CS157A

Data Mining-Association Rules and Clustering

Prof. Sin-Min Lee

What can be inferred?

What can be inferred?

- I purchase diapers

What can be inferred?

- I purchase diapers
- I purchase a new car

What can be inferred?

- I purchase diapers
- I purchase a new car
- I purchase OTC cough medicine

What can be inferred?

- I purchase diapers
- I purchase a new car
- I purchase OTC cough medicine
- I purchase a prescription medication

What can be inferred?

- I purchase diapers
- I purchase a new car
- I purchase OTC cough medicine
- I purchase a prescription medication
- I don’t show up for class

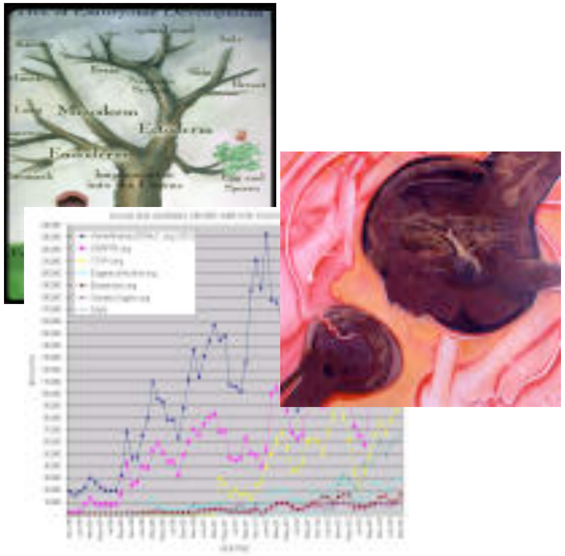
The Introduction to Data Mining

- The process of **extracting valid**, **previously unknown, comprehensible**, and **actionable information** from **large databases** and using it to make **crucial business decisions**.

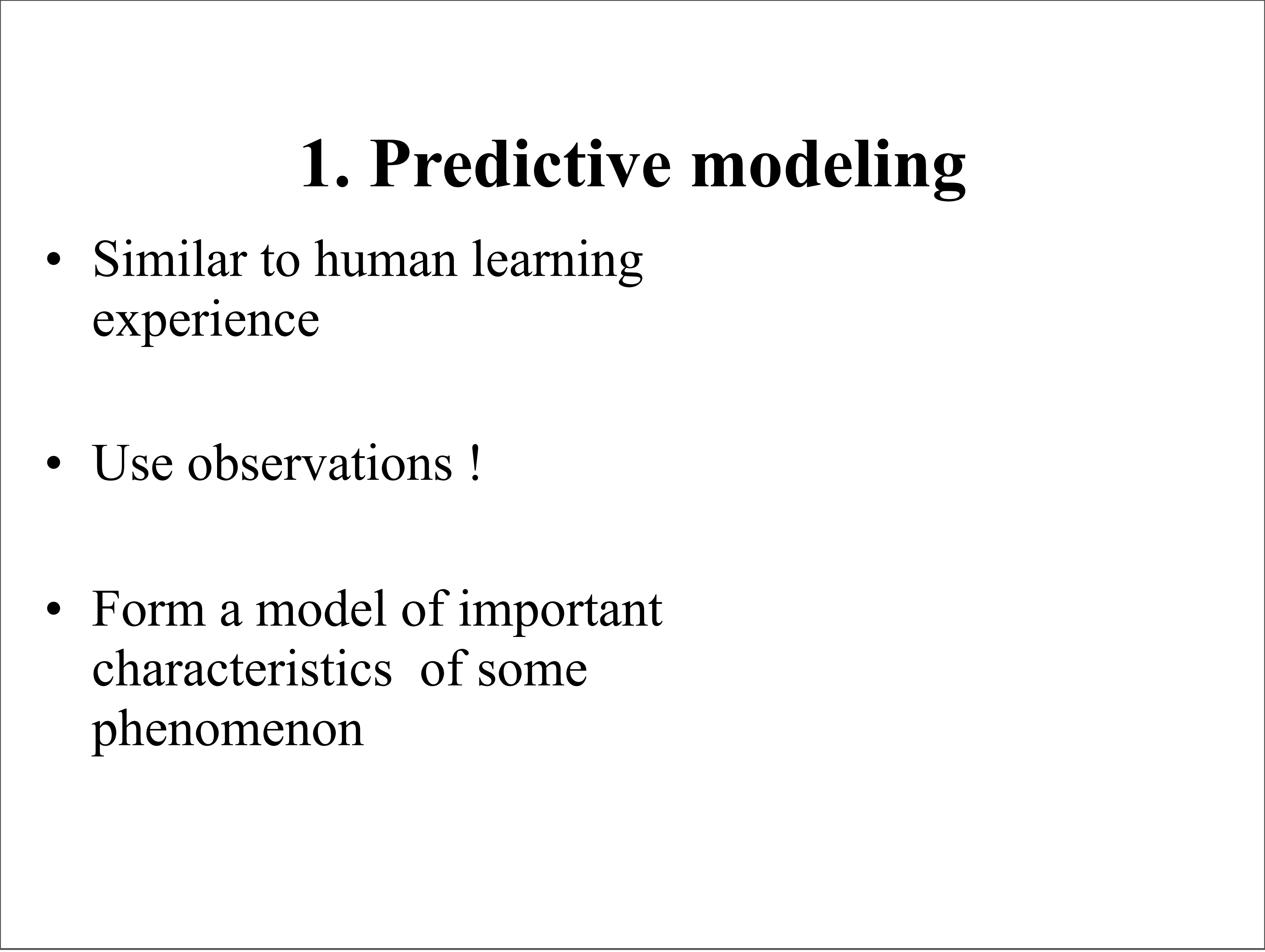
OVERVIEW

- **1. What is predictive modeling**
- **2. Two phases of predictive modeling**
  1)Training phase
  2)Testing phase
- **3. Two techniques of predictive modeling**
  1)Classification
  – Tree induction
  – Neural network

  2)Value Prediction
  – Linear regression
  – Nonlinear regression



OK, now, let me talk about predictive modeling, which is one of four techniques of data mining. I will cover three areas, which are the introduction of predictive modeling, two phases of predictive modeling, and two techniques of predictive modeling. As for the techniques of the predictive modeling, we are also going to discuss classification and value prediction.



1. Predictive modeling

- Similar to human learning experience
- Use observations !
- Form a model of important characteristics of some phenomenon

So, what’s the predictive modeling? It’s simple, we have predictors, and we are trying to make prediction value. It’s very similar to human learning experience. We use observations to form a model of the important characteristics of some phenomenon.

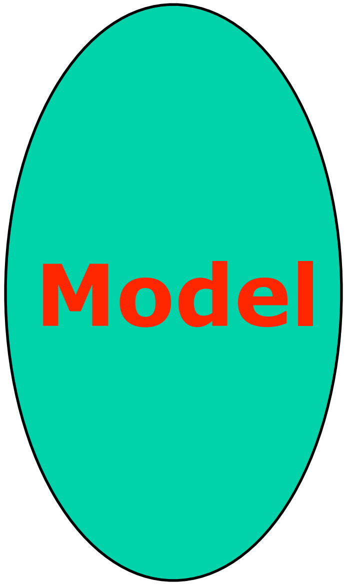
1. Predictive modeling (contd.)

- A “black box” that makes predictions about the future based on information from the past and present

- Application: customer retention management, credit approval, direct marketing.

