# The Self-Organising Maps for Data Visualisation and Principal Manifold Mapping

Hujun Yin

The University of Manchester

# SOM, ViSOM, Data Visualisation and Beyond

- ❑PCA, MDS, Principal Curve/Surface
- ❑SOM: Background & Data Visualisation
- ❑ViSOM & Principal Curve/Surface
- ❑Kernel Method, SOM & Mixture Model
- ❑Conclusions

# 1. PCA, MDS & Principal Curve/Surface

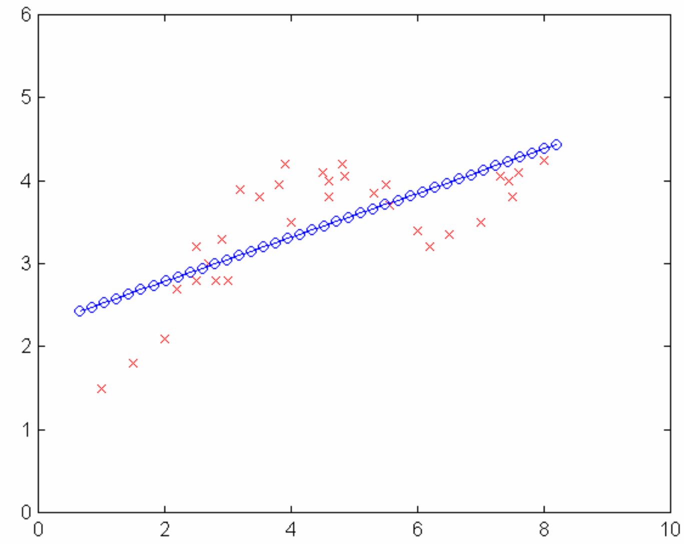
## *PCA is a linear coordinate transformation*

- To reduce the dimensionality of the data set
- To identify new "meaningful" (hidden) variables

$$\min \sum_{\mathbf{x}} \| X - \sum_{j=1}^{m} (\mathbf{q}_j^T X)\mathbf{q}_j \|^2$$

$$\max \{\mathbf{q}_i^T \mathbf{C} \mathbf{q}_i = \sigma_i^2\}, \ \mathbf{q}_i \perp \mathbf{q}_j, i \neq j$$

- $X$ : $n$-dimensional vector, zero-mean
- $\{\mathbf{q}_j\}$: orthogonal, eigenvectors of data covariance $\mathbf{C}=E[XX^T]$
- $m \leq n$




$|\mathbf{C}-\lambda_i\mathbf{I}|=0$

$(\mathbf{C}-\lambda_i\mathbf{I})\mathbf{q}_i=0$

### PCA decomposition

$$\mathbf{Q}^T E[XX^T]\mathbf{Q} = \mathbf{\Lambda}$$

- $\mathbf{Q}=[\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n]$
- $\mathbf{\Lambda}=\text{diag}\ [\lambda_1, \lambda_2, \ldots \ \ \lambda_n]$
- $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ eigenvalues or variances

☺ *simple, direct visualisation*

☺ *stable (fast) solution*

☹ *linear mapping*

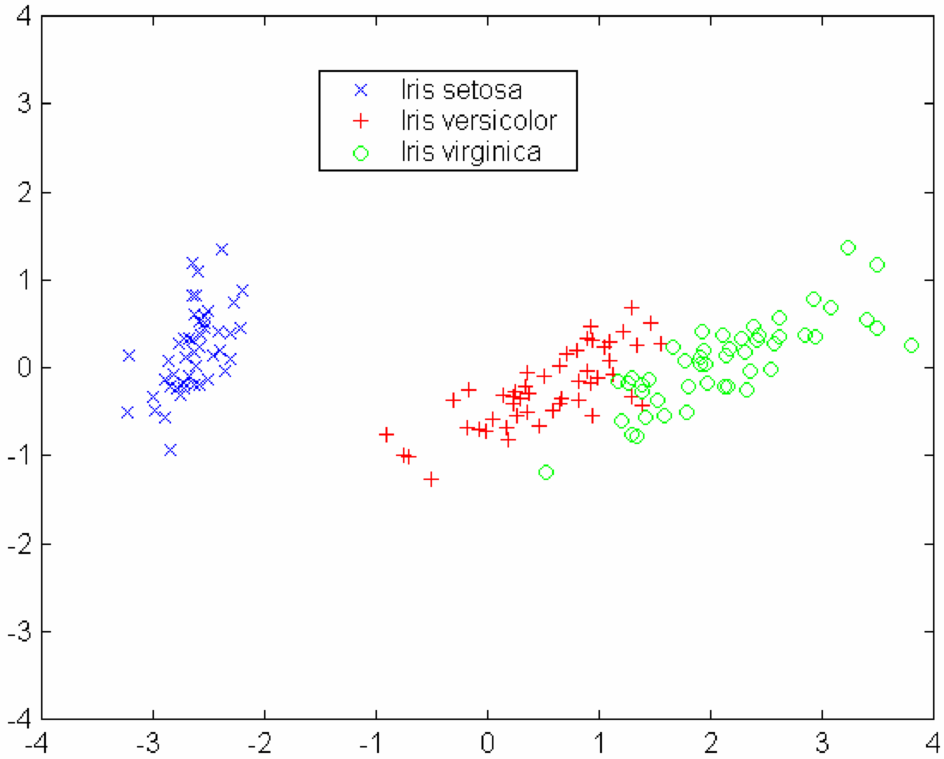
☹ *batch operation*

# 1. PCA, MDS & Principal Curve/Surface

## ***PCA**: Example –Iris data*

- 150 4-D vectors
- 3 categories, 50 points each

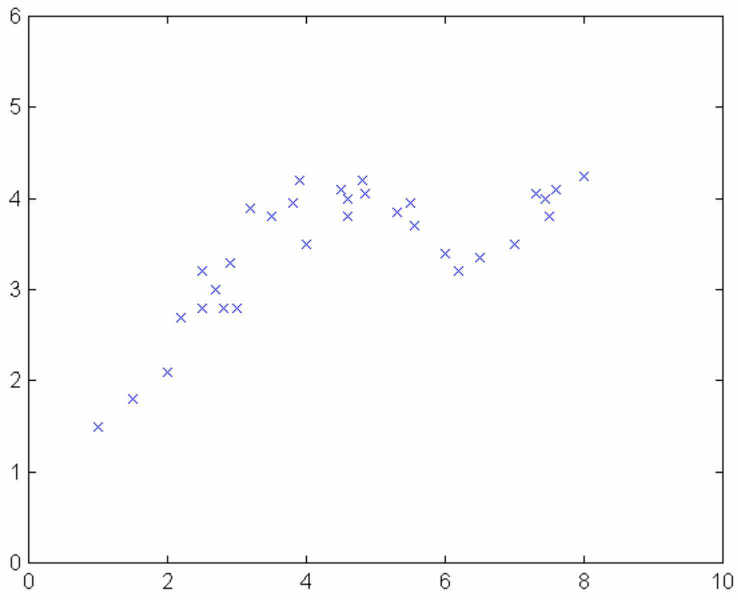
| | | | |
|---|---|---|---|
| 4.9 | 3.0 | 1.4 | 0.2 |
| 4.7 | 3.2 | 1.3 | 0.2 |
| 4.6 | 3.1 | 1.5 | 0.2 |
| 5.0 | 3.6 | 1.4 | 0.2 |
| 5.4 | 3.9 | 1.7 | 0.4 |
| 4.6 | 3.4 | 1.4 | 0.3 |
| …… | | | |
| 7.0 | 3.2 | 4.7 | 1.4 |
| 6.4 | 3.2 | 4.5 | 1.5 |
| 6.9 | 3.1 | 4.9 | 1.5 |
| 5.5 | 2.3 | 4.0 | 1.3 |
| 6.5 | 2.8 | 4.6 | 1.5 |
| 5.7 | 2.8 | 4.5 | 1.3 |
| …… | | | |
| 6.3 | 3.3 | 6.0 | 2.5 |
| 5.8 | 2.7 | 5.1 | 1.9 |
| 7.1 | 3.0 | 5.9 | 2.1 |
| 6.3 | 2.9 | 5.6 | 1.8 |
| 6.5 | 3.0 | 5.8 | 2.2 |
| 7.6 | 3.0 | 6.6 | 2.1 |
| …… | | | |
| …… | | | |




Projection onto the 1st×2nd eigenvectors

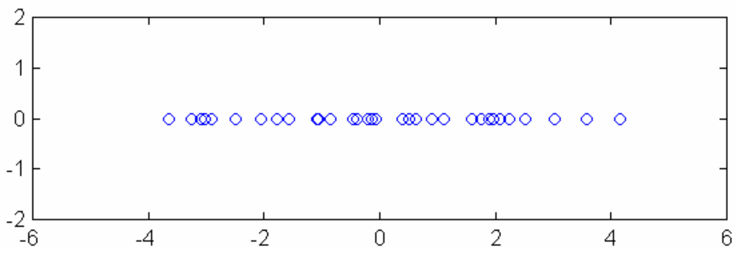
# 1. PCA, MDS & Principal Curve/Surface

## ***MDS**: Sammon Mapping*




$$S_{Sammon} = \frac{1}{\sum_{i<j} d_{ij}^{*}} \sum_{i<j} \frac{[d_{ij}^{*} - d_{ij}]^2}{d_{ij}^{*}}$$

- $d_{ij}^{*}$: *inter-point distance in original space*
- $d_{ij}$: *inter-point distance in projected plot*




