Chapter 5
Frequent Patterns and Association Rule Mining

Outline

- Frequent Itemsets and Association Rule
- APRIORI
- Post-processing
- Applications

Transactional Data

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk

- Definitions:
  - An *item*: an article in a basket, or an attribute-value pair
  - A *transaction*: set of items purchased in a basket; it may have TID (transaction ID)
  - A *transactional dataset*: A set of transactions

Itemsets & Frequent Itemsets

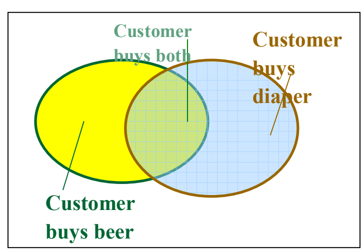
- itemset: A set of one or more items
- k-itemset $X = \{x_1, \ldots, x_k\}$
- *(absolute) support*, or, *support count* of X: Frequency or occurrence of an itemset X
- *(relative) support*, *s*, is the fraction of transactions that contains X (i.e., the probability that a transaction contains X)
- An itemset X is *frequent* if X's support is no less than a *minsup* threshold

Association Rule

- Find all the rules $X \rightarrow Y$ with minimum support and confidence
  - support, s, probability that a transaction contains $X \cup Y$
    - $s = P(X \cup Y)$
      $= \text{support count } (X \cup Y) / \text{number of all transactions}$
  - confidence, c, conditional probability that a transaction having X also contains Y
    - $c = P(X|Y)$
      $= \text{support count } (X \cup Y) / \text{support count } (X)$

Example

Tid	Items bought
10	Beer, Nuts, Diaper
20	Beer, Coffee, Diaper
30	Beer, Diaper, Eggs
40	Nuts, Eggs, Milk
50	Nuts, Coffee, Diaper, Eggs, Milk



- Let minsup = 50%, minconf = 50%
  - Number of all transactions = 5 & Min. support count = 5*50%=2.5 ⇒ 3
  - Items: Beer, Nuts, Diaper, Coffee, Eggs, Milk
  - Freq. Pat.: {Beer}:3, {Nuts}:3, {Diaper}:4, {Eggs}:3, {Beer, Diaper}:3
- Association rules (support, confidence):
  - *Beer → Diaper* (60%, 100%)
  - *Diaper → Beer* (60%, 75%)

Use of Association Rules

- Association rules do not necessarily represent causality or correlation between the two itemsets.
  - $X \Rightarrow Y$ does not mean X causes Y, no Causality
  - $X \Rightarrow Y$ can be different from $Y \Rightarrow X$, unlike correlation
- Association rules assist in Basket data analysis, cross-marketing, catalog design, sale campaign analysis, Web log (click stream) analysis, and DNA sequence analysis.
  - 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 classify web documents?

Computational Complexity of Frequent Itemset Mining

- How many itemsets are potentially to be generated in the worst case?
  - The number of frequent itemsets to be generated is senstive to the minsup threshold
  - When minsup is low, there exist potentially an exponential number of frequent itemsets
  - The worst case is close to M^N where M: # distinct items, and N: max length of transactions when M is large.
- The worst case complexty vs. the expected probability
  - Ex. Suppose Walmart has 10^4 kinds of products
    - The chance to pick up one product 10^{-4}
    - The chance to pick up a particular set of 10 products: $\sim 10^{-40}$
    - What is the chance this particular set of 10 products to be frequent 10^3 times in 10^9 transactions?

