_{k-2}=q.item_{k-2}, p.item_{k-1} < q.item_{k-1}}$
- Step 2: pruning

  forall ***itemsets c in*** $\boldsymbol{C_k}$ do

  forall ***(k-1)-subsets s of c*** do

  if *(s is not in* L_{k-1}*)* **then delete** *c* **from** C_k

Example of Generating Candidates:

- *from* L_3 *to* C_4
- L_3={*abc, abd, acd, ace, bcd*}
- Self-joining: $L_3 * L_3$
  - *abcd* from *abc* and *abd*
  - *acde* from *acd* and *ace*
- Pruning:
  - *acde* is removed because *ade* is not in L_3
- C_4={*abcd*}

The Final Step: Rule Generation
(from Frequent Itemsets)

- The *support* is used by the Apriori algorithm to mine the frequent itemsets while the *confidence* is used by the rule generation step to qualify the strength of the association rules
- The rule generation steps include:
  - For each frequent itemset L, generate all nonempty subsets of L
  - For every nonempty subset s of L, generate the rule $R: s \Rightarrow (L\text{-}s)$
  - If R satisfies the minimum confidence, i.e.

    $conf(s \Rightarrow L\text{-}s) = \text{support}(L)/\text{support}(s) \geq min\_conf$

    then rule R is a strong association rule and should be output

Rule Generation (cont.)

An example:

- For L_3={2,3,5}, we have six non-empty subsets: {2}, {3}, {5}, {2,3}, {2,5}, {3,5}. Thus, six candidate rules can be generated:
  - $\{2\} \Rightarrow \{3,5\}$; $\{3\} \Rightarrow \{2,5\}$; $\{5\} \Rightarrow \{2,3\}$;
  - $\{2,3\} \Rightarrow \{5\}$, $\{2,5\} \Rightarrow \{3\}$, $\{3,5\} \Rightarrow \{2\}$

  If any of them satisfies the minimum confidence, it will be output to the end user.

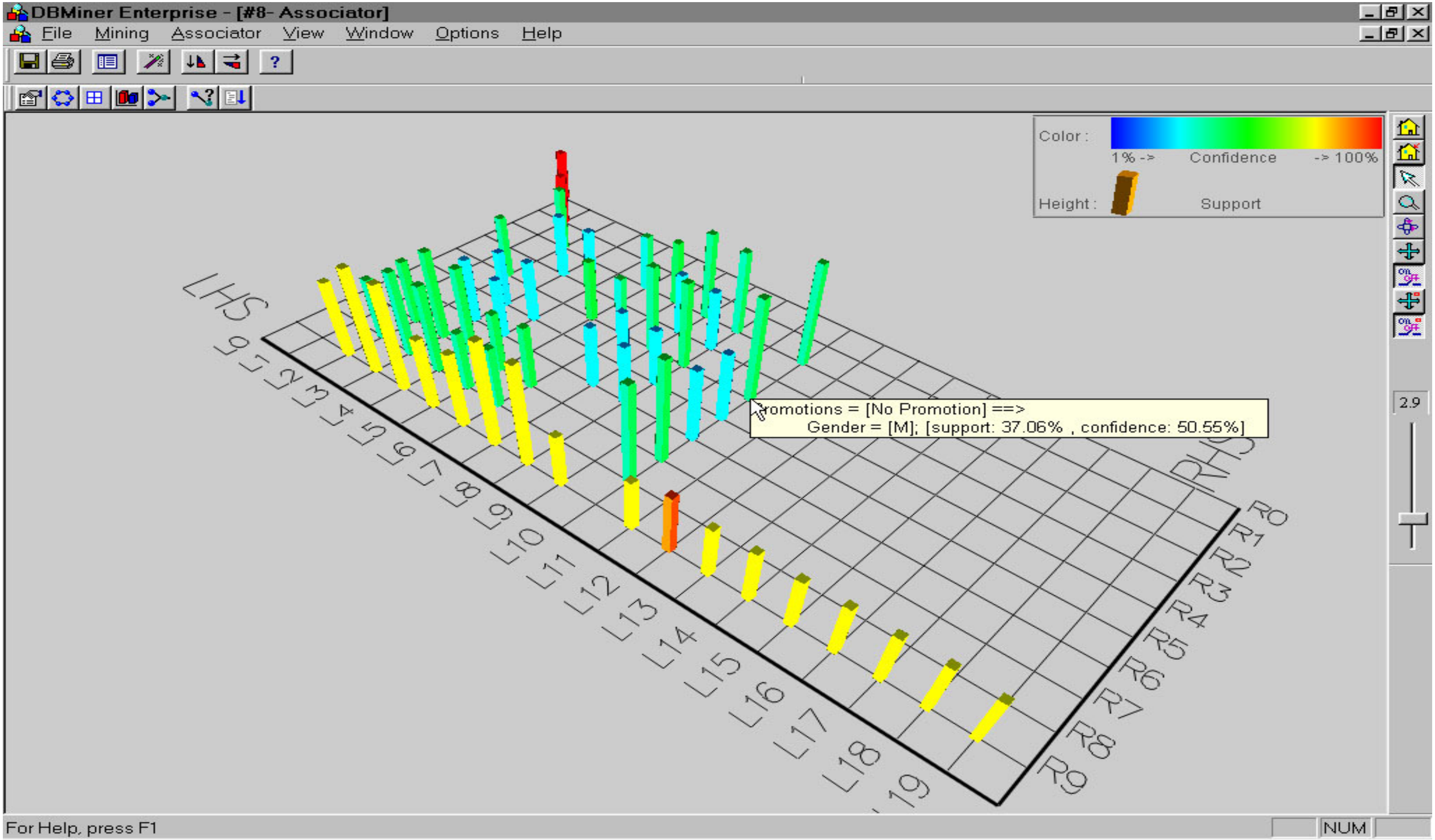
Presentation of Association Rules (Table Form): An Example

