Association Rule Mining

What is Market Basket Analysis?

- **Finding some useful information in 'market basket'**
- **What kinds of information?**
  - Who customers are
  - Which products tend to be purchased together
  - Why some products tend to be purchased together
- **Association rule: Information like "If item A then item B"** $(A \Rightarrow B)$

Point of Sale Transactions

- **Transaction and item**

Datetime	Customer	Items
2015-07-15 14:03	1	orange juice, banana
2015-07-15 16:20	2	orange juice, milk
2015-07-16 10:14	3	detergent, banana, orange juice
2015-07-25 19:34	2	milk, bread, soda
2015-07-29 09:41	4	detergent, window cleaner
2015-08-01 20:55	1	bread, milk

- Find pair of items that is more likely to be purchased together based on transactions
  - ✓ Banana and orange juice are more likely to be purchased together
  - ✓ Milk and bread are more likely to be purchased together

Association Rules

- **Association rules obtained from transactions are like "If item A, then item B"**
  - Rules are defined from co-occurrence of items in the same market basket

- **Three types of rules**
  - Useful: contains high quality, actionable information
    - ✓ On Thursday, customer who purchase diapers are likely to purchase beer
  - Trivial: already known by anyone familiar with the business
    - ✓ Customers purchasing paint buy pain brushes
  - Inexplicable: new but no explanation about customer behavior
    - ✓ When a new hardware store opens, one of the most commonly sold items is toilet rings

General Process for Finding Rules

1. • Gathering transactions for selected items
2. • Check co-occurrence of set of items
3. • Finding the most frequent combination from the matrix
4. • Making a distinction between ‘’condition’ and ‘result’ from the combination

Performance Measure for Rules

- **Rule: If 'condition', then 'result'**

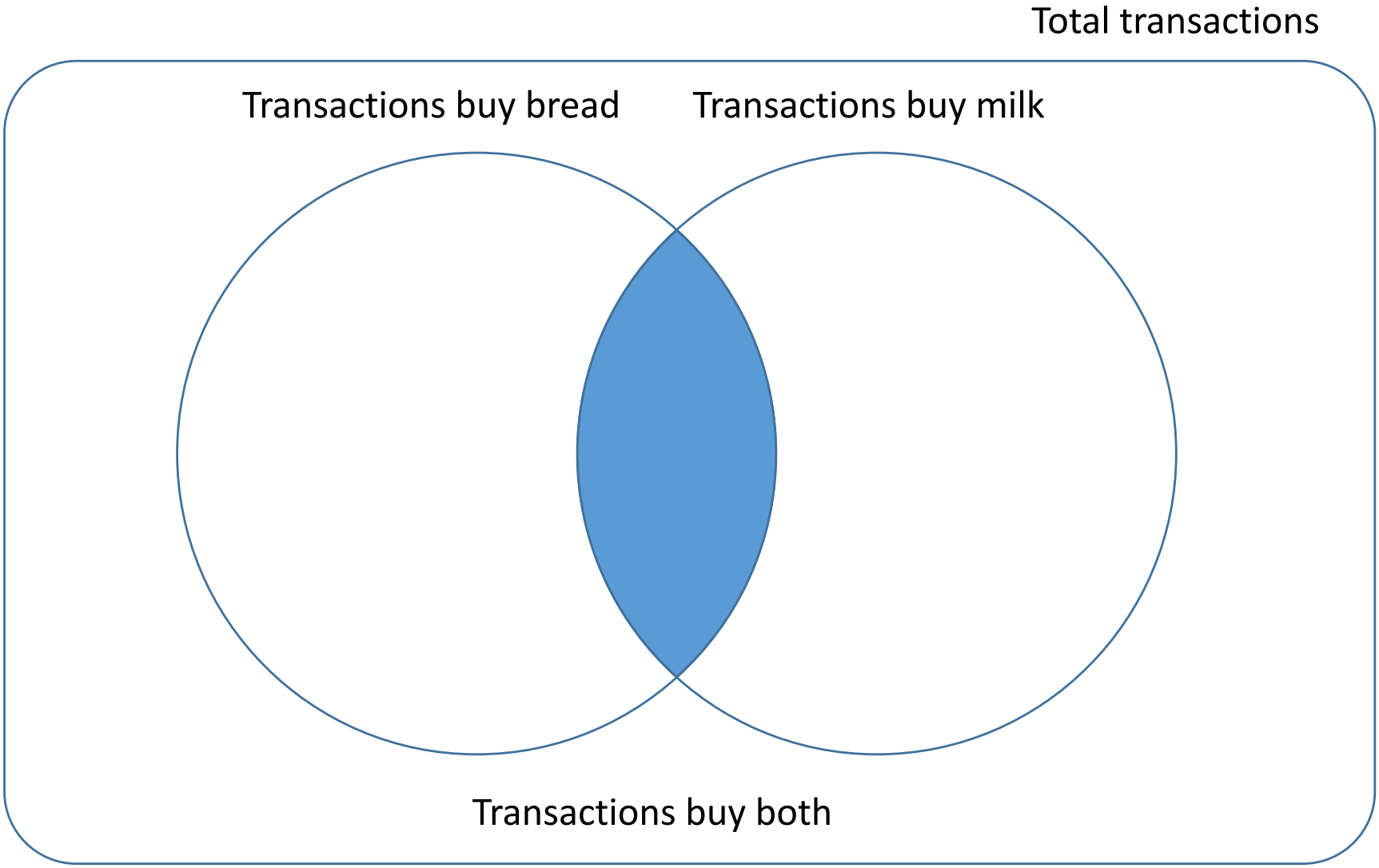
- **Support**
  - How many transactions that contain 'condition(X)' and 'result(Y)' simultaneously