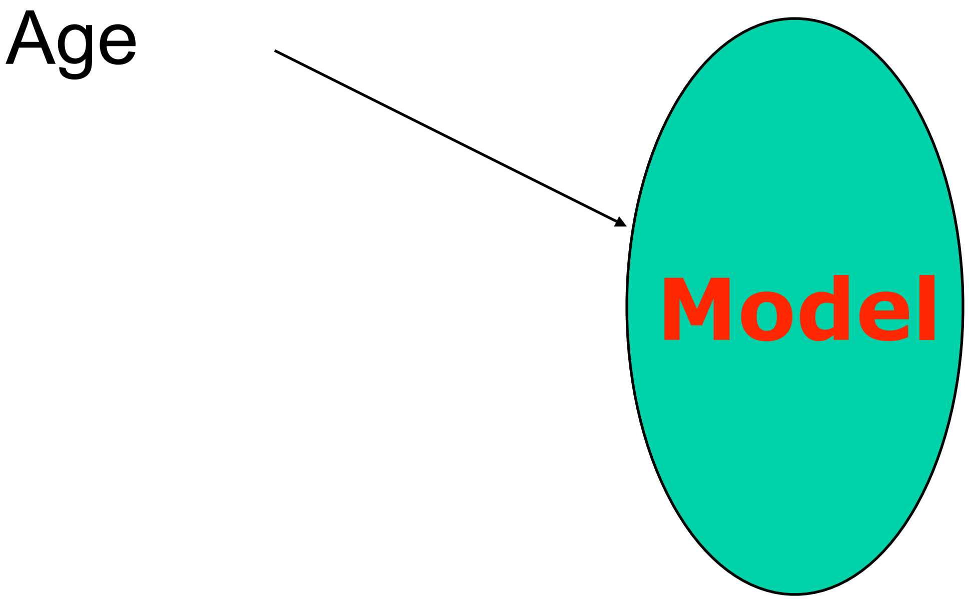
It’s pretty much like a “black box” that makes predictions about the future based on information from the past and present. And large number of inputs usually is available. And the predictive modeling is widely used in customer retention management, credit approval, and direct marketing areas

1. Predictive modeling (contd.)



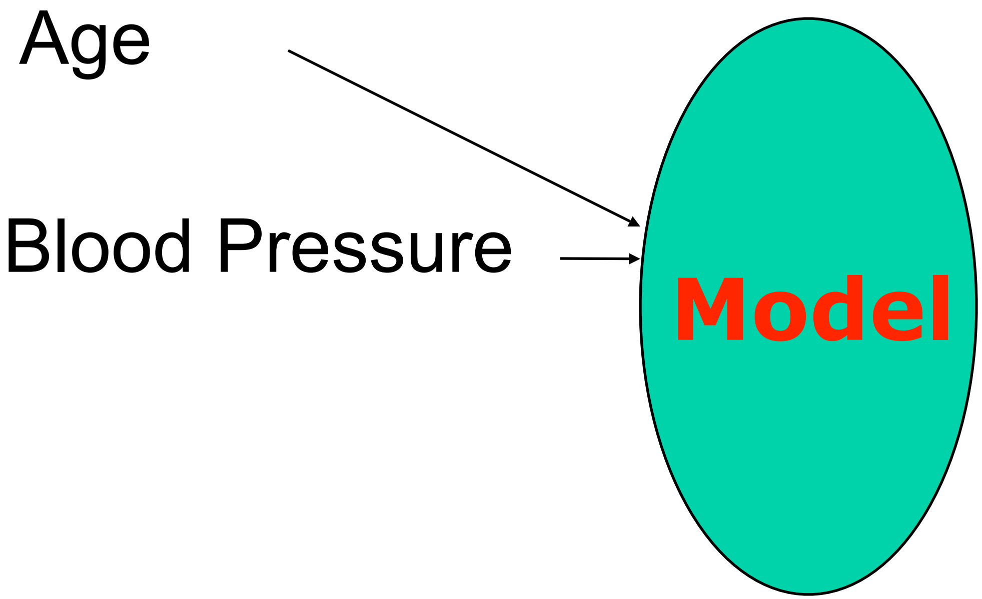
So, if we have a predictive model. And there are three predictors in this model, which are age, blood pressure, and eye color. And we try to figure out if the customers file bankruptcy. And that's the basic idea how a predictive model works.

1. Predictive modeling (contd.)



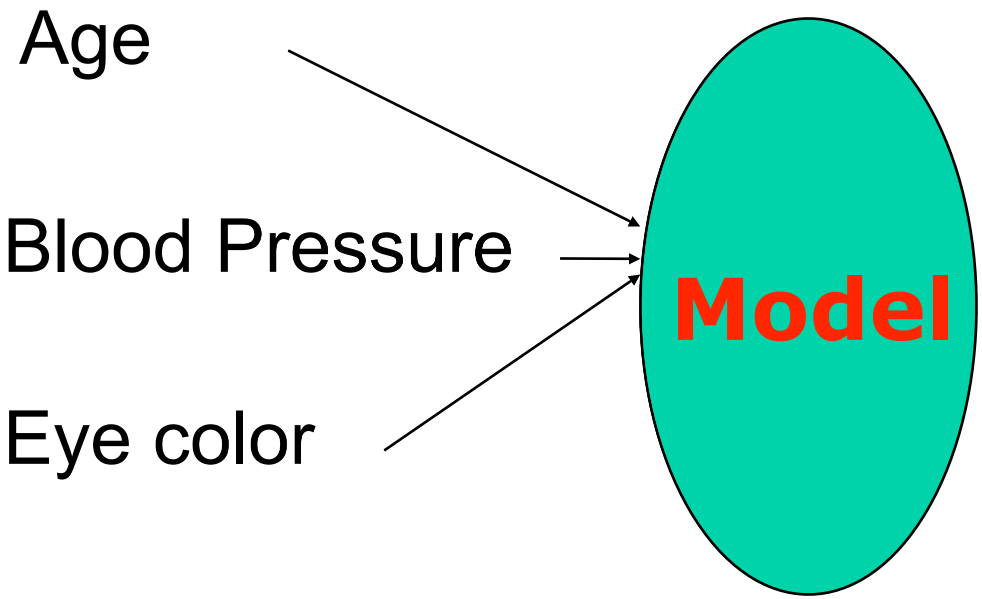
So, if we have a predictive model. And there are three predictors in this model, which are age, blood pressure, and eye color. And we try to figure out if the customers file bankruptcy. And that's the basic idea how a predictive model works.

1. Predictive modeling (contd.)



So, if we have a predictive model. And there are three predictors in this model, which are age, blood pressure, and eye color. And we try to figure out if the customers file bankruptcy. And that’s the basic idea how a predictive model works.

