Ulster University

Classification and Clustering

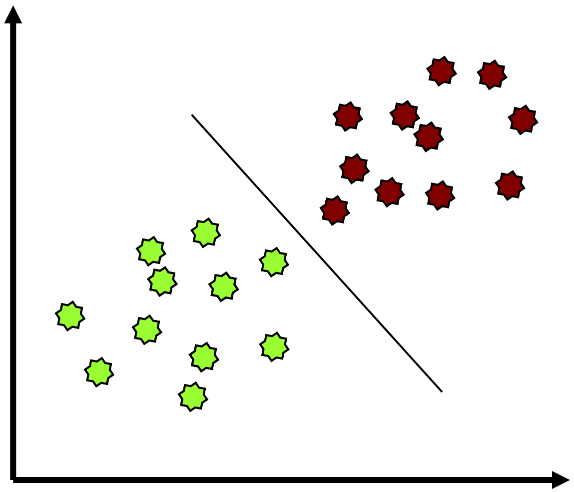
Professor Sally McClean
(si.mcclean@ulster.ac.uk)

Computer Science Research Institute
Coleraine

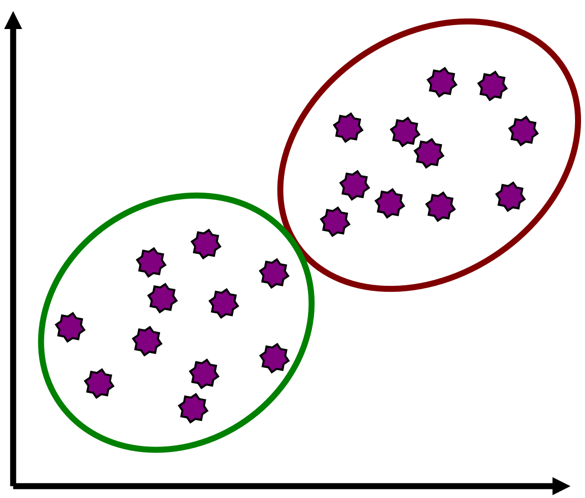
The Basic Idea

- Data characterized by one or more features

- Classification
  - We have labels for some points
  - Want a "rule" that will accurately assign labels to new points

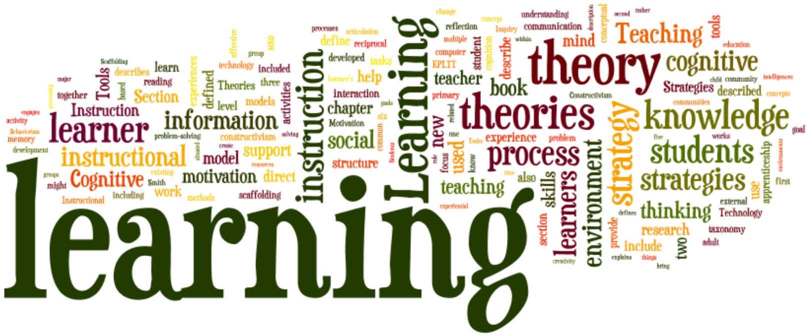


- Clustering
  - No labels
  - Clusters are based on how "near" the observations are to one another
  - Identify structure in data



Supervised versus Unsupervised Learning

- Supervised learning (classification)
  - Supervision: The training data (observations, measurements, etc.) are accompanied by **labels** indicating the class of the observations
  - New data is classified based on the training set
- Unsupervised learning (clustering)
  - The class labels of training data is unknown
  - We are given a set of data with the aim of finding classes or clusters in the data



Logistic Regression (LR): a simple approach

- Logistic regression(LR) is a type of regression that allows the prediction of discrete variables by a mix of continuous and discrete features.
- We model log odds as a linear combination of suitable features

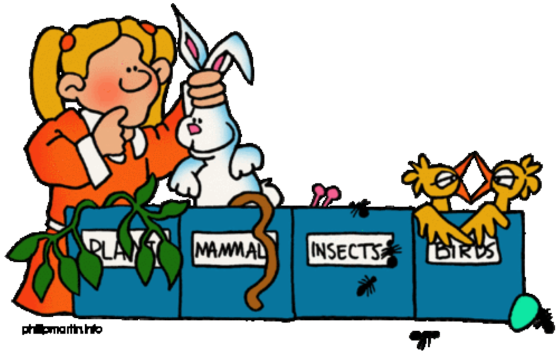
$$\log\left(\frac{\pi}{1-\pi}\right) = a_1 x_1 + \ldots + a_n x_n \text{ where}$$

π is the probability of class 1 and $(1-\pi)$ is the probability of class 2.

$\left(\frac{\pi}{1-\pi}\right)$ are the odds.

$x_1, \ldots, x_n$ are the features.

- Often we perform stepwise logistic regression which chooses the best features from the *n* available.

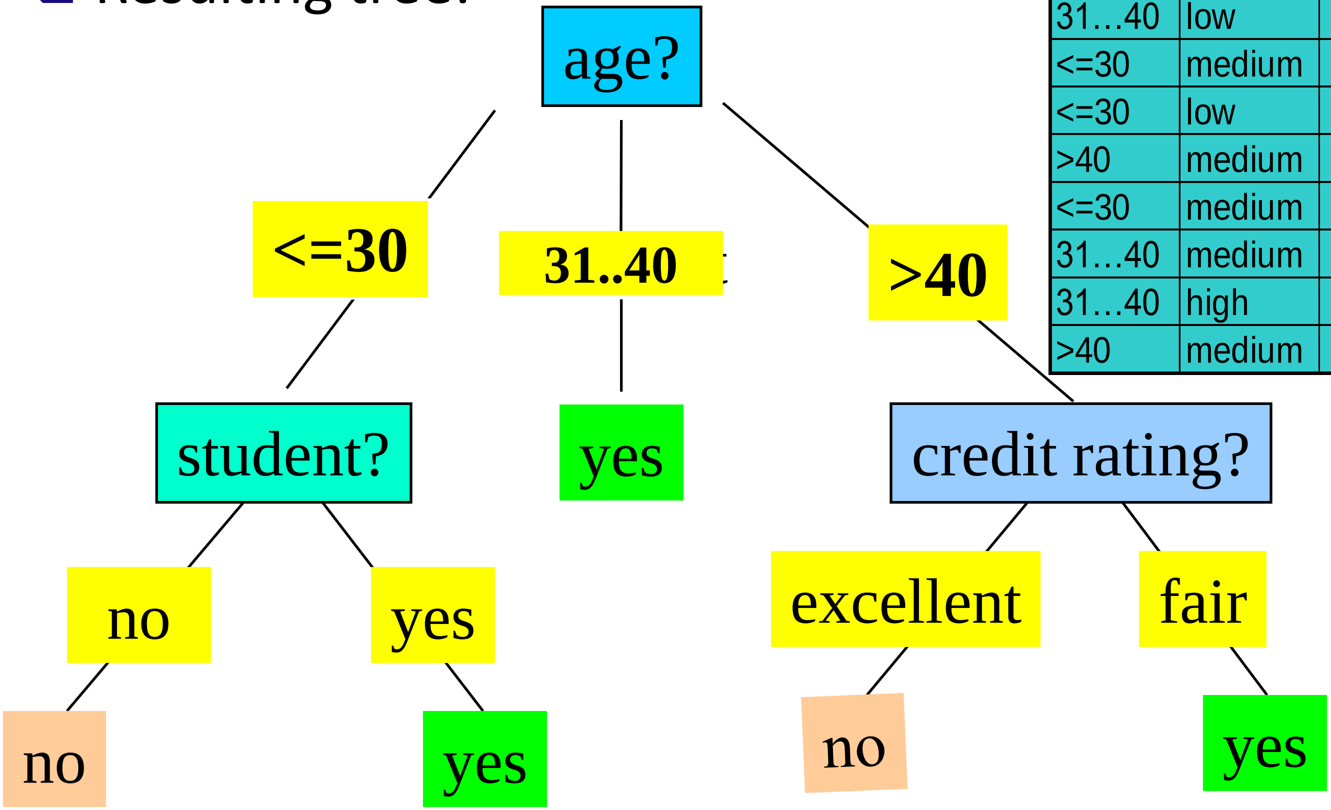


Classification using LR

- We use statistical tests to find the best model i.e. the best set of features and regression coefficients.
- Then we can use the equation to provide an estimate for π. This is the probability of class 1. (1- π) is the probability of class 2.
- We allocate an individual to the class with the higher probability.
- LR may be conducted in a forward or backward manner – to facilitate feature selection.

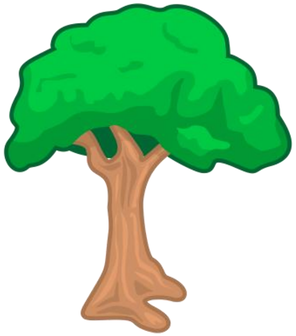
Decision Tree Induction: An Example

- Training data set: Buys_computer
- The data set follows an example of Quinlan's ID3
- Resulting tree:



age	income	student	credit_rating	buys_computer
<=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

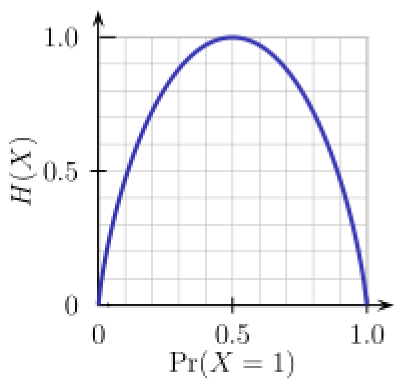
Algorithm for Decision Tree Induction



- Basic algorithm (a greedy algorithm)
  - Tree is constructed in a top-down recursive divide-and-conquer manner
  - Initially, all the training examples are at the root
  - Attributes are categorical (if continuous-valued, they are discretized in advance)
  - Examples are partitioned recursively based on selected attributes
  - Test attributes are selected on the basis of a heuristic or statistical measure (e.g., information gain)
- Conditions for stopping partitioning
  - All samples for a given node belong to the same class
  - There are no remaining attributes for further partitioning – majority voting is employed for classifying the leaf
  - There are no samples left

Definition of Entropy

- Entropy (Information Theory)
  - A measure of uncertainty associated with a random variable
  - Calculation: For a discrete random variable *Y* taking *m* distinct values $\{y_1, \ldots, y_m\}$,
    - $H(Y) = -\sum_{i=1}^{m} p_i \log(p_i)$, where $p_i = P(Y = y_i)$
  - Interpretation:
    - Higher entropy => higher uncertainty
    - Lower entropy => lower uncertainty
- Conditional Entropy
  - $H(Y|X) = \sum_x p(x) H(Y|X = x)$



m = 2

Attribute Selection Measure: Information Gain (ID3/C4.5)