Outline

- Frequent Itemsets and Association Rule
- APRIORI
- Post-processing
- Applications

Association Rule Mining

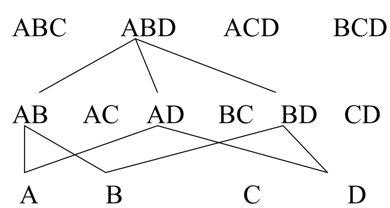
- Major steps in association rule mining
  - Frequent itemsets computation
  - Rule derivation
- Use of support and confidence to measure strength

The Downward Closure Property

- The downward closure property of frequent patterns
  - Any subset of a frequent itemset must be frequent
  - If **{beer, diaper, nuts}** is frequent, so is **{beer, diaper}**
  - i.e., every transaction having {beer, diaper, nuts} also contains {beer, diaper}

Apriori: A Candidate Generation & Test Approach

- A *frequent* (used to be called large) *itemset* is an itemset whose support (S) is $\geq$ minSup.
- Apriori pruning principle:
  - If there is any itemset which is infrequent, its superset should not be generated/tested!



APRIORI

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

The Apriori Algorithm—An Example

$Sup_{min} = 2$

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

L_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

L_2

Itemset	sup
{A, C}	2
{B, C}	2
{B, E}	3
{C, E}	2

C_3

Itemset
{B, C, E}

3^{rd} scan

L_3

Itemset	sup
{B, C, E}	2

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 !=$\emptyset$; 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 $\cup_k L_k$;

Implementation of Apriori

- How to generate candidates?
  - Step 1: self-joining L_k
  - Step 2: pruning
- Example of Candidate-generation
  - L_3={{a,b,c}, {a,b,d}, {a,c,d}, {a,c,e}, {b,c,d}}
  - Self-joining: L_3*L_3
    - *abcd* from *abc* and *abd*
    - *acde* from *acd* and *ace*
  - Pruning:
    - *abcd* is kept since *abc, abd, acd,* and *bcd* are in L_3
    - *acde* is removed because *cde* and *ade* are not in L_3
  - C_4 = {*abcd*}

Self-joining

- L_k is generated by joining L_{k-1} with itself
  - Apriori assumes that items within a transaction or itemset are sorted in lexicographic order.
  - l_1 from L_{k-1} = $\{l_1[1], l_1[2], \ldots, l_1[k-2], l_1[k-1]\}$
  - l_2 from L_{k-1} = $\{l_2[1], l_2[2], \ldots, l_2[k-2], l_2[k-1]\}$
  - Only when first **k-2** items of l_1 and l_2 are in common, and $l_1[k-1] < l_2[k-1]$, these two itemsets are joinable

Further Improvement of the Apriori Method

- Major computational challenges
  - Multiple scans of transaction database
  - Huge number of candidates
  - Tedious workload of support counting for candidates
- Improving Apriori: general ideas
  - Reduce passes of transaction database scans
  - Shrink number of candidates
  - Facilitate support counting of candidates
- Completeness: any association rule mining algorithm should get the same set of frequent itemsets.

Partition: Scan Database Only Twice

- Any itemset that is potentially frequent in DB must be frequent in at least one of the partitions of DB
  - Scan 1: partition database and find local frequent patterns
    - Since the sub-database is relatively smaller than original database, all frequent itemsets are tested in one scan.
  - Scan 2: consolidate global frequent patterns
- A. Savasere, E. Omiecinski, and S. Navathe. An efficient algorithm for mining association in large databases. In *VLDB'95*

DHP: Reduce the Number of Candidates

- A *k*-itemset whose corresponding hashing bucket count is below the threshold cannot be frequent
  - Candidates: a, b, c, d, e
  - Frequent 1-itemset: a, b, d, e
  - Using some hash function, get hash entries: {ab, ad, ae} with bucket count as 3, {bd, bd, be, de} with bucket count as 4…
  - ab is not a candidate 2-itemset if the count of bucket - {ab, ad, ae} is below support threshold
- J. Park, M. Chen, and P. Yu. An effective hash-based algorithm for mining association rules. In *SIGMOD'95*

Transaction Reduction

- A transaction that does not contain any frequent k-itemsets cannot contain any frequent (k+1)-itemsets.
- Such a transaction could be marked or removed from further consideration of (k+1)-itemsets.
  - Less support counting