$$\text{Supp}(X \Longrightarrow Y) = \text{Supp}(X \cup Y) =$$
$$= \frac{\#\ of\ transactions\ that\ include\ both\ condition\ and\ result}{\#\ of\ total\ transactions}$$

- **Confidence**
  - How many transactions that contain 'condition(X)' and 'result(Y)' among transactions including 'condition'

$$\text{Conf}(X \Longrightarrow Y) = P(Y|X) = \frac{\textbf{Supp}(X \cup Y)}{\text{Supp}(X)}$$
$$= \frac{\#\ of\ transactions\ that\ include\ both\ condition\ and\ result}{\#\ of\ transactions\ that\ include\ condition}$$

Performance Measure for Rules



Performance Measure for Rules

- **Low support**
  - This rule rarely happens → not interesting

- **High support, but low confidence**
  - Both ‘condition’ and ‘result’ are quite often observed, but comparing with the number of transactions that include condition are much more
  - The reason that support of the rule is high may be that the number of transactions that include condition is high

- **High support and high confidence**
  - This rule is significant rule
  - However, high support and high confidence do not guarantee usefulness of the rule

Example: Association Rule

- **Example rules from given transactions**

TID	Items
1	bread, milk, butter
2	bread, butter
3	bread, juice, butter
4	bread, beer
5	beer, juice

- If bread, then butter (bread ⇒ butter)

$$support = \frac{3}{5}, confidence = \frac{3}{4}$$

- If beer, then bread (beer ⇒ bread)

$$support = \frac{1}{5}, confidence = \frac{1}{2}$$

Performance Measure for Rules

- **Lift or improvement**
  - How much better a rule is at predicting the result than just guessing the result at random

$$\text{lift}(X \Longrightarrow Y) = \frac{P(Y|X)}{P(Y)} = \frac{\textbf{Supp}(X \cup Y)}{\textbf{Supp}(X) \times \textbf{Supp}(Y)}$$

$$= \frac{(\#\ of\ transactions\ that\ include\ both\ condition\ and\ result) \times (\#\ of\ transactions)}{(\#\ of\ transactions\ that\ include\ condition)(\#\ of\ transactions\ that\ include\ result)}$$

Improvement	Interpretation	Example
1	Two items are independent	pepper and cookies
>1	Complementary	Bread and butter
<1	Substitutional	Butter and margarine

In this case, If A, then NOT B is better than If A, then B

Performance Measure for Rules

- **Conviction**
  - Conviction measures the implication strength of the rule from statistical independence

$$\mathbf{c}\text{onv}(X \Longrightarrow Y) = \frac{1 - \text{supp}(Y)}{1 - \text{conf}(X \Longrightarrow Y)} = \frac{P(X) \times P(\sim Y)}{P(X \cup \sim Y)}$$

