Australian National University

Kohonen's Self Organising Feature Maps (SOMs)

Tom Gedeon

Research School of Computer Science
Australian National University
tom@cs.anu.edu.au

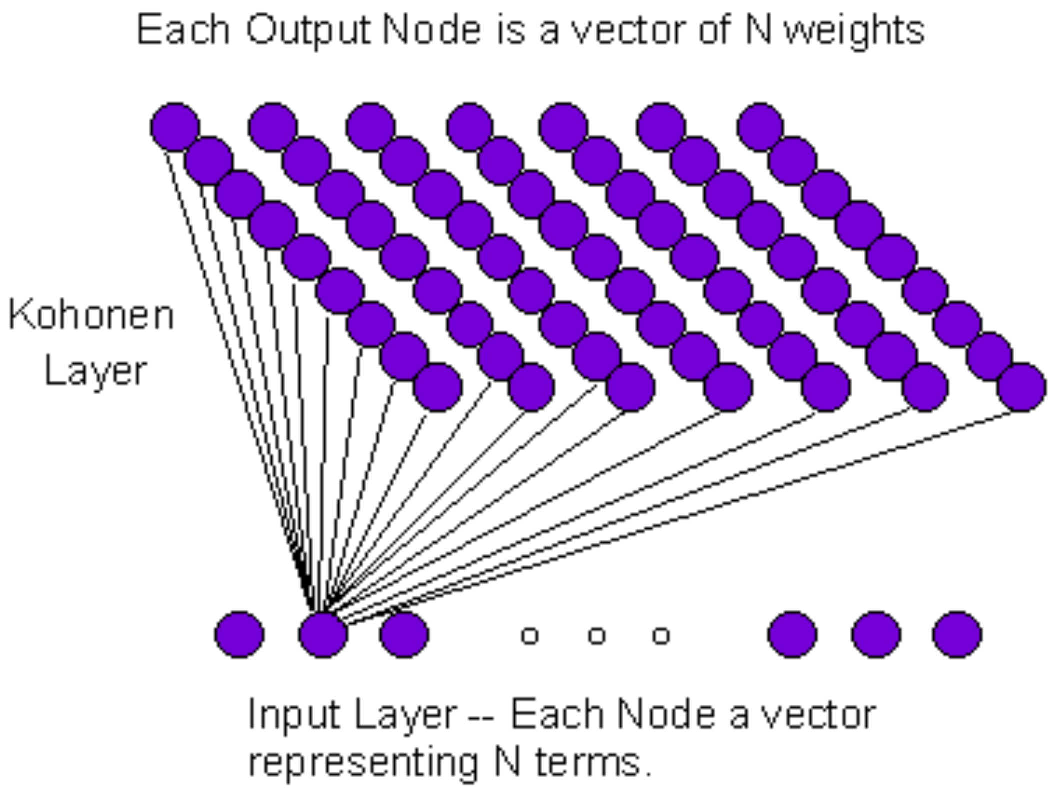
Based partly on:
Kohonen: IEEE Trans on Neural Nets v11 n3:3, May 2000
"ai-junkie" tutorial at http://www.ai-junkie.com/ann/som/som1.html