	Body	Implies	Head	Supp (%)	Conf (%)	F	G	H	I
1	cost(x) = '0.00~1000.00'	==>	revenue(x) = '0.00~500.00'	28.45	40.4				
2	cost(x) = '0.00~1000.00'	==>	revenue(x) = '500.00~1000.00'	20.46	29.05				
3	cost(x) = '0.00~1000.00'	==>	order_qty(x) = '0.00~100.00'	59.17	84.04				
4	cost(x) = '0.00~1000.00'	==>	revenue(x) = '1000.00~1500.00'	10.45	14.84				
5	cost(x) = '0.00~1000.00'	==>	region(x) = 'United States'	22.56	32.04				
6	cost(x) = '1000.00~2000.00'	==>	order_qty(x) = '0.00~100.00'	12.91	69.34				
7	order_qty(x) = '0.00~100.00'	==>	revenue(x) = '0.00~500.00'	28.45	34.54				
8	order_qty(x) = '0.00~100.00'	==>	cost(x) = '1000.00~2000.00'	12.91	15.67				
9	order_qty(x) = '0.00~100.00'	==>	region(x) = 'United States'	25.9	31.45				
10	order_qty(x) = '0.00~100.00'	==>	cost(x) = '0.00~1000.00'	59.17	71.86				
11	order_qty(x) = '0.00~100.00'	==>	product_line(x) = 'Tents'	13.52	16.42				
12	order_qty(x) = '0.00~100.00'	==>	revenue(x) = '500.00~1000.00'	19.67	23.88				
13	product_line(x) = 'Tents'	==>	order_qty(x) = '0.00~100.00'	13.52	98.72				
14	region(x) = 'United States'	==>	order_qty(x) = '0.00~100.00'	25.9	81.94				
15	region(x) = 'United States'	==>	cost(x) = '0.00~1000.00'	22.56	71.39				
16	revenue(x) = '0.00~500.00'	==>	cost(x) = '0.00~1000.00'	28.45	100				
17	revenue(x) = '0.00~500.00'	==>	order_qty(x) = '0.00~100.00'	28.45	100				
18	revenue(x) = '1000.00~1500.00'	==>	cost(x) = '0.00~1000.00'	10.45	96.75				
19	revenue(x) = '500.00~1000.00'	==>	cost(x) = '0.00~1000.00'	20.46	100				
20	revenue(x) = '500.00~1000.00'	==>	order_qty(x) = '0.00~100.00'	19.67	96.14				
21									
22									
23	cost(x) = '0.00~1000.00'	==>	revenue(x) = '0.00~500.00' AND order_qty(x) = '0.00~100.00'	28.45	40.4				
24	cost(x) = '0.00~1000.00'	==>	revenue(x) = '0.00~500.00' AND order_qty(x) = '0.00~100.00'	28.45	40.4				
25	cost(x) = '0.00~1000.00'	==>	revenue(x) = '500.00~1000.00' AND order_qty(x) = '0.00~100.00'	19.67	27.93				
26	cost(x) = '0.00~1000.00'	==>	revenue(x) = '500.00~1000.00' AND order_qty(x) = '0.00~100.00'	19.67	27.93				
27	cost(x) = '0.00~1000.00' AND order_qty(x) = '0.00~100.00'	==>	revenue(x) = '500.00~1000.00'	19.67	33.23				

Sheet1

Yet Another Example:

Visualization of Association Rule Using Plane Graph



Association Rule Mining

- Association rule mining
  - Problem, Concept, Measures
- AR Mining Algorithm – Apriori
- Comments on Apriori
- Criticism to support and confidence

