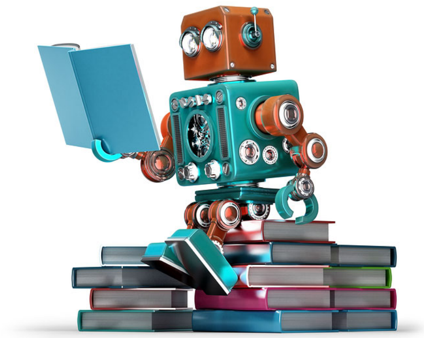
REASONING SYSTEMS
DAY 4



DAY 4 AGENDA

- Knowledge Discovery Using Data Mining Techniques
  - The Mining Process
  - Decision Tree
  - Cluster Analysis
  - Association Analysis

- Knowledge Discovery **Workshop**

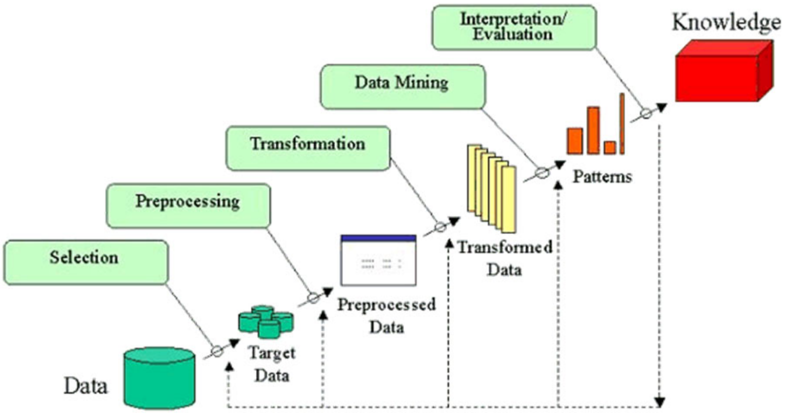
Knowledge Discovery in Databases (KDD)

Knowledge discovery in databases is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns or relationships in the data to make important decisions (Fayyad et al., 1996)

What Can Be Discovered?

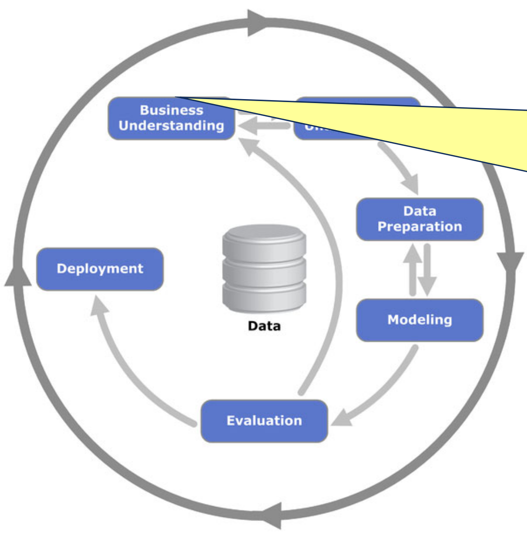
- **Predictive data mining – the task of building a model that can be used to predict the occurrence of an event**
  - A target variable to be predicted, therefore: **supervised learning**
  - Knowledge extracted from historic data, and the resulting model is applied to new situations
  - Classification and prediction using **decision trees**, regression, naïve Bayesian network, support vector machines, neural networks, etc.
- **Descriptive data mining – the task of providing a representation of the knowledge discovered without necessarily modelling a specific outcome**
  - No specific target variable, therefore: **unsupervised learning**
  - To identify patterns in the data that extend our knowledge and understanding of the world that the data reflects
  - **Cluster analysis** and **association rules**

KDD Process



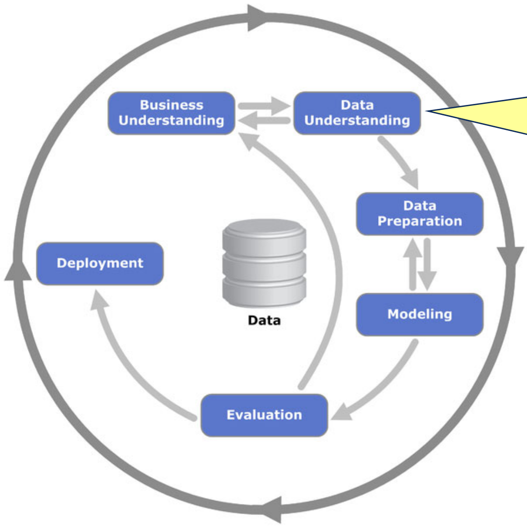
Fayyad et al, 1996

CRISP-DM Process



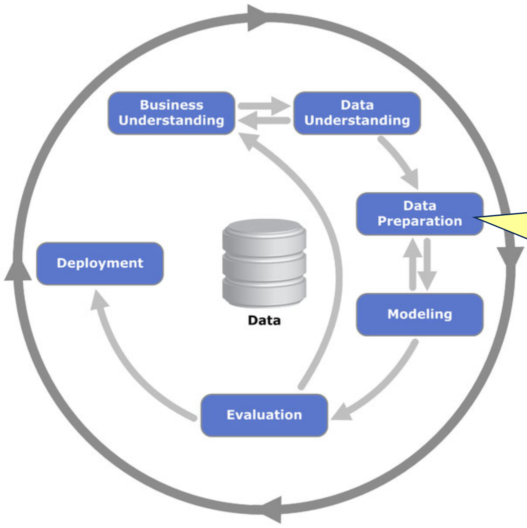
- Determine business objectives
- Assess situation
- Determine data mining goals
- Produce project plan