Human Centred Computing

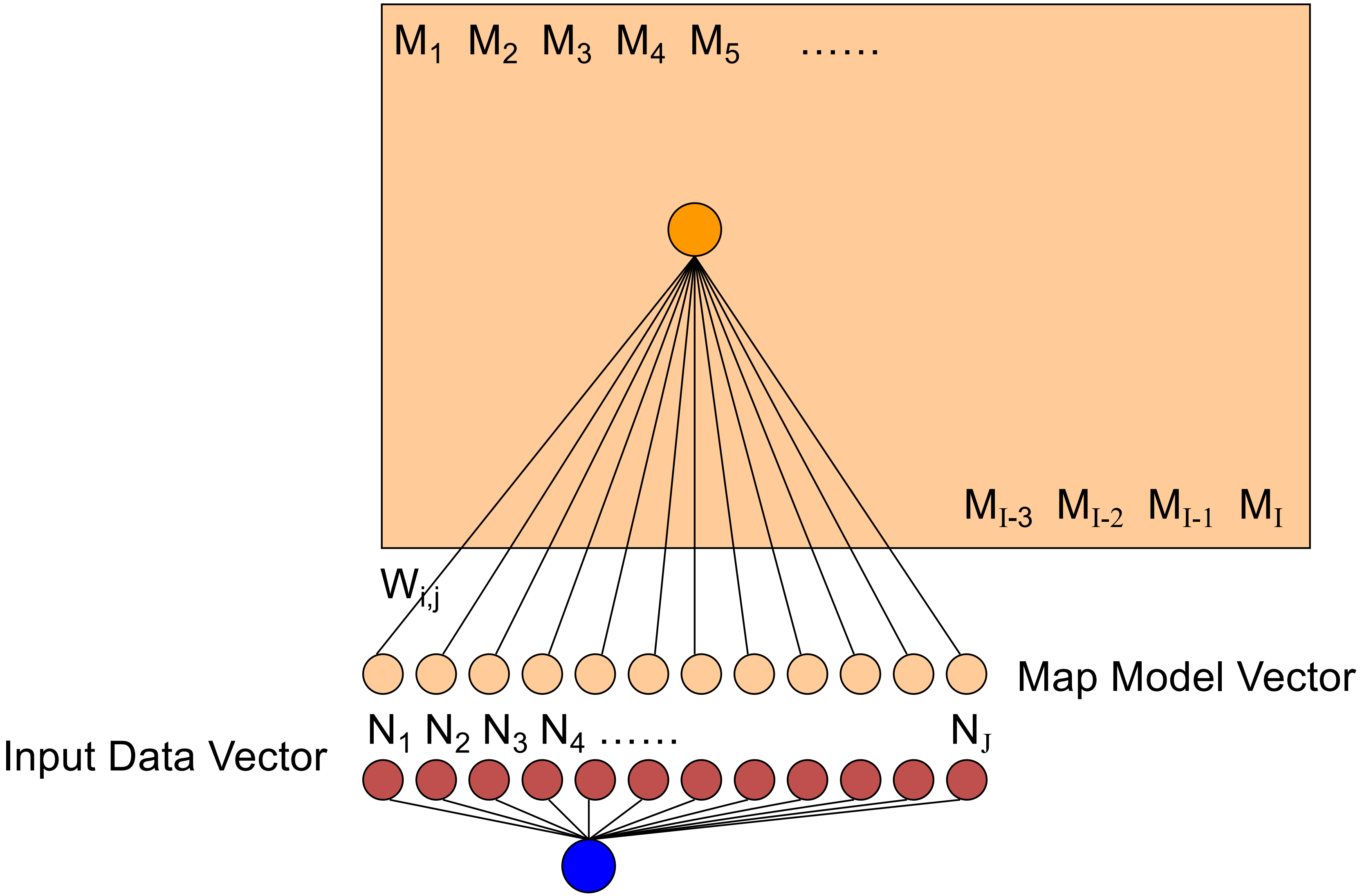
Purpose / function of SOM

- Map from higher number of dimensions down
  - Usually to 2D – Good for visualisation
- Unsupervised learning
  - Learn “similarity graph of input data”
    - Approximate probability density function of inputs
  - Neurons in a regular usually 2D grid
- Want similar input vectors represented by nearby neurons
  - Maintain topological relationships (as possible)

SOM algorithm



SOM algorithm



SOM algorithm

1. Initialize input nodes, output nodes, and connection weights

– input vector: top (most frequently occurring) *N* terms

– output nodes: create a 2-dimensional map (grid) of *M*

– Initialize weights w_{ij} *from N input nodes to M output nodes to small random values*

