



»» TRANSFORMING PROMISING IDEAS INTO COMMERCIAL REALITY

Exploring Safety and Diagnostic Data using Self-Organising Maps

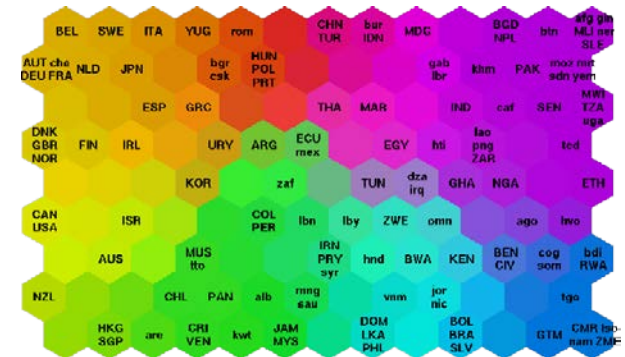
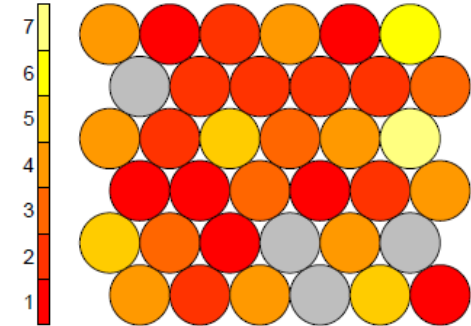
Presented by Colin de Klerk at the PhUSE Annual Conference, London 2014





Self-Organising Maps - Agenda

- Uses of Self-organising Maps
- Artificial Neural Networks (ANNs)
- Self-organising Maps
- Parameters
 - › Competitive Learning Algorithm
- Example: Numeric / RGB Data
- Distance Metric
 - › Dealing with Categorical Values
- Example: Numerical and Categorical Data
 - › Training
 - › Classifying
- Uses of Self-organising Maps - Review

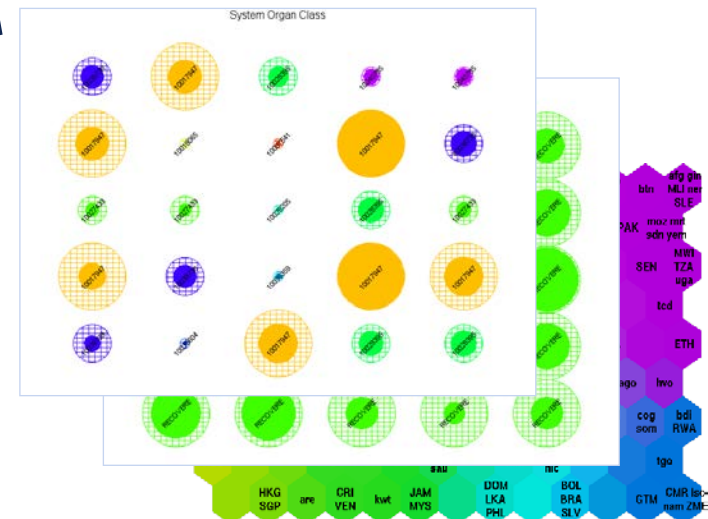
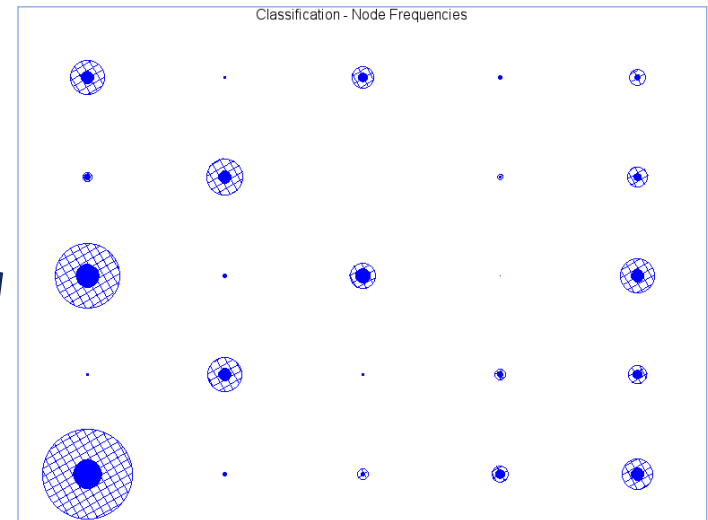




Uses of Self-Organising Maps

- Classification
 - › Group similar items using a distance metric
 - Whole-vector queries
-
- Visualisation
 - › Map multidimensional data onto a 2D grid
 - › Reveal hidden relationships
 - › Identify areas of interest for closer study

The table contains a large number of rows and columns, likely representing a dataset of features used for classification or visualization. The columns are labeled with various identifiers and numerical values.