☺ *nonlinear, direct visualisation*

☺ *stable solution*

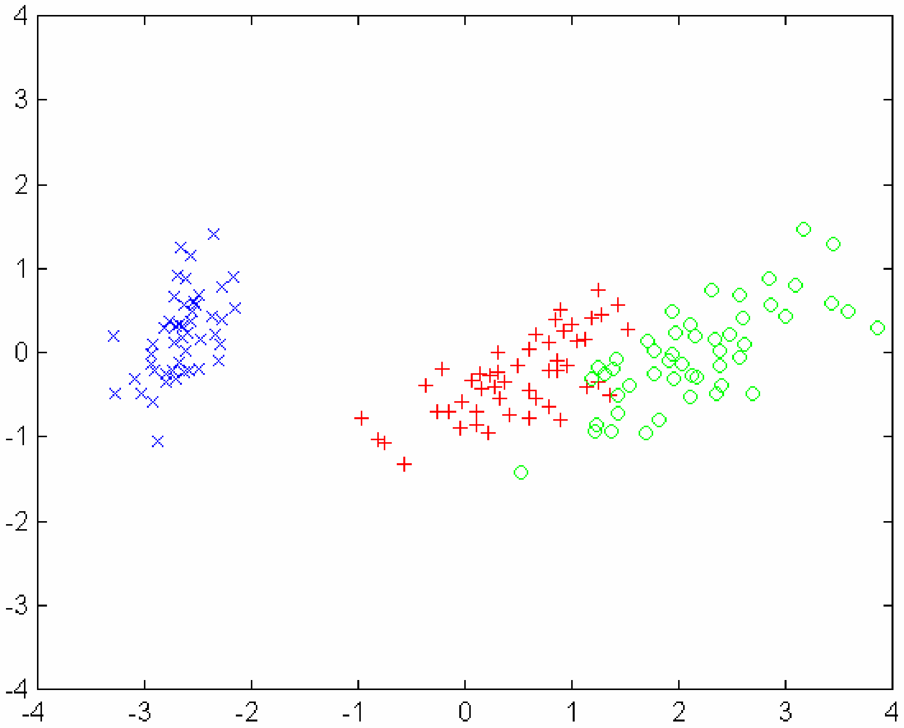
☹ *point-point mapping (no function)*

☹ *computational intensive*

# 1. PCA, MDS & Principal Curve/Surface

## ***MDS**: Sammon Mapping*

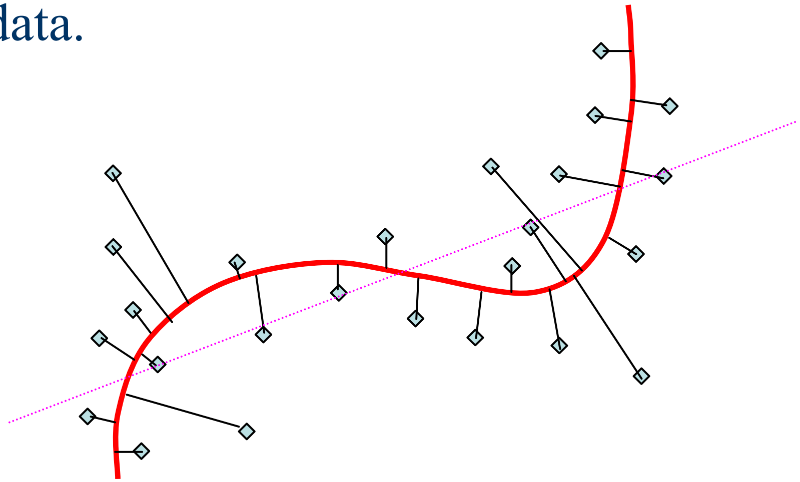
| | | | |
|---|---|---|---|
| 4.9 | 3.0 | 1.4 | 0.2 |
| 4.7 | 3.2 | 1.3 | 0.2 |
| 4.6 | 3.1 | 1.5 | 0.2 |
| 5.0 | 3.6 | 1.4 | 0.2 |
| 5.4 | 3.9 | 1.7 | 0.4 |
| 4.6 | 3.4 | 1.4 | 0.3 |
| …… | | | |
| 7.0 | 3.2 | 4.7 | 1.4 |
| 6.4 | 3.2 | 4.5 | 1.5 |
| 6.9 | 3.1 | 4.9 | 1.5 |
| 5.5 | 2.3 | 4.0 | 1.3 |
| 6.5 | 2.8 | 4.6 | 1.5 |
| 5.7 | 2.8 | 4.5 | 1.3 |
| …… | | | |
| 6.3 | 3.3 | 6.0 | 2.5 |
| 5.8 | 2.7 | 5.1 | 1.9 |
| 7.1 | 3.0 | 5.9 | 2.1 |
| 6.3 | 2.9 | 5.6 | 1.8 |
| 6.5 | 3.0 | 5.8 | 2.2 |
| 7.6 | 3.0 | 6.6 | 2.1 |
| …… | | | |
| …… | | | |



# 1. PCA, MDS & Principal Curve/Surface

## *Principal Curve/Surface*

Principal curve was defined by Hastie and Stuetzle (1989) as a smooth and self-consistent curve passing through the "middle" of the data.



Projection:

$$\rho_f(\mathbf{x}) = \sup_{\rho \in \Lambda}\{\rho : \|\mathbf{x} - f(\rho)\| = \inf_{\vartheta}\|\mathbf{x} - f(\vartheta)\|\}$$

Expectation:

$$f(\rho) = E[\mathbf{X} \mid \rho_f(\mathbf{X}) = \rho]$$

Kernel smoothing:

$$F(\rho) = \frac{\sum_i^S \mathbf{x}_i \kappa(\rho, \rho_i)}{\sum_i^S \kappa(\rho, \rho_i)}$$

- ☺ *principled nonlinear extension of PCA*
- ☺ *smooth mapping function*
- ☹ *lack good algorithm, esp. in 2D*
- ☹ *boundary problems*


h.yin@manchester.ac.uk

MANCHESTER 1824

The University of Manchester

# 2. SOM: Background

## ***SOM**: Background–Hebbian Learning*

*When an axon of cell A is near enough to excite a cell B and repeatedly or persistently takes part in firing it, some growth process or metabolic changes take place in one or both cells such that A's efficiency as one of the cells firing B, is increased.* (Donald Hebb, 1949)

*In mathematical term:* $\Delta w = \alpha x y$

*Oja' rule:*

$$w_i(t+1) = \frac{w_i(t) + \alpha x_i(t) y(t)}{\{\sum_{j=1}^{n} [w_j(t) + \alpha x_j(t) y(t)]^2\}^{1/2}} \approx w_i(t) + \alpha y(t)[x_i(t) - y(t) w_i(t)] + O(\alpha^2)$$

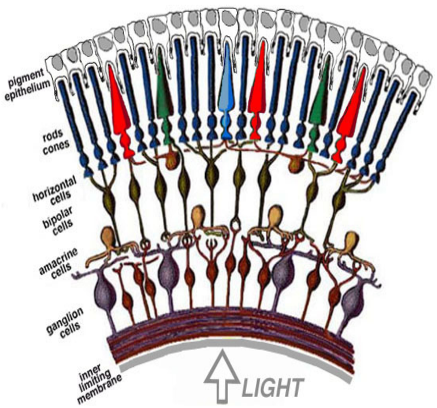
# 2. SOM: Background

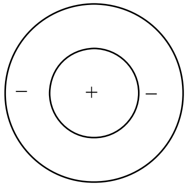
## ***SOM**: Background–Lateral Inhibition*

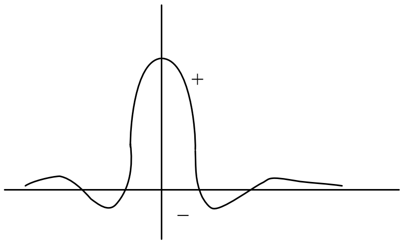
The University of Manchester

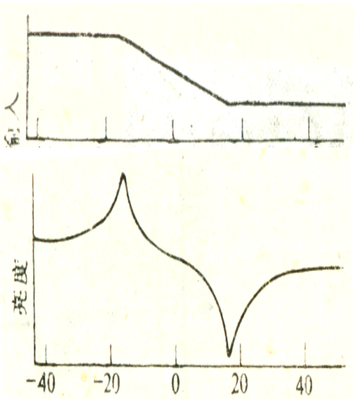
# 2. SOM: Background

## ***SOM**: Background–Lateral Inhibition*

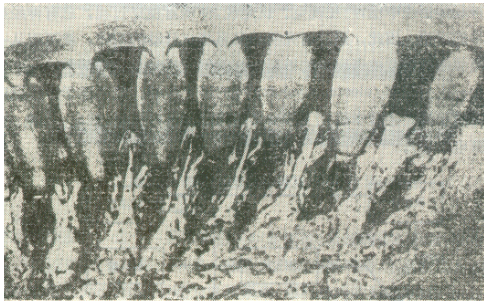


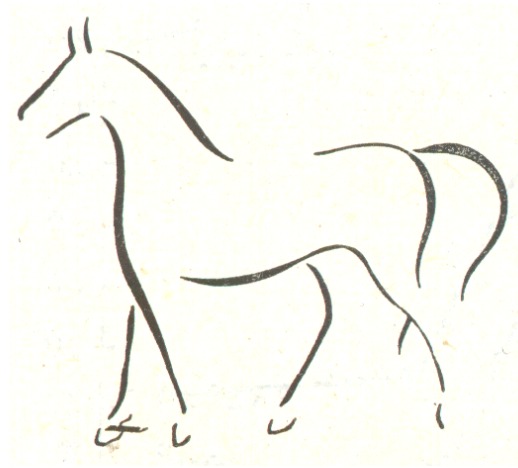




*Hartline, et al. 1960s*



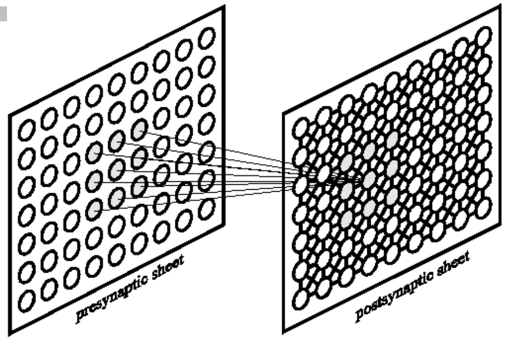


*from V. Bruce & P.R Green*

*It explains Mach-band effect and abstraction purpose*

# 2. SOM: Background

## ***SOM**: Background - Model*




**Hebbian learning (Hebb 1949)** $\Delta w = \alpha x y$

**von der Malsburg and Willshaw's model (1973, 1976)**

$$\frac{\partial y_i(t)}{\partial t} + c y_i(t) = \sum_j w_{ij}(t) x_i(t) + \sum_k e_{ik} y_k(t) - \sum_{k'} b_{ik'} y_{k'}(t)$$

$$y_j^*(t) = \begin{cases} y_j^*(t) - \theta, & \text{if } y_j^*(t) > \theta \\ 0 & \text{otherwise} \end{cases}$$

$$\frac{\partial w_{ij}(t)}{\partial t} = \alpha x_i(t) y_j^*(t), \quad \text{subject to } \sum w_{ij} = \text{constant}$$

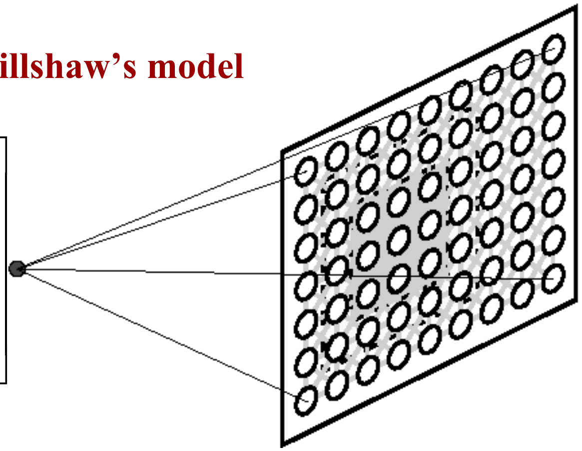
**Kohonen's model (1982) is an abstraction of von der Malsburg and Willshaw's model**

$$y_j(t+1) = \varphi[\mathbf{w}_j^T \mathbf{x}(t) + \sum_i h_{ij} y_i(t)]$$

$$\frac{\partial w_{ij}(t)}{\partial t} = \alpha y_j(t) x_i(t) - \beta y_j(t) w_{ij}(t)$$

$$= \alpha [x_i(t) - w_{ij}(t)] y_j(t) = \begin{cases} \alpha [x_i(t) - w_{ij}(t)], & \text{if } j \in \eta(t) \\ 0 & \text{if } j \notin \eta(t) \end{cases}$$

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ \vdots \\ x_n \end{bmatrix}$$



# 2. SOM: The Algorithm

## *SOM: Algorithm*

- *At each time* $t$, *present an input,* $\mathbf{x}(t)$, *select the winner.*

$$v = \arg\min_{c\in\Omega} \left\| \mathbf{x}(t) - \mathbf{w}_c \right\|$$

- *Updating the weights of winner and its neighbours.*

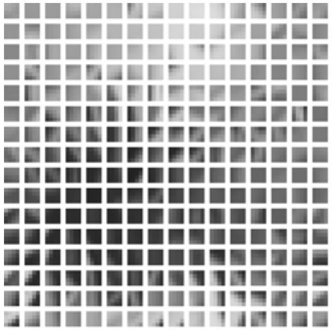
$$\Delta\mathbf{w}_k(t) = \alpha(t)\eta(v,k,t)[\mathbf{x}(t) - \mathbf{w}_v(t)]$$

- *Repeat until the map converges.*

**Typical neighbourhood function:** $\eta(v,k,t) \propto \exp[-\frac{\left\| v-k \right\|^2}{2\sigma(t)^2}]$

# 2. SOM: Interpretation

## ***SOM**: Quantisation, Topology & Cost Function*



Topologically "ordered" map

$$E(\mathbf{w}_1,\ldots\mathbf{w}_N) = \sum_i \int_{V_i} \sum_k h_{i,k} \left\| \mathbf{x} - \mathbf{w}_k \right\|^2 p(\mathbf{x}) d\mathbf{x}$$

(Heskes, 1999)