CRISP-DM Process



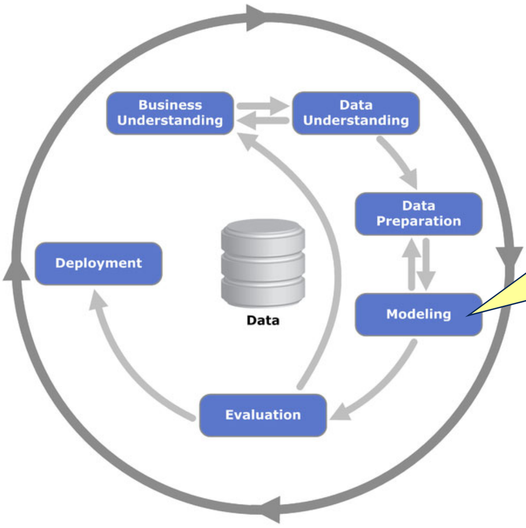
- Collect Initial data
- Describe data
- Explore data
- Verify data quality

CRISP-DM Process



- Select data
- Clean data
- Construct new data
- Integrate data
- Format data

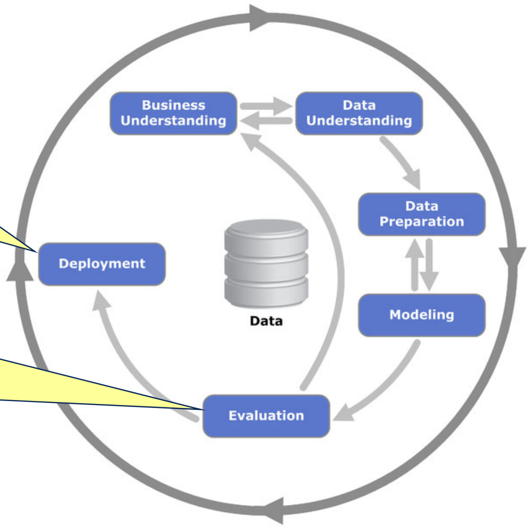
CRISP-DM Process



- Select modeling techniques
- Generate test design
- Build model
- Assess model

CRISP-DM Process

- Plan deployment
- Plan monitoring and maintenance
- Produce final report and presentation
- Review project



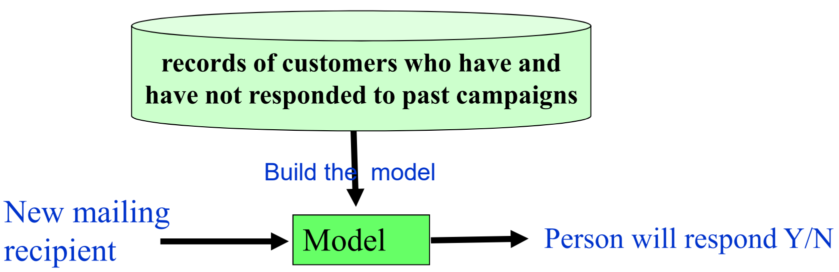
- Evaluate results
- Review process
- Determine next steps

DECISION TREE

KNOWLEDGE INDUCTION

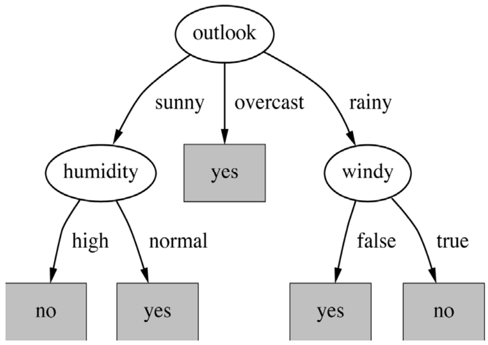
Induction

- Induction is a technology that automatically extracts knowledge from training examples in a structured form, such as a **decision tree** or a set of **rules**.
- It is an important technique for **predictive modelling**.
- Typical inductive algorithms - ID3/C4.5/C5.0



Decision Tree Example: Play or not Play?

Outlook	Temperature	Humidity	Windy	Play?
sunny	hot	high	false	No
sunny	hot	high	true	No
overcast	hot	high	false	Yes
rainy	mild	high	false	Yes
rainy	cool	normal	false	Yes
rainy	cool	normal	true	No
overcast	cool	normal	true	Yes
sunny	mild	high	false	No
sunny	cool	normal	false	Yes
rainy	mild	normal	false	Yes
sunny	mild	normal	true	Yes
overcast	mild	high	true	Yes
overcast	hot	normal	false	Yes
rainy	mild	high	true	No



ID3

- An early technique by Ross Quinlan
- ID3 uses a heuristics called ***information gain*** to find the most promising attribute on which to divide the data set.
- At each node in the decision tree, the inductive process evaluates the information gain for all the relevant slots.
- It then picks the one that, if answered, yields the highest increase of the information gain measure.
- The process is iterated onto the child nodes, until some stopping criteria are encountered:
  - No attributes left to consider
  - All data being considered at the node have the same value

ID3

- **Entropy**: the uncertainty about the value of the classification target *T*
- The **information gain** of an attribute *A* is the reduction of the entropy of *T* due to knowing the value of *A*.
- If *C* is the current case base, and the *k* values of the target *T* occur with relative frequencies $p_1,\ldots,p_k$ in *C,* then the **entropy** of *C* with respect to *T* is:

$$E_T(C) = \sum_{i=1}^{k} -p_i \log_2 p_i$$

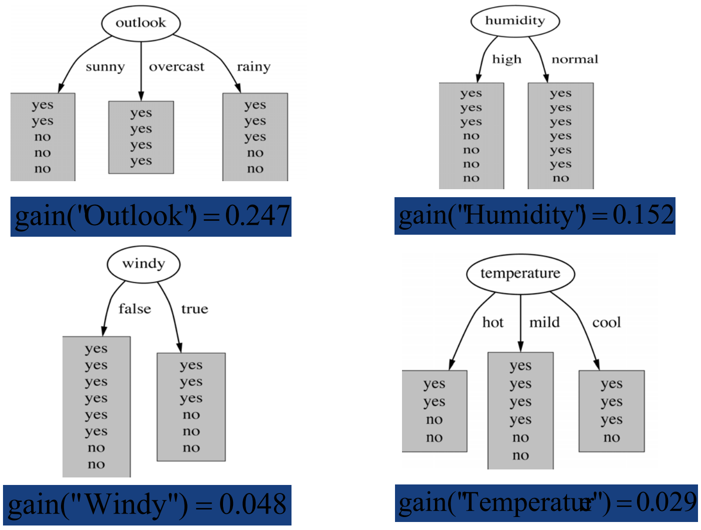
ID3

- If the values of *A* partition the case base C into *l* subsets $C_1,\ldots,C_l$, and, within subset C_j the *k* values of *T* occur with relative frequencies $q_{1j},\ldots,q_{kj}$, then the **conditional entropy** of C with respect to *T* when *A* is known is:

$$E_T(C \mid A) = \sum_{j=1}^{l} -p_j \sum_{i=1}^{k} q_{ij} \log_2 q_{ij}$$

- The **information gain** of *A* within C with respect to target *T* is:
  - $IG_T(A,C) = E_T(C) - E_T(C \mid A)$

Which Attribute to Select?



C4.5 algorithm

- C4.5 - extension of the basic ID3 algorithm designed by Quinlan to address the following issues:
  - Avoiding overfitting the data
    - Determining how deeply to grow a decision tree
    - Reduced error pruning
    - Rule post-pruning
  - Handling continuous attributes e.g. temperature
  - Choosing an appropriate attribute selection measure.
  - Handling training data with missing attribute values.
  - Handling attributes with differing costs
  - Improving computational efficiency

Test the Tree

- Divide the data into training and test sets randomly, e.g.

70% training —*Build Model and Apply to*→ **30% test**

- Overall accuracy – correct prediction/total (%)
- Confusion matrix

	Predicted buyer	Predicted Non-buyer	Total
Actual Buyer	200	100	300
Actual Non-Buyer	800	1900	2700
Total	1000	2000	3000

Decision Tree & Rule Sets

```
Na_to_K <= 14.64 [Mode: drugX] (109)
    BP = HIGH [Mode: drugA] (39)
        Age <= 50 [Mode: drugA] ⇨ drugA (23, 1.0)
        Age > 50 [Mode: drugB] ⇨ drugB (16, 1.0)
    BP = LOW [Mode: drugX] (34)
        Cholesterol = NORMAL [Mode: drugX] ⇨ drugX (18, 1.0)
        Cholesterol = HIGH [Mode: drugC] ⇨ drugC (16, 1.0)
    BP = NORMAL [Mode: drugX] ⇨ drugX (36, 1.0)
Na_to_K > 14.64 [Mode: drugY] ⇨ drugY (91, 1.0)
```

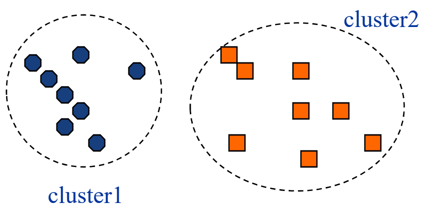
```
Rules for drugA - contains 1 rule(s)
    Rule 1 for drugA (23, 0.96)
        if    Age <= 50
        and   BP = HIGH
        and   Na_to_K <= 14.64
        then  drugA
Rules for drugB - contains 1 rule(s)
    Rule 1 for drugB (16, 0.944)
        if    Age > 50
        and   BP = HIGH
        and   Na_to_K <= 14.64
        then  drugB
Rules for drugC - contains 1 rule(s)
    Rule 1 for drugC (16, 0.944)
        if    BP = LOW
        and   Cholesterol = HIGH
        and   Na_to_K <= 14.64
        then  drugC
```

CLUSTER ANALYSIS

DATA PROFILING

Clustering: Definition

"Partition a database so that records that have similar characteristics are grouped together"



Why do Clustering?

- Learn something new about the data
  - Understanding the natural structure in the data may lead to knowledge discovery
- Simplify the data mining problem
  - Big databases often have too much complex structure for successful analysis. Analysis of smaller, homogenous clusters may yield better results
- Use the clusters as predictive models
  - E.g. cluster customer sales data to find groups of "typical" buyers. Predict new buyers by measuring their similarity to these clusters

E.g. "clustering showed that our customers are mostly either young + married + car owners or retired!"

Clustering Algorithms

- Clustering algorithms generally calculate the distance between different records and try to group the ones that are closest together
- Partitioning Clustering
  - K-Means Clustering
  - K-Medoids Clustering
  - ...
- Hierarchical Clustering
  - Agglomerative Clustering
  - Divisive Clustering
  - ...
- Other Algorithms – model-based, density-based, grid-based…

Measuring Similarity/Distance

Euclidean Distance is commonest for numerical variables

$$d_{xy} = \sqrt{\sum_{k=1}^{p} (x_i - y_i)^2}$$

ID	Age	Income
S1234567D	21	5600
S3456782X	56	4600
B1725353Y	39	7000

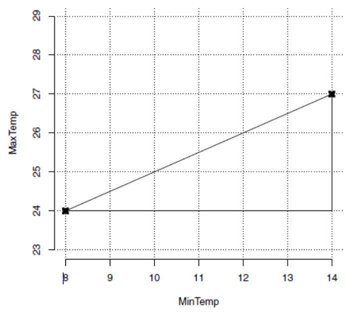
$\sqrt{(21-56)^2 + (5600-4600)^2}$

First normalise each variable to the range 0-1 to eliminate bias of "big" numbers – usually done by the tool

Other Distance Measures

- **Manhattan distance**:

$$d_{xy} = \sum_{i=1}^{p} |x_i - y_i|$$



- **Minkowski distance**

$$d_{xy} = \sqrt[q]{\sum_{i=1}^{p} (x_i - y_i)^q}$$

Measuring Similarity/Distance

- How to handle categorical fields?