Sampling for Frequent Patterns

- Select a sample of original database, mine frequent patterns within sample using Apriori with a lower support count threshold than min. support.
- Scan database once to verify frequent itemsets found in sample, only *borders* of closure of frequent patterns are checked
  - Example: check *abcd* instead of *ab, ac, …, etc.*
- Scan database again to find missed frequent patterns
- H. Toivonen. Sampling large databases for association rules. In *VLDB'96*

Outline

- Frequent Itemsets and Association Rule
- APRIORI
- Post-processing
- Applications

Derive rules from frequent itemsets

- Frequent itemsets != association rules
- One more step is required to find association rules
- For each frequent itemset *X*,

  For each proper nonempty subset *A* of *X*,
  - Let *B* = X - *A*
  - A $\Rightarrow$B is an association rule if
    - Confidence (A $\Rightarrow$ B) $\geq$ minConf,

      where support (A $\Rightarrow$ B) = support (AB) and

      confidence (A $\Rightarrow$ B) = support (AB) / support (A)

Example – deriving rules from frequent itemsets

- Suppose {2,3,4} is frequent, with supp=50%
  - Proper nonempty subsets: {2,3}, {2,4}, {3,4}, {2}, {3}, {4}, with supp = 50%, 50%, 75%, 75%, 75%, 75% respectively
  - These generate these association rules:
    - 2,3 => 4, confidence=100%
    - 2,4 => 3, confidence=100%
    - 3,4 => 2, confidence=67%
    - 2 => 3,4, confidence=67%
    - 3 => 2,4, confidence=67%
    - 4 => 2,3, confidence=67%
    - All rules have support = 50%

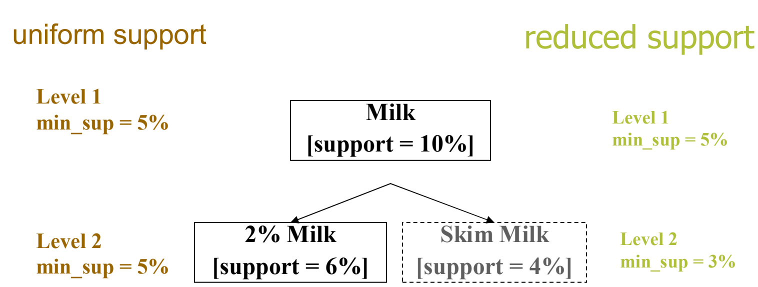
Mining Various Kinds of Association Rules

- Mining multilevel association
- Miming multidimensional association
- Mining quantitative association

Mining Multiple-Level Association Rules

- Items often form hierarchies
- A Top-down strategy is employed, any algorithm could be used for mining at each level.
- Flexible support settings
  - Uniform minimum support for all levels.
  - Items at the lower level are expected to have lower support.
  - Set up user-specific, item or group-based minimum support.
    - Setting particularly low support thresholds for laptop computers and flash drives.

Example



Multi-level Association: Redundancy Filtering

- Some rules may be redundant due to "ancestor" relationships between items
- Example
  - milk $\Rightarrow$ wheat bread [support = 8%, confidence = 70%]
  - 2% milk $\Rightarrow$ wheat bread [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 Multi-Dimensional Association

- Single-dimensional rules:

  buys(X, "milk") $\Rightarrow$ buys(X, "bread")
- Multi-dimensional rules: $\geq$ 2 dimensions or predicates
  - Inter-dimension assoc. rules (*no repeated predicates*)

    age(X,"19-25") $\wedge$ occupation(X,"student") $\Rightarrow$ buys(X, "coke")
  - hybrid-dimension assoc. rules (*repeated predicates*)

    age(X,"19-25") $\wedge$ buys(X, "popcorn") $\Rightarrow$ buys(X, "coke")

Mining Quantitative Associations

- Categorical Attributes and Quantitative Attributes
- Techniques can be categorized by how numerical attributes, such as age or salary are treated
  - Static discretization based on predefined concept hierarchies
  - Dynamic discretization based on data distribution.
  - Clustering: Distance-based association.
    - One dimensional clustering then association