SOM algorithm

2. Present each document in order

– Describe each document as an input vector of *N* coordinates

3. Compute distance to all nodes

$$d_i = \sum_{j=0}^{N-1} \left(x_j(t) - w_{ji}(t) \right)^2 \qquad (1)$$

SOM algorithm

- **4. Select winning node *i** and update weights to node *i** and its neighbors**
  - Update weights to nodes *i* and its neighbors to reduce the distances between them and the input vector $x_j(t)$:

$$w_{ij}(t+1) = w_{ij}(t) + \eta(t)\left(x_j(t+1) - w_{ij}(t)\right) \qquad (2)$$

  - $\eta(t)$ is an error-adjusting coefficient ($0 < \eta(t) < 1$) that decreases over time

SOM algorithm

- Recursive regression process
- t is the index of the regression step
- $\mathrm{x}(\mathrm{t})$ is the presentation of a sample of input x
- $h_{c(\mathrm{x}),i}(t)$ is the neighborhood function
- $m_{c(\mathrm{x})}(t)$ is the model ("winner") that matches best with

$$\mathrm{m}_i(t+1) = \mathrm{m}_i(t) + h_{c(\mathrm{x}),i}(t)\left[\mathrm{x}(t) - \mathrm{m}_i(t)\right] \qquad (3)$$

$$c(\mathrm{x}) = \arg\min_i \left\{ \left\| \mathrm{x}\text{-}\mathrm{m}_i \right\| \right\} \qquad (4)$$

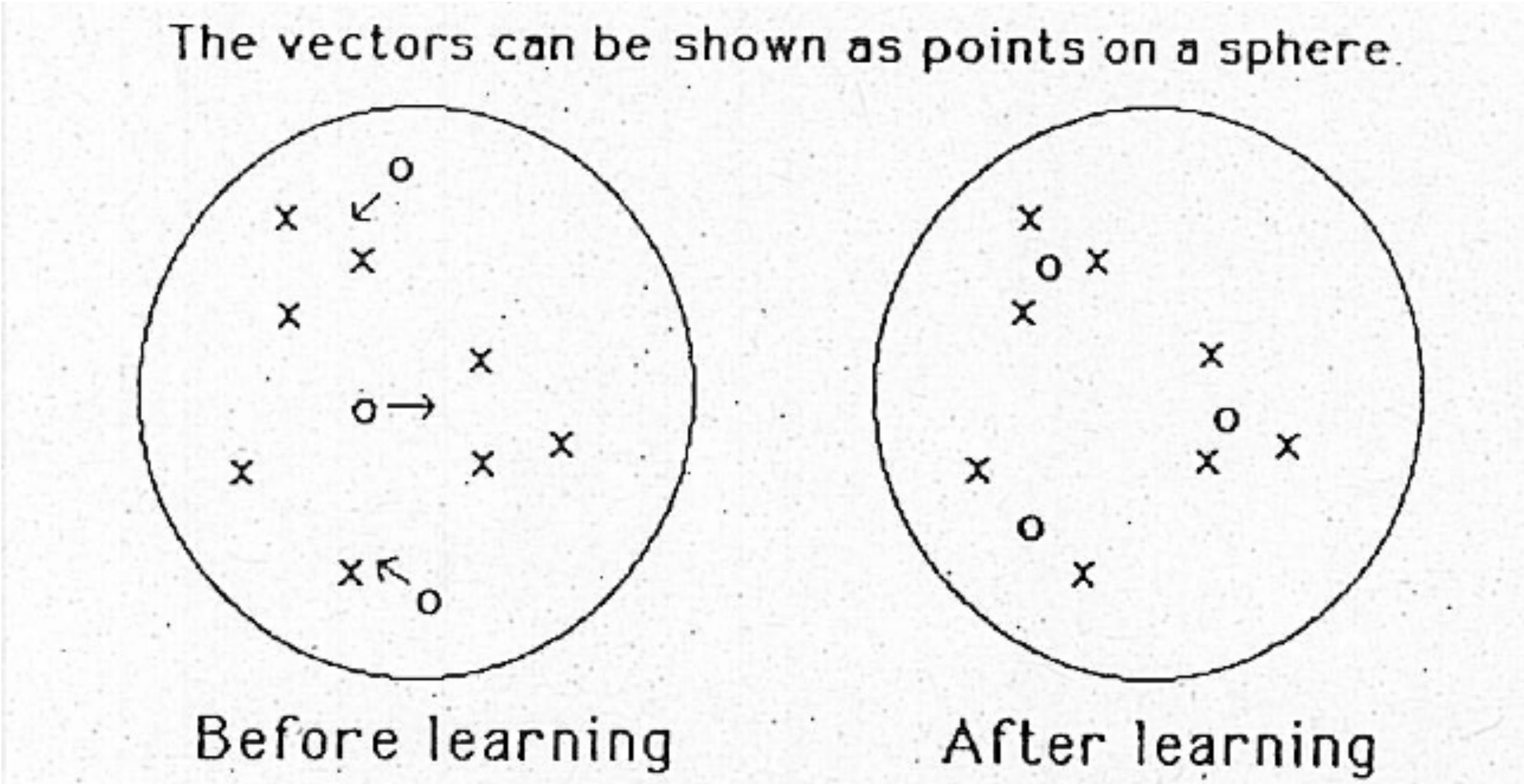
$$h_{c(\mathrm{x}),i}(t) = \alpha(t)\exp\left(-\frac{\left\| r_i - r_{c(\mathrm{x})} \right\|^2}{2\sigma^2(t)} \right) \qquad (5)$$