"*Error tolerant*" coding -HVQ (Luttrell, NC 1994)

"*Minimum wiring*" (Mitchison, NC 1995), (Durbin&Mitchson, Nature1990)

The University of Manchester

MANCHESTER 1824

# 2. SOM: Variants/Extensions

## *SOM: Variants & Extensions*

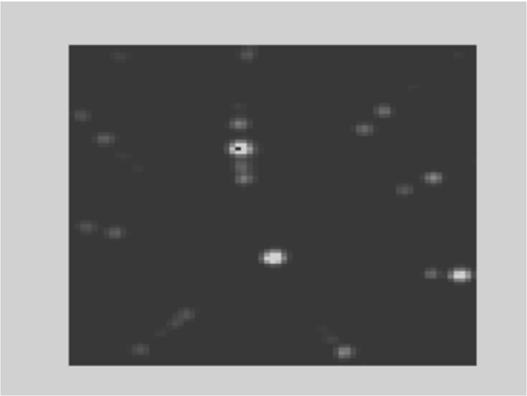
- **HVQ** (Luttrell 1989)
- **HSOM** (Miikkulainen 1990), **DISLEX** (1990, 1997)
- **PSOM** (Ritter 1993), **Hyperbolic SOM** (1999), **H$^2$SOM**
- **Temporal Kohonen Map** (Chappell &Taylor 1993)
- **Neural Gas**(Martinetz et al.1991) **Growing Grid**(Fritzke1995)
- **ASSOM** (Kohonen 1997)
- **Recurrent SOM** (Koskela, 1997)
- **Bayesian SOM** & **SOMN** (Yin&Allinson 1995,1997; Utsugi 1997)
- **GTM** (Bishop et al. 1998)
- **GHSOM** (Merkl et al. 2000)
- **PicSOM** (Laaksonen, Oja, et al., 2000)
- **ViSOM** (Yin 2001, 2002)

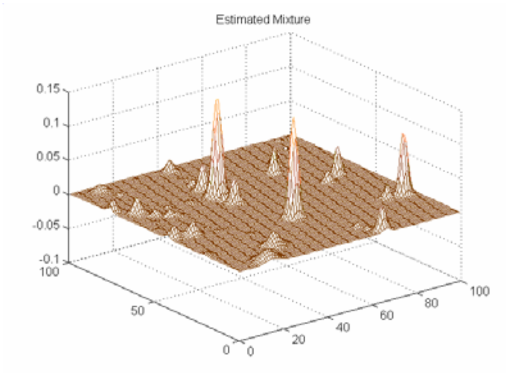
MANCHESTER 1824

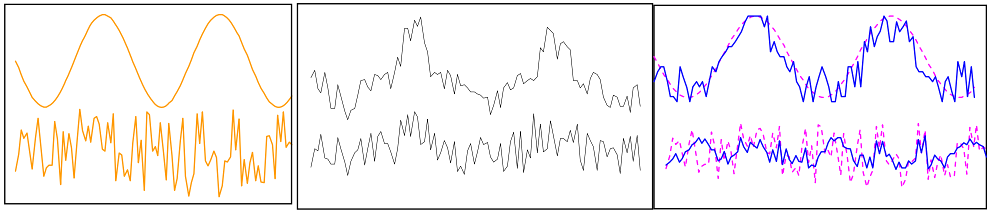
The University of Manchester

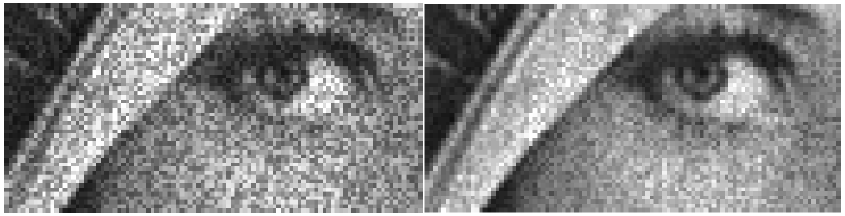
# 2. SOM: Applications

## ***SOM**: Applications -Snapshots*







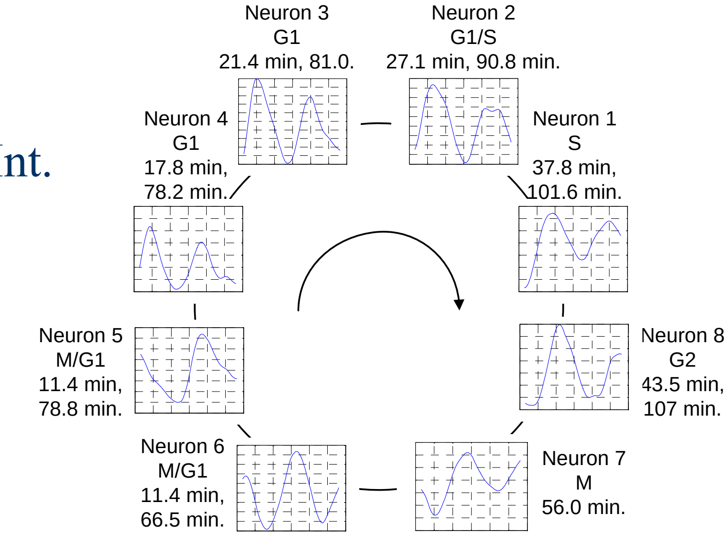
# 2. SOM: Applications

## ***SOM**: Applications -Snapshots*

A Temporal Shape Metric :

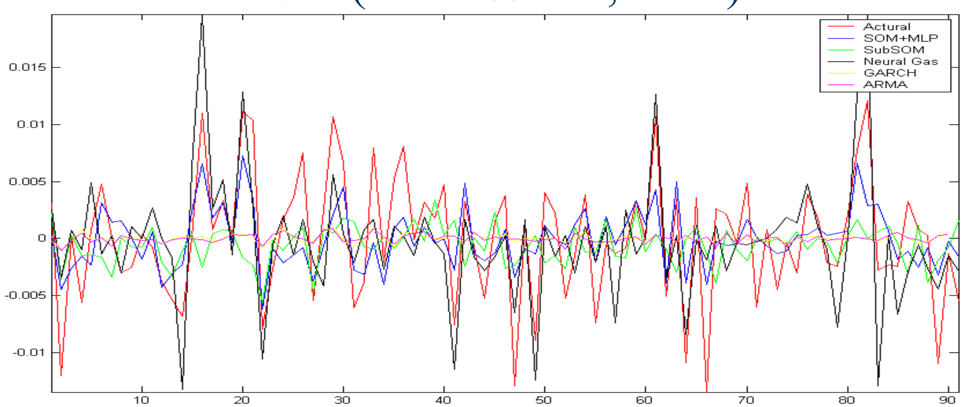
***Co-Expression Coefficient*** (Möller-Levet & Yin, Int. J. Neural Systems, 15: 311-322, 2005)

$$ce(x,y) = \frac{\int x' y' dt}{\sqrt{\int x'^2 dt \int y'^2 dt}}$$




Foreign exchange modelling :

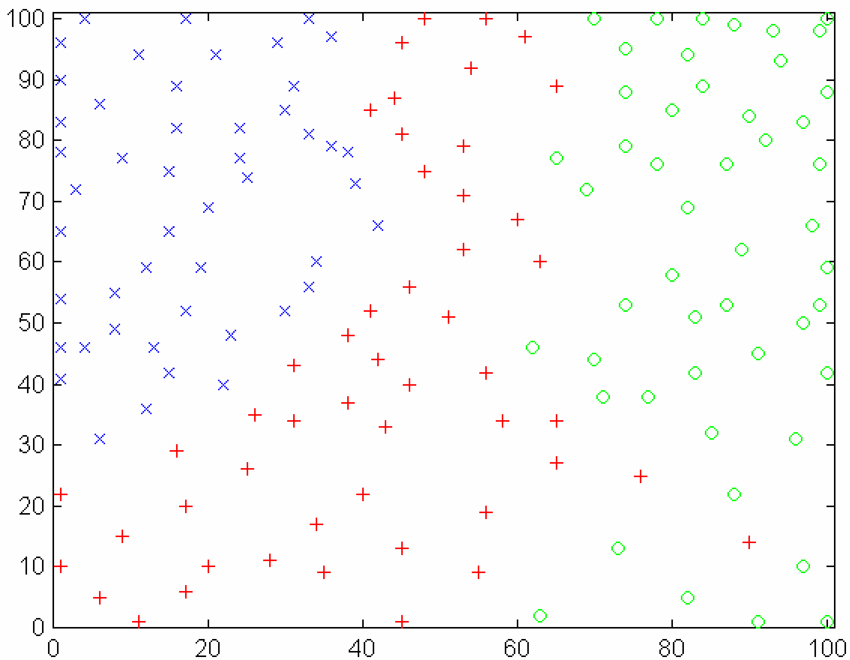
*SOM+local SVM* (H. Ni & Yin, 2006)

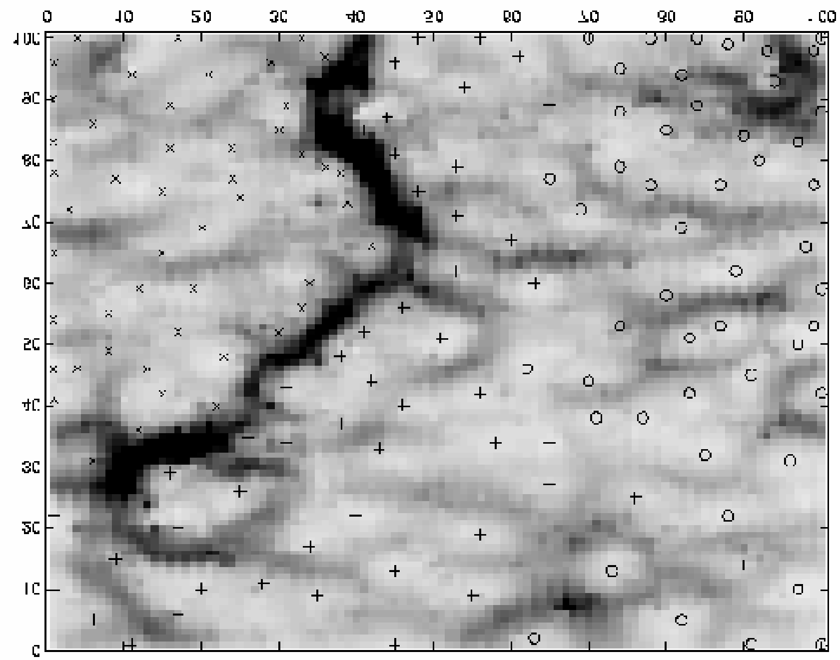



| | SOM+MLP | HSOM | Neural Gas | GARCH | ARMA |
|---|---|---|---|---|---|
| Mean Square Error (e-005) | 2.05 | 4.11 | 2.65 | 2.90 | 2.94 |
| Correct Prediction (%) | 73.62 | 50.55 | 65.38 | 51.11 | 52.2 |

# 2. SOM: Data Visualisation

## ***SOM**: Data Visualisation – Dimensionality Reduction*



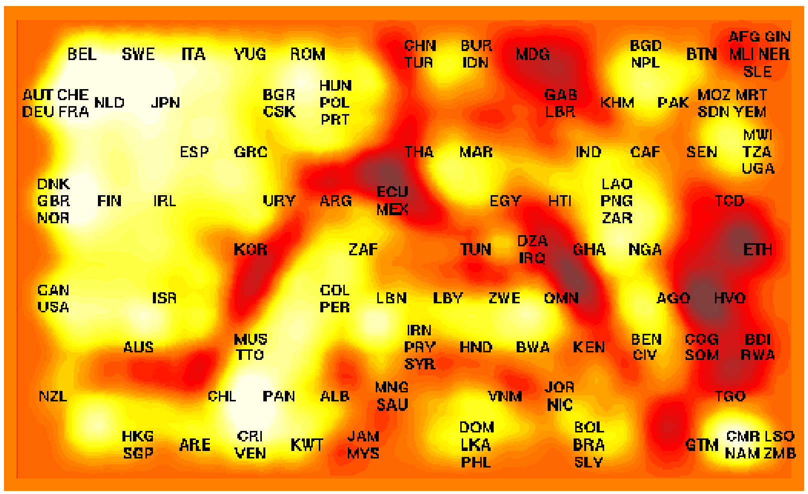
- ☺ *topology preserving mapping*
- ☺ *(discrete) mapping function*
- ☹ *non distance preserving*
- ☹ *boundary problems*