Id	Age	Income	Student	Credit_Rating	Buy_Computer
1	27	75,000	No	600	No
2	25	72,000	No	730	No
3	33	88,000	No	640	Yes
4	50	55,000	No	620	Yes
5	52	34,000	Yes	640	Yes
6	45	30,000	Yes	720	No
7	32	25,000	Yes	740	Yes
8	25	54,000	No	630	No
9	22	35,000	Yes	640	Yes
10	48	67,000	Yes	660	Yes
11	24	64,000	Yes	715	Yes
12	37	62,000	No	710	Yes
13	33	90,000	Yes	650	Yes
14	45	59,000	No	705	No

age	income	student	credit_rating	_comp
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

age	income	student	credit_rating	buy_computer
1	3	0	1	0
1	3	0	2	0
2	3	0	1	1
3	2	0	1	1
3	1	1	1	1
3	1	1	2	0
2	1	1	2	1
1	2	0	1	0
1	1	1	1	1
3	2	1	1	1
1	2	1	2	1
2	2	0	2	1
2	3	1	1	1
3	2	0	2	0

Mining Other Interesting Patterns

- Flexible support constraints (Wang, et al. @ VLDB'02)
  - Some items (e.g., diamond) may occur rarely but are valuable
  - Customized sup_{min} specification and application
- Top-K closed frequent patterns (Han, et al. @ ICDM'02)
  - Hard to specify sup_{min}, but top-k with $length_{min}$ is more desirable

Interestingness Measure: Correlations (Lift)

- The occurrence of itemset A is independent of the occurrence of itemset B if P(AUB)=P(A)P(B); otherwise, itemsets A and B are dependent and correlated as events.
- Measure of dependent/correlated events: lift

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

  - < 1, A is negatively correlated with B.
  - > 1, A and B are positively correlated.
  - = 1, A and B are independent.

Interestingness Measure: Correlations (Lift)

- *play basketball* $\Rightarrow$ *eat cereal* [40%, 66.7%] is misleading
  - The overall % of students eating cereal is 75% > 66.7%.
- *play basketball* $\Rightarrow$ *not eat cereal* [20%, 33.3%] is more accurate, although with lower support and confidence

$$lift(B,C) = \frac{2000/5000}{3000/5000 * 3750/5000} = 0.89$$

$$lift(B,\neg C) = \frac{1000/5000}{3000/5000 * 1250/5000} = 1.33$$

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

Outline

- Frequent Itemsets and Association Rule
- APRIORI
- Post-processing
- Applications

Synthetic Data on Purchase of Phone Faceplates

- A store that sells accessories for cellular phones runs a promotion of faceplates. Customers who purchase multiple faceplates from a choice of six different colors get a discount.
- The store manager wants to know what colors of faceplates customers are likely to purchase together.

Transactions for Purchase of Different-Colored Cellular Phone Faceplates

Transaction	Faceplate Colors Purchased
1	Red, white, green
2	White, orange
3	White, blue
4	Red, white, orange
5	Red, blue
6	White, blue
7	White, orange
8	Red, white, blue, green
9	Red, white, blue
10	Yellow

Phone Faceplate Data in Binary Matrix Format

Transaction	Red	White	Blue	Orange	Green	Yellow
1	1	1	0	0	1	0
2	0	1	0	1	0	0
3	0	1	1	0	0	0
4	1	1	0	1	0	0
5	1	0	1	0	0	0
6	0	1	1	0	0	0
7	1	0	1	0	0	0
8	1	1	1	0	1	0
9	1	1	1	0	0	0
10	0	0	0	0	0	1

Item Sets with Support Count of At Least Two (20%)

Item Set	Support (Count)
{red}	6
{white}	7
{blue}	6
{orange}	2
{green}	2
{red, white}	4
{red, blue}	4
{red, green}	2
{white, blue}	4
{white, orange}	2
{white, green}	2
{red, white, blue}	2
{red, white, green}	2