- Select the attribute with the highest information gain
- Let p_i be the probability that an arbitrary example belongs to class C_i
- Expected information (entropy) needed to classify a tuple in D:

$$Info(D) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

- Information needed (at each branch) to classify D:

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j)$$

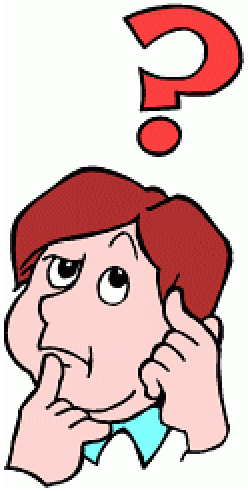
- Information gained by branching on attribute A

$$Gain(A) = Info(D) - Info_A(D)$$

Bayesian Classification

- Bayesian classification performs *probabilistic prediction, i.e.,* predicts class membership probabilities
- Foundation: Based on Bayes' Theorem and classical Probability Theory.
- Performance: A simple Bayesian classifier, *naïve Bayesian classifier*, has comparable performance with decision tree and selected neural network classifiers
- Incremental: Each training example can incrementally increase/decrease the probability that a hypothesis is correct — prior knowledge can be combined with observed data
- Standard: Even when Bayesian methods are computationally intractable, they can provide a standard of optimal decision making against which other methods can be measured

Prediction Based on Bayes' Theorem



- Given training data **X**, *the posteriori probability that a hypothesis* H, P(H|**X**), follows Bayes' theorem is:

$$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})} = P(\mathbf{X}|H) \times P(H) / P(\mathbf{X})$$

- Informally, this can be viewed as

    posteriori = likelihood x prior/evidence

- Predicts **X** belongs to C_i iff the probability $P(C_i|\mathbf{X})$ is the highest among all the $P(C_k|X)$ for all the *k* classes
- Practical difficulty: It requires initial knowledge of many probabilities, involving significant computational cost

Classification based on the Maximum Probability

- Let D be the training data and the associated class labels, where each observation is represented by an n-dimensional vector **X** = $(x_1, x_2, \ldots, x_n)$
- Suppose there are *m* classes $C_1, C_2, \ldots, C_m$.
- Classification is to derive the maximum probability posteriori, i.e., the maximal $P(C_i|\mathbf{X})$
- This can be derived from Bayes' theorem

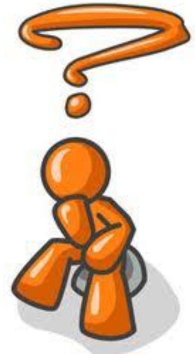
$$P(C_i|\mathbf{X}) = \frac{P(\mathbf{X}|C_i)P(C_i)}{P(\mathbf{X})}$$

- Since P(X) is constant for all classes, only

$$P(C_i|\mathbf{X}) = P(\mathbf{X}|C_i)P(C_i)$$

needs to be maximized

Naïve Bayes Classifier



- A simplified assumption: attributes are conditionally independent (i.e., no dependence relation between attributes):

$$P(\mathbf{X}|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \ldots \times P(x_n|C_i)$$

- This greatly reduces the computation cost: Only counts the class distribution
- If A_k is categorical, $P(x_k|C_i)$ is the # of tuples in C_i having value x_k for A_k divided by $|C_{i, D}|$ (# of tuples of C_i in D)
- If A_k is continuous-valued, $P(x_k|C_i)$ is usually computed based on Gaussian distribution with a mean μ and standard deviation σ

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

and $P(x_k|C_i)$ is

$$P(\mathbf{X}|C_i) = g(x_k, \mu_{C_i}, \sigma_{C_i})$$

Naïve Bayes Classifier: Training Dataset

Class:

C1:buys_computer = 'yes'

C2:buys_computer = 'no'

Data to be classified:

X = (age <=30,

Income = medium,

Student = yes

Credit_rating = Fair)

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

Naïve Bayes Classifier: An Example

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

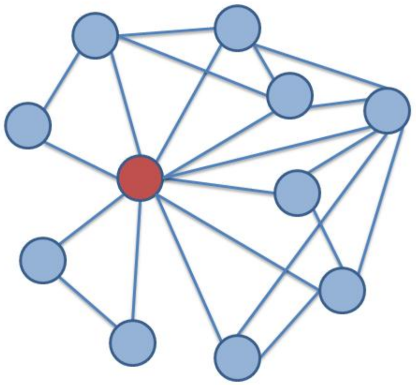
- $P(C_i)$: P(buys_computer = "yes") = 9/14 = 0.643
  P(buys_computer = "no") = 5/14= 0.357
- Compute $P(X|C_i)$ for each class
  P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222
  P(age = "<= 30" | buys_computer = "no") = 3/5 = 0.6
  P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444
  P(income = "medium" | buys_computer = "no") = 2/5 = 0.4
  P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667
  P(student = "yes" | buys_computer = "no") = 1/5 = 0.2
  P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667
  P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4
- **X = (age <= 30 , income = medium, student = yes, credit_rating = fair)**

$P(X|C_i)$: P(X|buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = 0.044
P(X|buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = 0.019

$P(X|C_i)*P(C_i)$: P(X|buys_computer = "yes") * P(buys_computer = "yes") = 0.028
P(X|buys_computer = "no") * P(buys_computer = "no") = 0.007

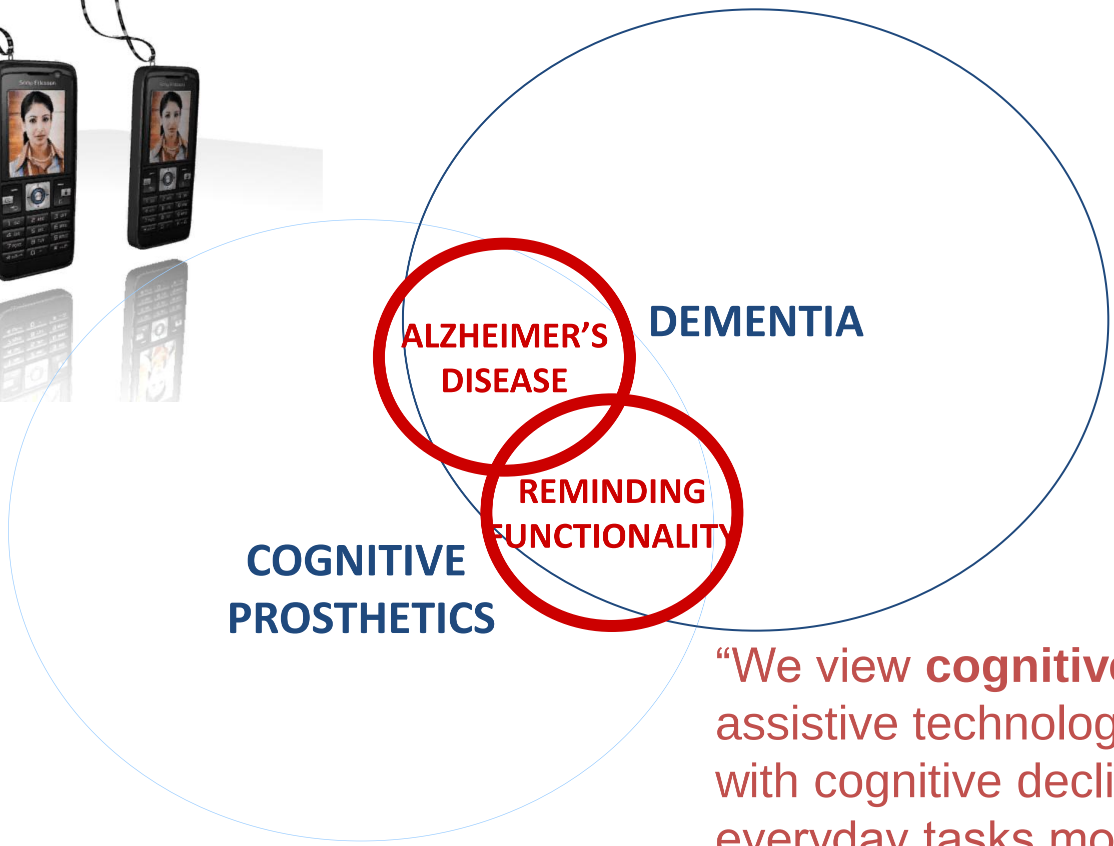
Therefore, X belongs to class ("buys_computer = yes")

Naïve Bayes Classifier: Comments



- Advantages
  - Easy to implement
  - Good results obtained in most cases
- Disadvantages
  - Assumption: class conditional independence, therefore loss of accuracy
  - Practically, dependencies exist among variables
    - e.g., for hospital patients features might be: age, family history, symptoms etc. Disease (class): lung cancer, diabetes, etc.
    - Dependencies among these cannot be modeled by Naïve Bayes Classifiers
- How to deal with these dependencies? Bayesian Belief Networks

A CASE STUDY: Cell-phone Video Streaming (CPVS) in Alzheimer's Disease



"We view **cognitive prosthetics** as assistive technologies that help people with cognitive decline to conduct everyday tasks more independently."

Donnelly, M., Nugent, C.D., McClean, S.I., Scotney, B.W., Mason, S., Passmore, P. & Craig, D. (2010). IEEE MultiMedia, 17 (2), pp. 42-51.

MATCH Plus

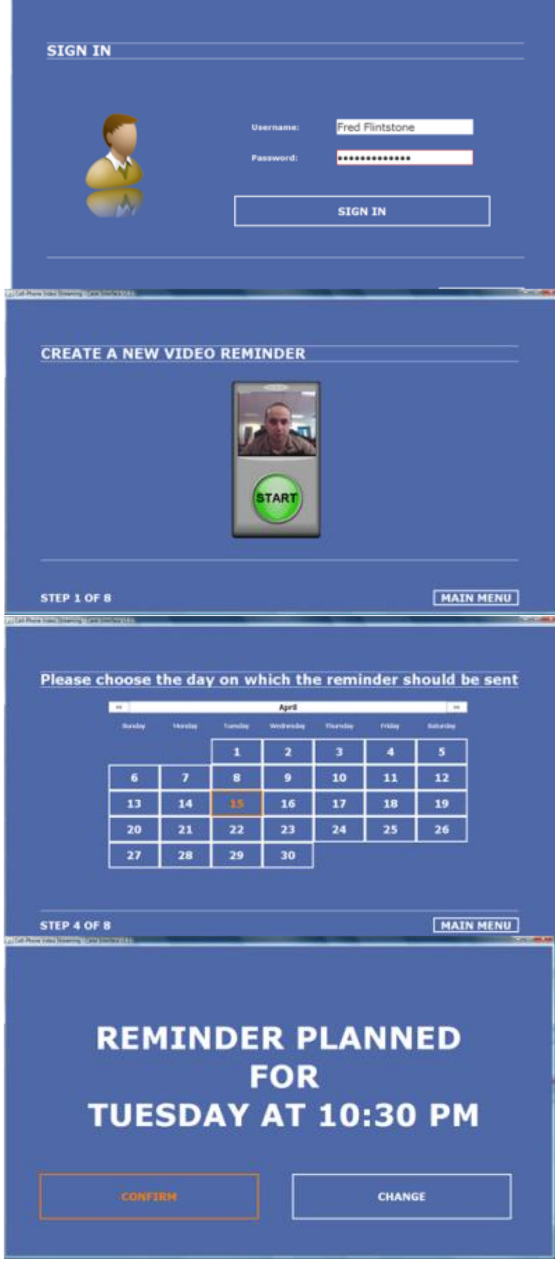
- Within MATCH+ further funding has been obtained to focus on user requirements
- At Ulster, MATCH+ is focussed on collecting and analysing data from users.
- This is in association with a previous project on developing a cell phone video streaming system for Alzheimer’s patients.