SOM algorithm

- $0 < \alpha(t) < 1$ is the learning-rate factor which decreases monotonically with the regression steps

- $\sigma(t)$ corresponds to the width of the neighborhood function, which is also decreasing monotonically with the regression steps

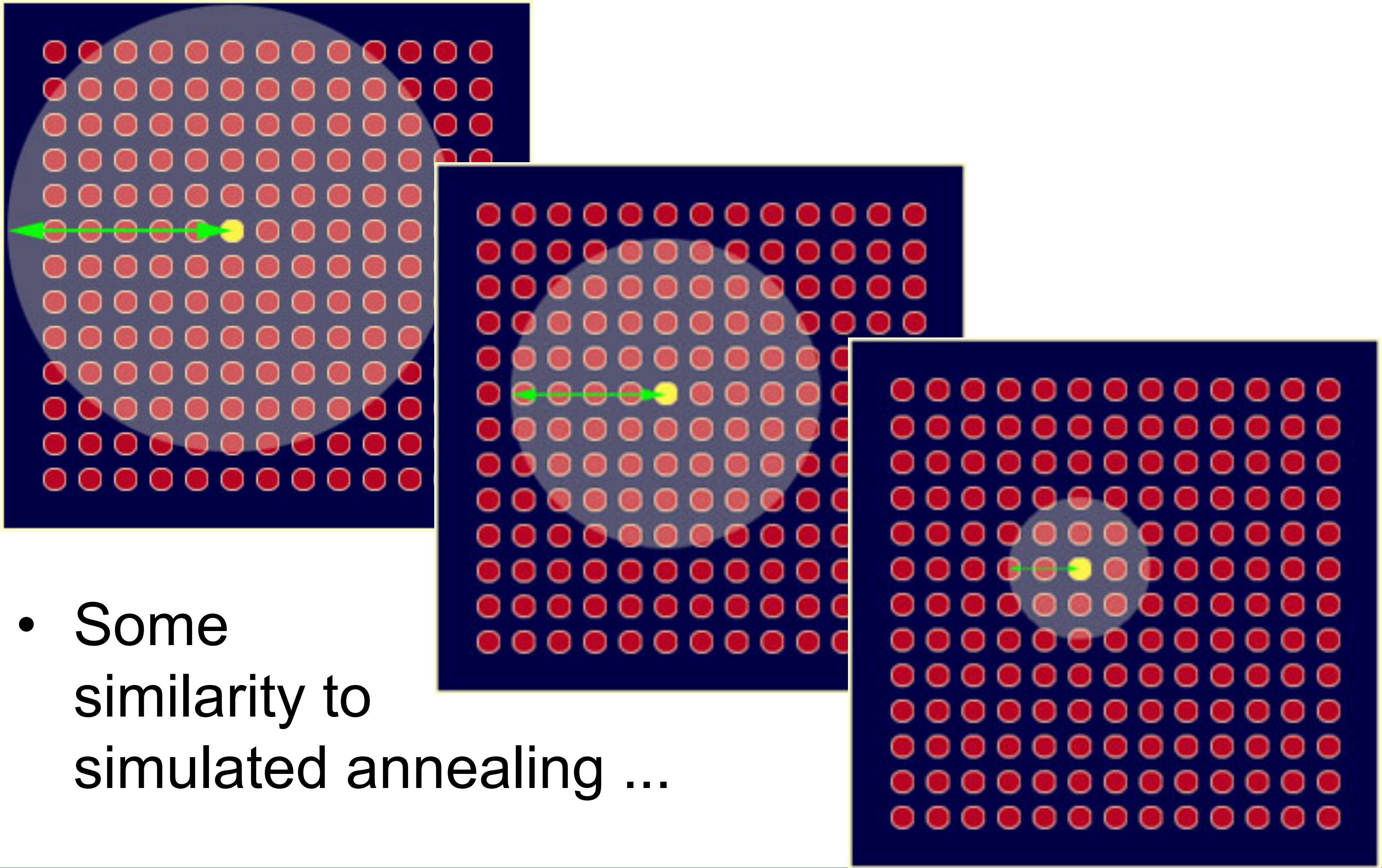
A geometric analogy



x = input vector
o = weight vector