1. Predictive modeling (contd.)



Age

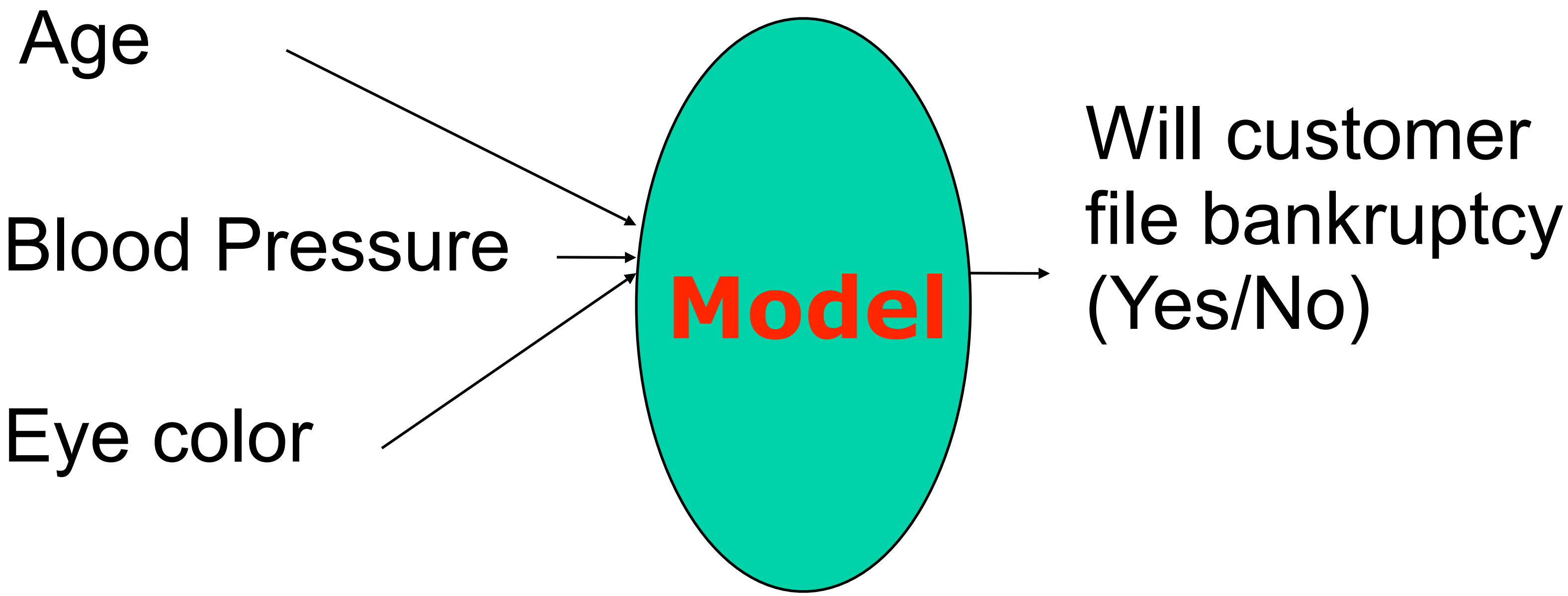
Blood Pressure

Eye color

Model

So, if we have a predictive model. And there are three predictors in this model, which are age, blood pressure, and eye color. And we try to figure out if the customers file bankruptcy. And that's the basic idea how a predictive model works.

1. Predictive modeling (contd.)



So, if we have a predictive model. And there are three predictors in this model, which are age, blood pressure, and eye color. And we try to figure out if the customers file bankruptcy. And that’s the basic idea how a predictive model works.

Definitions

Definitions

- Maps data item into one of several clusters, where clusters are natural groupings of data items based on similarity metrics or probability density models.

Definitions

- Maps data item into one of several clusters, where clusters are natural groupings of data items based on similarity metrics or probability density models.

Definitions

- Maps data item into one of several clusters, where clusters are natural groupings of data items based on similarity metrics or probability density models.

- Multi-scale representation of data refers to visualization of the data at different ‘scales’, where the term scale may signify either unit, frequency, radius, window size or kernel parameters.

Definitions

- Maps data item into one of several clusters, where clusters are natural groupings of data items based on similarity metrics or probability density models.
- Multi-scale representation of data refers to visualization of the data at different ‘scales’, where the term scale may signify either unit, frequency, radius, window size or kernel parameters.

Definitions

- Maps data item into one of several clusters, where clusters are natural groupings of data items based on similarity metrics or probability density models.
- Multi-scale representation of data refers to visualization of the data at different ‘scales’, where the term scale may signify either unit, frequency, radius, window size or kernel parameters.

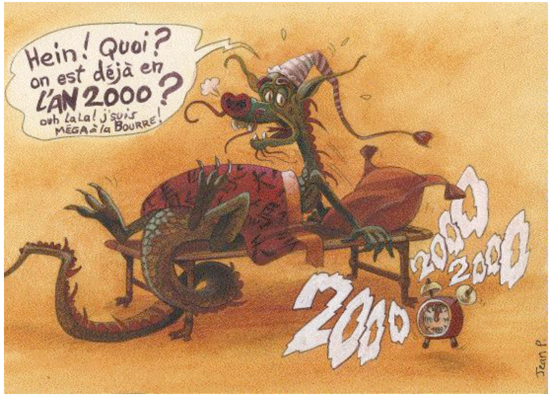
+

Definitions

- Maps data item into one of several clusters, where clusters are natural groupings of data items based on similarity metrics or probability density models.

- Multi-scale representation of data refers to visualization of the data at different ‘scales’, where the term scale may signify either unit, frequency, radius, window size or kernel parameters.

+

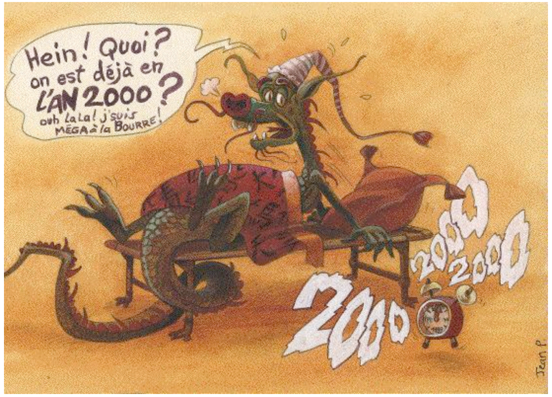

Hein! Quoi? on est déjà en L'AN 2000? ouh lala! j'suis MÉGA à la BOURRE!
2000
2000
2000
Jean P.

Definitions

- Maps data item into one of several clusters, where clusters are natural groupings of data items based on similarity metrics or probability density models.

- Multi-scale representation of data refers to visualization of the data at different ‘scales’, where the term scale may signify either unit, frequency, radius, window size or kernel parameters.

+


Hein! Quoi? on est déjà en L'AN 2000? ouh lala! j'suis MÉGA à la BOURRE!
2000
2000
2000
Jean P.

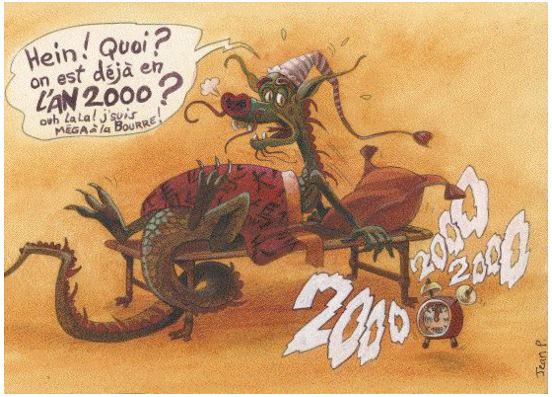
=

Definitions

- Maps data item into one of several clusters, where clusters are natural groupings of data items based on similarity metrics or probability density models.

- Multi-scale representation of data refers to visualization of the data at different ‘scales’, where the term scale may signify either unit, frequency, radius, window size or kernel parameters.

+



Hein! Quoi? on est déjà en L'AN 2000? ouh lala! j'suis MÉGA à la BOURRE!
2000
2000
2000
Jean P.

=

Definitions

Definitions

- The nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data

- Drawing circles and shapes around dots representing objects!

Overview of Clustering

Overview of Clustering

- Cluster analysis has wide application including market/customer segmentation, pattern recognition, biological studies, and Web document classification

- Clustering is a dynamic field of research in data mining. The algorithms can be categorized into partitioning, hierarchical, density-based, and model-based methods

Overview of Clustering

- Cluster analysis has wide application including market/customer segmentation, pattern recognition, biological studies, and Web document classification

- Clustering is a dynamic field of research in data mining. The algorithms can be categorized into partitioning, hierarchical, density-based, and model-based methods

Overview of Clustering

- Cluster analysis has wide application including market/customer segmentation, pattern recognition, biological studies, and Web document classification

- Clustering is a dynamic field of research in data mining. The algorithms can be categorized into partitioning, hierarchical, density-based, and model-based methods

Overview of Clustering

- Cluster analysis has wide application including market/customer segmentation, pattern recognition, biological studies, and Web document classification

- Clustering is a dynamic field of research in data mining. The algorithms can be categorized into partitioning, hierarchical, density-based, and model-based methods

Google™

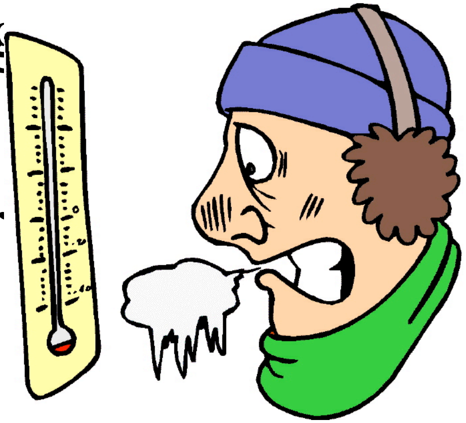
Data Types in Cluster Analysis

Data Types in Cluster Analysis

- Interval-Scaled Variables
  - Continuous measurements of a roughly linear scale
  - Weight, height, latitude, temperature
  - How to compute their differences?