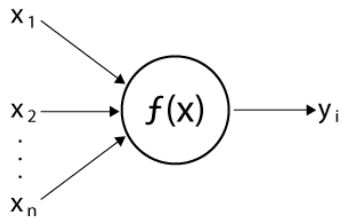
atp gln
 bin
 MLI ser
 SLE
 PAK
 moz ram
 son yem
 MRI
 TZA
 uga
 ted
 ETH
 ago
 lvo
 cog
 pom
 bdi
 RWA
 lgo
 GTM
 CDR
 lso-
 nem ZMC

HKG
 SGP
 are
 CRI
 VEN
 lout
 JAM
 MYE
 DOM
 LKA
 PHI
 BOL
 BRA
 SLV
 GTM
 CDR
 lso-
 nem ZMC

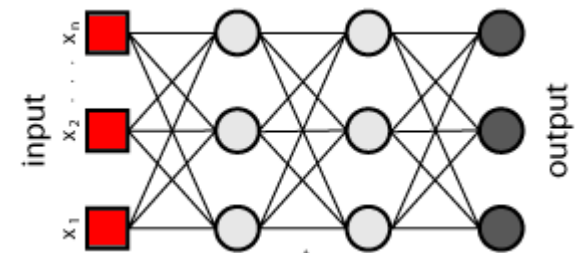
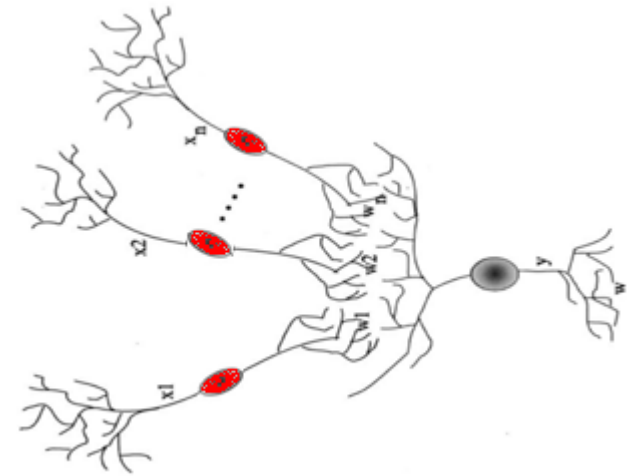


Artificial neural networks

- Model biological neural systems
 - › Signals transmitted if accumulated input signals exceed threshold value $f(x) \geq \theta$



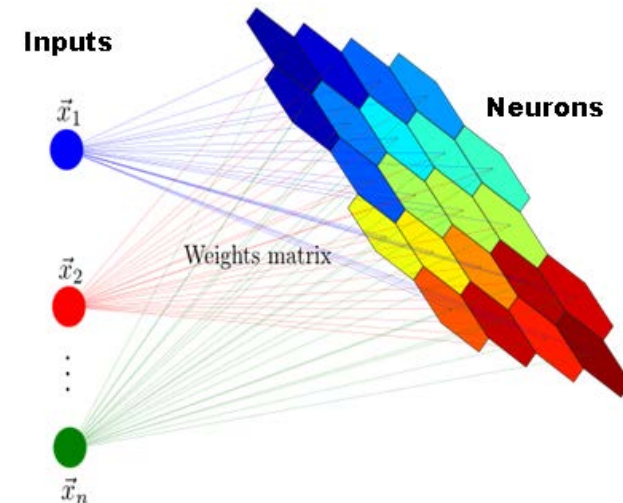
- Neurons or Nodes
 - › Arranged in layers
 - › Fully or partially connected
 - Connections are mathematically weighted
- Training
 - › The network „learns“ to recognise patterns
 - By adjusting connection weights
 - › Supervised Training $\equiv$ Explicit manual intervention
 - › Unsupervised Training $\equiv$ Self-training networks





