# Association Rules Discovery

# Affinity Analysis and Market Basket Analysis

- Affinity Analysis studies characteristics or attributes that "go together"
- Affinity Analysis also known as Market Basket Analysis
- Seeks to uncover associations between attributes
- Rules quantify relationship between two or more attributes
- Association Rules have form:

**IF antecedent THEN consequent**

- Rules include measure of Support and Confidence
- For example, of 1,000 customers shopping, 200 bought diapers. In addition, of the 200 buying diapers, 50 bought beer
- Thus, the rule "If buy diapers, then buy beer" has support = 5% and confidence = 25%

# Remember

- SUPPORT
  - The relative frequency where both left and right hand side of the association are found in the database
  - A low level of support may indicate that the pattern is not significant
  - Normally expressed as % or decimal e.g. A=>B

$$\frac{\text{No. of transactions containing both A and B}}{\text{Total no. of transactions}}$$

- CONFIDENCE
  - Given the occurrence condition A ( antecedent), how often does condition B occur(consequent)

$$\frac{\text{No. of transactions containing both A and B}}{\text{No. of transaction containing A}}$$

- LIFT (Apriori in Weka) Lift is Confidence divided by the proportion of all examples that are covered by the consequence. This is a measure of the importance of the association that is independent of support

$$\frac{\text{Confidence of Rule}}{\text{(No. of transaction containing B/Total no. of transactions)}}$$

# Affinity Analysis and Market Basket Analysis

- Algorithms seeking to mine association rules confronted with "curse of dimensionality"
  - Number of rules grows exponentially with number of attributes
- With *k* binary attributes, and only positive cases considered, there are $k \times 2^{k-1}$ possible association rules
- Typical applications of Market Basket Analysis may have thousands of attributes
  - Buy ITEM1, and ITEM2, and …, and ITEM1000?
- For example, suppose store sells only 100 different items
  - Customer may buy, or not buy, any combination of 100 items
  - This equals $100 \times 2^{99} = \sim 6.4 \times 10^{31}$ possible rules to interpret!
- Task searching for possible rules appears hopeless…

# Market Basket Analysis Example using the A Priori Algorithm

- A Priori Algorithm reduces search problem to manageable size
- Leverages rule structure to its advantage

- **Example**
  - Suppose farmer sells crops at roadside stand
  - Seven items available for purchase in set
  - $I$ = {**asparagus, beans, broccoli, corn, green peppers, squash, tomatoes**}
  - Customers purchase different subsets of $I$
  - Each customer transaction tracked, showing which items purchased

# Market Basket Analysis using the A Priori Algorithm

- Example Table shows transactions made at roadside stand, one particular day

| Transaction | Items Purchased |
|---|---|
| 1 | Broccoli, green peppers, corn |
| 2 | Asparagus, squash, corn |
| 3 | Corn, tomatoes, beans, squash |
| 4 | Green peppers, corn, tomatoes, beans |
| 5 | Beans, asparagus, broccoli |
| 6 | Squash, asparagus, beans, tomatoes |
| 7 | Tomatoes, corn |
| 8 | Broccoli, tomatoes, green peppers |
| 9 | Squash, asparagus, beans |
| 10 | Beans, corn |
| 11 | Green peppers, broccoli, beans, squash |
| 12 | Asparagus, beans, squash |
| 13 | Squash, corn, asparagus, beans |
| 14 | Corn, green peppers, tomatoes, beans, broccoli |

# A Priori Algorithm Data Representation

- Data Representation :
  - Market Basket Analysis data must be represented in Transactional or Tabular format
  - Transactional Format
    - Requires two fields: *ID* and *Content*
    - Each record represents single item
  - Tabular Format
    - Each record represents separate transaction
    - Items are flagged as 0 (not purchased) or 1 (purchased)
    - Only represents whether item purchased, not number of items purchased

| Transaction ID | Item |
|---|---|
| 1 | Broccoli |
| 1 | Green peppers |
| 1 | Corn |
| 2 | Asparagus |
| 2 | Squash |
| 2 | Corn |
| 3 | Corn |
| 3 | Tomatoes |
| ... | ... |

| Trans | Asparagus | Beans | Broccoli | Corn | Green Peppers | Squash | Toma-toes |
|---|---|---|---|---|---|---|---|
| **1** | 0 | 0 | 1 | 1 | 1 | 0 | 0 |
| **2** | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| **3** | 0 | 1 | 0 | 1 | 0 | 1 | 1 |

# Association Rule

- Let $D$ = set of transactions {$T1$, $T2$, …, $T14$} in Example Table
- Each $T$ represents set of items contained in $I$

- Suppose set of items A = {beans, squash} and B = {asparagus}
- Association Rule has the form:

**IF A THEN B**

**A → B**

**IF {beans, squash} THEN {asparagus}**

- A and B proper subsets of $I$
- A and B are mutually exclusive
  - Therefore, by definition, rules such as IF {beans, squash} THEN {beans} excluded
- SUPPORT and CONFIDENCE of Rule will also be calculated

# Frequent Itemsets

- Itemset is set of items contained in $I$
- *k*-itemset contains $k$ items
- For example, {beans, squash} = 2-itemset, from roadside stand set $I$
- Itemset Frequency is number of transactions containing specific itemset
- **Frequent Itemset** occurrence greater than or equal to minimum threshold

$$\text{A frequent itemset has itemset frequency} \geq \phi\text{, where}$$

$$\phi = \text{Minimum Threshold}$$

$$\text{Set of frequent } k\text{-itemsets denoted as } F_k$$

# Mining for Rules and the *A Priori* Property

- **Mining Association Rules**
  - Two-step process
  - (1) Find all frequent itemsets, where $(\text{itemset frequency} \geq \phi)$
  - (2) From list of frequent itemsets, generate association rules satisfying minimum support and confidence criteria

- ***A Priori* Property**

*If itemset Z not frequent, then for any item A, Z U A not frequent*

  - In other words, <u>no superset of Z</u> (itemset containing Z) will be frequent
  - *A Priori* algorithm uses this property to <u>significantly reduce the search space</u>

# *A Priori* Algorithm Part 1 - Generating Frequent Itemsets

- **Generating Frequent Itemset F1**
  - Let $\phi = 4$
  - Recall set of transactions *D* in Example Table
  - First find F1, frequent 1-itemsets, where itemset frequency >= 4
  - Calculating totals for each column determines all 1-itemsets are frequent
  - Therefore, **F1** = {asparagus, beans, broccoli, corn, green peppers, squash, tomatoes}

# *A Priori* Algorithm Part 1 - Generating Frequent Itemsets

## Generating Frequent Itemset F2

- *A Priori* derives F*k* by constructing a set of candidate k-itemsets C*k*, by joining F*k-1* with itself
- Next, C*k* is pruned using the *A Priori* property
- Remaining itemsets in C*k* form F*k*
- Table shows all candidate 2-itemsets C*2*

| Combination | Count | Combination | Count |
|---|---|---|---|
| Asparagus, beans | 5 | Broccoli, corn | 2 |
| Asparagus, broccoli | 1 | Broccoli, green peppers | 4 |
| Asparagus, corn | 2 | Broccoli, squash | 1 |
| Asparagus, green peppers | 0 | Broccoli, tomatoes | 2 |
| Asparagus, squash | 5 | Corn, green peppers | 3 |
| Asparagus, tomatoes | 1 | Corn, squash | 3 |
| Beans, broccoli | 3 | Corn, tomatoes | 4 |
| Beans, corn | 5 | Green peppers, squash | 1 |
| Beans, green peppers | 3 | Green peppers, tomatoes | 3 |
| Beans, squash | 6 | Squash, tomatoes | 2 |
| Beans, tomatoes | 4 | | |

- F2 = {{asparagus, beans}, {asparagus, squash}, {beans, corn}, {beans, squash}, {beans, tomatoes}, {broccoli, green peppers}, {corn, tomatoes}}

# *A Priori* Algorithm Part 1 - Generating Frequent Itemsets

## Generating Frequent Itemset F3

Next, frequent itemsets F2 used to generate C3 candidate 3-itemsets

- F2 joined with itself, where itemsets joined having first $k-1$ items in common (alphabetically /lexicographic order)
  - For example, {**asparagus**, beans} joined with {**asparagus**, squash}
- Have first $k-1=1$ items (asparagus) in common
- New candidate formed, {**asparagus**, beans, squash}
- Remaining candidate 3-itemsets generated
  - **C3 = {{asparagus, beans, squash}, {beans, corn, squash}, {beans, corn, tomatoes}, {bean, squash, tomatoes}}**

# *A Priori* Algorithm Part 1 - Generating Frequent Itemsets

- Finally, C3 is pruned using the *A Priori* Property
- For each itemset *s* in C3, its $k - 1$ subsets examined
- If <u>any</u> subsets not frequent, then *s* not frequent and pruned
  - For example, let $s$ = {asparagus, beans, squash}
  - Subsets $k - 1 = 2$ are {asparagus, beans}, {asparagus, squash}, and {beans, squash}
- Therefore $s$ = {asparagus, beans, squash} <u>not pruned</u> according to *A Priori* Property
- Examine the other Candidate-3 itemsets in the same way
  - Now, C3 = {**{asparagus, beans, squash}**, **{beans, corn, tomatoes}**}