    - ✓ $P(\sim Y)$ is the probability that Y does not appear in a transaction
  - Conviction compares the probability that X appears without Y if they were dependent with the actual frequency of the appearance of X without Y
  - Unlike confidence, conviction factors in both $P(X)$ and $P(Y)$ and always has a value 1 when the relevant items are completely unrelated.
  - In contrast to lift, conviction is directed measure because it also uses the information of the absence of the consequent

Question

- **Calculate performance measures of the rule**
  - Rule1: $a \Longrightarrow g$
  - Rule2: $e \Longrightarrow f$
  - Rele3: b and $c \Longrightarrow g$

TID	Items
1	b, c, g
2	a, b, d, e, f
3	a, b, c, g
4	b, c, e, f
5	b, c, e, f, g

1) Calculate support of above three rules

2) Calculate confidence of above three rules

3) Calculate lift of above three rules

4) Calculate conviction of above three rules

Apriori Algorithm

Practical Issues on Market Basket Analysis

- **Exponential growth on distinct combinations as the number of items increases**
  - If 100 items are sold in the store, the number of combinations with 3 items

$$C(100,3) = \frac{100!}{3!\,97!} = \frac{100\times99\times98}{3\times2} = 161{,}700$$

- **Methods to solve rapid growth on problem size**
  - Use the taxonomy: generalize items that can meet criterion
    - ✓ Vanilla ice cream ∈ Ice cream ∈ Frozen food ∈ Food
    - ✓ When there are too many items to handle, use higher level of category instead to reduce combinations
  - Use pruning: throw out item or combination of items that do not meet criterion
    - ✓ Minimum support pruning is the most common method

Practical Issues on Market Basket Analysis

- **List of the items**

StockCode	Description
22418	10 COLOUR SPACEBOY PEN
22436	12 COLOURED PARTY BALLOONS
21448	12 DAISY PEGS IN WOOD BOX
22282	12 EGG HOUSE PAINTED WOOD
23442	12 HANGING EGGS HAND PAINTED
21447	12 IVORY ROSE PEG PLACE SETTINGS
22906	12 MESSAGE CARDS WITH ENVELOPES
20973	12 PENCIL SMALL TUBE WOODLAND
20975	12 PENCILS SMALL TUBE RED RETROSPOT
20974	12 PENCILS SMALL TUBE SKULL
20984	12 PENCILS TALL TUBE POSY
20983	12 PENCILS TALL TUBE RED RETROSPOT
20982	12 PENCILS TALL TUBE SKULLS
20981	12 PENCILS TALL TUBE WOODLAND
84461	12 PINK HEN+CHICKS IN BASKET
21445	12 PINK ROSE PEG PLACE SETTINGS
21446	12 RED ROSE PEG PLACE SETTINGS
84465	15 PINK FLUFFY CHICKS IN BOX

Pros and Cons of Market Basket Analysis

Pros	Cons
• Produces understandable and clear results (association rules)	• Require much more computation resource as the problem size grows
• Handle transactions themselves	• Sometimes require to utilize the taxonomy for mining better rules and reducing complexity
• Computational method is simple to implement and understand	• Discount rare items

Apriori Algorithm

- **Apriori is the algorithm to mine rules from transactions**
  - Key idea is that any subsets of a frequent item set are also frequent item sets

$\{1,2,3\}$ **is frequent item set** $\Rightarrow \{1\},\{2\},\{3\},\{1,2\},\{1,3\},\{2,3\}$ **are frequent item set**