- Assumed 3 inputs, with
  - Vector length normalised

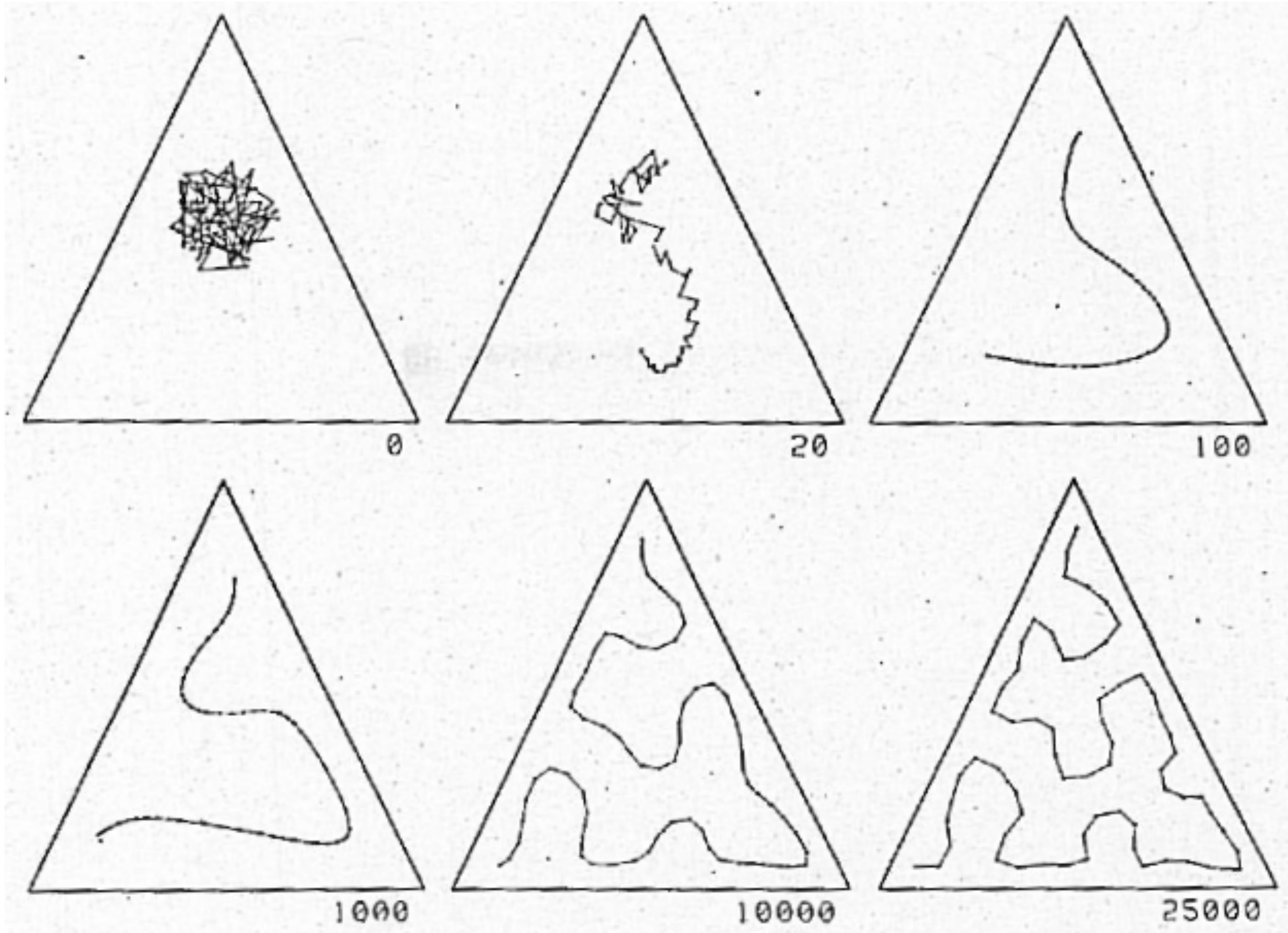
Neighbourhood decreases over time



- Some similarity to simulated annealing ...

Example with evenly distrib. data

- Illustrates ability to do topological mapping
  - Not otherwise a useful feature of SOMs
  - 2D 'triangle' of data
  - Linear topology used here