Self-Organising Maps or Kohonen Feature Maps

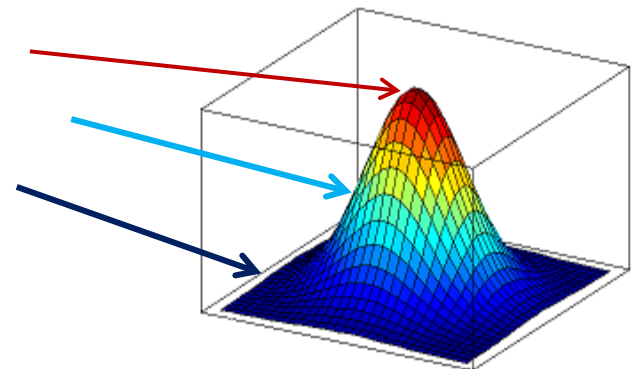
- A Self-Organising Map (SOM) is a neural network
- Single 2D layer of nodes
 - › Hexagonal / Rectangular configuration
- Fully-connected
 - › Every input is connected to every SOM node
 - › Connections are weighted
- Every node has a reference vector
 - › $\mathbf{p}_i(w_{i1}x_1, w_{i2}x_2, \dots, w_{in}x_n)$
 - › One weighted component for each input
 - › Training the network implies adjusting the weights
- Self-training
 - › SOMs use a competitive learning algorithm





Self-Organising Maps – Parameters

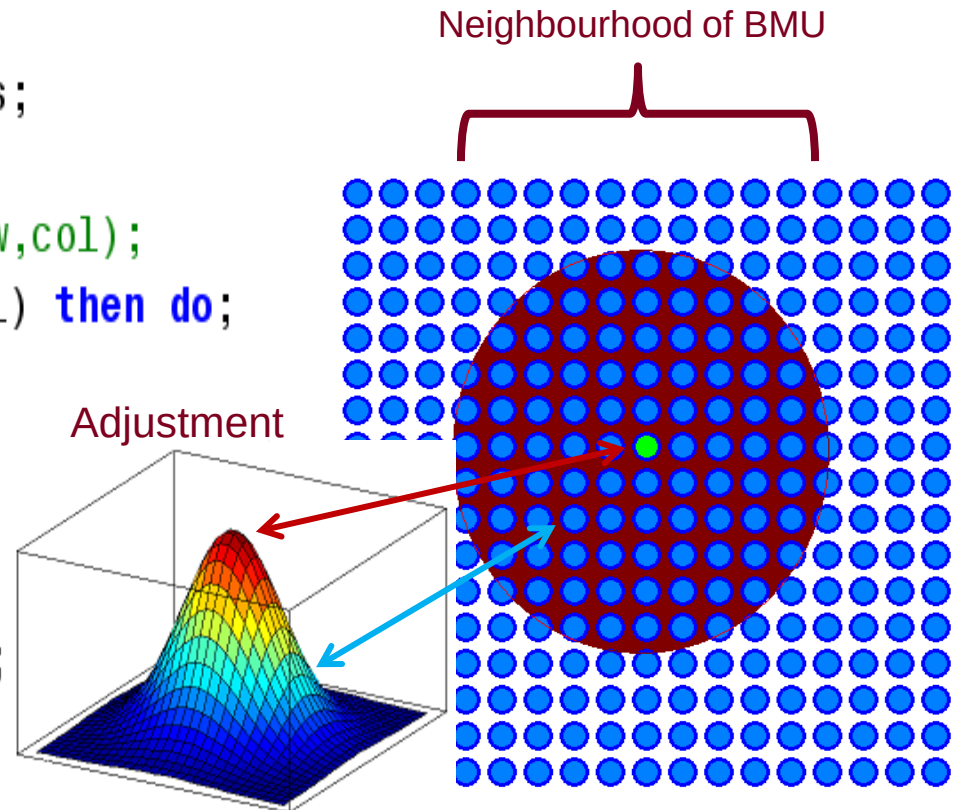
- Number of:
 - › Nodes in 2D SOM grid (rows x columns)
 - › Vector components / Input variables
 - › Training Records
 - › Training iterations
- Radii:
 - › Search radius to find Best-Matching Unit (BMU) in SOM grid
 - BMU := the node with the minimum **distance** from the input vector
 - Euclidean distance metric
 - › Neighbourhood radius to adjust weights
 - Maximum adjustment for the BMU itself
 - Less for other nodes within the neighbourhood
 - No adjustment beyond the neighbourhood
- Learning Rate reduces:
 - › Search radius
 - › Neighbourhood radius