# *A Priori* Algorithm Part 1 - Generating Frequent Itemsets

- Process continues, *A Priori* Property applied to remaining $C_3$ candidate 3-itemsets

$$\phi = 4$$

- **$C_3$ = {{asparagus, beans, squash}, {beans, corn, tomatoes}}**
- Itemset {asparagus, beans, squash} occurs 4 times; however, {beans, corn, tomatoes} occurs in 3 transactions and is <u>pruned</u>

- Therefore **$F_3$ = {asparagus, beans, squash}**

# *A Priori* Algorithm Part 2 - Generating Association Rules

- Generating Association Rules
  - Association Rules generated from Frequent Itemsets
  - Two-step process

  (1) For each frequent itemset *s*

  Generate all subsets of *s*

  - Let *ss* represent non-empty subset of *s*

  (2) For each subset ss

  Consider Association Rule *R*: $ss \rightarrow (s - ss)$

  - Association Rule *R* generated, if *R* fulfills minimum confidence criterion

    Note: single-item consequent desired for simplicity

# *A Priori* Algorithm Part 2 - Generating Association Rules

- For example, recall F3 = {asparagus, beans, squash}
- Let *ss* = {asparagus, beans}; it follows (*s* – *ss*) = {squash}
- Consider *R*: if {asparagus, beans} then {squash}
- Table shows *R* support = 28.6% and confidence = 80%

| If *Antecedent* then *Consequent* | Support | Confidence |
|---|---|---|
| If buy asparagus and beans, then buy squash | 4/14 = 28.6% | 4/5 = 80% |
| If buy asparagus and squash, then buy beans | 4/14 = 28.6% | 4/5 = 80% |
| If buy beans and squash, then buy asparagus | 4/14 = 28.6% | 4/6 = 66.7% |

  - Recall, support is proportion of transactions where both {asparagus, beans} and {squash} occur = 4/14
  - For confidence, {asparagus, beans} occurs in 5 transactions, of which 4 <u>also</u> contain {squash} = 4/5
- Note, additional rules in Table generated similarly

# *A Priori* Algorithm **Part** 2 - Generating Association Rules

- Next, single-antecedent/consequent rules evaluated
- Itemsets in F2 used for association rule generation
- Candidate association rules generated from F2 shown below

| **If *Antecedent* then *Consequent*** | **Support** | **Confidence** |
|---|---|---|
| If buy asparagus, then buy beans | 5/14 = 35.7% | 5/6 = 83.3% |
| If buy beans, then buy asparagus | 5/14 = 35.7% | 5/10 = 50% |
| If buy asparagus, then buy squash | 5/14 = 35.7% | 5/6 = 83.3% |
| If buy squash, then buy asparagus | 5/14 = 35.7% | 5/7 = 71.4% |
| If buy beans, then buy corn | 5/14 = 35.7% | 5/10 = 50% |
| If buy corn, then buy beans | 5/14 = 35.7% | 5/8 = 62.5% |
| If buy beans, then buy squash | 6/14 = 42.9% | 6/10 = 60% |
| If buy squash, then buy beans | 6/14 = 42.9% | 6/7 = 85.7% |
| If buy beans, then buy tomatoes | 4/14 = 28.6% | 4/10 = 40% |
| If buy tomatoes, then buy beans | 4/14 = 28.6% | 4/6 = 66.7% |
| If buy broccoli, then buy green peppers | 4/14 = 28.6% | 4/5 = 80% |
| If buy green peppers, then buy broccoli | 4/14 = 28.6% | 4/5 = 80% |
| If buy corn, then buy tomatoes | 4/14 = 28.6% | 4/8 = 50% |
| If buy tomatoes, then buy corn | 4/14 = 28.6% | 4/6 = 66.7% |

# Association Rules Supervised or Unsupervised Learning?

- Recall most data mining methods represent supervised learning
  - (1) Target \Response variable specified
  - (2) Algorithm provided examples, and learns relationships between predictor and target variables
- In contrast, unsupervised methods search for patterns and structure among all variables
- Association rule mining either supervised or unsupervised
- For example, in market basket analysis **NO target/ response variable** specified => **Unsupervised** Learning Approach
- Simply interested in "which items purchased together"

# Association Rules Supervised or Unsupervised Learning? *(cont'd)*

- Conversely, some data sets naturally structured, with one attribute particularly suited as consequent
- For example, political pollsters collect demographic exit poll data
  - Each subject's voting preference also collected
  - Association rule mining uses demographic attributes as antecedents, and voting preference as single consequent
  - **Supervised Learning Approach**
    - i.e. antecedent – **predictor variables** and consequent – **target variable/response variable**
- Rules may uncover (classify) voting preferences, according to certain demographic characteristics