Sex	Marital Status	Job
M	single	lawyer
M	divorced	doctor
F	married	lawyer

- Could preprocess into lots of 0/1 variables
  - is-male 0/1, is-female 0/1
  - is-single 0/1, is-married 0/1, is-divorced 0/1, is-widowed 0/1
  - is-lawyer 0/1, is-doctor 0/1 ……. etc…

Measuring Similarity/Distance

- If ordering is important then a better solution may be to assign numbers that reflect that ordering

  Which of the below do you think is reasonable?
  - *cold, warm, hot* $\rightarrow$ *0, 1, 2*
  - *single, married, divorced, widowed* $\rightarrow$ *0, 1, 2, 3*
  - *lawyer, doctor, engineer, teacher, ..* $\rightarrow$ *0, 1, 2, 3, 4, 5, 6, 7, 8,…*

- For vector objects, cosine similarity measure can be used.

Partitioning Algorithms: Basic Concept

- Partitioning method: Construct a partition of a database ***D*** of ***n*** objects into a set of ***k*** clusters
- Given a *k*, find a partition of *k clusters* that optimizes the chosen partitioning criterion
  - Global optimal: exhaustively enumerate all partitions
  - Heuristic methods: ***k-means*** and ***k-medoids*** algorithms
  - *k-means* (MacQueen'67): Each cluster is represented by the center of the cluster
  - *k-medoids* or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

The K-Means Clustering Method

- Given *k*, the *k-means* algorithm is implemented in four steps:

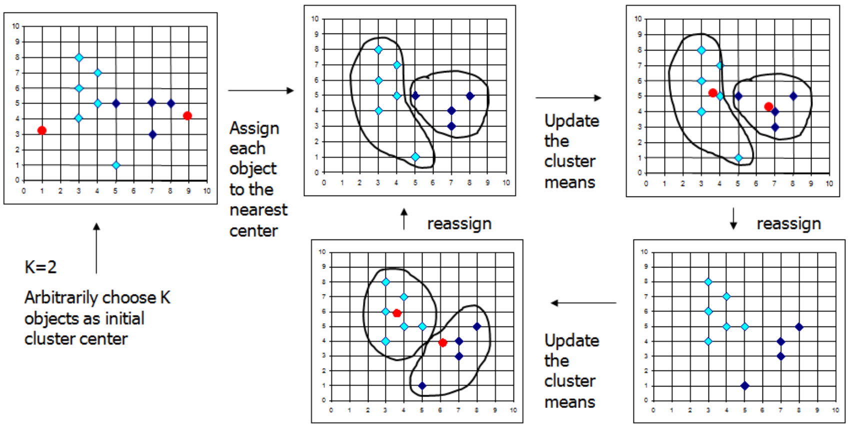
Step 1: Arbitrarily partition objects into *k* nonempty subsets

Step 2: Compute seed points as the centroids of the clusters of the current partition (the centroid is the center, i.e., *mean point*, of the cluster)

Step 3: Assign each object to the cluster with the nearest seed point

Step 4: Go back to Step 2, stop when no more new assignment

The K-Means Clustering Method

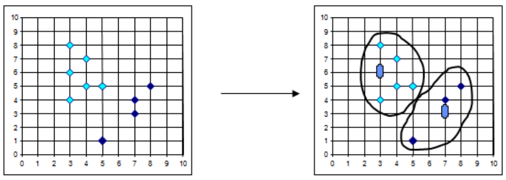


Pros and Cons of the K-Means Method

- **Strength** of the k-means:
  - Relatively efficient: O(tkn), where n is # of objects, k is # of clusters, and t is # of iterations. Normally, k, t << n
- **Issues** of the k-means:
  - Applicable only when mean is defined, then what about categorical data?
  - Need to specify *k*, the number of clusters, in advance
  - Unable to handle noisy data and outliers
  - Not suitable to discover clusters with non-convex shapes
  - Often terminates at a local optimum, with resulting clusters may not be the best and highly dependent upon initial partitioning or division of the data set
  - Difficult to determine best clustering

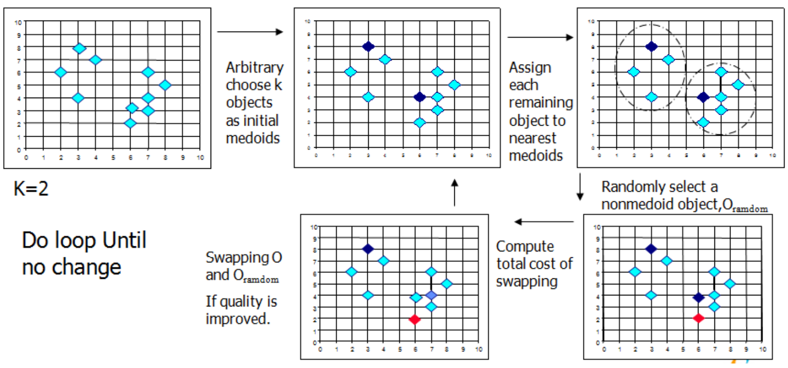
The K-Medoids Clustering Method

- The *k-means* algorithm is sensitive to outliers.
  - Since an object with an extremely large value may substantially distort the distribution of the data.
- *K-Medoids*: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster.