Is Apriori Fast Enough? — Performance Bottlenecks

- The core of the Apriori algorithm:
  - Use frequent ($k-1$)-itemsets to generate candidate frequent k-itemsets
  - Use database scaning and pattern matching to collect counts for the candidate itemsets
- The bottleneck of *Apriori*: candidate generation
  - Huge candidate sets:
    - 10^4 frequent 1-itemset will generate $>10^7$ candidate 2-itemsets
    - To discover a frequent pattern of size 100, e.g., $\{a_1, a_2, \ldots, a_{100}\}$, one needs to generate $2^{100} \approx 10^{30}$ candidates.
  - Multiple scans of database:
    - Needs ($n + 1$) scans, where n is the length of the longest pattern

Methods to Improve Apriori's Efficiency

- Hash-based itemset counting: A *k*-itemset whose corresponding hashing bucket count is below the threshold cannot be frequent
- Transaction reduction: A transaction that does not contain any frequent k-itemset is useless in subsequent scans
- Partitioning: Any itemset that is potentially frequent in DB must be frequent in at least one of the partitions of DB
- Sampling: mining on a subset of given data, lower support threshold + a method to determine the completeness
- Dynamic itemset counting: add new candidate itemsets only when all of their subsets are estimated to be frequent

Association Rule Mining

- Association rule mining
  - Problem, Concept, Measures
- AR Mining Algorithm – Apriori
- Comments on Apriori
- Criticism to support and confidence

Interestingness Measurements

- Objective measures

  Two popular measurements:

  - *support* and *confidence*

- Subjective measures (Silberschatz & Tuzhilin, KDD95)

  A rule (pattern) is interesting if

  - it is *unexpected* (surprising to the user); and/or
  - *actionable* (the user can do something with it)

Criticism to Support and Confidence

- Example 1: (Aggarwal & Yu, PODS98)
  - Among 5000 students
    - 3000 play basketball
    - 3750 eat cereal
    - 2000 both play basket ball and eat cereal
  - *play basketball* $\Rightarrow$ *eat cereal* [40%, 66.7%] is misleading because the overall percentage of students eating cereal is 75% which is higher than 66.7%.
  - *play basketball* $\Rightarrow$ *not eat cereal* [20%, 33.3%] is far more accurate, although with lower support and confidence

	basketball	not basketball	sum(row)
cereal	2000	1750	3750
not cereal	1000	250	1250
sum(col.)	3000	2000	5000

Criticism to Support and Confidence (Cont.)

- Example 2:
  - X and Y: positively correlated,
  - X and Z, negatively related
  - support and confidence of $X \Rightarrow Z$ dominates

item	Trans#1	Trans#2	Trans#3	Trans#4	Trans#5	Trans#6	Trans#7	Trans#8
X	1	1	1	1	0	0	0	0
Y	1	1	0	0	0	0	0	0
Z	0	1	1	1	1	1	1	1

Rule	Support	Confidence
$X \Rightarrow Y$	25%	50%
$X \Rightarrow Z$	37.50%	75%

Other Interestingness Measures: Interest

- Interest (): $$\frac{P(A \wedge B)}{P(A)P(B)}$$
  - taking both P(A) and P(B) in consideration
  - P(A^B)=P(B)*P(A), if A and B are independent events
  - A and B negatively correlated, if the value is less than 1; otherwise A and B positively correlated
  - a kind of correlation analysis → correlation ≠ association
  - is also called the *lift (ratio)* of the association rule $A \Rightarrow B$ (lift the likelihood of B by a factor of the value returned)

X	1	1	1	1	0	0	0	0
Y	1	1	0	0	0	0	0	0
Z	0	1	1	1	1	1	1	1

Itemset	Support	Interest
X,Y	25%	2
X,Z	37.50%	0.9
Y,Z	12.50%	0.57

References

- R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. SIGMOD'93, 207-216, Washington, D.C.
- **R. Agrawal and R. Srikant. Fast algorithms for mining association rules. VLDB'94 487-499, Santiago, Chile.**
- M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen, and A.I. Verkamo. Finding interesting rules from large sets of discovered association rules. CIKM'94, 401-408, Gaithersburg, Maryland.
- **R. Agrawal and R. Srikant. Mining sequential patterns. ICDE'95, 3-14, Taipei, Taiwan.**
- S. Brin, R. Motwani, and C. Silverstein. Beyond market basket: Generalizing association rules to correlations. SIGMOD'97, 265-276, Tucson, Arizona.
- S. Brin, R. Motwani, J. D. Ullman, and S. Tsur. Dynamic itemset counting and implication rules for market basket analysis. SIGMOD'97, 255-264, Tucson, Arizona, May 1997.
- J. Han, G. Dong, and Y. Yin. Efficient mining of partial periodic patterns in time series database. ICDE'99, Sydney, Australia.
- J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. VLDB'95, 420-431, Zurich, Switzerland.
- J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. SIGMOD'00, 1-12, Dallas, TX, May 2000.
- http://www.rsrikant.com/publications.html