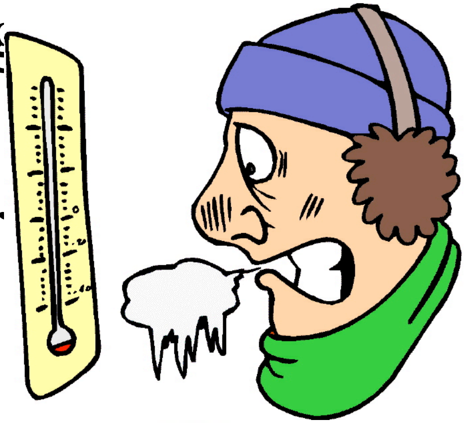
Data Types in Cluster Analysis

- Interval-Scaled Variables
  - Continuous measurements of a roug scale
  - Weight, height, latitude, temperatur
  - How to compute their differences?



Data Types in Cluster Analysis

- Interval-Scaled Variables
  - Continuous measurements of a roug scale
  - Weight, height, latitude, temperatur
  - How to compute their differences?



$$d(i,j) = \sqrt{|x_{i1} - x_{j1}|^2 + |x_{i2} - x_{j2}|^2 + \cdots + |x_{ip} - x_{jp}|^2},$$

Data Types in Cluster Analysis

Data Types in Cluster Analysis

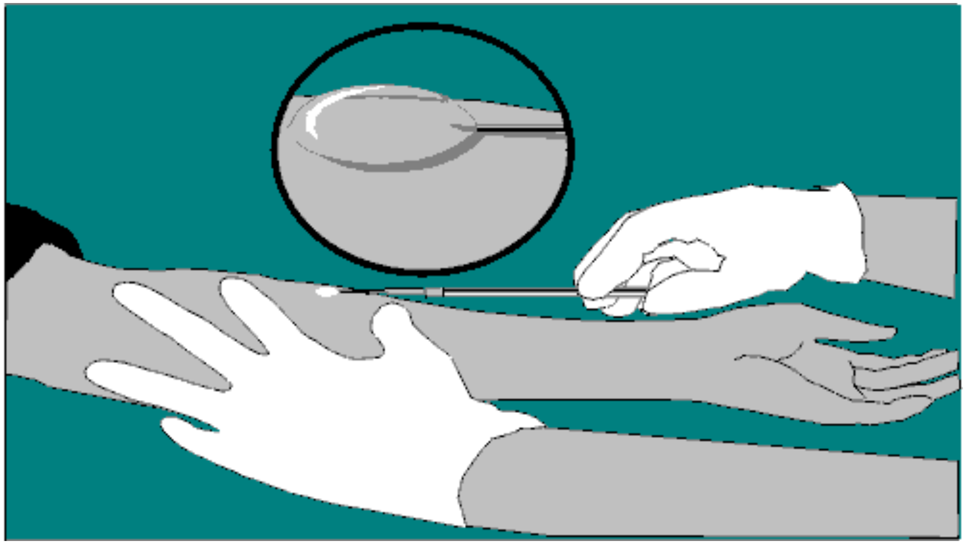
- Binary Variables
  - Only two states: 0 or 1
  - However, it can be symmetric/asymmetric
    - Symmetric – gender
    - Asymmetric – outcome of a disease test

Data Types in Cluster Analysis

- Binary Variables
  - Only two states: 0 or 1
  - However, it can be symmetric/asymmetric
    - Symmetric – gender
    - Asymmetric – outcome of a disease test

Data Types in Cluster Analysis

- Binary Variables
  - Only two states: 0 or 1
  - However, it can be symmetric/asymmetric
    - Symmetric – gender
    - Asymmetric – outcome of a disease test

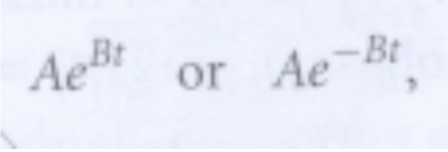


Data Types in Cluster Analysis

Data Types in Cluster Analysis

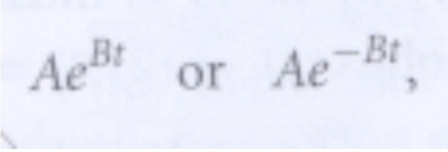
- Nominal Variables
  - A generalization of the binary variable in that it can take on more than two states.
  - For example, a color be white, green, blue, red.
  - How is dissimilarity computed?
    - Matching approach d(i,j)=(p-m)/p
    - M is the number of similar attributes between I and j
    - P is the number of total attributes between I and j

Data Types in Cluster Analysis



Ae^{Bt} or Ae^{-Bt},

Data Types in Cluster Analysis



Ae^{Bt} or Ae^{-Bt},

- Ratio-Scaled Variables
  - A positive measurement on a nonlinear scale, such as an exponential scale
  - Growth of bacteria population
  - Decay of radioactive element
  - How to compute dissimilarity?
    - Just like Interval-based variables
    - But needs a transformation:
      - Apply logarithmic transformation to a linearly ratio-scaled variable
      - Some times we may need to use log-log, log-log-log, and so on... Very exciting!

K-Mean Method

K-Mean Method

- **K-mean algorithm creates clusters by determining a central mean for each cluster**

- The algorithm starts by randomly select K entities as the means of K clusters and randomly adds entities to each clusters

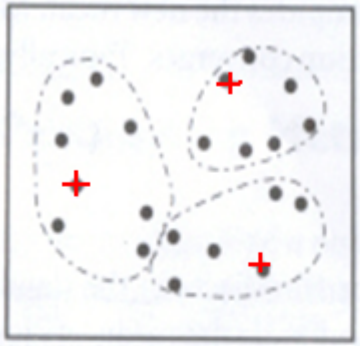
- Then, it re-computes cluster mean and re-assigns entities to clusters to which it is most similar, based on the distance between entity and the cluster mean.

K-Mean Method

- **K-mean algorithm creates clusters by determining a central mean for each cluster**

- The algorithm starts by randomly select K entities as the means of K clusters and randomly adds entities to each clusters

- Then, it re-computes cluster mean and re-assigns entities to clusters to which it is most similar, based on the distance between entity and the cluster mean.



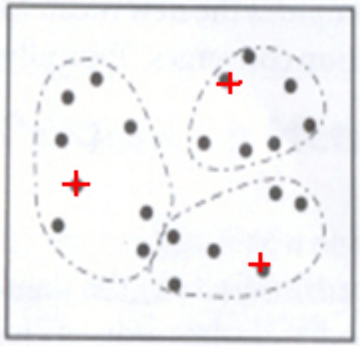
(a)

K-Mean Method

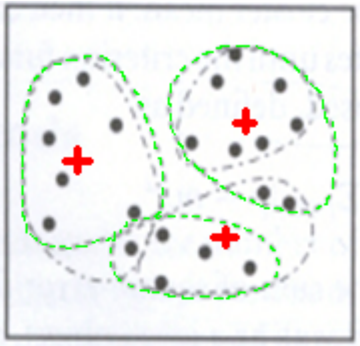
- **K-mean algorithm creates clusters by determining a central mean for each cluster**

- The algorithm starts by randomly select K entities as the means of K clusters and randomly adds entities to each clusters

- Then, it re-computes cluster mean and re-assigns entities to clusters to which it is most similar, based on the distance between entity and the cluster mean.