Self-Organising Maps – Training by Competitive Learning

```
%Setup(numRows,numCols,numIterations,searchRadius,neighbourRadius,learnRate);  
*** Outer loop;  
do outerCount=1 to numIterations;  
    %ScrambleInputVectors(numInputVectors); *** Input vectors in random order;  
    *** Inner loop;  
    do innerCount=1 to numInputVectors;  
        %findBMU(innerCount);  
        *** For each grid node at (row,col);  
        if WithinNeighbourhood(row,col) then do;  
            %AdjustGridNode(row,col);  
        end;  
    end; *** Inner loop;  
    %ReduceSearchRadius(learnRate);  
    %ReduceNeighbourRadius(learnRate);  
end; *** Outer loop;
```



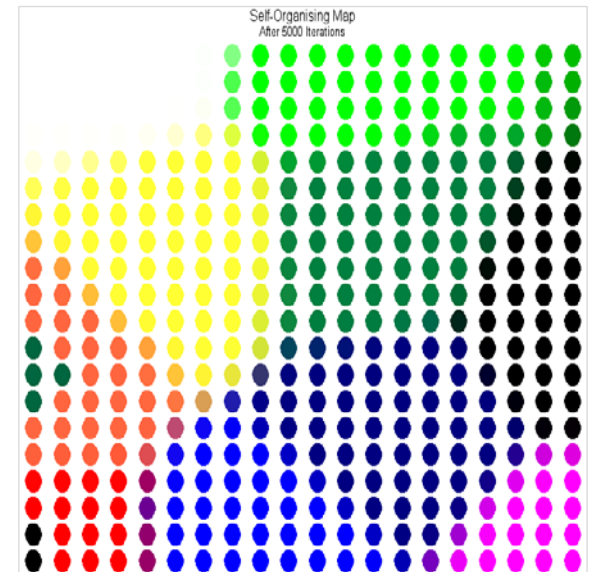


Example: Colour Map – Numeric data only

- Set up 20 x 20 nodes in the rectangular SOM grid
- Set up system parameters, e.g. Vector size = 3 (red,green,blue)
- Initialise each node to a random state
- Introduce the training set of 10 colours, e.g. White = (255,0,0)
- Run the competitive learning algorithm 5000 times
- Result: System converges to just the training colours in zones



Training
Vector





Competitive Learning – Distance Metric

- Competitive learning algorithm
 - › Step 1 : Find the BMU
 - The BMU is the node with the lowest **distance** ...
 - ... between its reference vector **p** and the input vector **q**
 - › Step 2 : Adjust the weights of the BMU and its neighbours
 - Aim: Minimise the total **distance** between the input vector and the BMU's
- Euclidean distance metric between 2 vectors

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2}$$

- › Numeric quantities – just plug them into the formula
- › Categorical values



1	DRUG WITHDRAWN
2	DRUG INTERRUPTED
3	DOSE REDUCED
4	DOSE NOT CHANGED
5	DOSE INCREASED
991	NOT APPLICABLE
992	UNKNOWN



Distances : Handling Categorical Data

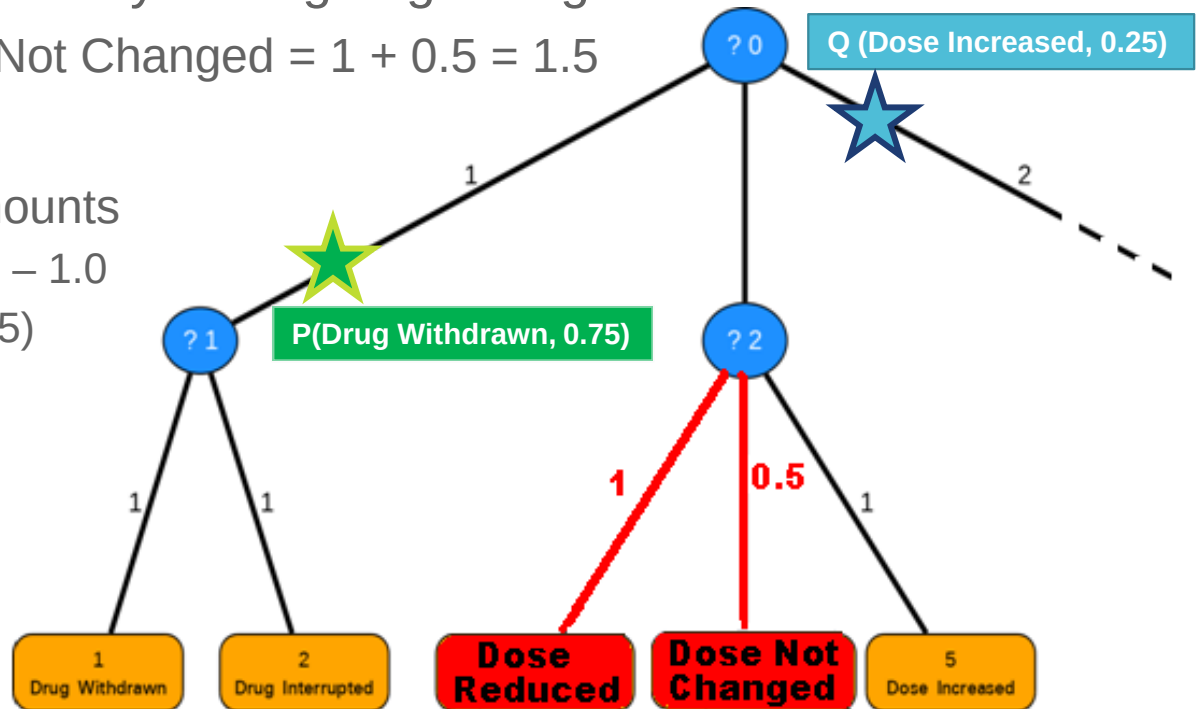
	Numeric Data	Categorical Data
Can be Ordered	Yes ($1.6 < 2.5$)	Sometimes (mild $<$ severe) But (male $?$ female)
Can be adjusted by arbitrary amounts	Yes ($1.6 + 0.3 = 1.9$)	Difficult (mild $+$ $?$ = severe) (green $+$ $?$ = blue)