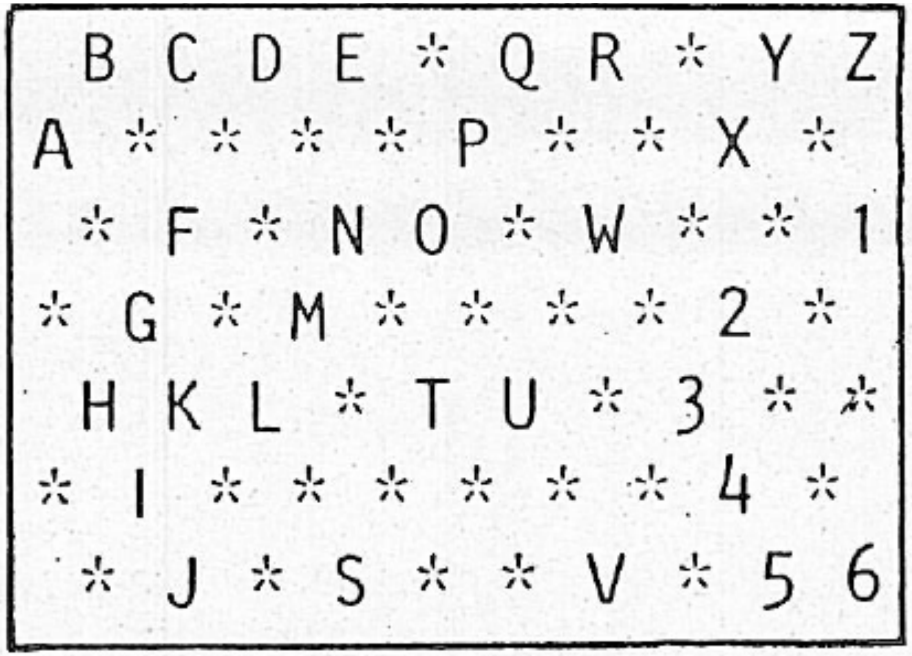
Example with hierarchical data

Table 1 Input Data Matrix

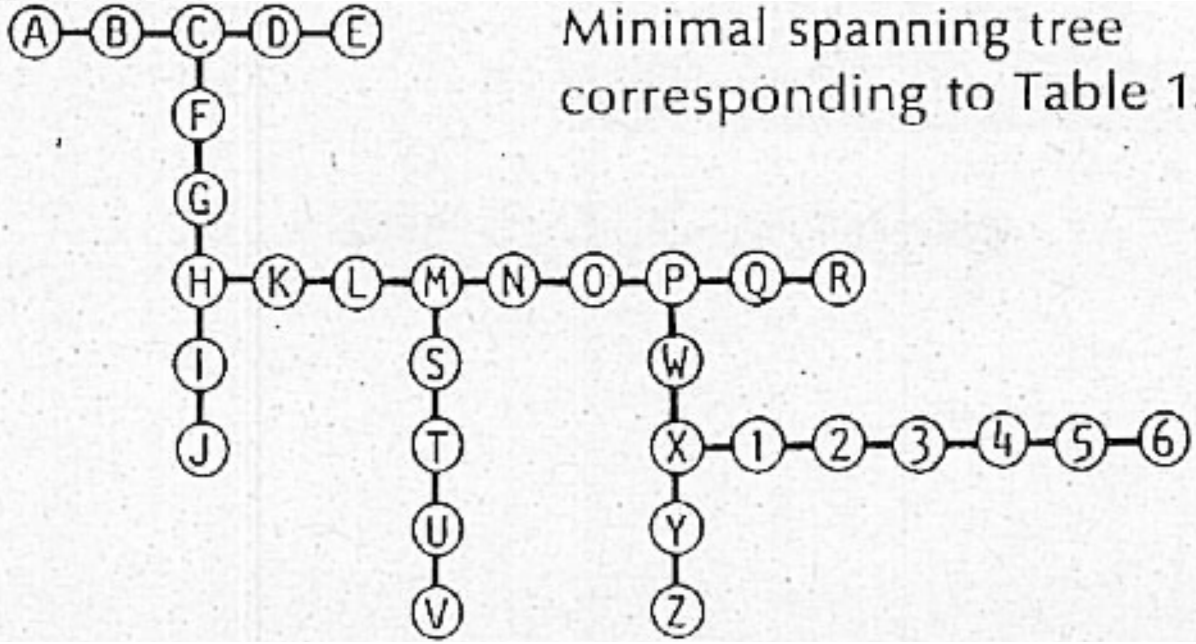
Attribute	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
a_1	1	2	3	4	5	3	3	3	3	3	3	3	3	3	3	3
a_2	0	0	0	0	0	1	2	3	4	5	3	3	3	3	3	3
a_3	0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	6
a_4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
a_5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Q	R	S	T	U	V	W	X	Y	Z	1	2	3	4	5	6
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
7	8	3	3	3	3	6	6	6	6	6	6	6	6	6	6
0	0	1	2	3	4	1	2	3	4	2	2	2	2	2	2
0	0	0	0	0	0	0	0	0	0	1	2	3	4	5	6

Hierarch. example continued



Self-organized map of the data matrix of Table 1.