Motivation

- Observation in a local memory clinic highlighted the need for proactive reminding solutions

- The aim was to support persons with mild Alzheimer's
  - ✓ potential to improve independence and QoL.
  - ✓ potential to reduce caregiver burden.
  - ✓ potential to delay the need for care home.

- The focus is on using a truly *'everyday'* technology:
  - ✓ Use a **familiar face** to offer the reminders
  - ✓ Provide a **'virtual Caregiver'** throughout the day.

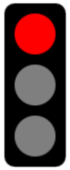


Technology Adoption & Usage Tool (TAUT)

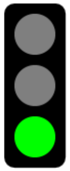
- Are there common factors to predict who will readily adopt the technology and who will drop out?
- What is the influence of the carer on this process?
- What could be done to facilitate adoption for likely drop-outs?
- How generalisable are these finding?

The Patients

- 40 patients
  - MMSE>16, with mean=28 (out of 30)
- from both patients and carers
- Adopter vs. Non-adopter: 70% : 30%



NA={dropped out; non-compliant}



A={compliant, eager to keep}

We need a simple prediction tool: TAUT

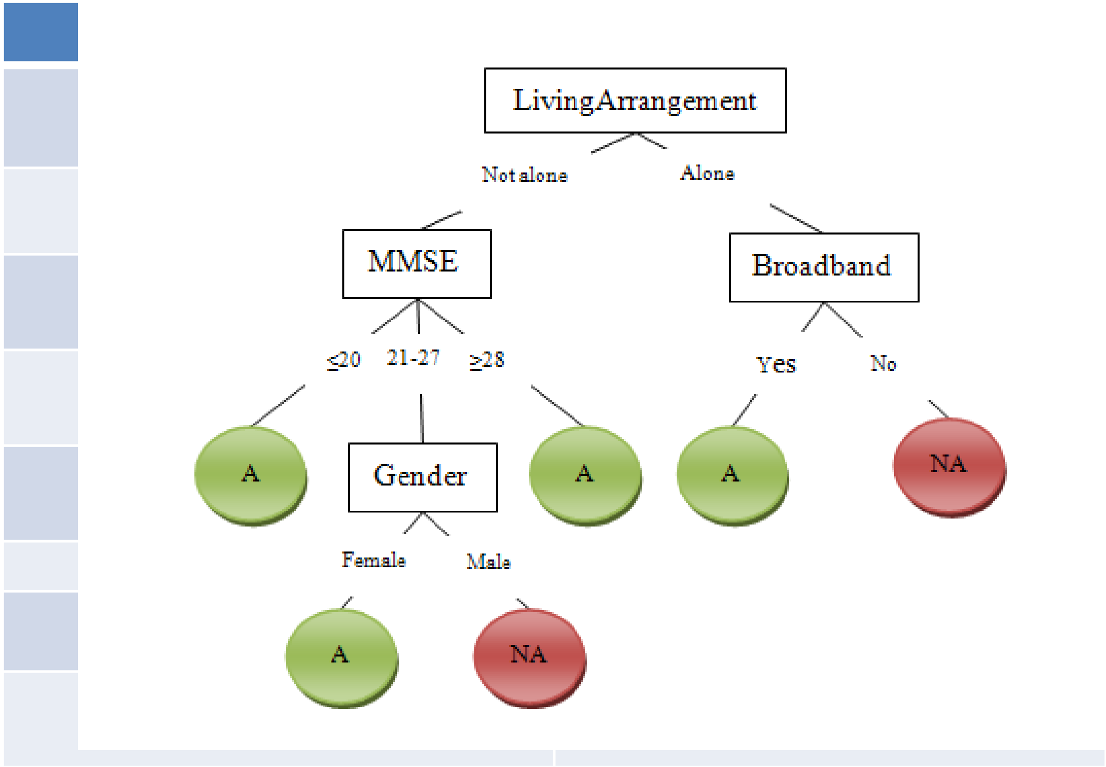
Feature selection

Index	Parameters
1	Age
2	Gender
3	MMSE
4	Previous Profession
5	Technology Experience Patient
6	Broadband
7	Mobile_Reception
8	Carer
9	Living_Arrangement
10	Extra Support
11	Physical Health
If there is a carer	
12	~~Age Carer~~
13	~~Gender of Carer~~
14	~~Previous Profession Carer~~
15	~~Health of Carer~~

- Discarded information about carers
- Pair-wise significance tests on an individual feature vs. output
- Principle component analysis

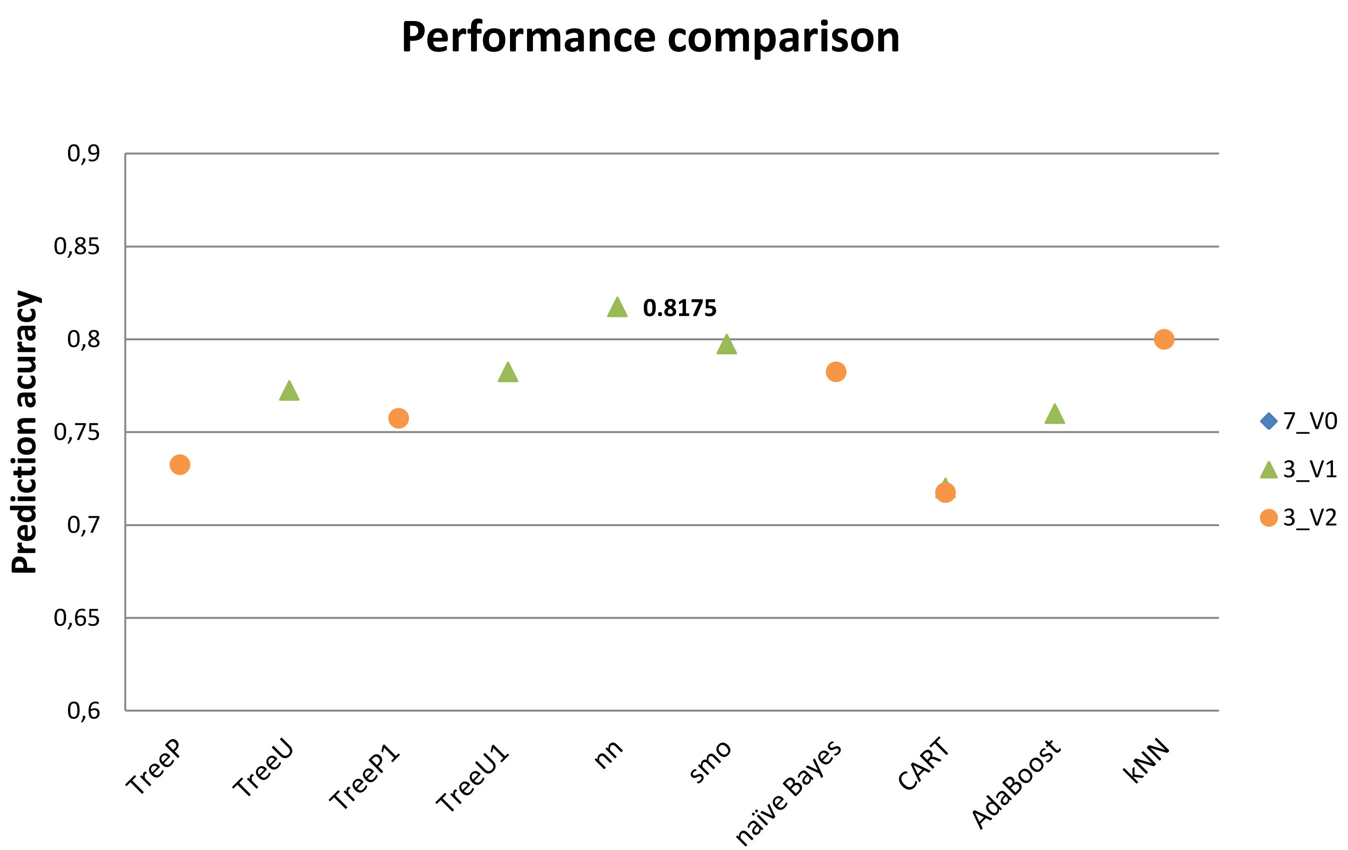
The Decision Tree Prediction model

- C4.5 decision tree
- Discretisation



(Pruned, minNumObj=2)

Best Performing Feature Sets Overall



7 Features set V_0= {Gender, Living, MMSE, Broadband, Age, MobileRec, and Carer}

3 Features set V_1= {Gender, Living, MMSE}

3 Features set V_2= {Gender, Living, Broadband}

Which is the best classifier?

Classifier	Accuracy	Robustness	Bias	Interpretation
kNN_7V_0_S	★★★★★	★★★★★	★★★★	★★
nn_3V_1	★★★★	★★★★★	★★★★★	★★★
smo_3V_1	★★★★	★★★	★★	★★★
TreeU1_3V_1	★★	★★★★	★★★★★	★★★★★
Bayes_3V_2	★★★	★★★★★	★★	★★★★★
TreeU_3V_1	★★★	★★★★	★★★★★	★★★★★
AdaBoost_3V_1	★★	★★	★	★★
TreeP1_3V_2	★★	★★★★	★★	★★★★★
CART_3V_1_S	★★★	★★	★★★★★	★★★★★
TreeP_3V_2	★★	★★	★★★	★★★★★

TAUT Project Team

Technology development of cognitive prosthetics

Ulster University

Sally McClean

Bryan Scotney

Shuai Zhang

Mark Donnelly

Chris Nugent

Ian Cleland

Phillip Hartin

UtahState University

Memory in Ageing. Epidemiological and statistical analyses

JoAnn Tschanz

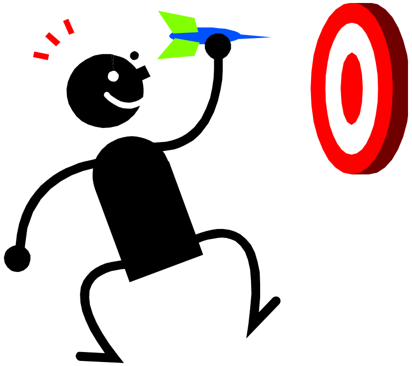
Maria Norton

Ken Smith

THE UNIVERSITY OF UTAH

Epidemiological and statistical analyses

Aims of the ETAC Project

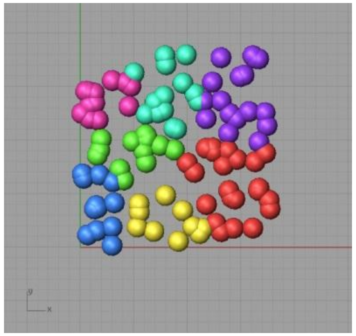


"To identify the factors and parameters that influence technology adoption and as a result develop a predictive model which can be used to assess dyads and predict whether they are likely to adopt assistive technology."