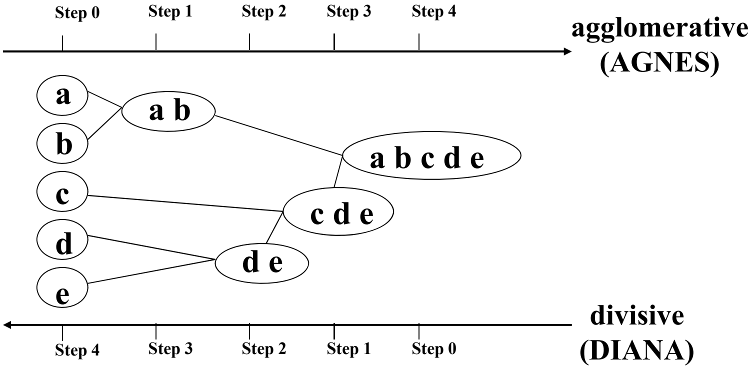
The K-Medoids Clustering Method

- PAM (Partitioning Around Medoids, 1987)
  - Starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering



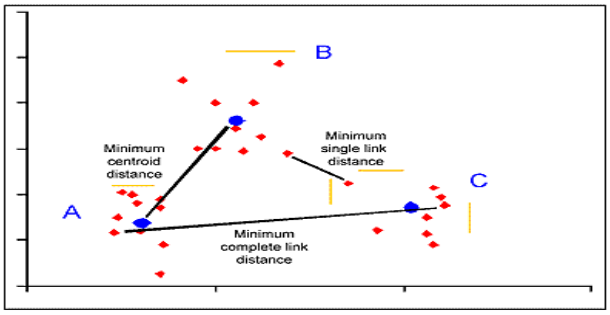
Hierarchical Clustering

- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition



Distance Measures between Clusters

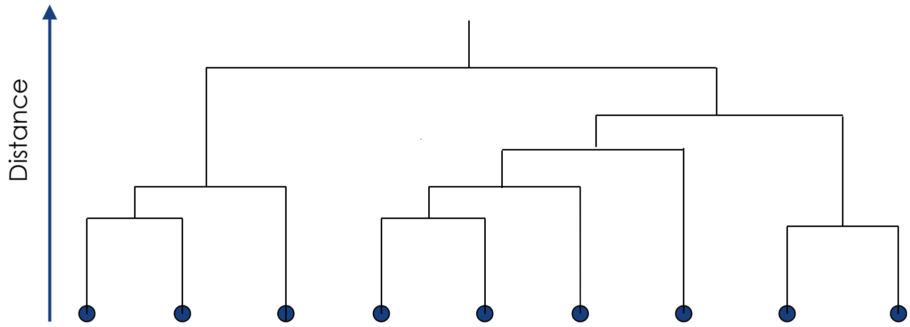
- ***Single link method***- The distance between two clusters is equal to the distance between the two closest records in them, aka *nearest neighbor method*.
- ***Complete link method***- The distance between two clusters is equal to the distance between the two most distant records in them, aka *furthest neighbor method*.
- ***Centroid method-*** The distance between two clusters is equal to the distance between their centroids.



Dendrogram

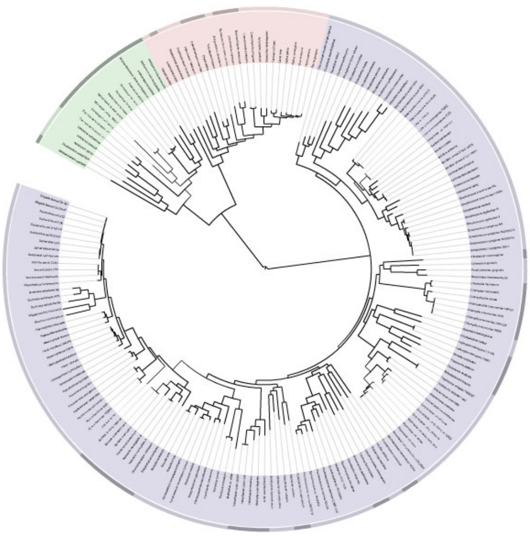
Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram.

A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.



Hierarchical Clustering Example

- Tree of Life: a hierarchical clustering of RNA sequences



http://itol.embl.de/itol.cgi

Hierarchical Clustering Methods

- Major weakness of agglomerative clustering methods
  - Do not scale well: time complexity of at least $O(n^2)$, where n is the number of total objects
  - Cannot undo what was done previously
- Integration of hierarchical with other clustering methods
  - BIRCH (1996): incrementally adjusts the quality of sub-clusters
  - CHAMELEON (1999): hierarchical clustering using dynamic modeling

Clustering Issues

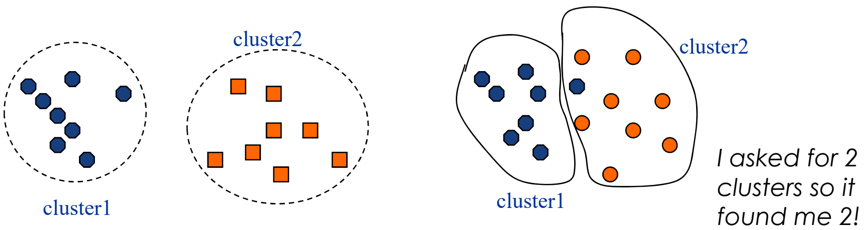
- Clustering is a very challenging data mining activity
- Problems and issues include:
  - Variable selection
  - Understanding the resulting clusters
  - Assessing the quality of the clusters
  - Utilising the clusters

Clustering Issues

- Variable selection
  - Clustering with too many variables produces poor clusters that are not homogeneous within clusters and heterogeneous between clusters
  - Clustering is "unsupervised learning" - there is no target variable to guide the selection of relevant versus non-relevant variables

Assessing the quality of the clusters

- Different clustering techniques can produce widely varying results leading analysts to ask the question: "If every technique results in a different answer, how do I know which one is correct?".
- Do the found clusters represent natural structure in the data or merely a by-product of the clustering algorithm?

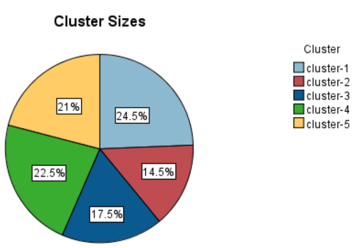


Quality of Clusters

- A good clustering method will produce high quality clusters in which:
  - the intra-class similarity is high.
  - the inter-class similarity is low.
- The quality of clusters derived can be measured using
  - *Cluster cohesion*: such as within cluster sum of squares, the squares of the differences between the observations within each of the clusters.
  - *Cluster separation*: such as the proximity of a cluster centroid to the overall centroid multiplied by the number of objects in the cluster.