MANCHESTER 1824

The University of Manchester

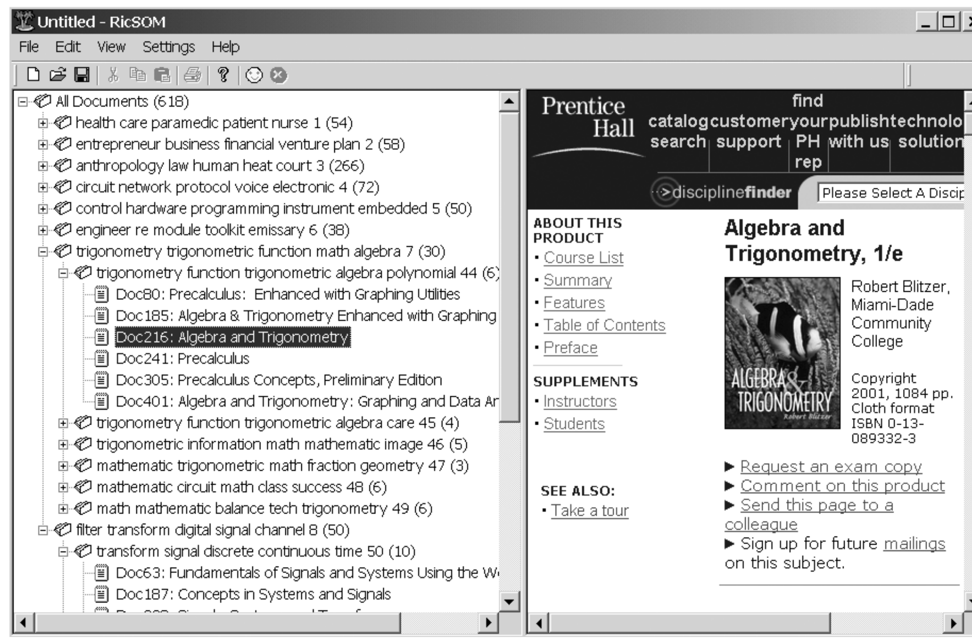
# 2. SOM: Data Visualisation

## ***SOM**: Data Visualisation –Knowledge Management*

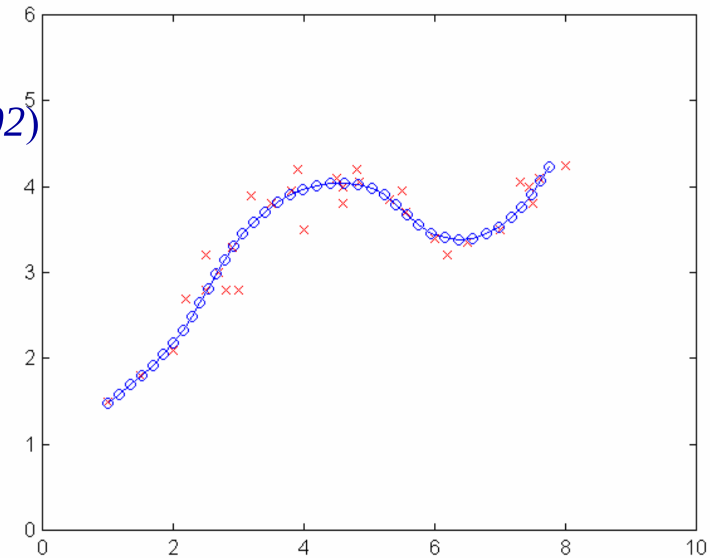



*courtesy of S. Kaski and T. Kohonen*

### *Tree-View SOM*



# 3. ViSOM & Principal Curve/Surface




## *ViSOM: Visualisation induced SOM*

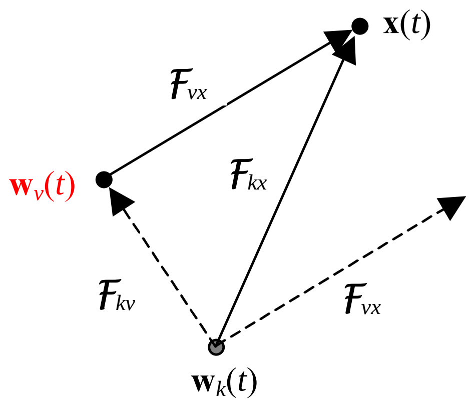
*(Yin, IEEE Trans Neural Networks,13: 237-243, 2002)*

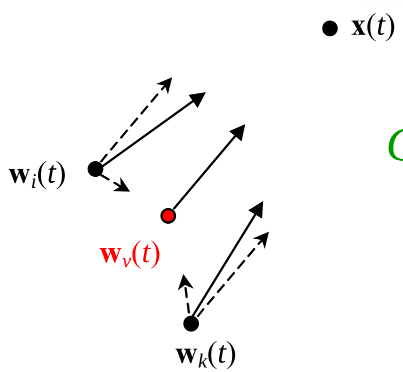
- To preserve distance/metric on the map
- To extrapolate smoothly

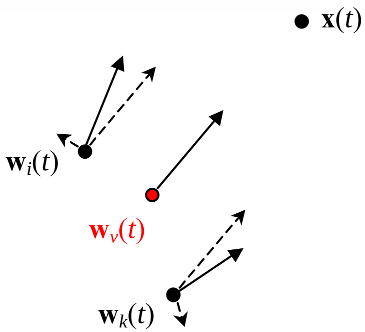
## *Principle*

SOM update:

$$\mathbf{w}_k(t+1) = \mathbf{w}_k(t) + \alpha(t)\eta(v,k,t)[\mathbf{x}(t) - \mathbf{w}_k(t)]$$




*Contraction*

*Expansion*

ViSOM

# 3. ViSOM & Principal Curve/Surface

## *ViSOM: Algorithm*

- *Grid structure and winner selection same to SOM*

- *Updating*

$$\Delta \mathbf{w}_k(t) = \alpha(t)\eta(v,k,t)\left([\mathbf{x}(t)-\mathbf{w}_v(t)] + [\mathbf{w}_v(t)-\mathbf{w}_k(t)]\frac{(d_{vk}-\Delta_{vk}\lambda)}{\Delta_{vk}\lambda}\right)$$

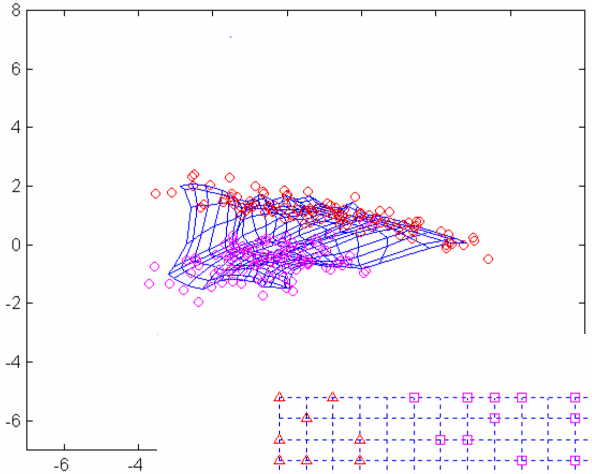
- *Refreshing*

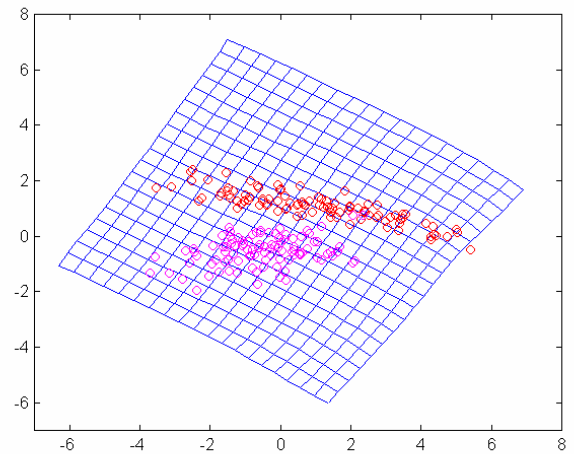
At certain iterations (e.g. 20%), choosing a neuron randomly and using its weight as an alternative input.

$$\Delta \mathbf{w}_k = \mathbf{w}_k(t) + \alpha(t)\eta(v,k,t)\left([\mathbf{x}(t)-\mathbf{w}_v(t)] + [\xi + (1-\xi)(\frac{d_{vk}}{\Delta_{vk}\lambda} - 1)][\mathbf{w}_v(t)-\mathbf{w}_k(t)]\right)$$