Research Questions

1. Are there common factors to predict who will readily adopt the technology and who will drop out?
2. What is the influence of the carer on this process?
3. What can be undertaken to facilitate adoption for non-adopters?

Clustering Algorithms

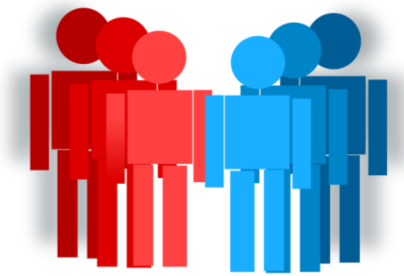


- A clustering algorithm attempts to ***discover*** clusters in the training data.
- Here training data consists of records but ***without*** the class labels.
- The algorithm tries to group objects so that
  - those belonging to a cluster are *more like each other* than they are like objects in *another* cluster.
- This requires the algorithm to be able to assess how *close* or *similar* objects are in their descriptions.
  - For this it needs a *measure* of similarity.

Clusters versus Classes

- In Machine Learning Concept learning assumes a *pre-determined* class attribute.
  - Training examples aid the learning algorithm by indicating each example's class.
- In many situations we do not have or know the classes to use.
- Instead we wish to group objects that appear ***similar*** in some way.
- Such a grouping is called a ***cluster*** or ***segment***.
  - The term segment is used (particularly in marketing) because a population of objects is often divided into non-overlapping regions.

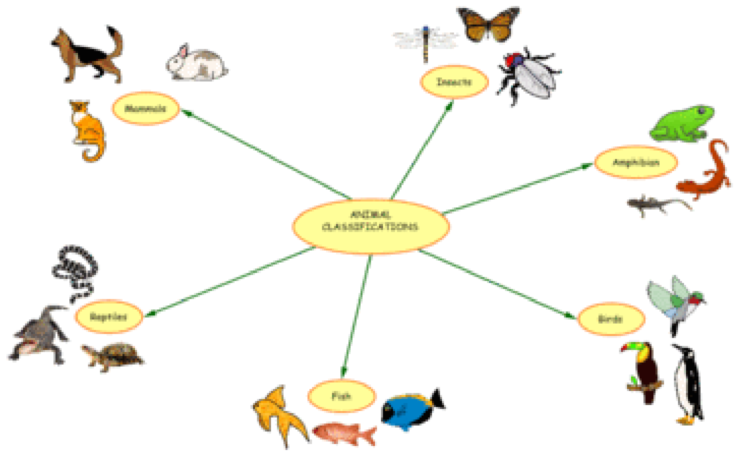
Measuring Similarity



- Similarity is a context-dependent notion.
  - Objects may be similar in some ways but not in others.
    - eg, when buying from a web-site, customers may exhibit similar purchasing behaviour but be very different as people – in terms of *age*, and *where they live*.
- Thus the context for similarity must be decided and which attributes will be represented in that context.
- Typically the attributes used to assess similarity include some discrete and some continuous, as for classification algorithms.
- Sometime we use distance instead of similarity; a small distance means low similarity.

Issues in Assessing Similarity

- Are all the attributes equally important in determining similarity?
  - Perhaps some attributes should be weighted more highly than others.
- If similarity is a measure of closeness of value, how do we assess the closeness for discrete attributes?
  - eg, for attribute *animal* with values such as: number of legs, presence of fur, ability to fly, which values are closest to each other?



How Clustering Algorithms work

- Can be divided into those producing
  - a single ***flat*** collection of clusters
  - a **hierarchy** of clusters
- Some clustering algorithms work with a *logical,* others with a *probabilistic,* definition of similarity.
- Clusters can be formed through different approaches:
  - *divide and conquer*
    - clusters are refined until a termination condition is reached.
  - *separate and conquer*
    - examples are processed one at a time
    - each example is added to an existing cluster or a new cluster is created to contain it
      - Clusters can also be merged
    - the outcome can depend on the order in which examples are presented.

Similarity Measure for Nominal Attributes

- If object attributes are all nominal (categorical), then proximity measure are used to compare objects
- Can take 2 or more states, e.g., red, yellow, blue, green (generalization of a binary attribute)
- Method 1: Simple matching
  - *m*: # of matches, between *i* and *j*; *p*: total # of variables

$$\boxed{d(i,j) = \frac{p-m}{p}}$$

- Method 2: Convert to Standard Spreadsheet format
  - For each attribute *A* create *M* binary attribute for the *M* nominal states of *A*
  - Then use standard vector-based similarity or distance metrics

Normalizing or Standardizing Numeric Data

- Z-score:
  - *x*: raw value to be standardized, *μ*: mean of the population, *σ*: standard deviation
  - the distance between the raw score and the population mean in units of the standard deviation
  - negative when the value is below the

$$z = \frac{x - \mu}{\sigma}$$

ID	Gender	Age	Salary
1	F	27	19,000
2	M	51	64,000
3	M	52	100,000
4	F	33	55,000
5	M	45	45,000

ID	Gender	Age	Salary
1	1	0.00	0.00
2	0	0.96	0.56
3	0	1.00	1.00
4	1	0.24	0.44
5	0	0.72	0.32

Common Distance Measures for Numerical Data

- Consider two vectors
  - Rows in the data matrix $\boxed{X = \langle x_1, x_2, \cdots, x_n \rangle}$ $\boxed{Y = \langle y_1, y_2, \cdots, y_n \rangle}$
- Common Distance Measures:
  - Manhattan distance:

$$\boxed{dist(X,Y) = |x_1 - y_1| + |x_2 - y_2| + \cdots + |x_n - y_n|}$$

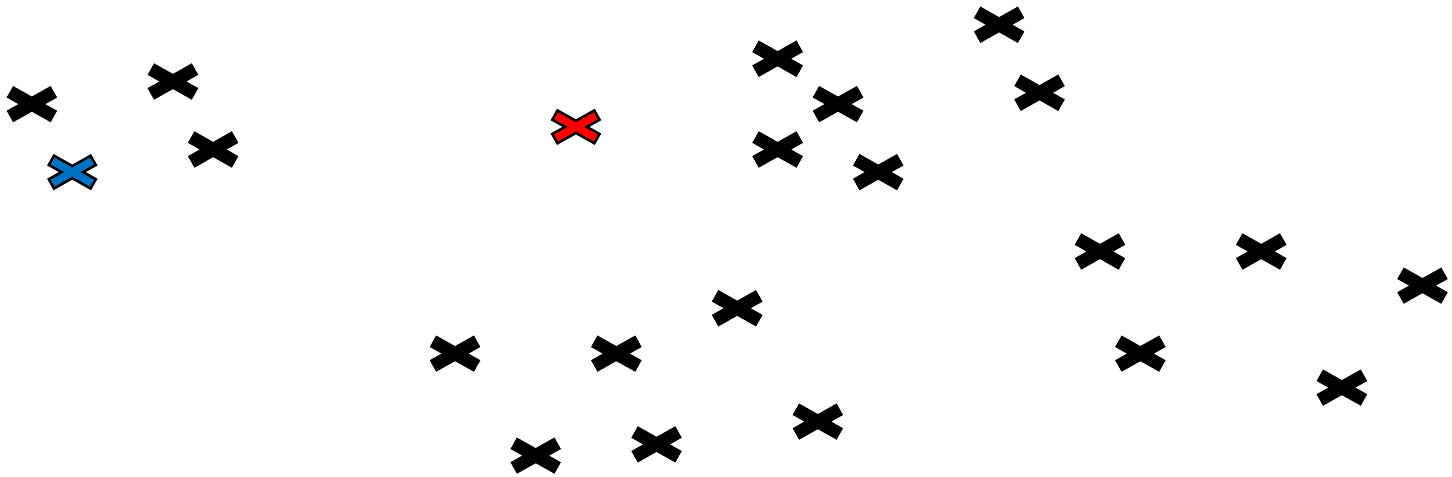
  - Euclidean distance:

$$\boxed{dist(X,Y) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_n - y_n)^2}}$$

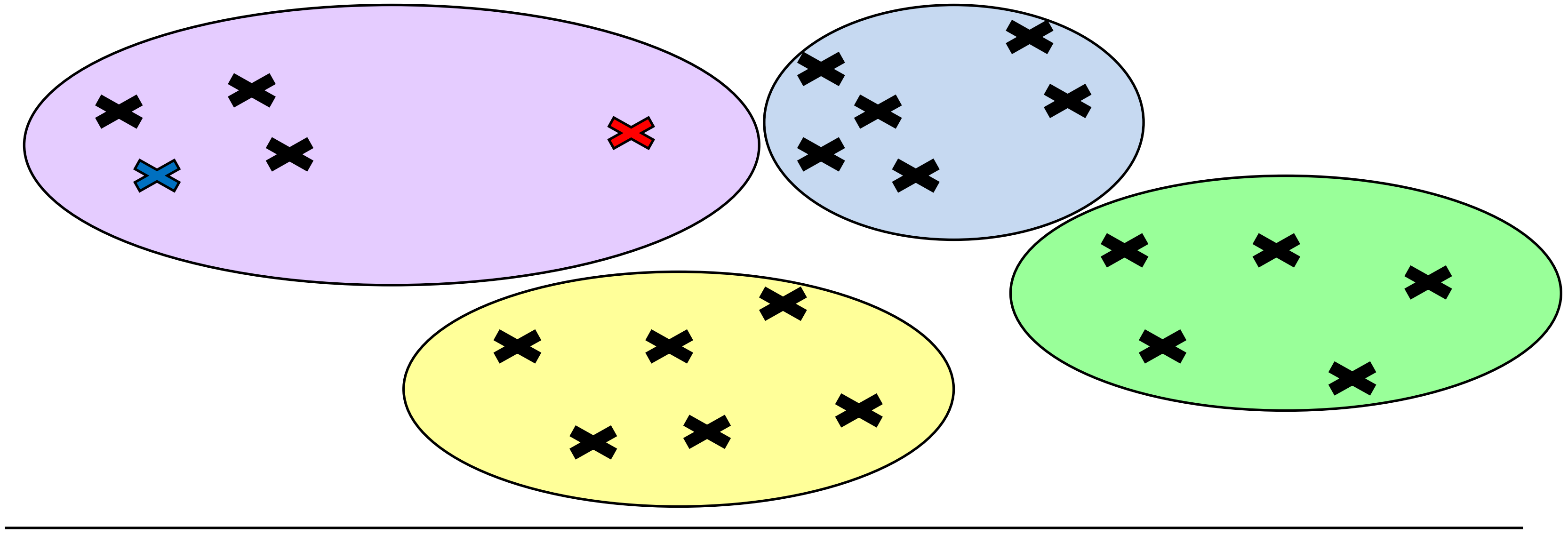
  - Distance can be defined as a dual of a similarity measure