Phase 1

- Find all frequent item sets having specified minimum support s_{min}

Phase 2

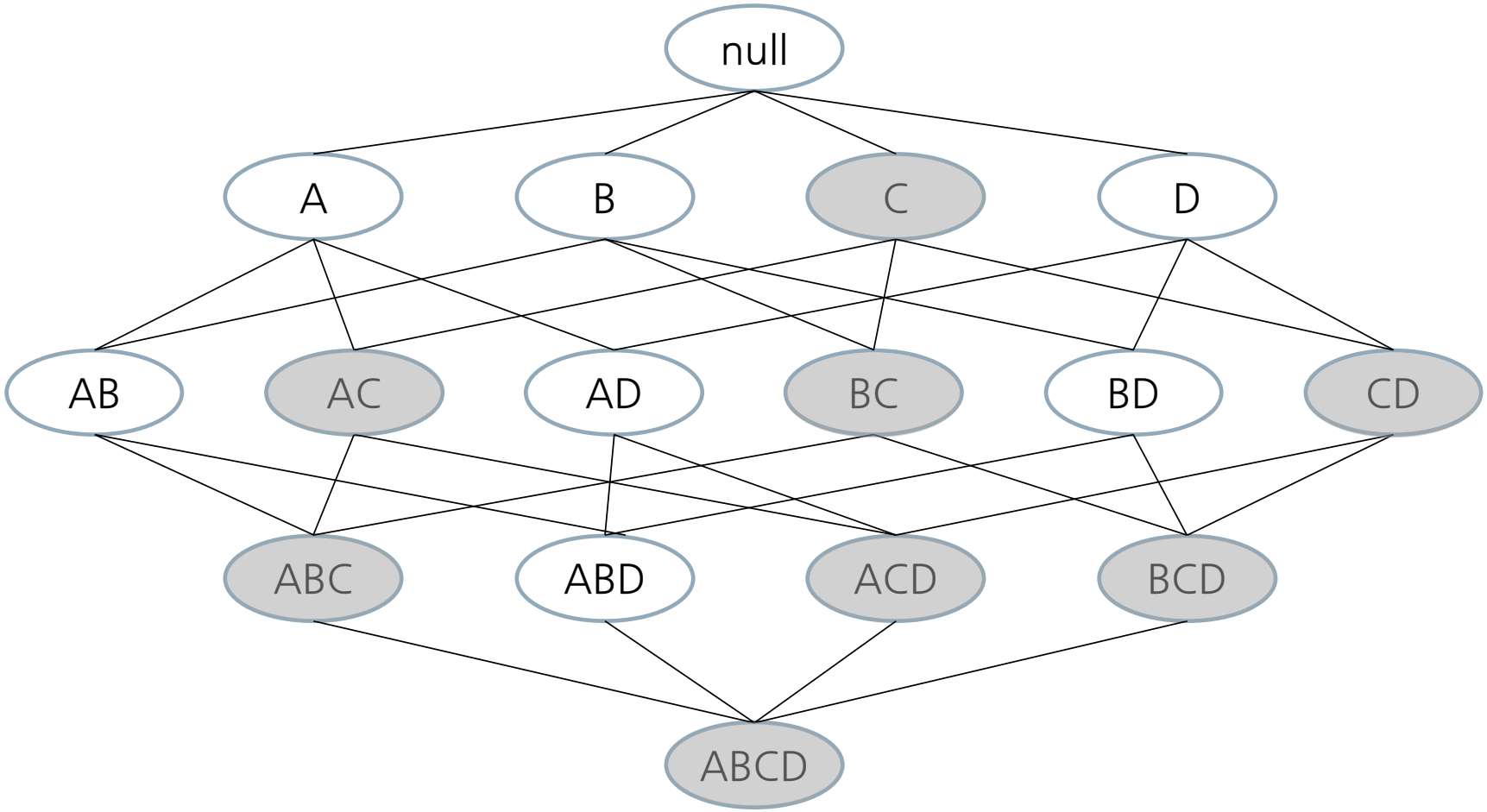
- Consider a subset A of a frequent item set L
  - For a specified confidence c_{min}
    if $support(L)/support(A) \geq c_{min}$,
    then generate a rule $R: A \Rightarrow (L - A)$

Apriori Algorithm – Phase 1

1. [initial step] Specify the minimum support s_{min} and set $k = 1$
   - $C_1 = \{\{i_1\}, \{i_2\}, \ldots, \{i_n\}\}$ $\quad L_1 = \{c \in C_1 | support(c) \geq s_{min}\}$
2. Set $k = k$ and Generate new candidate item sets C_k from L_{k-1}
   - Generate item sets C_k by joining like $C = L_{k-1} \times L_{k-1}$
   - Delete all item sets whose any subsets are not in L_{k-1} from C_k
3. Generate L_k such that $L_k = \{c \in C_k | support(c) \geq s_{min}\}$
4. Repeat step 2 and 3 until $C_k = \phi$