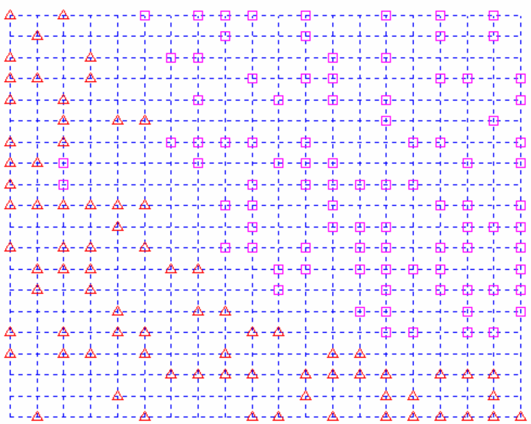
# 3. ViSOM & Principal Curve/Surface

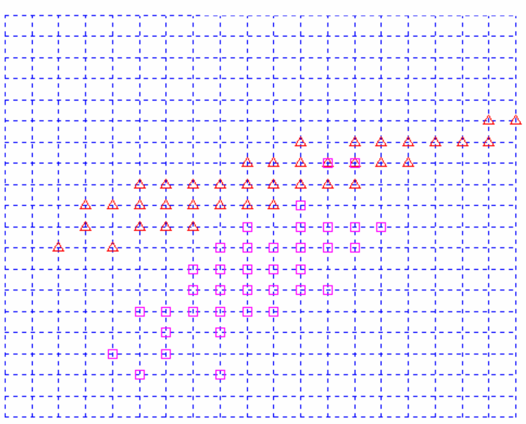
## ***ViSOM**: Examples*





SOM





ViSOM

# 3. ViSOM & Principal Curve/Surface

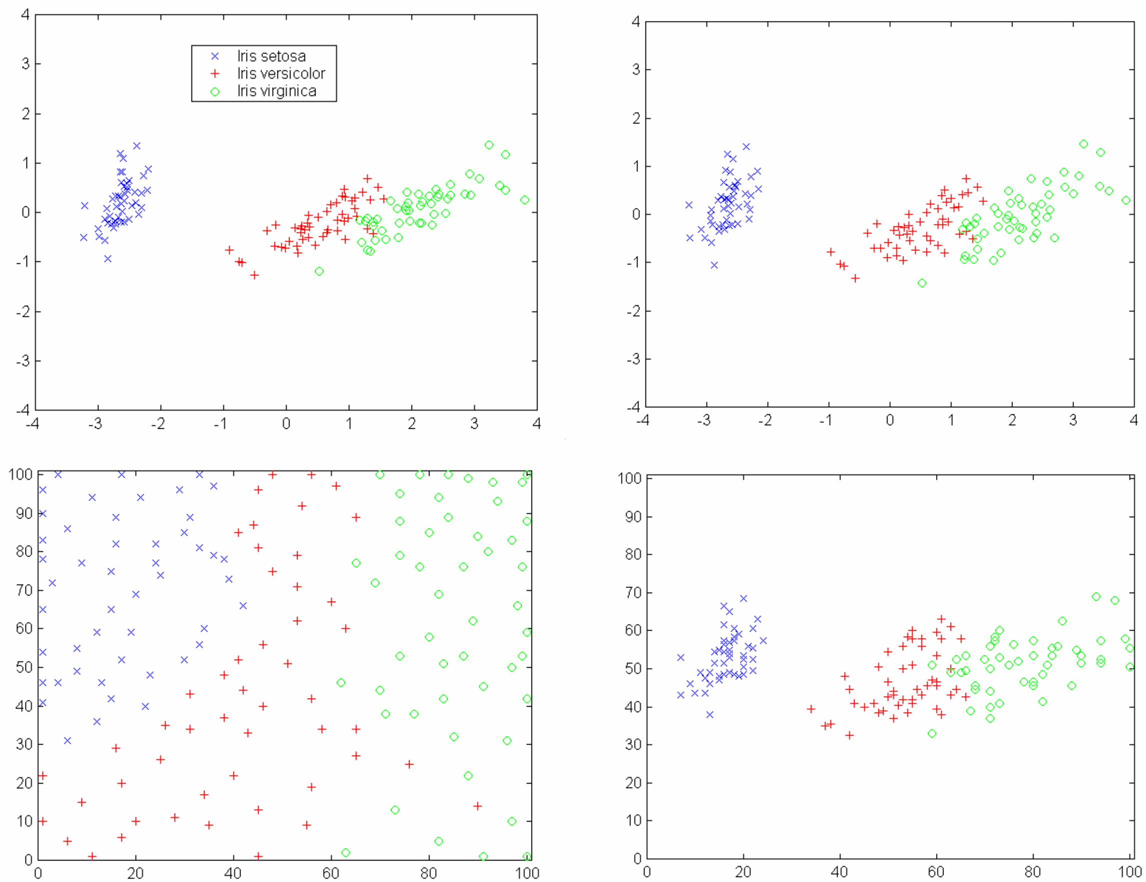
## *ViSOM: Examples*

PCA

Sammon

SOM

ViSOM



# 3. ViSOM & Principal Curve/Surface

## ***ViSOM***: *Examples*

## *Ranking table of UK universities*

*- source: The Sunday Times, 18 September 2000*

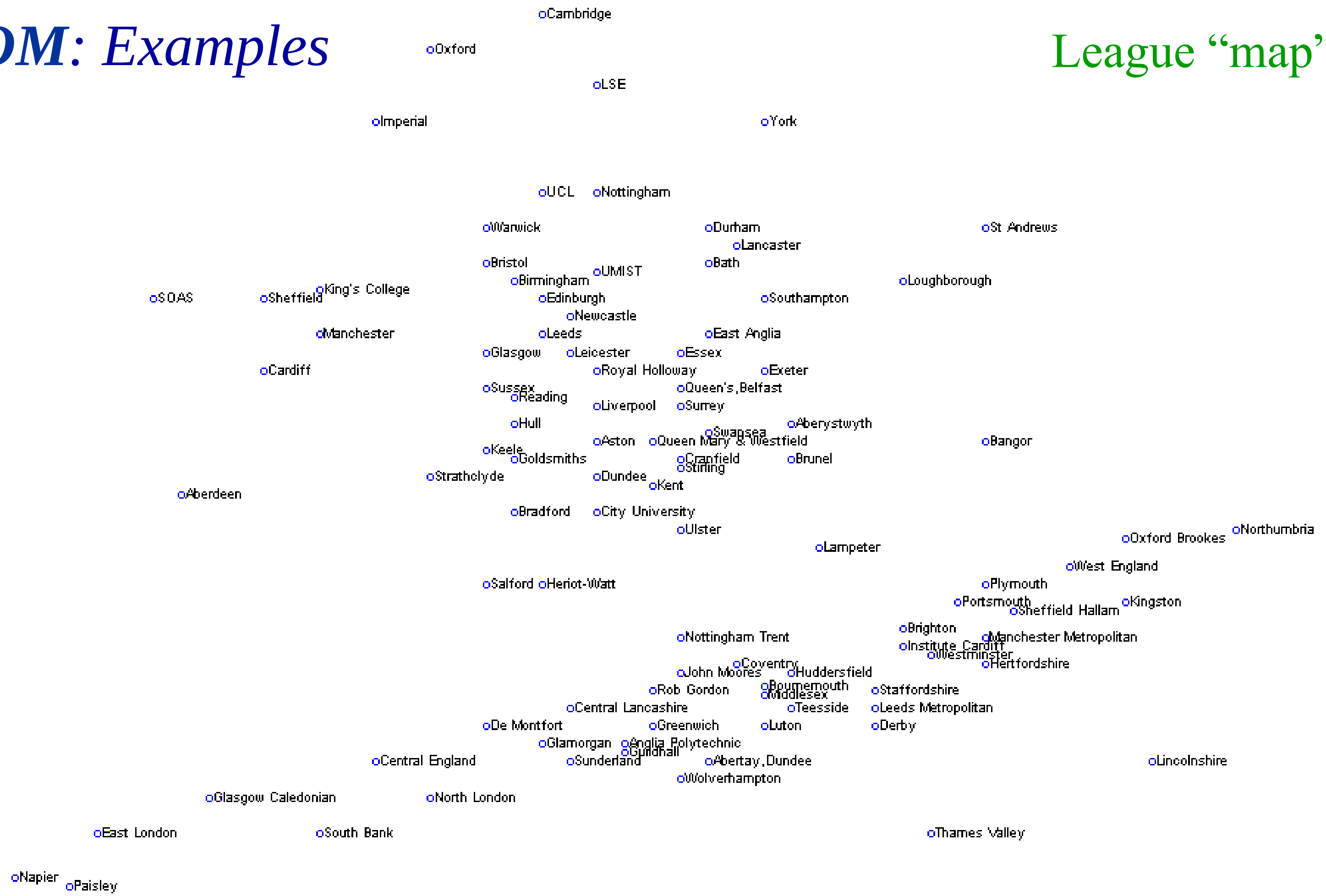
| *Ranking* | *University* | *F1* | *F2* | *F3* | *F4* | *F5* | *F6* | *F7* | ***Total*** |
|---|---|---|---|---|---|---|---|---|---|
| 1 | Cambridge | 241 | 182 | 247 | 97 | 88 | 100 | 50 | **1005** |
| 2 | Oxford | 214 | 175 | 244 | 97 | 81 | 100 | 30 | **941** |
| 3 | LSE | 200 | 175 | 233 | 97 | 68 | 100 | 50 | **923** |
| 4 | Imperial | 203 | 154 | 232 | 98 | 67 | 100 | 10 | **864** |
| 5 | York | 206 | 143 | 208 | 94 | 63 | 76 | 60 | **850** |
| 6 | UCL | 172 | 152 | 210 | 95 | 71 | 100 | 30 | **830** |
| 7 | St Andrews | 139 | 131 | 194 | 96 | 73 | 91 | 100 | **824** |
| 8 | Warwick | 153 | 155 | 215 | 97 | 69 | 86 | 20 | **795** |
| 9 | Bath | 132 | 142 | 211 | 97 | 66 | 83 | 60 | **791** |
| 9 | Nottingham | 176 | 125 | 218 | 96 | 74 | 72 | 30 | **791** |
| 11 | Bristol | 145 | 131 | 218 | 96 | 75 | 94 | 20 | **779** |
| 11 | Durham | 163 | 132 | 207 | 91 | 64 | 72 | 50 | **779** |
| 11 | Edinburg | 106 | 145 | 218 | 96 | 74 | 100 | 40 | **779** |
| 14 | Lancaster | 156 | 144 | 186 | 95 | 62 | 63 | 50 | **756** |
| 15 | UMIST | 135 | 144 | 188 | 97 | 58 | 100 | 30 | **752** |
| 16 | Birmingham | 146 | 127 | 204 | 96 | 67 | 87 | 20 | **747** |
| 17 | Loughborough | 162 | 115 | 177 | 95 | 57 | 66 | 60 | **732** |
| 18 | Southampton | 143 | 124 | 180 | 93 | 55 | 71 | 50 | **716** |
| 19 | King's College | 135 | 126 | 204 | 96 | 63 | 100 | -10 | **714** |
| 20 | Newcastle | 134 | 117 | 193 | 97 | 60 | 87 | 20 | **708** |
| 21 | Manchester | 125 | 134 | 198 | 96 | 66 | 98 | -10 | **707** |
| 22 | Leeds | 122 | 127 | 199 | 97 | 61 | 74 | 20 | **700** |
| 23 | Sheffield | 143 | 125 | 213 | 97 | 61 | 72 | -20 | **691** |
| 24 | East Anglia | 125 | 127 | 176 | 96 | 63 | 60 | 40 | **687** |
| 24 | Leicester | 125 | 120 | 183 | 94 | 52 | 93 | 20 | **687** |

*F1:Research*
*F2:Teaching*
*F3:A-levels*
*F4:Employment*
*F5:S/S ratio*
*F6:1st/2:1s*
*F7:Dropout rate*

# 3. ViSOM & Principal Curve/Surface

## ***ViSOM**: Examples*

League "map"



The University of Manchester

MANCHESTER 1824

# 3. ViSOM & Principal Curve/Surface

## ***ViSOM**: A Discrete Principal Curve/Surface*

*(Yin, Neural Networks, 15: 1005-1016, 2002)*

Projection:

$$\rho_f(\mathbf{x}) = \sup_{\rho \in \Lambda} \{\rho : \|\mathbf{x} - f(\rho)\| = \inf_{\vartheta} \|\mathbf{x} - f(\vartheta)\|$$

Expectation:

$$f(\rho) = E[\mathbf{X} \mid \rho_f(\mathbf{X}) = \rho]$$

Kernel smoothing:

$$F(\rho) = \frac{\sum_i^S \mathbf{x}_i \kappa(\rho, \rho_i)}{\sum_i^S \kappa(\rho, \rho_i)}$$

SOM/ViSOM smoothing:

$$\mathbf{w}_k = \frac{\sum_i^S \mathbf{x}_i h(k,i)}{\sum_i^S h(k,i)}$$

SOM: $||k\text{-}i|| \neq ||\mathbf{w}_k\text{-}\mathbf{w}_i||$

ViSOM: $||k\text{-}i|| \approx ||\mathbf{w}_k\text{-}\mathbf{w}_i||$

The University of Manchester

MANCHESTER 1824

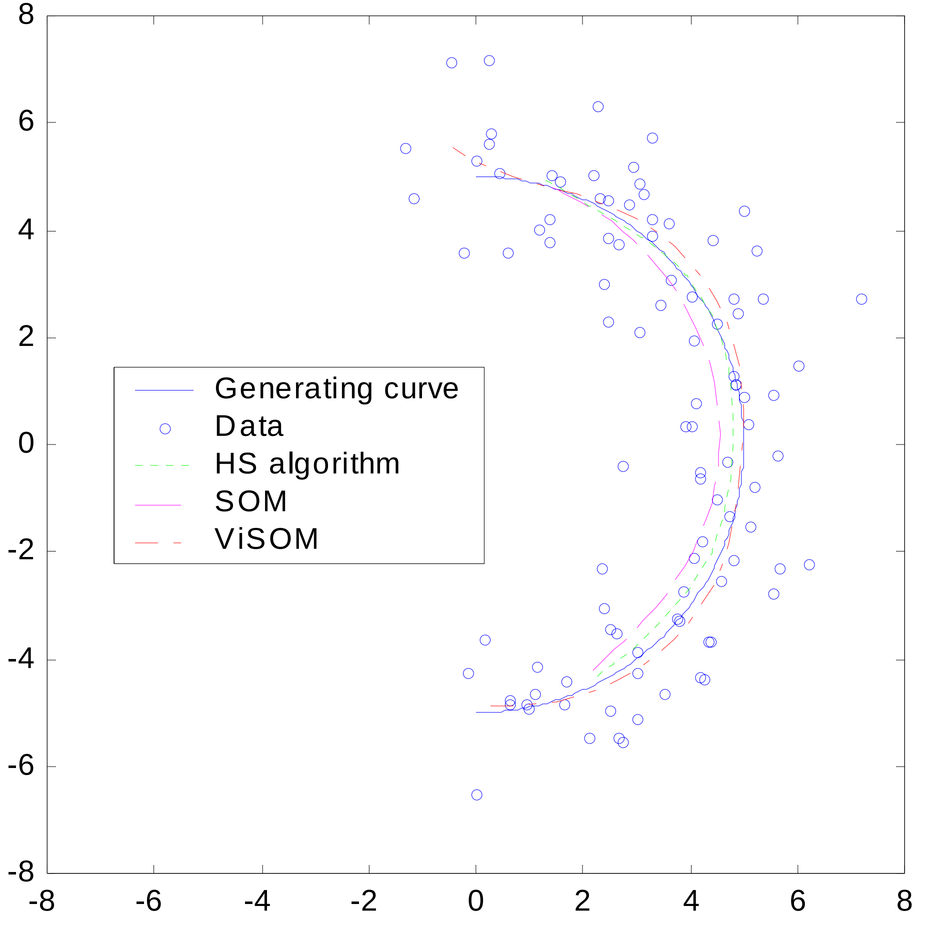
# 3. ViSOM & Principal Curve/Surface

***ViSOM**: STVQ (Graeple, Burger&Obermayer,Phys. Rev. E 1997)*
*+ViSOM → PRSOM(Wu&Chow IEEE-TNN 16(6),2005)*

$$w_j(t+1) = w_j(t)+\varepsilon(t)p_j'\left(x(t)\right)\left[\sum_{i=1}^{N} p_i\left(x(t)\right)\left([x(t)-w_i(t)] + \gamma\left[w_i(t)-w_j(t)\right]\left(\frac{d_{ij}^2-\lambda\Delta_{ij}^2}{\lambda\Delta_{ij}^2+I_{ij}}\right)\right)\right].$$

$$E = F_{vq}+\gamma F_{reg} = \frac{1}{2}\sum_{t=1}^{M}\left\|\sum_{j=1}^{N} p_j\left(x(t)\right)\left[x(t)-w_j\right]\right\|^2 + \frac{\gamma}{8}\sum_{t=1}^{M}\sum_{j=1}^{N}\sum_{m=1}^{N} p_j\left(x(t)\right)p_m\left(x(t)\right)\frac{\left(d_{jm}^2-\lambda\Delta_{jm}^2\right)^2}{\left(\lambda\Delta_{jm}^2+I_{jm}\right)}$$

# 3. ViSOM & Principal Curve/Surface



# 3. ViSOM & Principal Curve/Surface

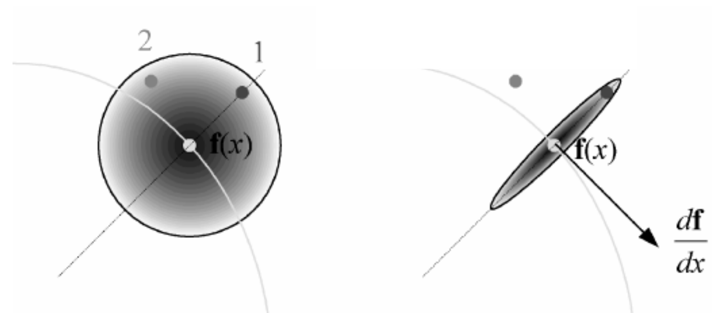
## *Other PC/S algorithms:*

- **SOM** has been related to PC/S and termed discrete PC/S by *Ritter, Martinetz & Schulten in 1992*. However, the differences are:
  - Projection onto nodes instead of curve/surface
  - Smoothing is governed by indexes in the map space, not the input space

$$\mathbf{w}_k = \frac{\sum_i^S \mathbf{x}_i h(k,i)}{\sum_i^S h(k,i)} \qquad F(\rho) = \frac{\sum_i^S \mathbf{x}_i \kappa(\rho,\rho_i)}{\sum_i^S \kappa(\rho,\rho_i)}$$

SOM: $||k-i|| \neq ||\mathbf{w}_k - \mathbf{w}_i|| = ||\rho - \rho_i||$

ViSOM: $||k-i|| \approx ||\mathbf{w}_k - \mathbf{w}_i||$




More importantly for the SOM, one cannot get the curve/surface at any point other than the nodes, even with interpolations.

**GTM** (generative topographic mapping) and **PPS** (probabilistic principal surface) are parametrised SOMs with GTM using spherical and PPS oriented Gaussians for the nodes.

The University of Manchester

MANCHESTER 1824

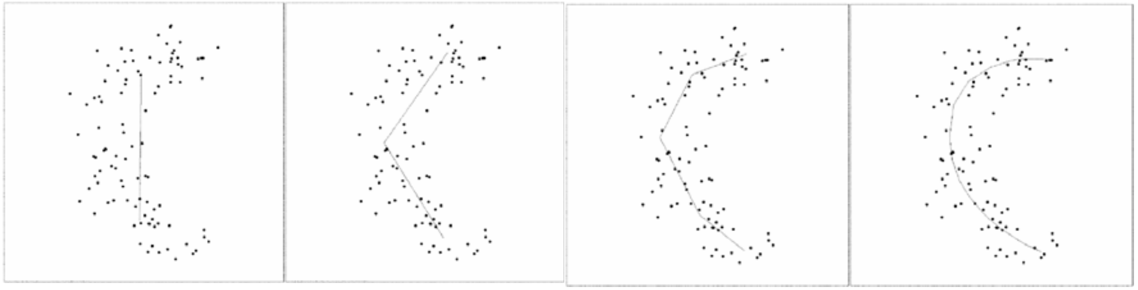
# 3. ViSOM & Principal Curve/Surface

## *Other PC/S algorithms:*

- **Polygonal Algorithm:** proposed by *Kégl, et al 1999* for incrementally constructing PC:
  - Consist of (connected) line segments and vertexes with total length constant.
  - The number of segments or vertexes is increasing to a certain level.

$$\Delta(\mathbf{x}, f) = \min_{\rho} \| \mathbf{x} - f(\rho) \|^2 \qquad F = \arg\min_{f \in S} \{\frac{1}{n}\sum_{i=1}^{n} \Delta(\mathbf{x}_i, f)\}$$

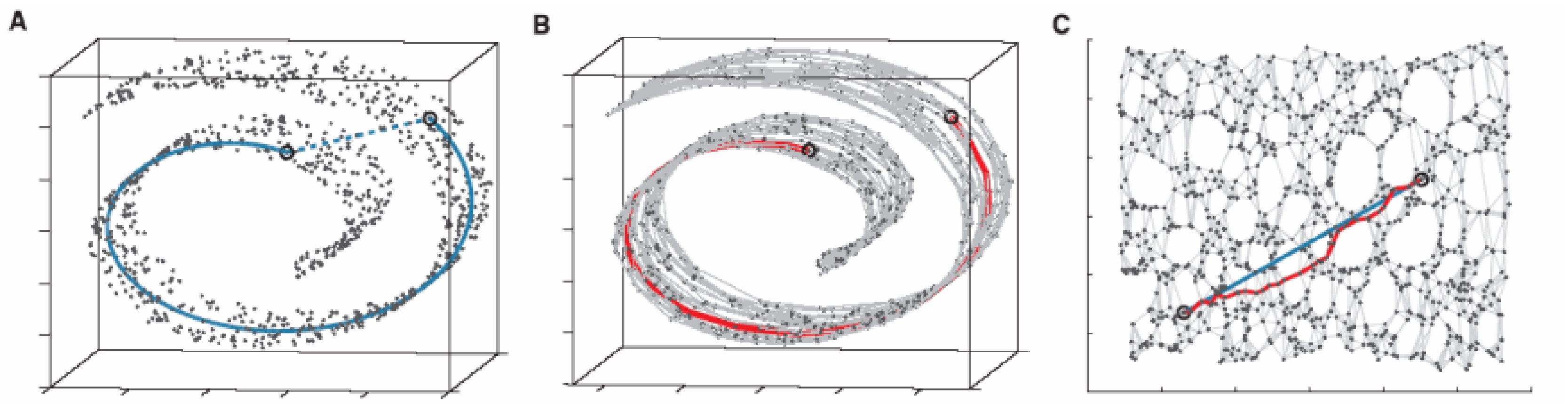
  - Projection (most data points) on segments instead of nodes (vertexes).
  - New vertex is added to the longest segment (middle point).



# 3. ViSOM & Principal Curve/Surface

## *Other PC/S algorithms:*

- **Isomap:** proposed by *Tenenbaum, Silva and Langford 2000* for nonlinear dimensionality reduction.
  - **Construct neighbourhood graph:** by $d_X(i,j)<\varepsilon$ or $K$ nearest neighbours.
  - **Compute the shortest (geodesic) paths:** $\min\{d_G(i,j),\ d_G(i,k)+d_G(k,j)\}$.
  - **Construct low dimension embedding:** by applying MDS,



# 3. ViSOM & Principal Curve/Surface

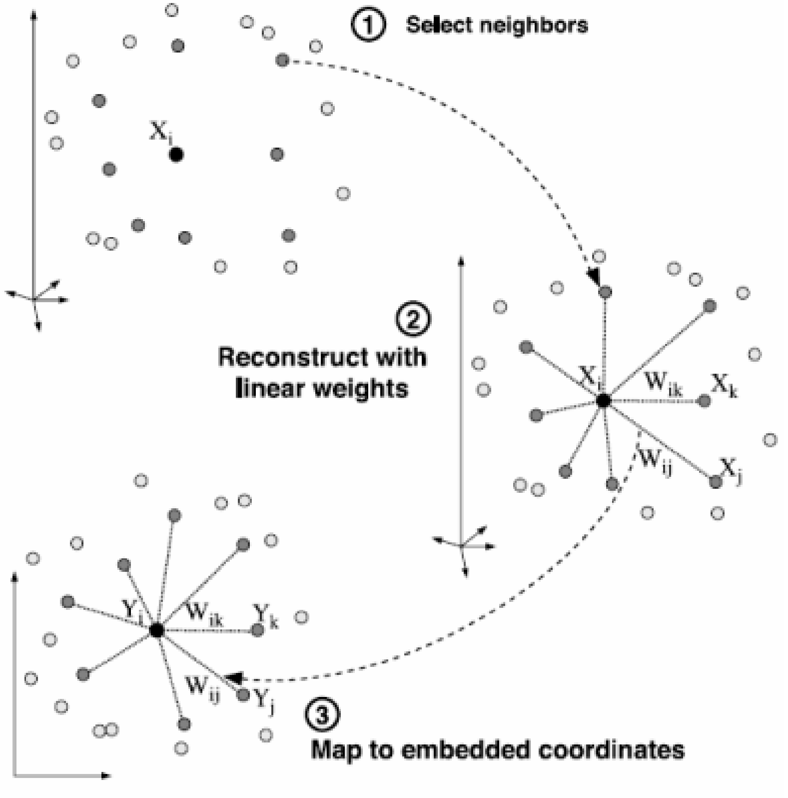
## *Other PC/S algorithms:*

- **Local Linear Embedding:** proposed by *Roweis and Saul 2000* also for dimensionality reduction.
  - **Select neighbourhood graph:**