(a)



(b)

K-Mean Method

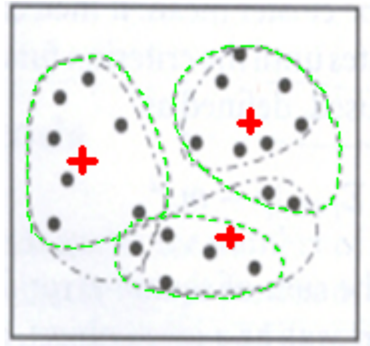
K-Mean Method

- Then, the mean is recomputed at each cluster, and previous entities either stay / move to a different cluster, and one iteration completes

- Algorithm iterates until there is no change of the means at each clusters.

K-Mean Method

- Then, the mean is recomputed at each cluster, and previous entities either stay / move to a different cluster, and one iteration completes

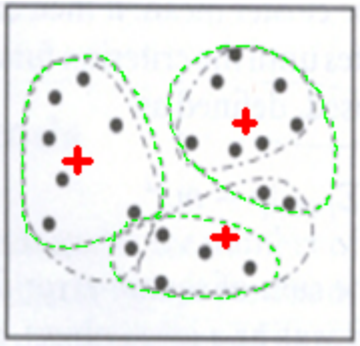


(b)

- Algorithm iterates until there is no change of the means at each clusters.

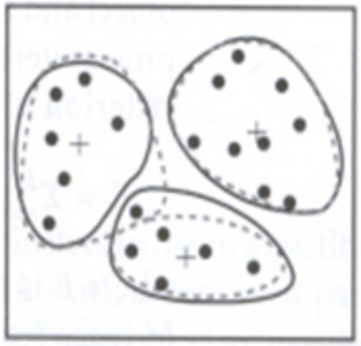
K-Mean Method

- Then, the mean is recomputed at each cluster, and previous entities either stay / move to a different cluster, and one iteration completes



(b)

- Algorithm iterates until there is no change of the means at each clusters.



(c)

K-Mean Method

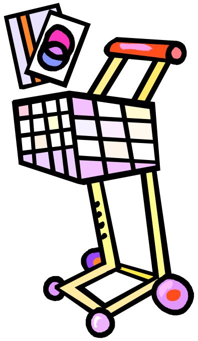
K-Mean Method

- K-mean is fast!
  - Computation complexity is O(K*n*t)
  - K is the number of clusters
  - N is the total number of objects
  - T is the number of iterations
- But K-mean are sensitive to outliers!
  - Outliers at the edge of the cluster may cause the cluster creates a skewed mean.

K-Mean Method

- K-mean is fast!
  - Computation complexity is O(K*n*t)
  - K is the number of clusters
  - N is the total number of objects
  - T is the number of iterations
- But K-mean are sensitive to outliers!
  - Outliers at the edge of the cluster may cause the cluster creates a skewed mean.

Market Basket Analysis



- Retail – each customer purchases different set of products, different quantities, different times
- MBA uses this information to:
  - Identify who customers are (not by name)
  - Understand why they make certain purchases
  - Gain insight about its merchandise (products):
    - Fast and slow movers
    - Products which are purchased together
    - Products which might benefit from promotion
  - Take action:
    - Store layouts
    - Which products to put on specials, promote, coupons…
- Combining all of this with a customer loyalty card it becomes even more valuable

Transactional Data

Market basket example:

Basket1: {bread, cheese, milk}

Basket2: {apple, eggs, salt, yogurt}

…

Basketn: {biscuit, eggs, milk}

Definitions:

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

Itemsets and Association Rules

- An *itemset* is a set of items.
  - E.g., {milk, bread, cereal} is an itemset.
- A *k-itemset* is an itemset with k items.
- Given a dataset *D*, an itemset *X* has a (frequency) *count* in *D*
- An *association rule* is about relationships between two disjoint itemsets *X* and *Y*

$$X \Rightarrow Y$$

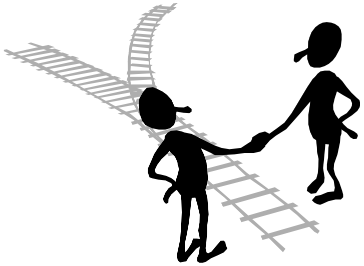
- It presents the pattern when *X* occurs, *Y* also occurs

Use of Association Rules

- Association rules do not represent any sort of causality or correlation between the two itemsets.
  - $X \Rightarrow Y$ does not mean X causes Y, so no Causality
  - $X \Rightarrow Y$ can be different from $Y \Rightarrow X$, unlike correlation
- Association rules assist in marketing, targeted advertising, floor planning, inventory control, churning management, homeland security, …

A 100-year old furniture giant (W…) claimed bankruptcy, who started the brochure shopping in the US.

Association Rules



- DM technique most closely allied with Market Basket Analysis
- AR can be automatically generated
  - AR represent patterns in the data without a specified target variable
  - Good example of undirected data mining
  - Whether patterns make sense is up to humanoids (us!)

Association Rules Apply Elsewhere

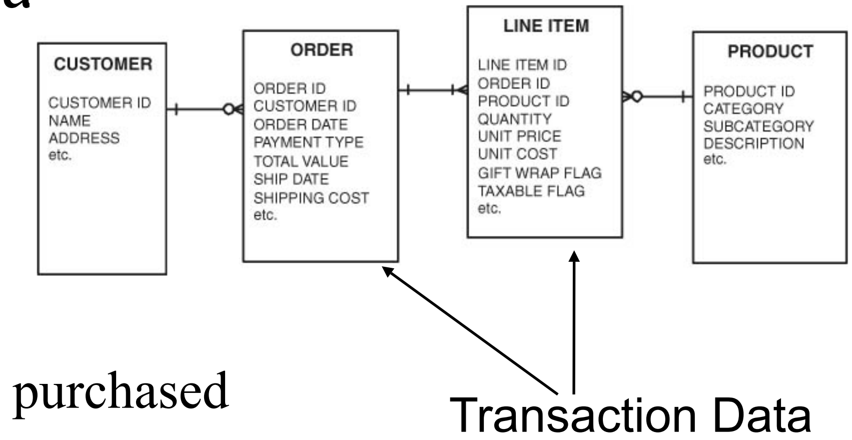
- Besides retail – supermarkets, etc…
- Purchases made using credit/debit cards
- Optional Telco Service purchases
- Banking services
- Unusual combinations of insurance claims can be a warning of fraud
- Medical patient histories

Market Basket Analysis Drill-Down

- MBA is a set of techniques, Association Rules being most common, that focus on point-of-sale (p-o-s) transaction data
- 3 types of market basket data (p-o-s data)
  - Customers
  - Orders (basic purchase data)
  - Items (merchandise/services purchased)