Apriori Algorithm – Phase 1

- **Key idea of phase 1**



Example: Apriori Algorithm – Phase 1

- **Generate C_k and L_k**
  - Set s_{min}=0.4

TID	Items
1	b, c, g
2	a, b, d, e, f
3	a, b, c, g
4	b, c, e, f
5	b, c, e, f, g

$$C_1 = \{\{a\},\{b\},\{c\},\{d\},\{e\},\{f\},\{g\}\}$$

Remove infrequent item sets

$$L_1 = \{\{a\},\{b\},\{c\},\{e\},\{f\},\{g\}\}$$

Generate item sets by joining

$$C_2 = \{\{a,b\},\{a,c\},\{a,e\},\{a,f\},\{a,g\},\{b,c\},\{b,e\},\{b,f\},\{b,g\},\{c,e\},\{c,f\},\{c,g\},\{e,f\},\{e,g\},\{f,g\}\}$$

$$L_2 = \{\{a,b\},\{b,c\},\{b,e\},\{b,f\},\{b,g\},\{c,e\},\{c,f\},\{c,g\},\{e,f\}\}$$

$$C_3 = \{\{b,c,e\},\{b,c,f\},\{b,c,g\},\{b,e,f\},\{c,e,f\}\}$$

$$L_3 = \{\{b,c,e\},\{b,c,f\},\{b,c,g\},\{b,e,f\},\{c,e,f\}\}$$

$\{a,b,c\}$ is removed from C_3 because $\{a,c\}$ does not belong to L_2

$$C_4 = \{\{b,c,e,f\}\}$$

$$L_4 = \{\{b,c,e,f\}\}$$

Question

- **Generate C_k and L_k**
  - Set s_{min}=0.4

TID	Items
1	bread, milk, butter
2	bread, butter
3	bread, juice, butter
4	bread, beer
5	beer, juice

1) Generate C_1 and L_1

2) Generate C_2 and L_2

Example: Apriori Algorithm – Phase 2

- **Rule generation**
  - Candidate frequent item set $L = \{b, c, g\}$
  - Rules having 1 item in result

$$R_1: \{b, c\} \Rightarrow \{g\}$$
$$R_2: \{b, g\} \Rightarrow \{c\}$$
$$R_3: \{c, g\} \Rightarrow \{b\}$$

TID	Items
1	b, c, g
2	a, b, d, e, f
3	a, b, c, g
4	b, c, e, f
5	b, c, e, f, g

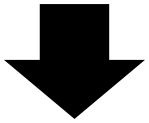
Rule	Support($\{b, c, g\}$)	Support(condition)	Confidence
$R_1: \{b, c\} \Rightarrow \{g\}$	0.6	0.8	0.6/0.8=0.75
$R_2: \{b, g\} \Rightarrow \{c\}$	0.6	0.6	0.6/0.6=1
$R_3: \{c, g\} \Rightarrow \{b\}$	0.6	0.6	0.6/0.6=1

How to Efficiently Generate Rules

- **Confidence is not anti-monotonic**
  - $confidence(ABC \Longrightarrow D)$ can be larger or smaller than $confidence(AB \Longrightarrow D)$

- **However, confidence of rules generated from the same item set is anti-monotonic with respect to the number of items in result**
  - All conditions should be subsets of the largest condition

$$confidence(ABC \Rightarrow D) \geq confidnece(AB \Rightarrow CD) \geq confidence(A \Rightarrow BCD)$$



If the rule $ABC \Rightarrow D$ has lower confidence than certain value

Then, $BC \Rightarrow AD$, $AC \Rightarrow BD$, $BD \Rightarrow CD$, $C \Rightarrow ABD$, $B \Rightarrow ACD$, $A \Rightarrow BCD$ have lower confidence than certain value

Set Up s_{min}

- **If s_{min} is too high, we can miss item sets containing interesting rare items (e.g., expensive products)**

- **If s_{min} is too low, the number of frequent item sets increases and computational cost becomes expensive**

Always, it is really hard to set “appropriate” parameter

It is not effective to use a single s_{min}

Thank you!