$$\boxed{dist(X,Y) = 1 - sim(X,Y)}$$

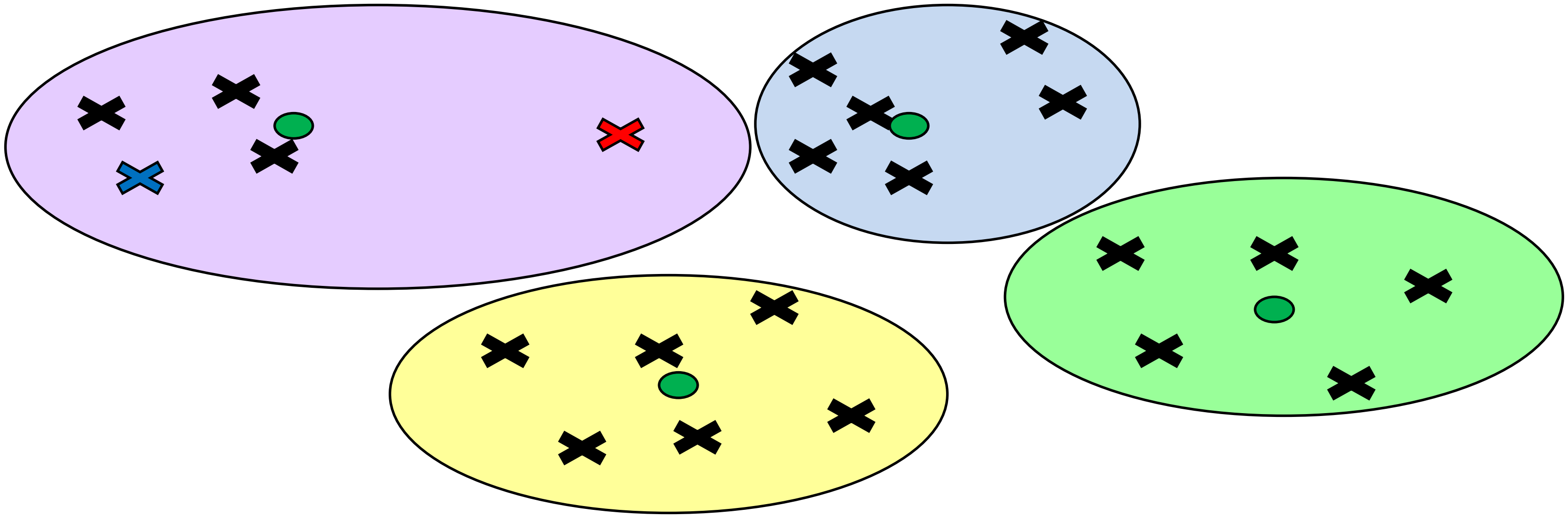
Divide and Conquer (i)



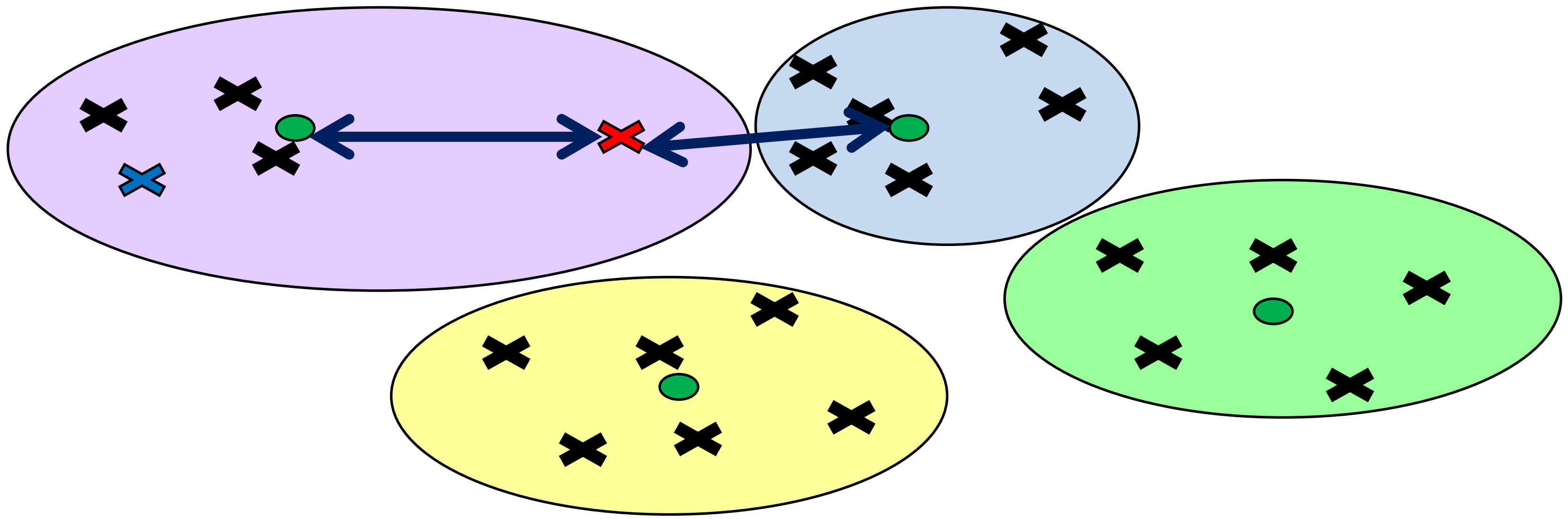
Divide and Conquer (ii)



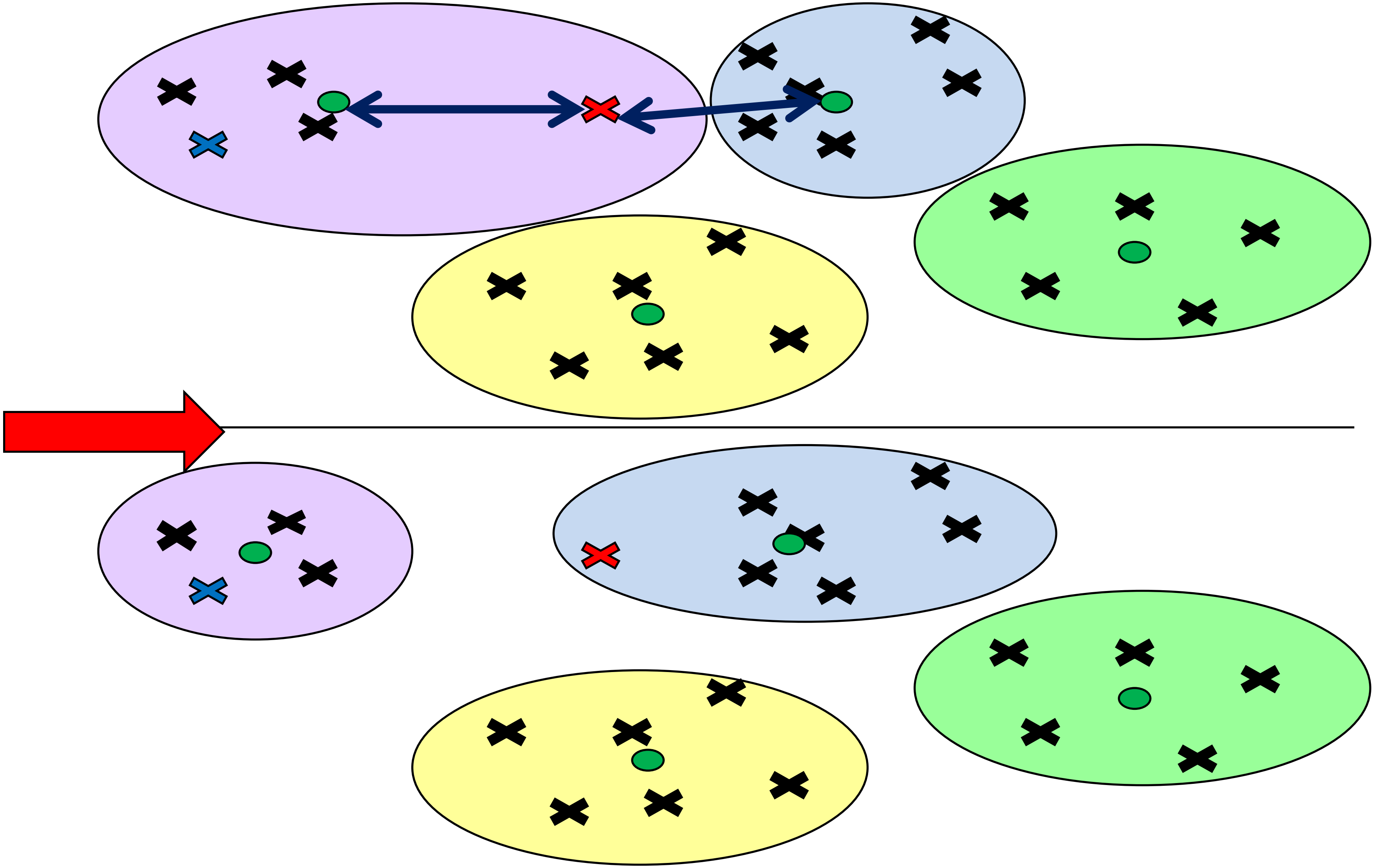
Divide and Conquer (iii)