Generating Association Rule

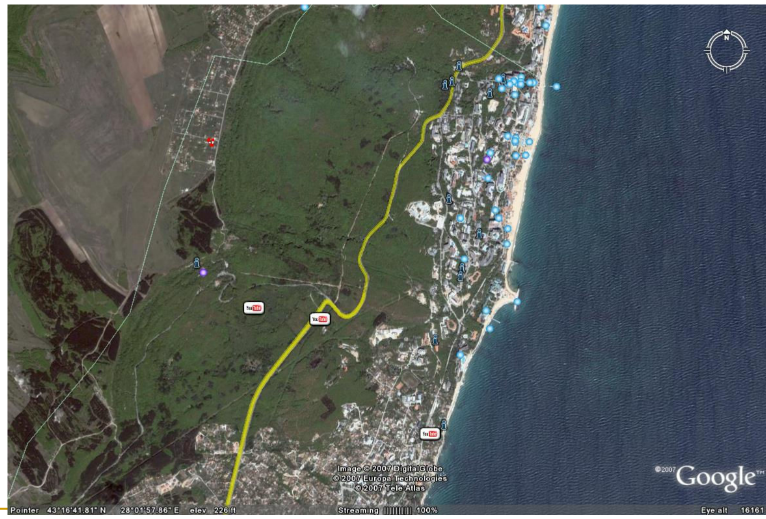
- For itemset {red, white, green}
  - Rule 1: {red, white} => {green},
    - conf = sup {red, white, green} / sup {red, white} = 2/4 = 50%
  - ***Rule 2: {red, green} => {white},***
    - ***conf = sup {red, white, green} / sup {red, green} = 2/2 = 100%***
  - ***Rule 3: {white, green} => {red},***
    - ***conf = sup {red, white, green} / sup {white, green} = 2/2 = 100%***
  - Rule 4: {red} =>{white, green},
    - conf = sup {red, white, green} / sup {red} = 2/6 = 33%
  - Rule 5: {white} => {red, green},
    - conf = sup {red, white, green} / sup {white} = 2/7 = 29%
  - ***Rule 6: {green} => {red, white}***
    - ***conf = sup {red, white, green} / sup {green} = 2/2 = 100%***
- If the desired min_conf is 70%, we got Rule 2, 3, 6.

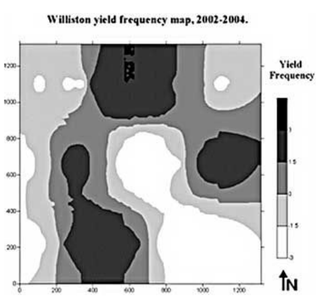
Final Results for Phone Faceplate Transactions

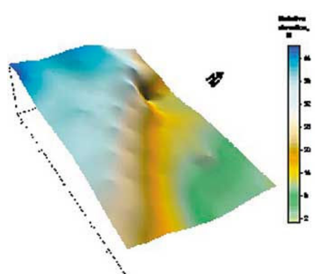
Rule #	Conf.%	X	Y	Supp.(X)	Supp.(Y)	Supp.(XUY)	Lift
1	100	Green	Red, White	2 (20%)	4 (40%)	2 (20%)	2.5
2	100	Green	Red	2 (20%)	6 (60%)	2 (20%)	1.67
3	100	Green, White	Red	2 (20%)	6 (60%)	2 (20%)	1.67
4	100	Green	White	2 (20%)	7 (70%)	2 (20%)	1.43
5	100	Green, Red	White	2 (20%)	7 (70%)	2 (20%)	1.43
6	100	Orange	White	2 (20%)	7 (70%)	2 (20%)	1.43

- The **support** for the rule indicates its impact in terms of overall size: What proportion of transactions is affected?
- The **confidence** indicates what rate Y will be found, is useful in determining the business or operational usefulness of a rule.
- The **lift ratio** indicates how efficient the rule is in finding Y, compared to random selection.
- The more records the rule is based on, the more solid the conclusion since the key evaluative statistics are based on ratios and proportion.