- We need to be able to treat categorical values like numeric values
 - › Ordering
 - › Adjustment by arbitrary amounts



Data Hierarchies for Categorical Data

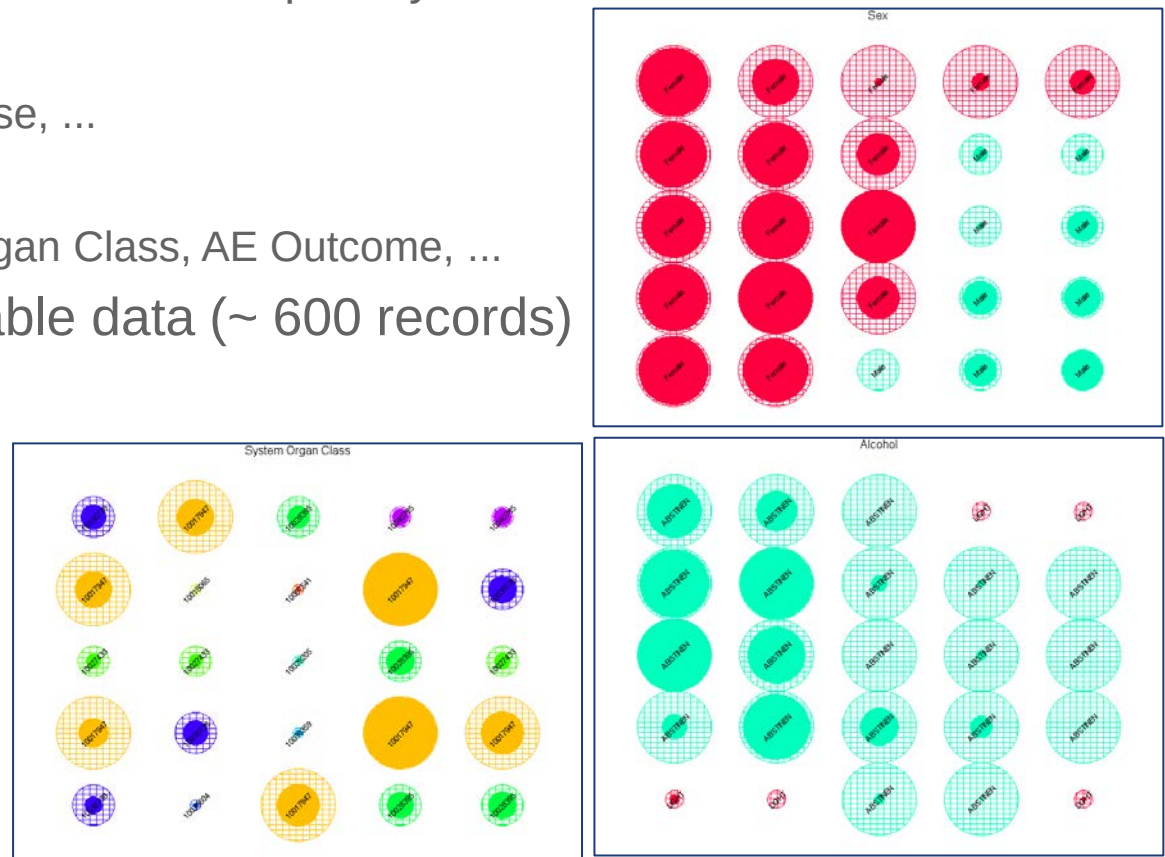
- Tree structure with weighted edges
- Items arranged by semantic proximity
 - › Similar concepts closer together
 - › Dose Reduced closer to Dose Not Changed than to Dose Increased
- Distances can be simulated by adding edge weights
 - › Dose Reduced to Dose Not Changed = $1 + 0.5 = 1.5$
- Distances can be
 - › Adjusted by arbitrary amounts
 - $P(\text{Drug Withdrawn}, 0.75) - 1.0 = Q(\text{Dose Increased}, 0.25)$
 - › Ordered
 - $P(\text{Drug Withdrawn}, 0.75) < R(\text{Drug Withdrawn}, 0.9)$