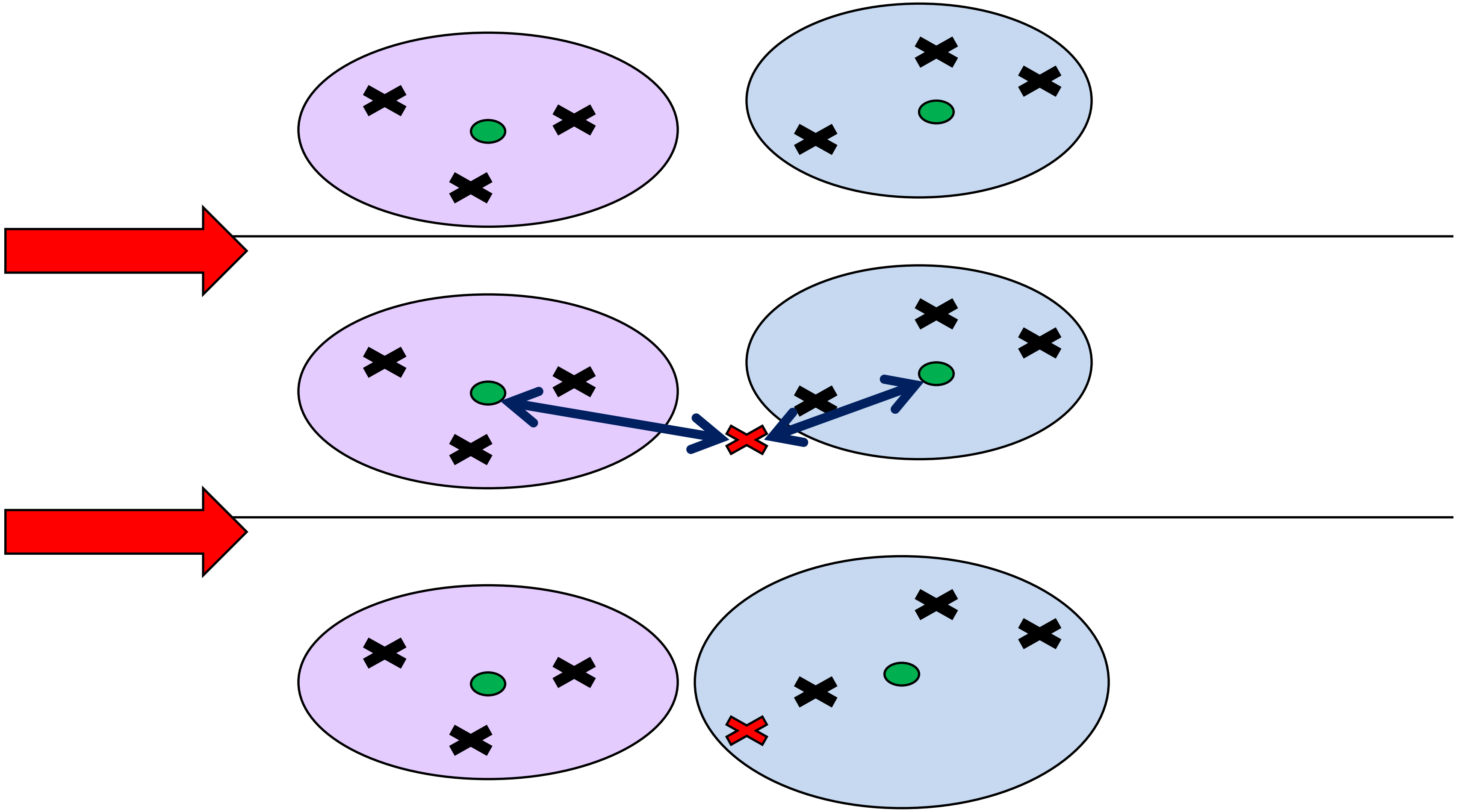
Divide and Conquer (iv)



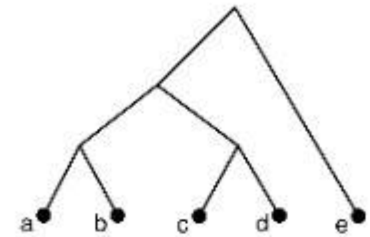
Divide and Conquer (v)



Separate and Conquer (i)

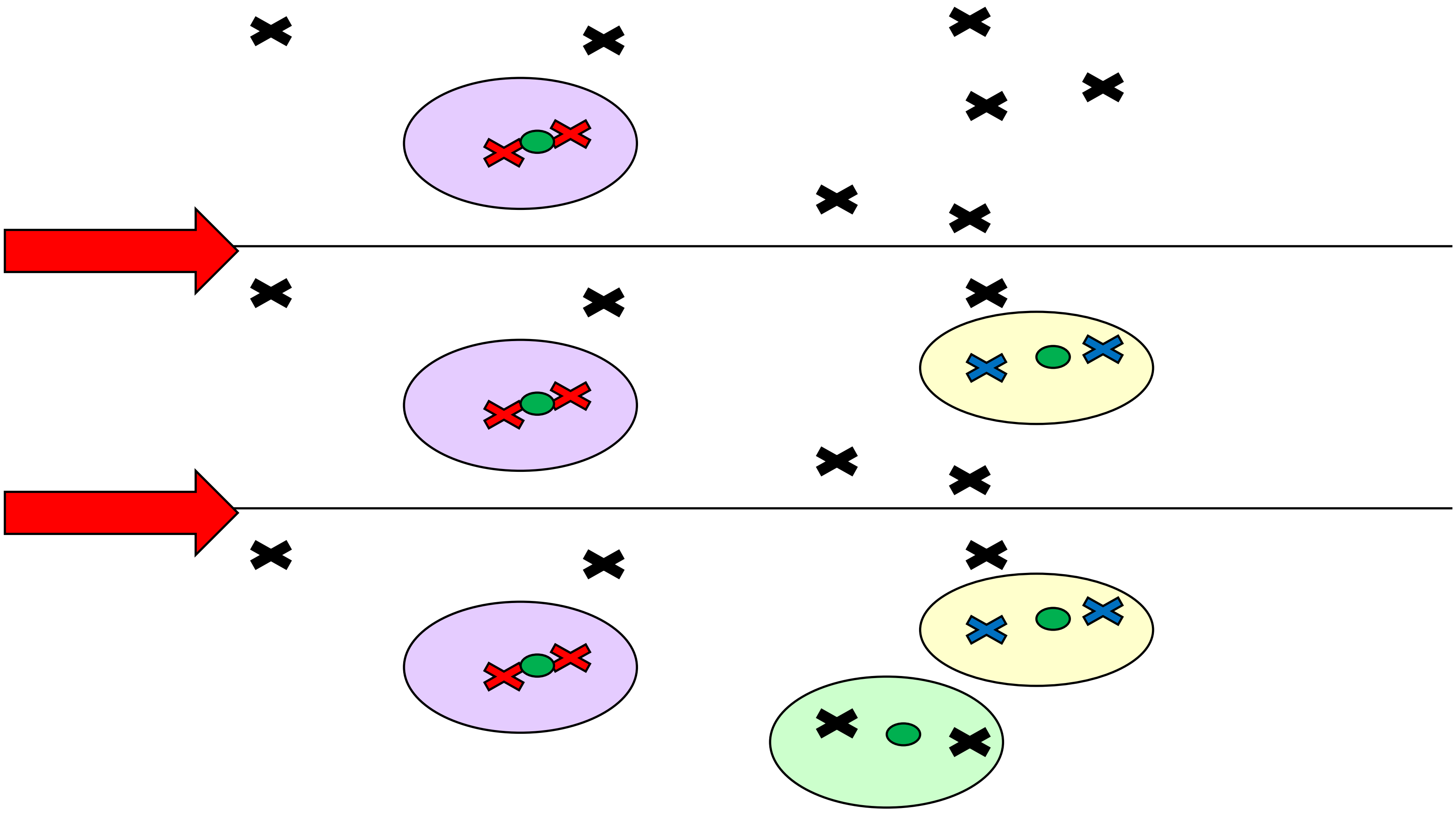


Hierarchical Clustering

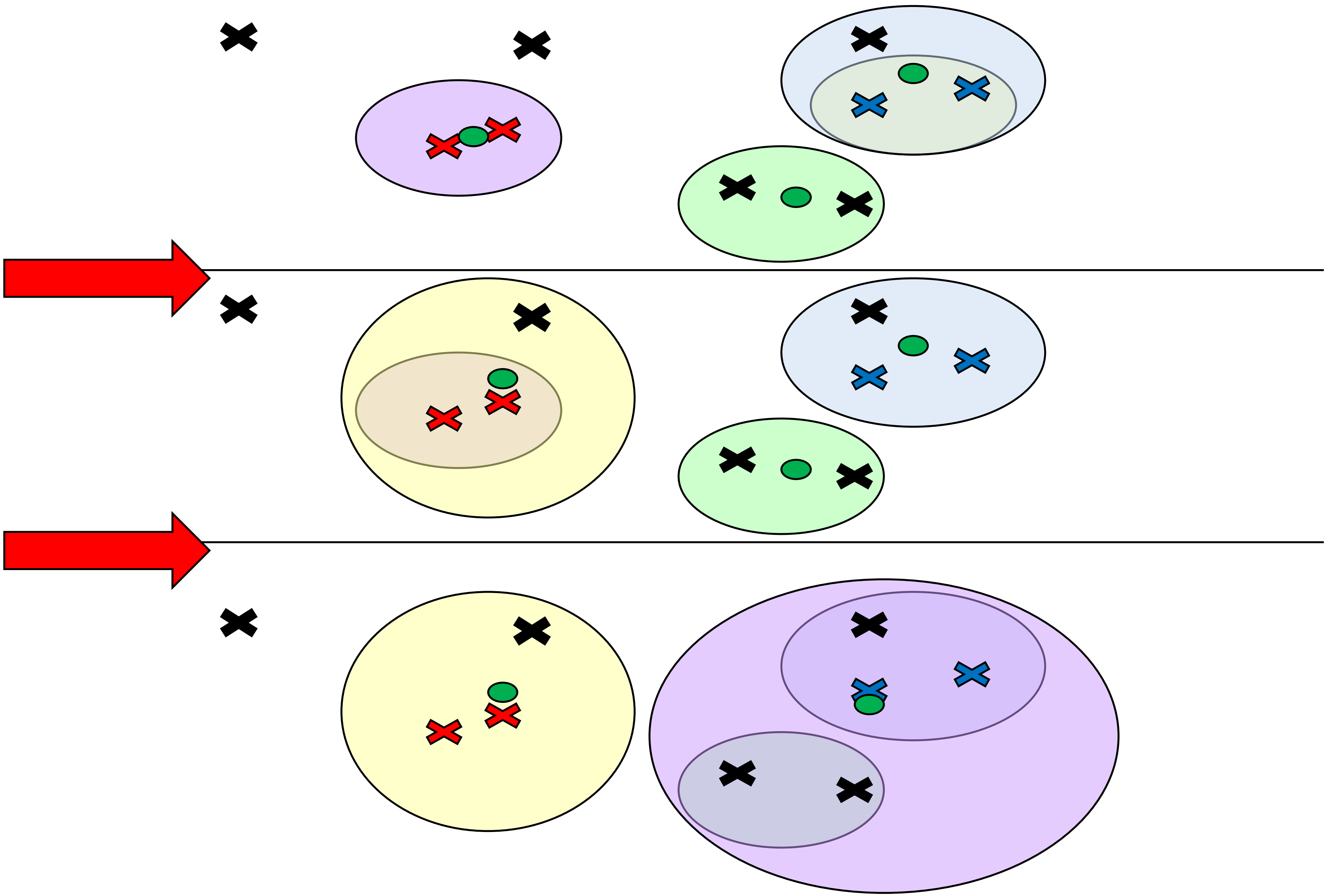


- A *taxonomy* with different levels of clusters is created
  - the user can choose the level from which to extract clusters
  - can be illustrated as a ***dendogram*** *(tree)*.
- *Agglomerative clustering* builds a binary decision tree from leaves upwards to the root
  - at each stage the two closest examples are combined into a cluster.