Clustering Understanding

- Cluster Understanding
  - Clustering algorithms assign a cluster label (typically a number) to each record. How do we interpret what this means?



Looking at the number of clusters and their relative size is not really very informative if the goal is knowledge discovery!

  - DM tools provide various aids to help cluster understanding – visualisation aids are particularly useful

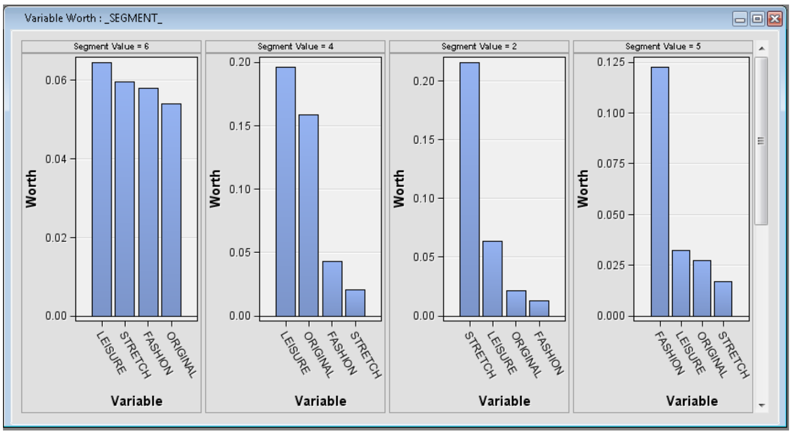
Cluster Visualisation

- Main issue:
  - Two or Three Dimensions are plausible but can we handle more than that?

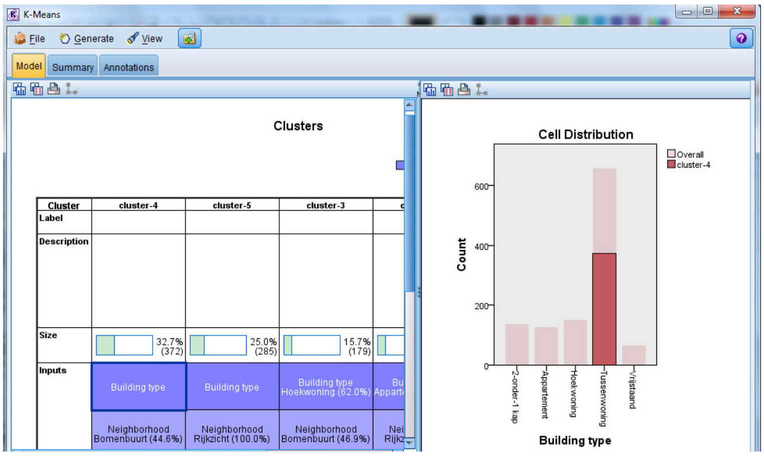
- Several approaches have been attempted with different tools:
  - SAS Enterprise Miner
  - IBM SPSS Modeler

SAS EM – Cluster Profile

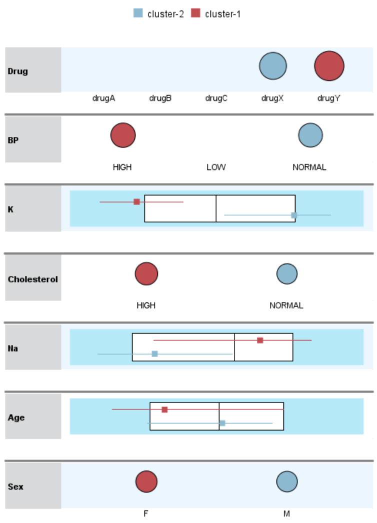
SAS EM – Cluster Profile



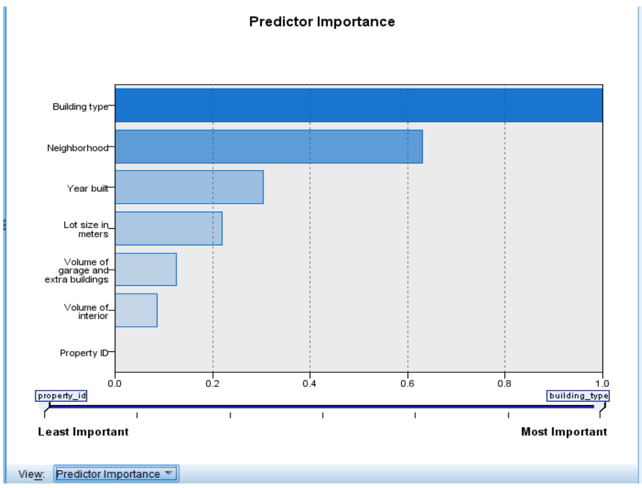
IBM SPSS Modeler– Cluster Profile



IBM SPSS Modeler– Cluster Comparison

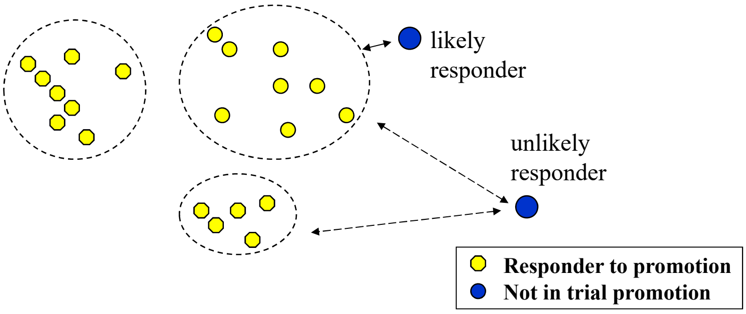


IBM SPSS Modeler– Predictor Importance



Utilising the clusters

- Analyse clusters for knowledge discovery
- Use of clusters as predictive (or other) models
  - E.g. to generate a mailing list given a list of responders to a previous mailing campaign