Example: Demographic and Safety Data

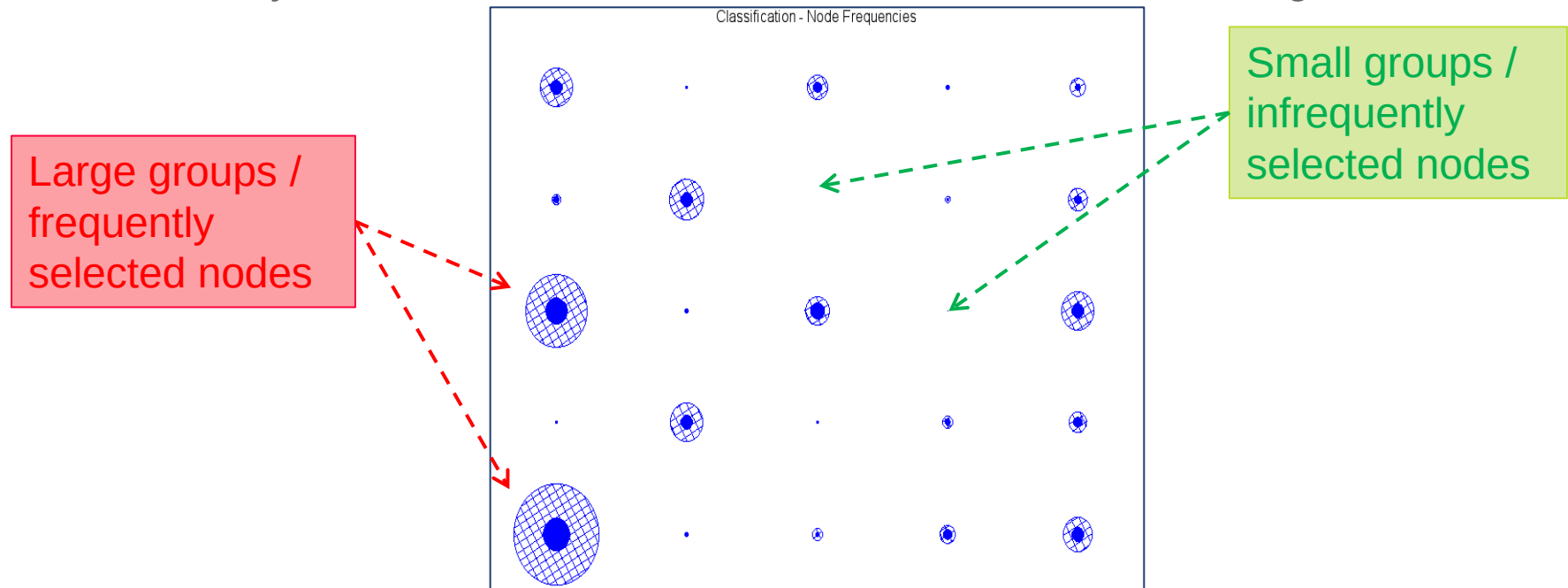
- 5 x 5 node rectangular SOM grid
- 14 variables from ADSL and ADAE
 - › Values scaled to influence results equitably
 - › Numeric variables
 - Age, BMI, Smoking dose, ...
 - › Categorical variables
 - Sex, Race, System Organ Class, AE Outcome, ...
- Training : half of available data (~ 600 records)
- 5000 iterations





Example: Demographic and Safety Data

- Trained SOM used as a classifier to group remaining ~ 600 records
- Sizes indicate how often a node was found to be the BMU
 - › Minimum distance between the BMU and the input
 - › Small circles represent residual error / confidence interval
- Each node represents a group of similar items
 - › Similarity extends over entire vector - all the variables taken together