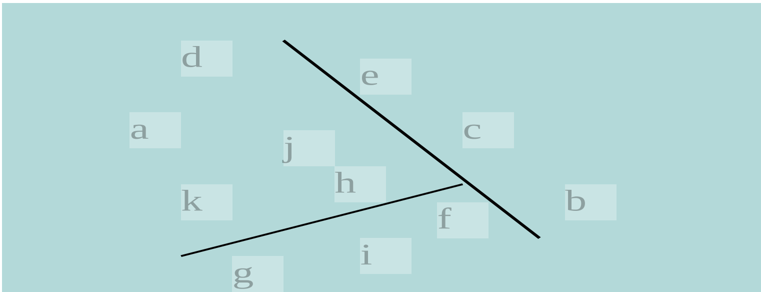
Agglomerative Clustering



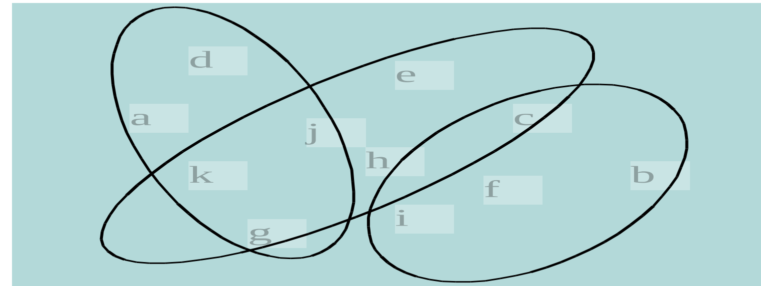
Agglomerative Clustering (ctd)



Representing Clusters



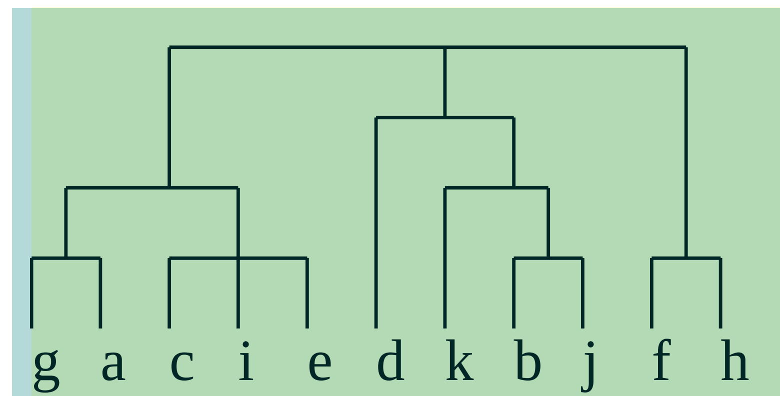
Non-overlapping sets



Overlapping sets

	1	2	3
a	0.4	0.1	0.5
b	0.1	0.8	0.1
c	0.3	0.3	0.4
d	0.1	0.1	0.8
e	0.4	0.2	0.4
f	0.1	0.4	0.5
g	0.7	0.2	0.1
h	0.5	0.4	0.1

Probabilities of membership

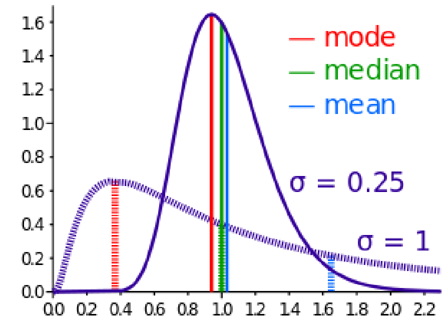


Hierarchical clusters - dendogram

The Number of Clusters

- For some methods the number of clusters, *k*, to be discovered must be set in advance by the user.
  - This is not easy to stipulate and often not appropriate for a business application.
  - We can try an initial value of *k* and inspect the clusters that are obtained
    - then repeat, if necessary, with a different value of *k*.
- The *larger* the number of clusters, the *more similar* the objects within clusters will tend to be.
  - Clusters may become too specialised.
- For other methods, where the number of clusters to be discovered is not set in advance by the user, a suitable number of clusters will emerge from the discovery process.

The k-means algorithm



- A commonly used clustering method
  - best suited to numeric values.
- Assumes that the number of clusters, *k*, to be discovered is specified in advance.
- Represents a cluster by its *centroid* (ie, the average value of each field).

- **The algorithm**:
  1. select *k* examples at random as cluster seeds;
  2. assign each example to the closest cluster;
  3. determine the centroid of each cluster;
  4. re-assign each example to the cluster with the closest centroid;
  5. if cluster membership has changed, go to 3
     else stop.

Classes from Clusters

- After formation, clusters can be *named*
  - eg, *traditional purchaser*.
- A new example can be assigned to a cluster
  - eg, for *k*-means clustering, determine the cluster whose centroid is closest to the example.
- Here clusters are being used as *classes*.
  - Can learn classification rules to *describe* clusters in terms of *other* attributes that were *not* used in the clustering
    - eg, shoppers clustered on *purchase behaviour* attributes, could be described by rules that use *personal details* attributes:
      - (if *age* > 50) and (*gender* = male) and (*loyalty card* = no) and (*payment method* = cash) then
      (*customer type* = traditional purchaser)

Ulster University

Using Fitt’s Law to Model Arm Motion Tracked in 3D by a Leap Motion Controller for Virtual Reality Upper Arm Stroke Rehabilitation

Dominic Holmes, Darryl Charles, Philip Morrow, Sally McClean and Suzanne McDonough

BACKGROUND

- Rehabilitation is capable of improving arm function for stroke recovery and can help survivors with activities of daily living. Therapy must be intense with repetition of relevant functional tasks.
- Virtual Reality (VR) can provide patients with a fun and flexible interactive environment, helping to guide high quality physiotherapy. VR and games have been shown to be beneficial in improving upper limb function and active daily.

INTRODUCTION

- We HAVE developed a VR system called TAGER as a 3D reaching and pointing exercise system for upper limb rehabilitation. A core component of TAGER is the Leap Motion which tracks hands. TAGER also enables the wearing of an Oculus Rift (VR headset).

- We present results of an experiment with healthy users to evaluate the effectiveness of Fitts's law and three other popular variants for modelling movement performance in 3D virtual environments using a Leap Motion to track hand motion.

THE MODELS

- Fitts' law provides an established model to profile user's motor skills
- The time (MT) to reach and touch a target, to a target's size (W) and distance (D) from an origin (1)
- The logarithmic element, is known as the "Index of Difficulty" (ID).

$$MT = a + blog_2\left(2\frac{D}{W}\right) \tag{1}$$

- Two adaptations of Fitts' Law are Shannon/McKenzie (2) and Welford (3)
- Are originally tailored to quantify human movement behaviour for 1D and 2D tasks.
- Murata/Iwase recently adapted Shannon/McKenzie (2) for 3D by accounting for a user's angle of motion (θ) from an origin to a target (4).

$$MT = a + blog_2\left(\frac{D}{W} + 1\right) \tag{2}$$

$$MT = a + blog_2\left(\frac{D}{W} + 0.5\right) \tag{3}$$

$$MT = a + blog2\left(\frac{D}{W} + 1\right) + \sin\theta \tag{4}$$

TAGER

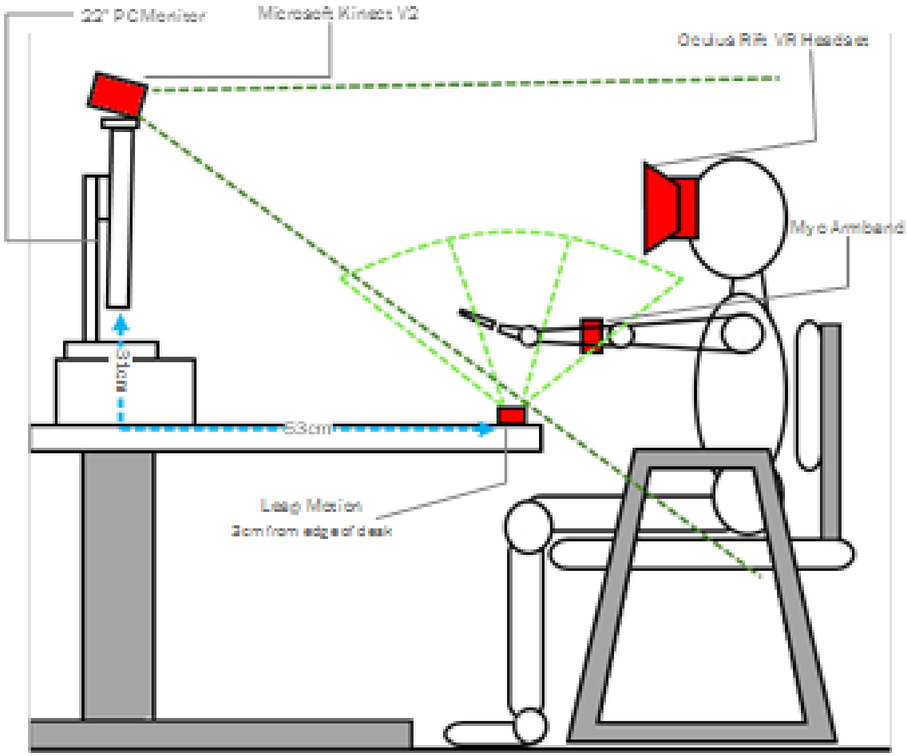
Technology:

- Leap Motion
- Microsoft Kinect V2
- Myo Armband
- Oculus Rift (VR headset)

Application Environment

- User sees a basic walled room.
- 4 repetitions per level containing 27 icosahedrons to target, at different fixed locations.
- When started a single icosahedron appears randomly at any of the 27 locations.
- User moves hand around the environment selecting each target and destroying it.
- Another target appears on the floor (origin), for consistent movement trajectories.
- After each repetition and level the user is given a fixed rest period.
- Icosahedron were chosen for their geometric properties and visual clarity.