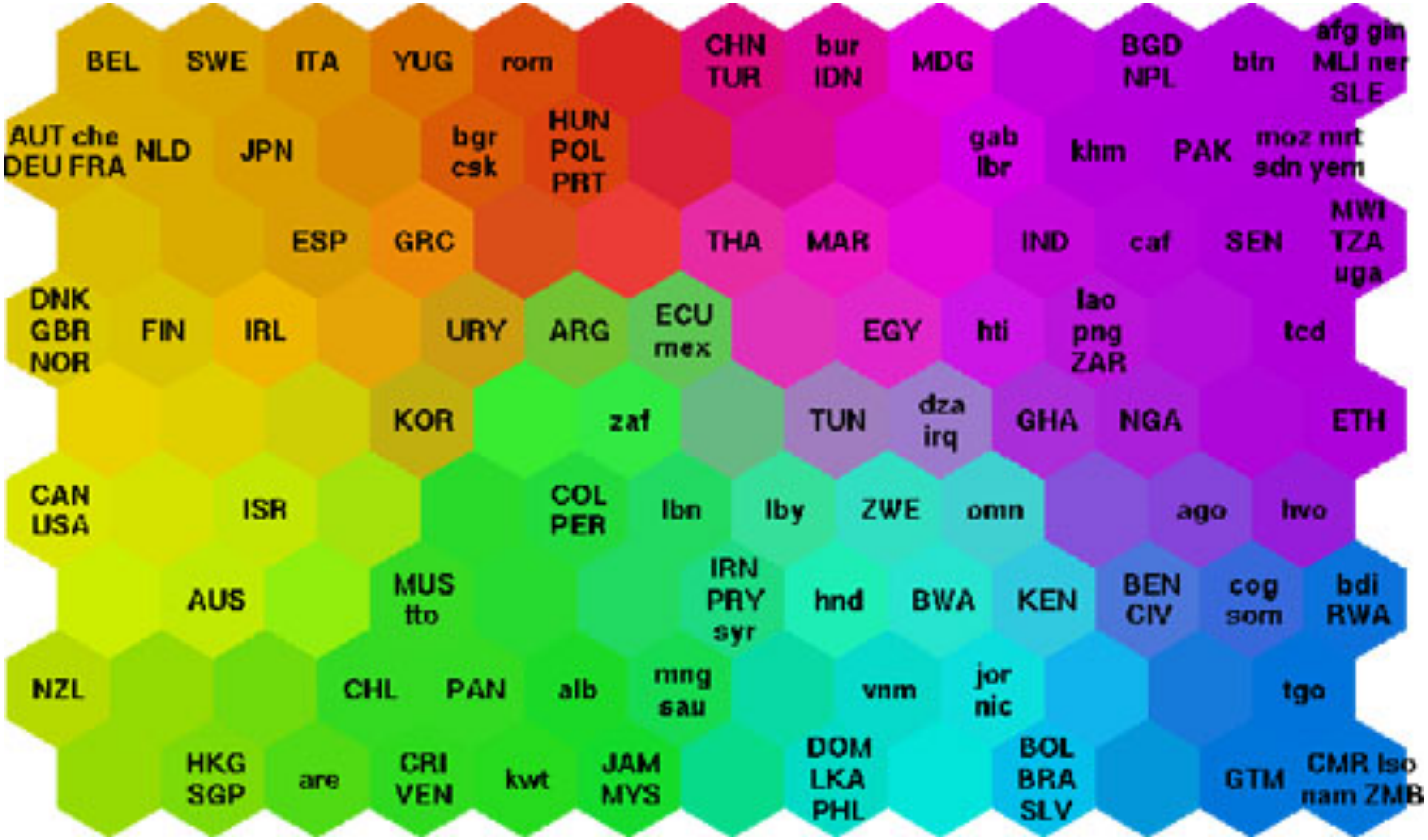


Minimal spanning tree corresponding to Table 1.

Australian National University

Applic. World Poverty Map

- Various quality-of-life factors such as state of health, nutrition, educational services etc.
- A hex is a neuron in the SOM



Poverty map on World map

- Has found reasonably similarities