Words of caution

- **Most cluster analysis methods are heuristics**
  - Plausible methods for generating clusters
- **Different clustering methods can and do generate different solutions to the same data set**
  - Inherent bias with each clustering process
- **Clustering methods may impose spurious structure**
  - Clusters can even be formed out of random data

ASSOCIATION ANALYSIS

FREQUENTLY CO-OCCURRING PATTERNS

Association Analysis

- Has roots in analysis of point-of-sale (POS) transactions
  - Determine what products are purchased together or likely to be purchased by the same person
- Common applications
  - Cross-sell - make the purchasers of one product the targets for another
  - Up-sell – target customers likely to upgrade their product or service
- In general, when customers do multiple things in close proximity then there is a potential application

Example Applications

- Items purchased on a credit card (e.g. rental cars, hotel rooms) give insight into the next product the customer may buy
- Optional services bought by telecom customers (call waiting, forwarding, auto-roam etc) show how best to bundle these services
- Banking services used by retail customers (investment services, car loans, home loans, money market accounts etc) show possible cross-sells
- Unusual combinations of insurance claims may indicate fraud
- May find associations between certain combinations of medical treatments and complications in medical patients

Example Applications

- Targeted advertisement: product recommendation systems

Basic MBA

- Requires a list of transactions
- E.g. transactions at a convenience store
  - Transaction1: frozen pizza, cola, milk
  - Transaction2: milk, potato chips
  - Transaction3: cola, frozen pizza
  - Transaction4: milk, peanuts
  - Transaction5: cola, peanuts
  - Transaction6: cola, potato chips, peanuts

The Co-occurrence Table

- Cross-tabulate into a table to show how often each possible pair of products were sold together

Strong Cross-Sell opportunity:
Pizza buyers (2) always also buy cola (2)

	Pizza	Milk	Cola	Chips	P/nuts
Pizza	2	1	2	0	0
Milk	1	3	1	1	1
Cola	2	1	4	1	2
Chips	0	1	1	2	1
P/nuts	0	1	2	1	3

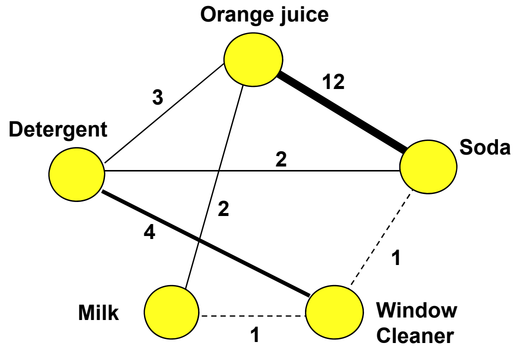
Milk sells well with everything!

Cola buyers (4) do not always buy pizza (2)

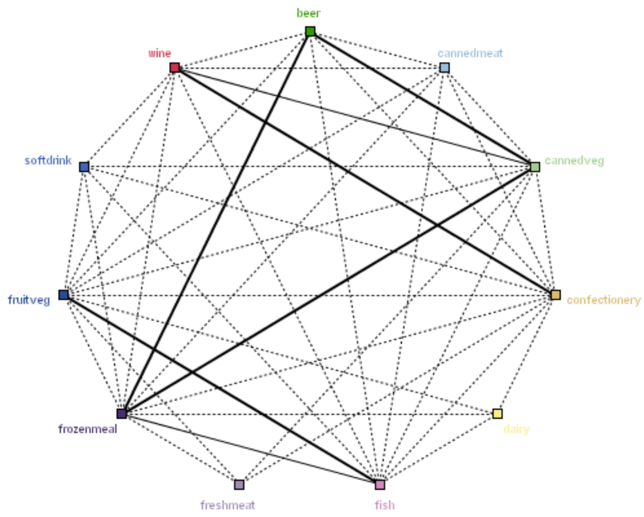
Weaker Cross-Sell opportunity:
Peanut buyers (3) nearly always also buy cola (2)

Link Analysis

- Nodes represent items, thickness of joining line indicates number of times they occurred together



Another Example (IBM SPSS Modeler)



Link Analysis: data format

- Convert transaction records into summary records for each customer

custID	fruitveg	freshmeat	dairy	cannedveg	beer	wine	softdrink	fish
39808	F	T	T	F	F	F	F	F
67362	F	T	F	F	F	F	F	F
10872	F	F	F	T	T	F	F	T
26748	F	F	T	F	F	T	F	F
91609	F	F	F	F	F	F	F	F
26630	F	T	F	F	F	T	F	T
62995	T	F	F	F	F	F	T	F
38765	F	F	F	F	T	F	F	F
28935	T	F	F	F	F	F	F	F
41792	T	F	F	F	F	F	F	T
59480	T	T	T	T	F	T	F	T
60755	T	F	F	F	F	F	F	T

The transformation function usually available in tools, e.g. Modeler has a SetToFlag node to do this

Association Rules

- Algorithmically search for associations between categorical variables and express as rules:

  If a customer buys Pizza *then he will also buy Cola*

  LHS *RHS*

  If Coffee and Milk then Sugar

  If BBQ charcoal then Sausages and Steak

Note: Some tools may restrict RHS to one product

- Analysis is based on generating ***frequent item sets***
- Algorithm is straightforward, generally finds the same associations as visual inspection & link analysis, but can take a long time to execute

Association Rule Algorithm

- Examine each possible rule and select those whose "goodness" score is above a threshold
  - E.g. If there are three items A, B, C, then possible rules are

    If A and B then C

    If A and C then B

    If B and C then A
- Typical scores
  - ***Support*** : probability of getting that combination, also known as *coverage*
  - **Confidence** : support(rule)/support(LHS) , also known as *accuracy*