    *K* nearest neighbours.
  - **Reconstruct linear weights:**

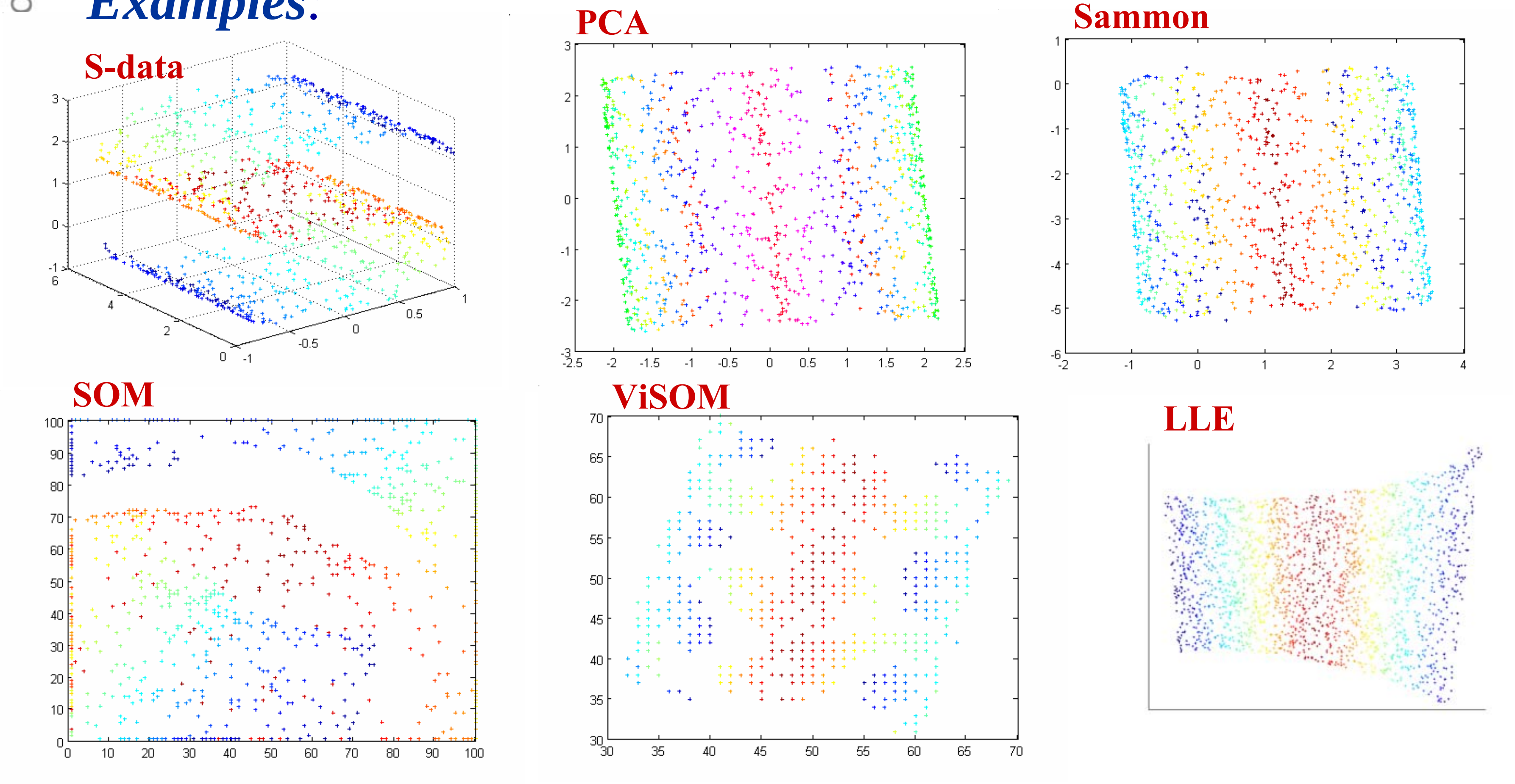
    $$\varepsilon(W) = \min \sum_i \| X_i - \sum_j W_{ij} X_j \|^2$$
  - **Compute embedding coordinates Y:**

    $$\Phi(Y) = \min \sum_i \| Y_i - \sum_j W_{ij} Y_j \|^2$$



# 3. ViSOM & Principal Curve/Surface

***Examples:***



MANCHESTER 1824

The University of Manchester

# 4. Kernel SOM & Mixture Model

## ***Kernel SOM**: Background*

- Kernel method has become popular.

$$\phi : X \rightarrow F, \qquad \mathbf{x} \mapsto \phi(\mathbf{x})$$

$$\kappa : X \times X \in \Re, \qquad \kappa(\mathbf{x};\mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle$$

- PCA

$$\mathbf{Cq} = \lambda \mathbf{q}, \qquad \mathbf{C} = \frac{1}{n}\sum_i \mathbf{x}_i \mathbf{x}_i^T, \quad \mathbf{q} = \sum_i \alpha_i \mathbf{x}_i,$$

- Kernel PCA

$$\mathbf{K}\boldsymbol{\alpha} = \lambda \boldsymbol{\alpha}, \qquad K_{ij} := \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle, \qquad \boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots\ldots \alpha_n]^T,$$

$$\mathbf{q} = \sum_i \alpha_i \phi(\mathbf{x}_i), \qquad \langle \phi(\mathbf{x}_k), \mathbf{q} \rangle = \sum_i \alpha_i \kappa(\mathbf{x}_k, \mathbf{x}_i),$$

# 4. Kernel SOM & Mixture Model

## *KM-Kernel SOM (MacDonald & Fyfe 2000):*

$$\phi : \mathbf{x} \to F \qquad \mathbf{x} \mapsto \phi(\mathbf{x}), \qquad \mathbf{m}_i = \sum_n \alpha_{i,n}\phi(\mathbf{x}_n),$$

$$\|\phi(\mathbf{x}) - \mathbf{m}_i\|^2 = \|\phi(\mathbf{x}) - \sum_n \alpha_{i,n}\phi(\mathbf{x}_n)\|^2$$

$$= \kappa(\mathbf{x},\mathbf{x}) - 2\sum_n \alpha_{i,n}\kappa(\mathbf{x},\mathbf{x}_n) + \sum_{n,m}\alpha_{i,n}\alpha_{i,m}\kappa(\mathbf{x}_n,\mathbf{x}_m)$$

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \Lambda[\phi(\mathbf{x}) - \mathbf{m}_i(t)], \qquad \Lambda = \frac{\zeta_{i(\mathbf{x}),j}}{\sum_{n=1}^{t+1}\zeta_{i,n}}$$

$$\alpha_{i,n}(t+1) = \begin{cases} \alpha_{i,n}(t)(1-\Lambda), & \text{for } n \neq t+1 \\ \zeta, & \text{for } n = t+1 \end{cases}$$

# 4. Kernel SOM & Mixture Model

## *GD-Kernel SOM (Andras 2002; Pan et al. 2004):*

$$v = \arg\min_i \|\mathbf{x} - \mathbf{m}_i\|^2 \qquad v = \arg\min_i \|\phi(\mathbf{x}) - \phi(\mathbf{m}_i)\|^2$$

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t) h(v(\mathbf{x}), i) \nabla J(\mathbf{x}, \mathbf{m}_i)$$

$$J(\mathbf{x}, \mathbf{m}_i) = \|\phi(\mathbf{x}) - \phi(\mathbf{m}_i)\|^2 = \kappa(\mathbf{x}, \mathbf{x}) + \kappa(\mathbf{m}_i, \mathbf{m}_i) - 2\kappa(\mathbf{x}, \mathbf{m}_i)$$

$$\nabla J(\mathbf{x}, \mathbf{m}_i) = \frac{\partial \kappa(\mathbf{m}_i, \mathbf{m}_i)}{\partial \mathbf{m}_i} - 2\frac{\partial \kappa(\mathbf{x}, \mathbf{m}_i)}{\partial \mathbf{m}_i}$$

$$v = \arg\min_i J(\mathbf{x}, \mathbf{m}_i) = \arg\min_i [-2\kappa(\mathbf{x}, \mathbf{m}_i)] = \arg\min_i [-\exp(-\frac{\|\mathbf{x} - \mathbf{m}_i\|^2}{2\sigma^2})]$$

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t) h(v(\mathbf{x}), i) \frac{1}{2\sigma^2} \exp(-\frac{\|\mathbf{x} - \mathbf{m}_i\|^2}{2\sigma^2})(\mathbf{x} - \mathbf{m}_i)$$

# 4. Kernel SOM & Mixture Model

## *Kernel SOM:*

**Table**: *Classification errors on UCI colon cancer dataset. M, A and V denote the minimum distance, average distance and majority voting methods to label the nodes.*

| *Kernel* | *Type I Kernel SOM* | | | *Type II Kernel SOM* | | | *SOM* | | |
|---|---|---|---|---|---|---|---|---|---|
| | M | A | V | M | A | V | M | A | V |
| Gaussian | 5.6 | 5.8 | 5.6 | 5.3 | 5.3 | 5.7 | 4.3 | 7.0 | 3.8 |
| Cauchy | 5.5 | 5.6 | 5.5 | 5.5 | 5.5 | 4.8 | | | |
| Log | 4.6 | 4.6 | 4.6 | 5.2 | 5.2 | 4.6 | | | |

# 4. Kernel SOM & Mixture Model

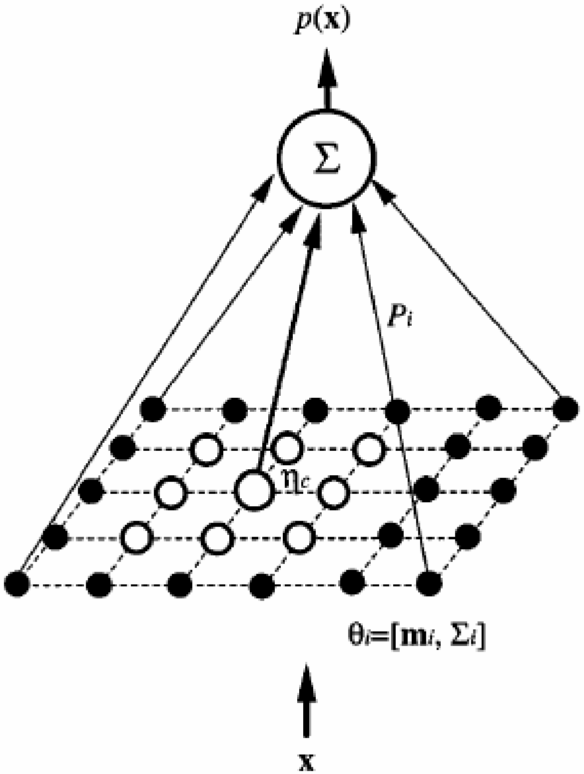
## *Mixture Model:*

$$p(\mathbf{x}\,|\,\Theta) = \sum_{i=1}^{K} p_i(\mathbf{x}\,|\,\theta_i)P_i$$

Kullback-Leibner divergence:

$$\mathbf{I} = -\int \log \frac{\hat{p}(\mathbf{x})}{p(\mathbf{x})} p(\mathbf{x}) d\mathbf{x}$$

$$\frac{\partial \mathbf{I}}{\partial \theta_i} = -\int \left[\frac{1}{\hat{p}(\mathbf{x}\,|\,\hat{\Theta})} \frac{\partial \hat{p}(\mathbf{x}\,|\,\hat{\Theta})}{\partial \theta_i}\right] p(\mathbf{x}) d\mathbf{x}$$



# 4. Kernel SOM & Mixture Model

## *Self Organising Mixture Network*

*(Yin & Allinson IEEE Trans Neural Networks, 12:405-411, 2001)*

$$\hat{\theta}_i(t+1) = \hat{\theta}_i(t) + \alpha(t)h(v(\mathbf{x}),i)[\frac{1}{\hat{p}(\mathbf{x}\,|\,\hat{\Theta})}\frac{\partial\hat{p}(\mathbf{x}\,|\,\hat{\Theta})}{\partial\theta_i}]$$

$$= \hat{\theta}_i(t) + \alpha(t)h(v(\mathbf{x}),i)[\frac{\hat{P}_i(t)}{\sum_j \hat{P}_i(t)\hat{p}_j(\mathbf{x}\,|\,\theta_j)}\frac{\partial\hat{p}_i(\mathbf{x}\,|\,\hat{\theta}_i)}{\partial\theta_i}]$$

$$\hat{P}_i(t+1) = \hat{P}_i(t) + \alpha(t)[\frac{\hat{p}_i(\mathbf{x}\,|\,\hat{\theta}_i)\hat{P}_i(t)}{\hat{p}(\mathbf{x}\,|\,\hat{\Theta})} - \hat{P}_i(t)] = \hat{P}_i(t) - \alpha(t)h(v(\mathbf{x}),i)[\hat{P}(i\,|\,\mathbf{x}) - \hat{P}_i(t)]$$

$$v = \arg\max_i\{\hat{P}(i\,|\,\mathbf{x}) = \frac{\hat{P}_i\hat{p}_i(\mathbf{x}\,|\,\hat{\theta}_i)}{\hat{p}(\mathbf{x}\,|\,\hat{\Theta})}\}$$