Example: Demographic and Safety Data

- The 2 indicated nodes share these characteristics

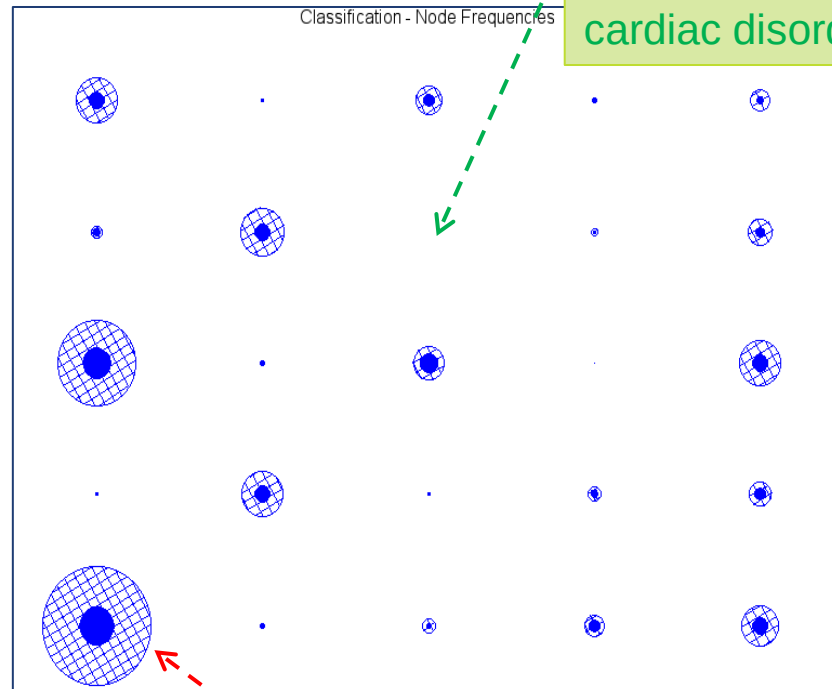
- > Subject / Demography

- White
 - Female
 - Moderate BMI

- > Adverse Event:

- Outcome = Recovered
 - Severity = Mild
 - Serious = No
 - Related = No

- Boxes show disparate Characteristics



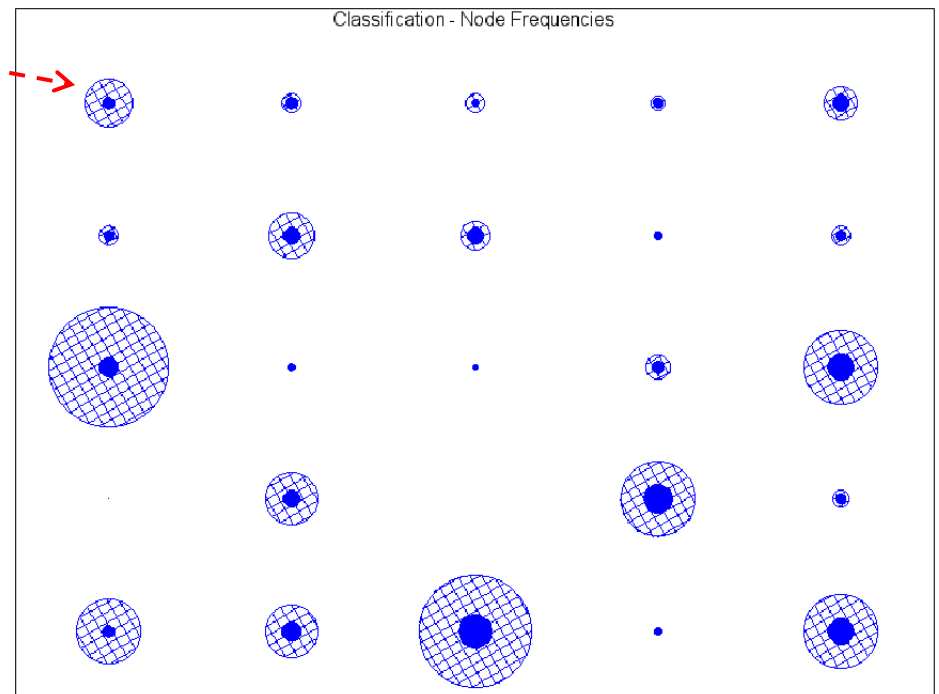
Younger,
some tobacco use,
non-drinkers
cardiac disorders