    E.g. confidence(A->B) = support(A & B)/support(A) , or P(B | A)
  - ***Lift*** : the increased likelihood in seeing C in a transaction containing A & B

    E.g. confidence(A -> B)/support(B), or P(B | A)/P(B)

Rule Evaluation Example

- How good are the rules below?
  - **If a customer buys Pizza then they will also buy Cola (R1)**
  - **If a customer buys Peanuts then they will also buy Cola (R2)**
- Data:

  Total (100), Pizza (25), Peanut (40), Cola (40)

  Pizza & Cola (20), Peanut & Cola (20)
- **Support**
  - Support ~ probability of getting that combination
  - R1 support = 20% (20 trans. out of 100 included pizza & cola)
  - R2 support = 20% (20 trans. out of 100 included peanuts & cola)

Rule Evaluation Example

- **Confidence**
  - Confidence = Support combination/ Support condition (LHS)
  - R1: 80% (20 out of 25 transactions that contain pizza also contain cola)
  - R2: 50% (20 out of 40 trans that contain peanuts also contain cola)
- **Lift**
  - Lift = confidence (rule) / support (RHS)
  - R1: 2 (rule confidence 80% / support of cola 40%)
  - R2: 1.25 (rule confidence 50% / support of cola 40%)

Problems with Association Rules

- The basic algorithm is combinatorially explosive

E.g. If 100 products are for sale

Num. items	Num. combinations
1	100
2	4,950
3	161,700
4	3,921,255
5	75,287,520
6	1,192,052,400
8	186,087,894,300

Apriori Algorithm

- Reduces the number of rules to consider by…
  1. Find the **large item sets** from the transaction data
  2. Generate the association rules from the **large item sets**
- **Large item sets**, or **frequent item sets**: item sets that appear *frequently enough* (threshold parameter) in the data
- Based on the simple observation that all subsets of a frequent item set must also be frequent
  - If {*milk, bread, cheese*} is a frequent item set, so is each of the smaller item sets, {*milk, bread*}, {*milk, cheese*}, {*bread, cheese*}, {*milk*}, {*bread*}, and {*cheese*}
- Significantly reduces search space

Apriori Algorithm

Finding the large item sets

Scan	Candidates	Large item sets
1	{milk}{cola}{pizza} {peanuts}{chips}{mints}	{milk}{cola}{pizza} {peanuts}{chips}
2	{milk cola}{milk pizza} {milk peanuts} {milk chips}{cola pizza} {cola peanuts}{cola chips} {peanuts chips}	{cola peanuts} {cola pizza}
3	{cola peanuts pizza}	

Only consider item sets with size > N

Problems with Association Rules

- Hence can generate a huge number of rules, often trivial and with repetition:

  If coffee and milk then sugar
  If milk and sugar then coffee
  If sugar and coffee then milk

- Define minimum support and minimum confidence for rule pruning/filtering to get "strong" rules

- Analyst must make decisions regarding validity & importance of rules to be accepted (subjective)

Association Rules Examples

Modeler rules showing *support* & *confidence*

Rules have been sorted by *support*

Consequent	Antecedent	Support %	Confidence %
frozenmeal	cannedveg	30.300	57.100
beer	cannedveg	30.300	55.120
cannedveg	frozenmeal	30.200	57.280
beer	frozenmeal	30.200	56.290
frozenmeal	beer	29.300	58.020
confectionery	wine	28.700	50.170
wine	confectionery	27.600	52.170
beer	cannedveg frozenmeal	17.300	84.390
cannedveg	frozenmeal beer	17.000	85.880
frozenmeal	cannedveg beer	16.700	87.430

…

Association Rules Examples

Sorted by *confidence*

Consequent	Antecedent	Support %	Confidence %
cannedveg	freshmeat frozenmeal beer	3.000	96.670
frozenmeal	freshmeat cannedveg beer	3.100	93.550
cannedveg	cannedmeat frozenmeal beer	4.000	90.000
beer	fruitveg cannedveg frozenmeal	4.500	88.890
beer	freshmeat cannedveg frozenmeal	3.300	87.880
frozenmeal	cannedveg beer	16.700	87.430
frozenmeal	fruitveg cannedveg beer	4.600	86.960
beer	dairy cannedveg frozenmeal	2.300	86.960
frozenmeal	dairy cannedveg beer	2.300	86.960
cannedveg	frozenmeal beer	17.000	85.880

…

Example of a Misleading "Strong" Rule

- Transactions with respect to the purchase of computer games and videos
  - Total 10,000 transactions
  - 6,000 transactions included computer games
  - 7,500 transactions included videos
  - 4,000 included both
- With min support=30%, min confidence=60%, an association rule is discovered:
  - "buy computer games" => "buy videos" [support=40%, confidence=66%]
- However:
  - Probability of buying videos is actually 75%, even larger than 66%!
  - The association is in fact negative: buying computer games decreases the likelihood of buying videos.

Correlation Analysis Using Lift

- Uses lift to help filter out misleading "strong" association rules
- **Lift** – a simple correlation measure
  - Lift(A,B) = P(B | A)/P(B) = P({A, B})/(P(A)P(B))
  - A is independent of the occurrence of B if P({A,B})=P(A)P(B), ie lift=1
  - Otherwise, A and B are dependent and correlated.
    - Lift >1: positively correlated
    - Lift <1: negatively correlated
- For the rule in the previous slide
  - lift = P({game, video}) / (P(game)P(video))= 0.40/(0.60x0.75)=0.89
    => negative correlation
- Alternative method, the χ^2 measure

What about numerical variables?

- Binning is required, partitioning the ranges of quantitative variables into intervals
  - Equal-width binning
    The interval size of each bin is the same
  - Equal-frequency binning
    Each bin has approximately the same number of tuples assigned to it
  - Clustering-based binning
    Clustering is performed on the variable to group neighboring points (judged based on various distance measures) into the same bin