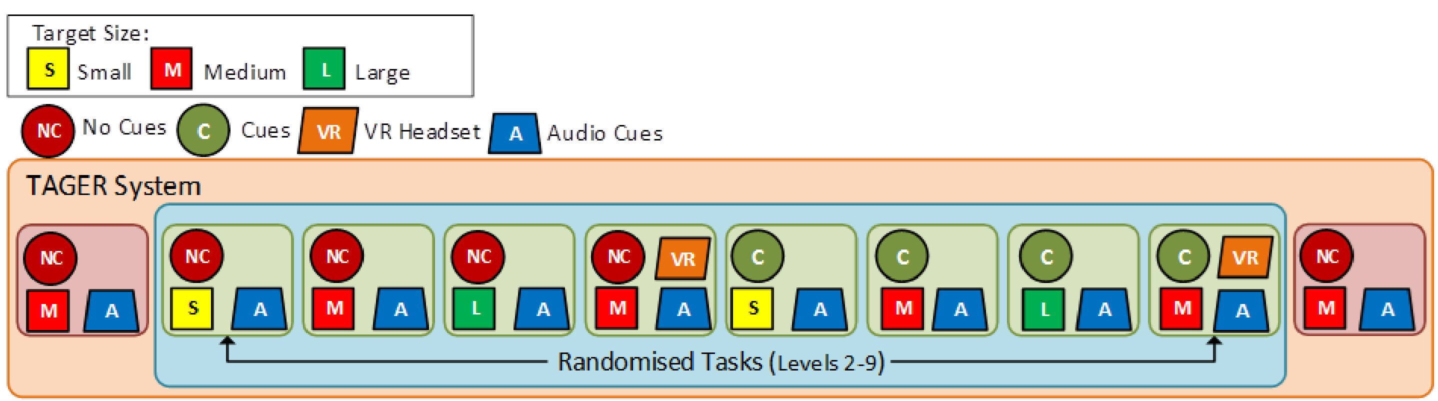
EXPERIMENTAL DESIGN



Three stages:

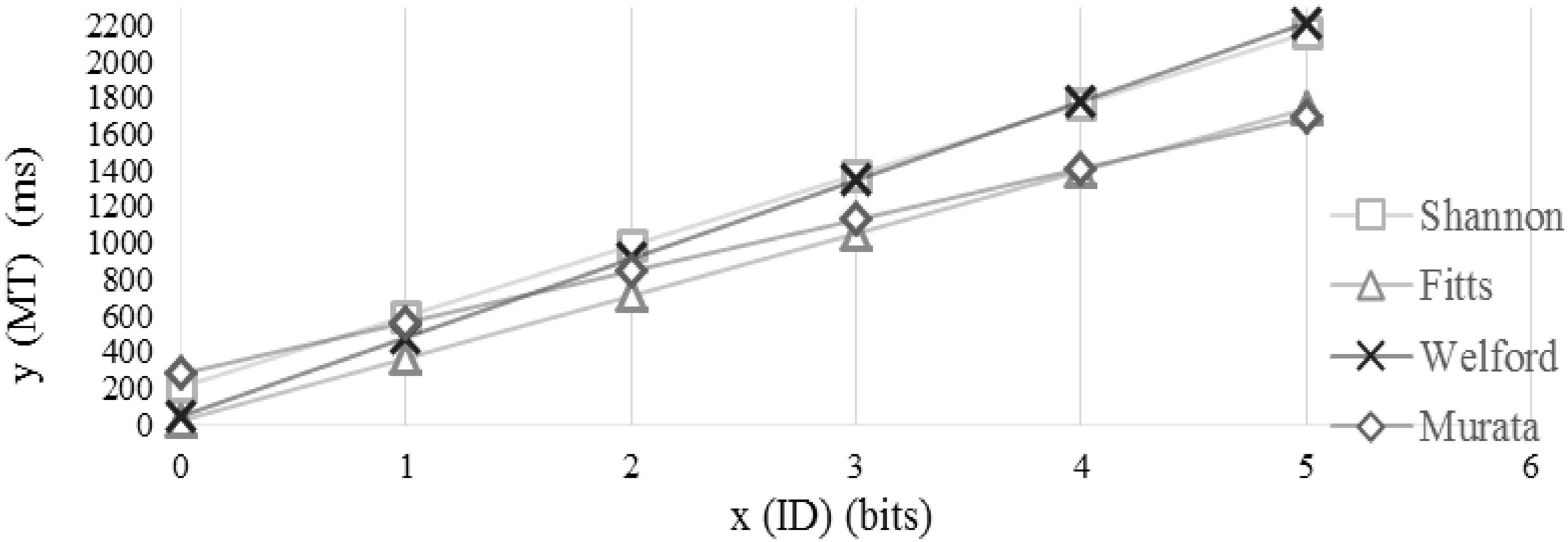
1. A training stage which gives the participant ten minutes practice interaction.
2. Stage 2, consists of the complete TAGER trial.
3. Stage 3, after Completion, a short discussion takes place any gathering comments made and for investigators to ask questions related to the users experience of the system.

Level Layout:



RESULTS

- Evaluation of Fitts' Law Models



Murata's 3D equation (4) :

- Intercept =196 to 363ms
- Gives more gradual slope

While all models provide similar results we focus on Murata for further analysis

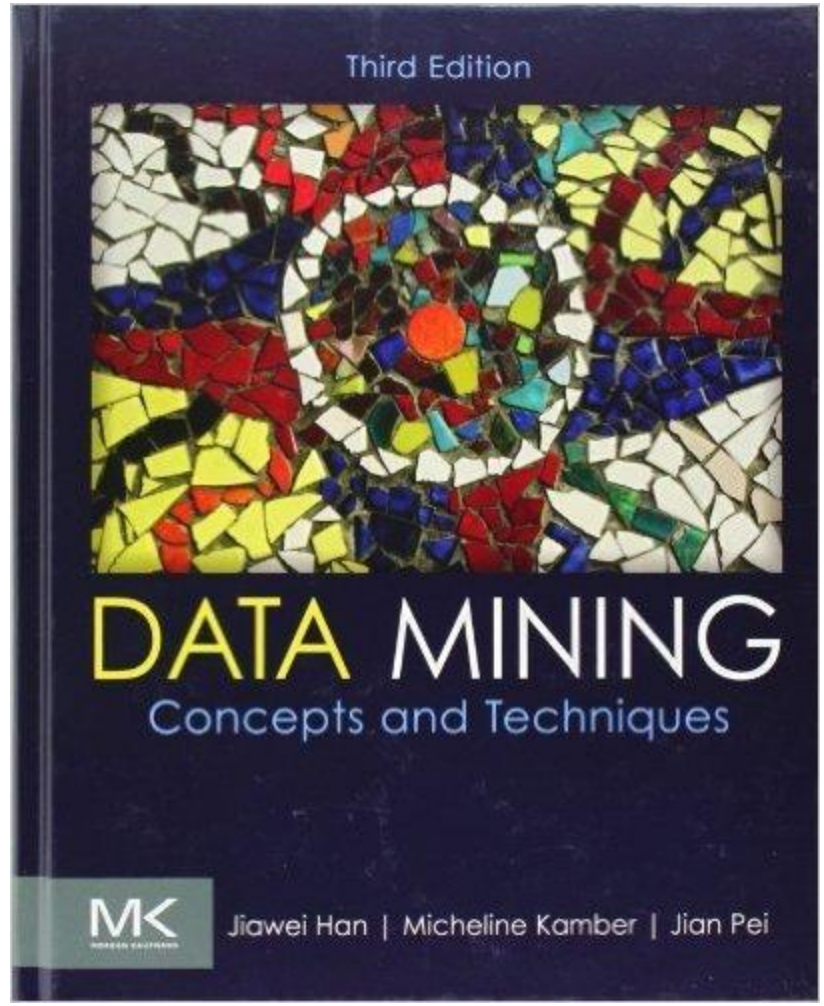
User Performance

User	1001	1002	1009	1013	1019	1023	1026
Gender	20:F	20:F	42:M	24:M	66:F	54:F	24:F
Start (Residual)							
Standard Deviation	0.354	0.461	0.257	0.368	0.313	0.408	0.378
Kurtosis	0.085	-1.141	0.031	-0.238	-0.331	-0.262	0.376
Skewness	0.986	0.394	0.996	0.286	0.394	0.785	0.980
Range	1.519	1.522	1.036	1.558	1.431	1.594	1.528
End (Residual)							
Standard Deviation	0.111	0.275	0.194	0.242	0.420	0.285	0.235
Kurtosis	5.852	0.985	2.540	-0.627	1.927	1.197	0.498
Skewness	1.834	1.227	1.515	0.187	0.952	1.143	1.289
Range	0.657	1.259	1.016	0.983	2.227	1.340	0.920
Start Regression							
R^2	0.115	0.076	0.228	0.453	0.343	0.112	0.189
P-Value	4.05E-03	9.28E-02	8.60E-07	1.78E-08	1.79E-08	1.17E-03	3.92E-05
Slope	0.268	0.312	0.273	0.602	0.517	0.348	0.388
Intercept	0.352	0.360	0.268	-0.289	-0.212	0.475	-0.121
End Regression							
R^2	0.266	0.024	0.348	0.446	0.224	0.392	0.004
P-Value	3.80E-05	2.54E-01	2.83E-09	7.42E-09	3.18E-06	2.44E-11	5.77E-01
Slope	0.133	-0.109	0.293	0.445	0.481	0.509	0.033
Intercept	0.359	1.220	0.092	-0.070	0.116	-0.216	0.703
Performance							
Targets Hit(1080)	794	554	848	638	828	900	788
Start Hits	70	38	96	55	78	91	83
End Hits	73	57	85	59	88	92	74
% Change Hits	4.29	50.00	-11.46	7.27	12.82	1.10	-10.84
Start Mean Time	1.167	1.292	1.063	1.522	1.391	1.497	1.055
End Mean Time	0.768	0.884	0.952	1.270	1.580	1.262	0.802
% Change Mean Time	-34.19	-31.59	-10.42	-16.54	13.64	-15.66	-23.96

ARE WE CLASSIFYING OR CLUSTERING?

- A more complex profiling system helps to personalise training requirements and distinguish between loss of interest and mental or physical fatigue.
- We can also profile the type of game-player to improve engagement.
- We will develop the profiling system with further experiments and subsequently investigate the system with impaired users.

- **Slides in PowerPoint**
- Chapter 1. Introduction
- Chapter 2. Know Your Data
- Chapter 3. Data Preprocessing
- Chapter 4. Data Warehousing and On-Line Analytical Processing
- Chapter 5. Data Cube Technology
- Chapter 6. Mining Frequent Patterns, Associations and Correlations: Basic Concepts and Methods
- Chapter 7. Advanced Frequent Pattern Mining
- Chapter 8. Classification: Basic Concepts
- Chapter 9. Classification: Advanced Methods
- Chapter 10. Cluster Analysis: Basic Concepts and Methods
- Chapter 11. Cluster Analysis: Advanced Methods
- Chapter 12. Outlier Detection
- Chapter 13. Trends and Research Frontiers in Data Mining



Jiawei Han, Micheline Kamber and Jian Pei (2011) Data Mining: Concepts and Techniques, 3rd ed. Morgan Kaufmann.

http://hanj.cs.illinois.edu/bk3/bk3_slidesindex.htm