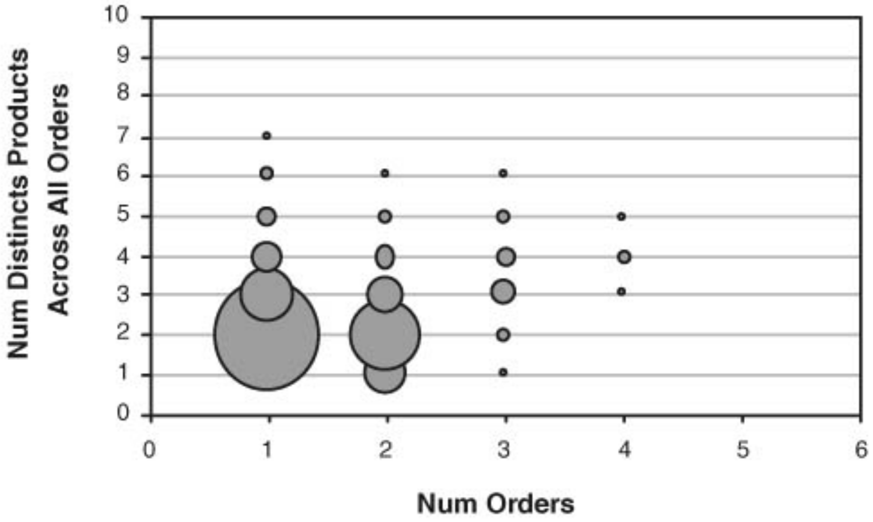
Typical Data Structure (Relational Database)

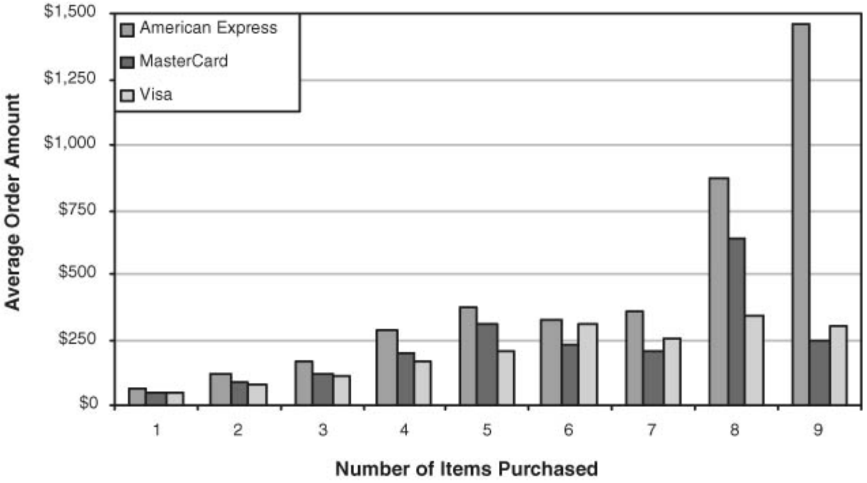
- Lots of questions can be answered
  - Avg # of orders/customer
  - Avg # unique items/order
  - Avg # of items/order
  - For a product
    - What % of customers have purchased
    - Avg # orders/customer include it
    - Avg quantity of it purchased/order
  - Etc…
- Visualization is extremely helpful…next slide



Transaction Data

Sales Order Characteristics

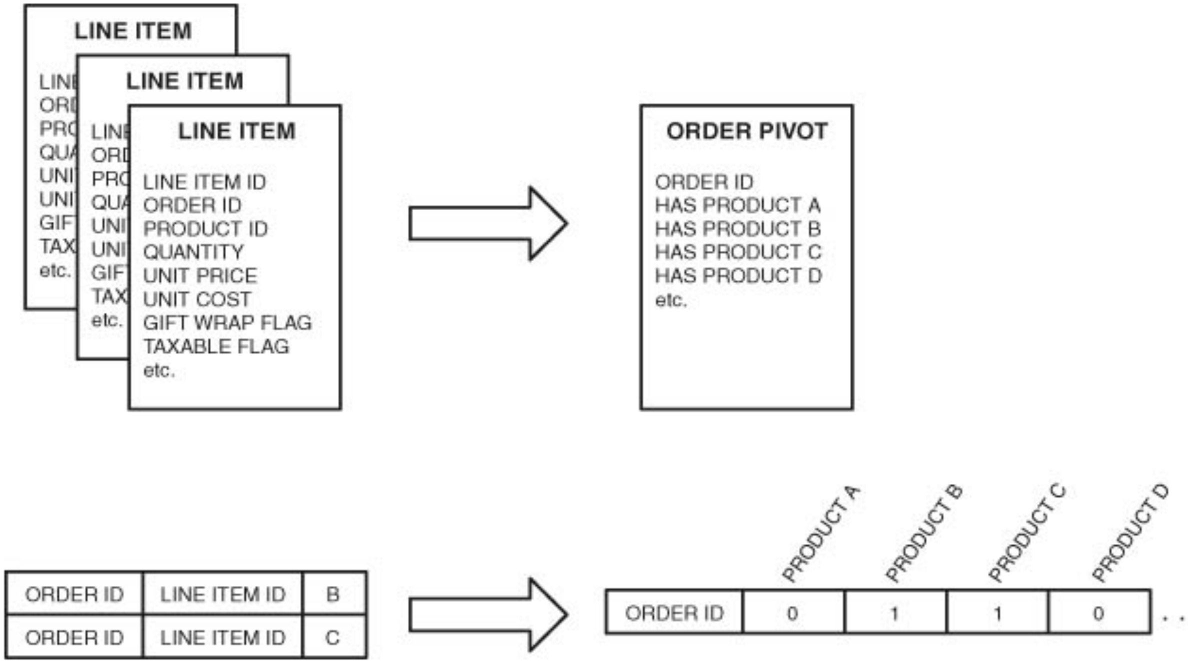




Sales Order Characteristics

- Did the order use gift wrap?
- Billing address same as Shipping address?
- Did purchaser accept/decline a cross-sell?
- What is the most common item found on a one-item order?
- What is the most common item found on a multi-item order?
- What is the most common item for repeat customer purchases?
- How has ordering of an item changed over time?
- How does the ordering of an item vary geographically?
- Yada…yada…yada…

Pivoting for Cluster Algorithms



LINE ITEM

LINE ITEM

LINE ITEM

LINE ITEM ID
ORDER ID
PRODUCT ID
QUANTITY
UNIT PRICE
UNIT COST
GIFT WRAP FLAG
TAXABLE FLAG
etc.

ORDER PIVOT

ORDER ID
HAS PRODUCT A
HAS PRODUCT B
HAS PRODUCT C
HAS PRODUCT D
etc.

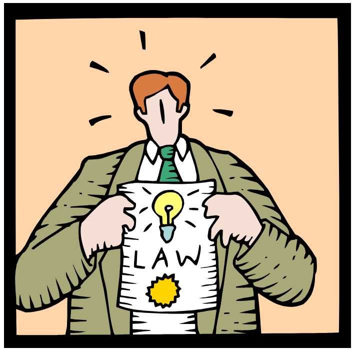
ORDER ID	LINE ITEM ID	B
ORDER ID	LINE ITEM ID	C

	PRODUCT A	PRODUCT B	PRODUCT C	PRODUCT D	
ORDER ID	0	1	1	0	. .

Association Rules

- Wal-Mart customers who purchase Barbie dolls have a 60% likelihood of also purchasing one of three types of candy bars [*Forbes*, Sept 8, 1997]
- Customers who purchase maintenance agreements are very likely to purchase large appliances (author experience)
- When a new hardware store opens, one of the most commonly sold items is toilet bowl cleaners (author experience)
- So what…

Association Rules



- Association rule types:
  - Actionable Rules – contain high-quality, actionable information
  - Trivial Rules – information already well-known by those familiar with the business
  - Inexplicable Rules – no explanation and do not suggest action
- Trivial and Inexplicable Rules occur most often

How Good is an Association Rule?

POS Transactions

Customer	Items Purchased
1	OJ, soda
2	Milk, OJ, window cleaner
3	OJ, detergent
4	OJ, detergent, soda
5	Window cleaner, soda

Co-occurrence of Products

	OJ	Window cleaner	Milk	Soda	Detergent
OJ	4	1	1	2	2
Window cleaner	1	2	1	1	0
Milk	1	1	1	0	0
Soda	2	1	0	3	1
Detergent	2	0	0	1	2

How Good is an Association Rule?