Older,
non-smokers,
light drinkers,
respiratory disorders



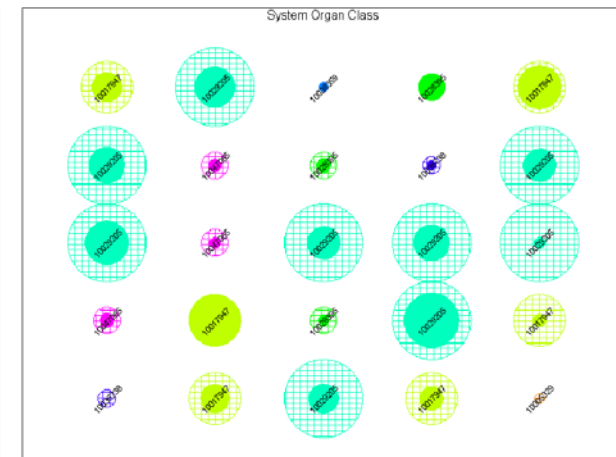
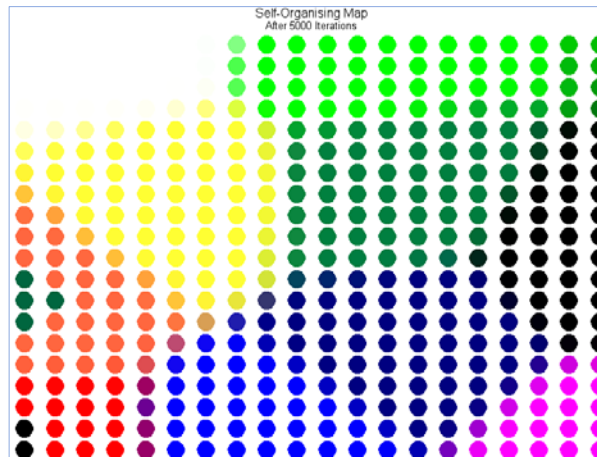
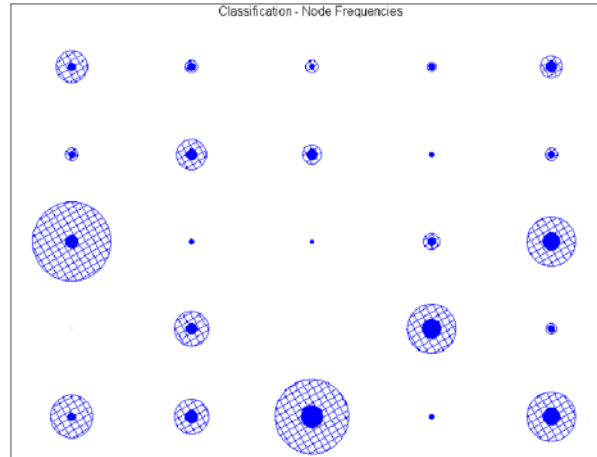
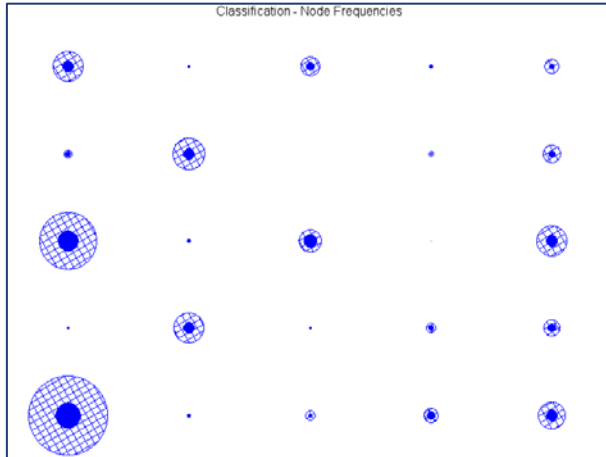
Uses of Self-Organising Maps - Review

- Classification
 - › Group similar items using a distance metric
- Whole-vector queries
 - › Queries can be formulated by tweaking the values of some reference vectors, e.g.
 - Serious AEs
 - SOC = 10029205
 - › (nervous system disorders)
 - Men
 - High BMI
 - Over 50
 - Smokers
 - › Then re-running the classification
 - › Arrowed node „attracts“ a group of similar input records





Self-Organising Maps – Q & A



Colin de Klerk

Principal Statistical Programmer

inVentiv Health Germany GmbH

T +49 (0) 6123 70437 14

colin.deklerk@inventivhealth.com