Image Processing: Image Scene







Yield Map

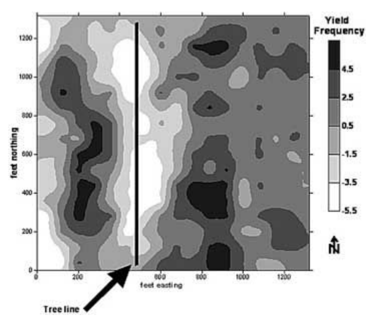
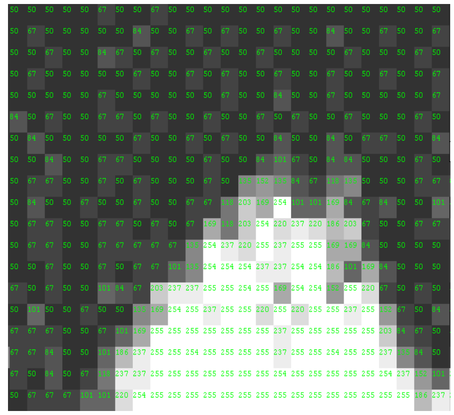
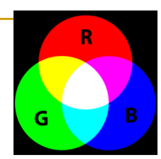


Image Processing : Data



http://www.cs.washington.edu/research/metip/about/digital.html

Color Image : Data



- A color image can be represented by a two-dimensional array of Red, Green and Blue triples. Typically, each number in the triple also ranges from 0 to 255, where 0 indicates that none of that primary color is present in that pixel and 255 indicates a maximum amount of that primary color.

Pixel	Band1 (Red)	Band2 (Green)	Band3 (Blue)
1	40	140	200
2	50	130	210

Data

- Three bands of Yield map were converted into a gray scale.

Pixel	Band1 (Red)	Band2 (Green)	Band3 (Blue)	Band4 (Gray Scale) Yield
1	40	140	200	240
2	50	130	210	250

- The problem is to discover the associations between band1, band2, band3 and band4. This will help farmers to understand what combination of spectral bands will have a high crop yield.

Preprocessing of Data

- Data Discretization: divide the range of a continuous attribute into intervals.

	[0,63]	[64,127]	[128,191]	[192,255]
Band1	B11	B12	B13	B14
Band4	B41	B42	B43	B44

	[0,31]	[32,63]	[64,95]	[96,127]	[128,159]	[160,191]	[192,225]	[226,255]
Band2	B21	B22	B23	B24	B25	B26	B27	B28
Band3	B31	B32	B33	B34	B35	B36	B37	B38

Pixel	Band1 (Red)	Band2 (Green)	Band3 (Blue)	Band4 (Gray Scale) Yield
1	B11(40)	B25(140)	B37(200)	B44(240)
2	B11(50)	B25(130)	B37(210)	B44(250)

Improvement of ARM Algorithm Based on Domain Knowledge

- Dimensions: B11- B14, B21- B28, B31- B38, B41- B44.
- These candidates should not be generated since the combination of ***k*** intervals from same band has support zero.
  - Possible 2-itemsets candidates: {B11, B14} if B11 and B14 are frequent 1-itemsets.

Evaluate Rules

- From the domain knowledge, we know that band1, band2, and band3 refer to reflectance data and band4 refers to yield data. The association rules the user likes to mine are of the form: band1 $\wedge$ band2 $\wedge$ band3 => band4.
- When minsup = 40%, we found two rules:
  - B12 $\wedge$ B26 $\wedge$ B32 => B42
  - B11 $\wedge$ B25 $\wedge$ B37 => B44

Reference

- “The application of Association Rule Mining to Remotely Sensed Data”, J. Dong, W. Perrizo, Q. Ding and J. Zhou, SAC’2000.