	OJ	Window cleaner	Milk	Soda	Detergent
OJ	4	1	1	2	2
Window cleaner	1	2	1	1	0
Milk	1	1	1	0	0
Soda	2	1	0	3	1
Detergent	2	0	0	1	2

Simple patterns:

1. OJ and soda are more likely purchased together than any other two items
2. Detergent is never purchased with milk or window cleaner
3. Milk is never purchased with soda or detergent

How Good is an Association Rule?

← POS Transactions

Customer	Items Purchased
1	OJ, soda
2	Milk, OJ, window cleaner
3	OJ, detergent
4	OJ, detergent, soda
5	Window cleaner, soda

- What is the confidence for this rule:
  - If a customer purchases soda, then customer also purchases OJ
  - 2 out of 3 soda purchases also include OJ, so 67%
- What about the confidence of this rule reversed?
  - 2 out of 4 OJ purchases also include soda, so 50%
- **Confidence** = Ratio of the number of transactions with all the items to the number of transactions with just the “if” items

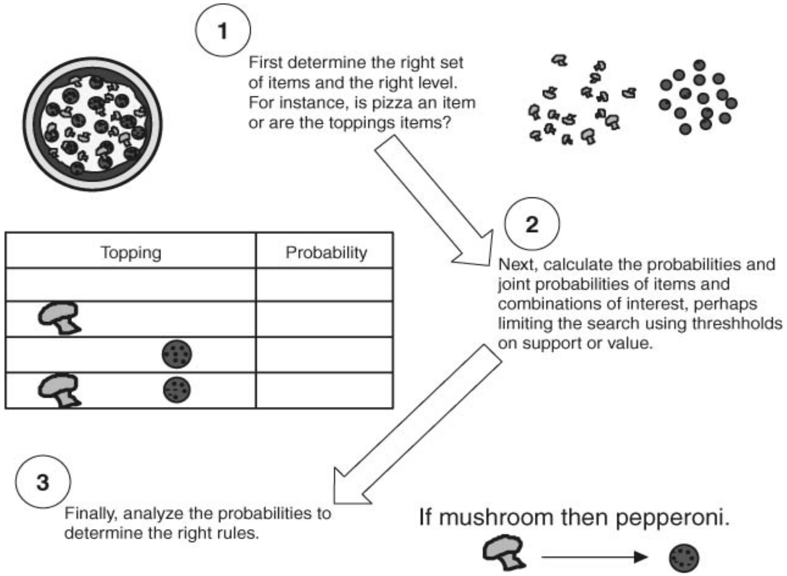
How Good is an Association Rule?

- How much better than chance is a rule?
- Lift (improvement) tells us how much better a rule is at predicting the result than just assuming the result in the first place
- **Lift** is the ratio of the records that support the entire rule to the number that would be expected, assuming there was no relationship between the products
- Calculating lift…p 310…When lift > 1 then the rule is better at predicting the result than guessing
- When lift < 1, the rule is doing worse than informed guessing and using the **Negative Rule** produces a better rule than guessing

- Co-occurrence can occur in 3, 4, or more dimensions…

Creating Association Rules

1. Choosing the right set of items
2. Generating rules by deciphering the counts in the co-occurrence matrix
3. Overcoming the practical limits imposed by thousands or tens of thousands of unique items



1 First determine the right set of items and the right level. For instance, is pizza an item or are the toppings items?

2 Next, calculate the probabilities and joint probabilities of items and combinations of interest, perhaps limiting the search using threshholds on support or value.

3 Finally, analyze the probabilities to determine the right rules.

If mushroom then pepperoni.

Overcoming Practical Limits for Association Rules

1. Generate co-occurrence matrix for single items…"*if OJ then soda"*
2. Generate co-occurrence matrix for two items…"*if OJ and Milk then soda"*
3. Generate co-occurrence matrix for three items…"*if OJ and Milk and Window* Cleaner" then soda
4. Etc…

Final Thought on Association Rules: The Problem of Lots of Data

- Fast Food Restaurant…could have 100 items on its menu
  - How many combinations are there with 3 different menu items? 161,700 !
- Supermarket…10,000 or more unique items
  - 50 million 2-item combinations
  - 100 billion 3-item combinations
- Use of product hierarchies (groupings) helps address this common issue
- Finally, know that the number of transactions in a given time-period could also be huge (hence expensive to analyze)

Support and Confidence

- *support* of X in D is *count*(X)/|D|
- For an association rule $X \Rightarrow Y$, we can calculate
  - support ($X \Rightarrow Y$) = support (XY)
  - confidence ($X \Rightarrow Y$) = support (XY)/support (X)
- Relate Support (S) and Confidence (C) to Joint and Conditional probabilities
- There could be exponentially many A-rules
- Interesting association rules are (for now) those whose S and C are greater than minSup and minConf (some thresholds set by data miners)

- How is it different from other algorithms
  - Classification (supervised learning -> classifiers)
  - Clustering (unsupervised learning -> clusters)
- Major steps in association rule mining
  - Frequent itemsets generation
  - Rule derivation
- Use of support and confidence in association mining
  - S for frequent itemsets
  - C for rule derivation

Example

- Data set *D*

TID	Itemsets
T100	1 3 4
T200	2 3 5
T300	1 2 3 5
T400	2 5

Count, Support, Confidence:

Count(13)=2

$|D| = 4$

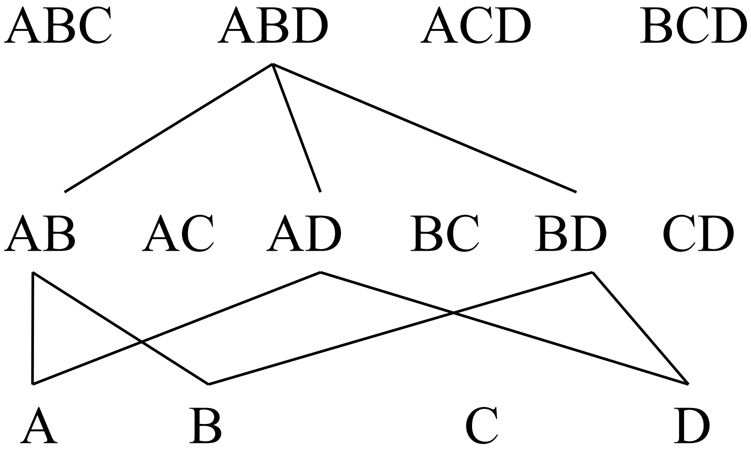
Support(13)=0.5

Support(3→2)=0.5

Confidence(3→2)=0.67

Frequent itemsets

- A *frequent* (used to be called large) *itemset* is an itemset whose support (S) is ≥ minSup.
- Apriori property (downward closure): any subsets of a frequent itemset are also frequent itemsets



APRIORI

- Using the downward closure, we can prune unnecessary branches for further consideration
- APRIORI
  1. k = 1
  2. Find frequent set L_k from C_k of all candidate itemsets
  3. Form C_{k+1} from L_k; k = k + 1
  4. Repeat 2-3 until C_k is empty
- Details about steps 2 and 3
  - Step 2: scan *D* and count each itemset in C_k , if it’s greater than minSup, it is frequent
  - Step 3: next slide

Apriori's Candidate Generation

- For k=1, C_1 = all 1-itemsets.
- For k>1, generate C_k from L_{k-1} as follows:
  - *The join step*

    C_k = k-2 way join of L_{k-1} with itself

    If both $\{a_1, \ldots, a_{k-2}, a_{k-1}\}$ & $\{a_1, \ldots, a_{k-2}, a_k\}$ are in L_{k-1}, then add $\{a_1, \ldots, a_{k-2}, a_{k-1}, a_k\}$ to C_k

    (We keep items **sorted**).
  - *The prune step*

    Remove $\{a_1, \ldots, a_{k-2}, a_{k-1}, a_k\}$ if it contains a non-frequent (k-1) subset

Example – Finding frequent itemsets

Dataset D

TID	Items
T100	a1 a3 a4
T200	a2 a3 a5
T300	a1 a2 a3 a5
T400	a2 a5

minSup=0.5

1. scan D ➔ C_1: a1:2, a2:3, a3:3, a4:1, a5:3

➔ L_1: a1:2, a2:3, a3:3, a5:3

➔ C_2: a1a2, a1a3, a1a5, a2a3, a2a5, a3a5

2. scan D ➔ C_2: a1a2:1, a1a3:2, a1a5:1, a2a3:2, a2a5:3, a3a5:2

➔ L_2: a1a3:2, a2a3:2, a2a5:3, a3a5:2

➔ C_3: a2a3a5

➔ Pruned C_3: a2a3a5

3. scan D ➔ L_3: a2a3a5:2

Order of items can make difference

Dataset D

TID	Items
T100	1 3 4
T200	2 3 5
T300	1 2 3 5
T400	2 5

minSup=0.5

1. scan D ➔ C_1: 1:2, 2:3, 3:3, 4:1, 5:3

➔ L_1: **1**:2, **2**:3, **3**:3, **5**:3

➔ C_2: 12, 13, 15, 23, 25, 35

2. scan D ➔ C_2: 12:1, **13:2**, 15:1, **23:2**, **25:3**, **35:2**

Suppose the order of items is: 5,4,3,2,1

➔ L_2: **31**:2, **32**:2, **52:**3, **53**:2

➔ C_3: 321, 532

➔ Pruned C_3: 532

3. scan D ➔ L_3: 532:2

The change of order in step 2 is to illustrate the concept of pruning.

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 ⇒B is an association rule if

    - Confidence (A ⇒ B) ≥ minConf,

      where support (A ⇒ B) = support (AB), and

      confidence (A ⇒ B) = support (AB) / support (A)

Example – deriving rules from frequent itemses

- Suppose 234 is frequent, with supp=50%
  - Proper nonempty subsets: 23, 24, 34, 2, 3, 4, with supp=50%, 50%, 75%, 75%, 75%, 75% respectively
  - These generate these association rules:
    - 23 => 4, confidence=100%
    - 24 => 3, confidence=100%
    - 34 => 2, confidence=67%
    - 2 => 34, confidence=67%
    - 3 => 24, confidence=67%
    - 4 => 23, confidence=67%
    - All rules have support = 50%

Deriving rules

- To recap, in order to obtain A ⇒B, we need to have Support(AB) and Support(A)
- This step is not as time-consuming as frequent itemsets generation
  - Why?
- It's also easy to speedup using techniques such as parallel processing.
  - How?
- Do we really need candidate generation for deriving association rules?
  - Frequent-Pattern Growth (FP-Tree)

Efficiency Improvement

- Can we improve efficiency?
  - Pruning without checking all k - 1 subsets?
  - Joining and pruning without looping over entire L_{k-1}?.
- Yes, one way is to use hash trees.
- One hash tree is created for each pass *k*
  - Or one hash tree for k-itemset, k = 1, 2, …

Hash Tree

- Storing all candidate *k*-itemsets and their counts.
- Internal node *v* at level *m* “contains” bucket pointers
  - Which branch next? Use hash of m^{th} item to decide
  - Leaf nodes contain lists of itemsets and counts
- E.g., C_2: 12, 13, 15, 23, 25, 35;   use identity hash function

```
                        {}                        ** root
                  /1              |2         \3              ** edge+label
    /2   |3   \5        /3   \5        /5
  [12:][13:] [15:]  [23:] [25:]   [35:]           ** leaves
```

- How to join using hash tree?
  - Only try to join frequent k-1 itemsets with *common parents* in the hash tree
- How to prune using hash tree?
  - To determine if a k-1 itemset is frequent with hash tree can avoid going through all itemsets of L_{k-1}. (The same idea as the previous item)
- Added benefit:
  - No need to enumerate all *k*-subsets of transactions. Use traversal to limit consideration of such subsets.
  - Or enumeration is replaced by tree traversal.

Further Improvement

- Speed up searching and matching
- Reduce number of transactions (a kind of instance selection)
- Reduce number of passes over data on disk
- Reduce number of subsets per transaction that must be considered
- Reduce number of candidates

Speed up searching and matching

- Use hash counts to filter candidates (see example)
- Method: When counting candidate k-1 itemsets, get counts of “hash-groups” of k-itemsets
  - Use a hash function *h* on k-itemsets
  - For each transaction *t* and k-subset *s* of *t*, add 1 to count of *h*(s)
  - Remove candidates q generated by Apriori if *h*(q)’s count <= minSupp
  - The idea is quite useful for k=2, but often not so useful elsewhere. (For sparse data, k=2 can be the most expensive for Apriori. Why?)

Hash-based Example

1,3,4

2,3,5

1,2,3,5

2,5

- Suppose *h_2* is:
  - $h_2(x,y)$ = ((order of x) * 10 + (order of y)) mod 7
  - E.g., $h_2(1,4)$ = 0, $h_2(1,5)$ = 1, …

	bucket0	bucket1	bucket2	bucket3	bucket4	bucket5	bucket6
	14	15	23	*24*	25	12	13
	35						34
counts	3	1	2	*0*	3	1	3

The order is 1 for 1, 2 for 2 … in this example.

Working on transactions

- Remove transactions that do not contain any frequent *k*-itemsets in each scan
- Remove from transactions those items that are not members of any candidate k-itemsets
  - e.g., if 12, 24, 14 are the only candidate itemsets contained in 1234, then remove item 3
  - if 12, 24 are the only candidate itemsets contained in transaction 1234, then remove the transaction from next round of scan.
- Reducing data size leads to less reading and processing time, but extra writing time

Reducing Scans via Partitioning

- Divide the dataset D into *m* portions, D_1, D_2,…, D_m, so that each portion can fit into memory.
- Find frequent itemsets F_i in D_i, with support ≥ minSup, for each *i*.
  - *If it is frequent in D, it must be frequent in some* D_i.
- The union of all F_i forms a candidate set of the frequent itemsets in D; get their counts.
- Often this requires only two scans of D.

Unique Features of Association Rules

- vs. classification
  - Right hand side can have any number of items
  - It can find a classification like rule X ⇒ *c* in a different way: such a rule is not about differentiating classes, but about what (X) describes class *c*
- vs. clustering
  - It does not have to have class labels
  - For X ⇒ Y, if Y is considered as a cluster, it can form different clusters sharing the same description (X).

Other Association Rules

- Multilevel Association Rules
  - Often there exist structures in data
  - E.g., yahoo hierarchy, food hierarchy
  - Adjusting minSup for each level
- Constraint-based Association Rules
  - Knowledge constraints
  - Data constraints
  - Dimension/level constraints
  - Interestingness constraints
  - Rule constraints

Measuring Interestingness - Discussion

- What are interesting association rules
  - Novel and actionable
- Association mining aims to look for “valid, novel, useful (= actionable) patterns.” Support and confidence are **not** sufficient for measuring interestingness.
- Large support & confidence thresholds ➔ only a small number of association rules, and they are likely “folklores”, or known facts.
- Small support & confidence thresholds ➔ too many association rules.

Post-processing

- Need some methods to help select the (**likely**) “interesting” ones from numerous rules
- Independence test
  - A ⇒ BC is perhaps interesting if p(BC|A) differs greatly from p(B|A) * p(C|A).
  - If p(BC|A) is approximately equal to p(B|A) * p(C|A), then the information of A ⇒ BC is likely to have been captured by A ⇒ B and A ⇒C already. Not interesting.
  - Often people are more familiar with simpler associations than more complex ones.