# 4. Kernel SOM & Mixture Model

## *Self Organising Mixture Network:*

### Homoscedastic case

$$v = \arg\max_i \frac{\hat{p}_i(\mathbf{x}\,|\,\theta_i)}{\sum_j \hat{p}_i(\mathbf{x}\,|\,\theta_j)}$$

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t) h(v(\mathbf{x}), i) \frac{1}{\sum_j p_j(\mathbf{x}\,|\,\theta_j)} \frac{\partial p_i(\mathbf{x}\,|\,\theta_i)}{\partial \mathbf{m}_i}$$

# 4. Kernel SOM & Mixture Model

## *Self Organising Mixture Network:*

**Homoscedastic and Gaussian case**

$$v = \arg\max_i [\exp(-\frac{\|\mathbf{x}-\mathbf{m}_i\|^2}{2\sigma^2})]$$

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + \alpha(t) h(v(\mathbf{x}), i) \frac{1}{2\sigma^2} \frac{1}{\sum_j p_j(\mathbf{x}|\theta_j)} \exp(-\frac{\|\mathbf{x}-\mathbf{m}_i\|^2}{2\sigma^2})(\mathbf{x}-\mathbf{m}_i)$$

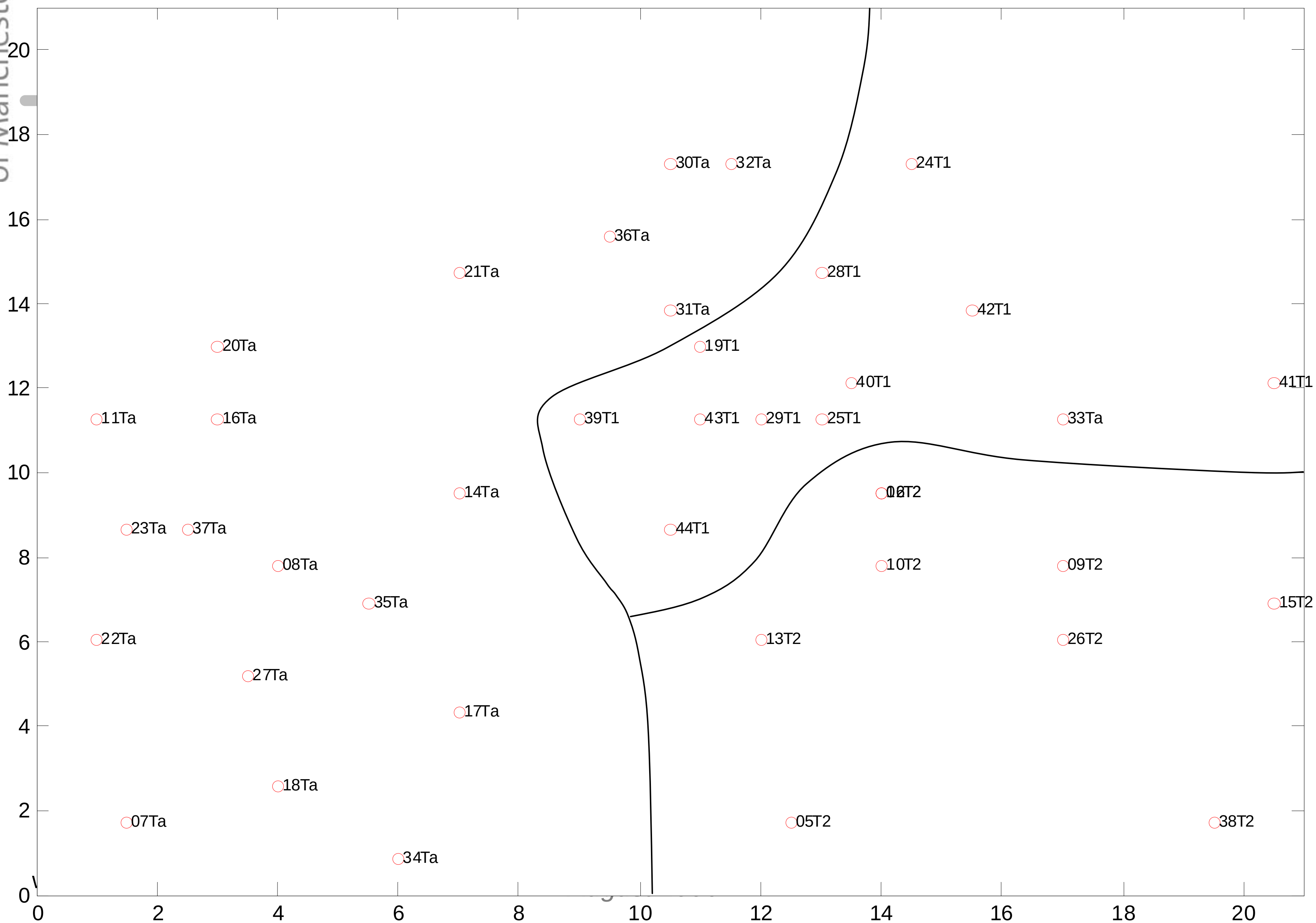
**The same as those of Kernel SOM !!**

(*Yin, Neural Networks, 19: 780-784, 2006*)

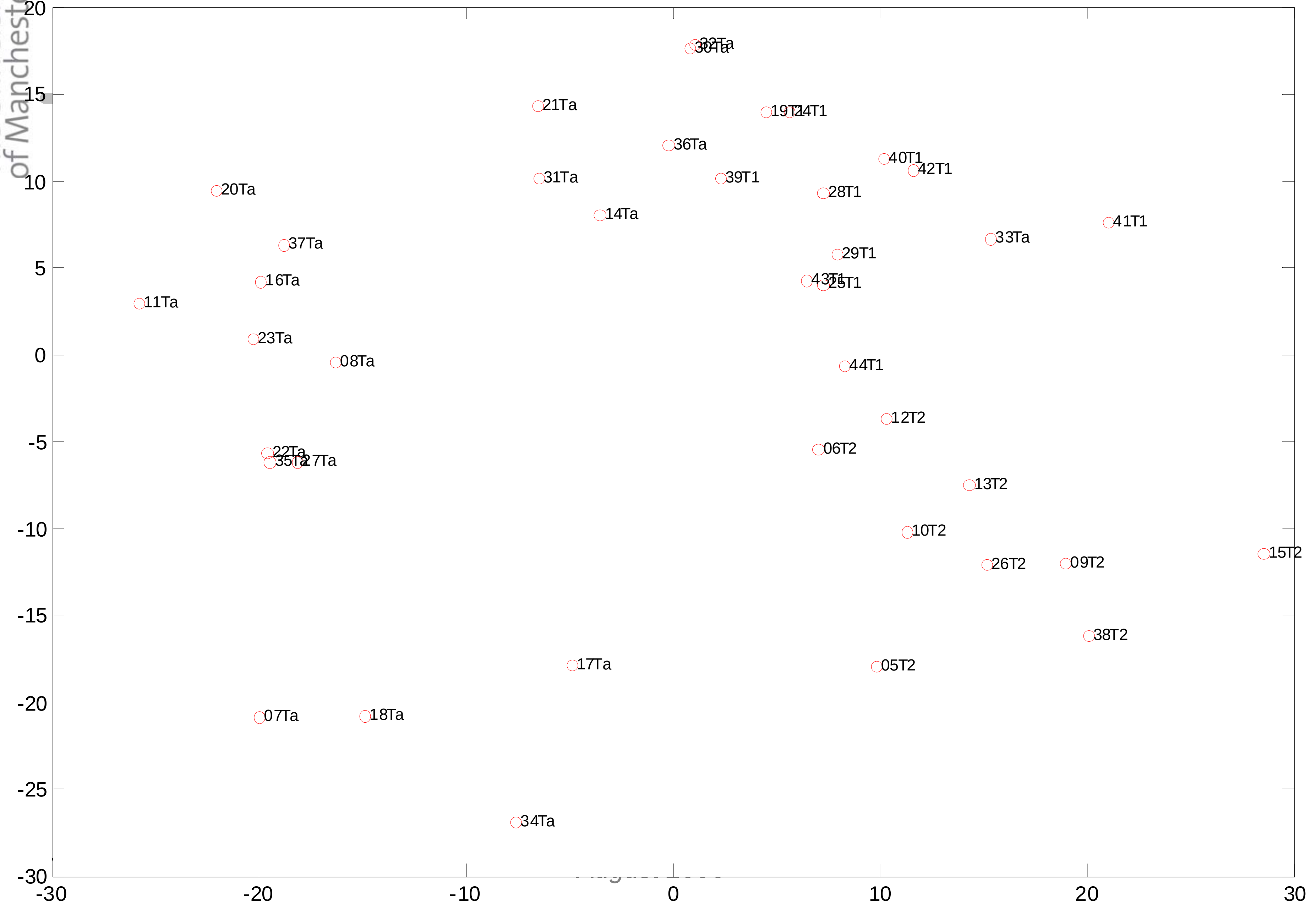
# 5. Summary

- SOMs are a useful tool for clustering and visualisation and management (organisation.
- ViSOM is particularly suited for direct data visualisation or manifold mapping where distance preserving (and topology) is important.
- Kernel SOM is linked to mixture model (probabilistic) and thus can outperform SOM in some cases when parameters are optimised.
- SOM approximates a natural kernel method.

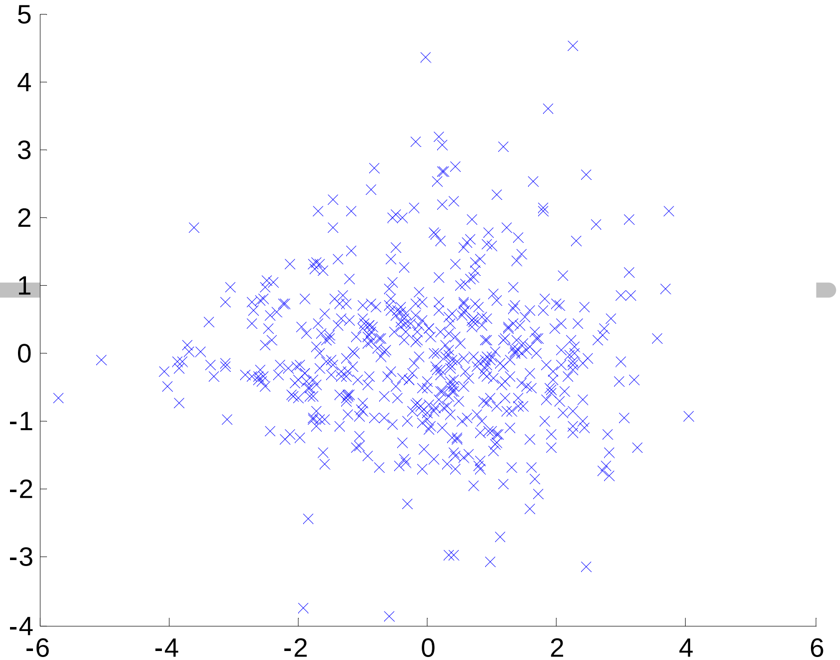
# Dataset II – samples: ViSOM

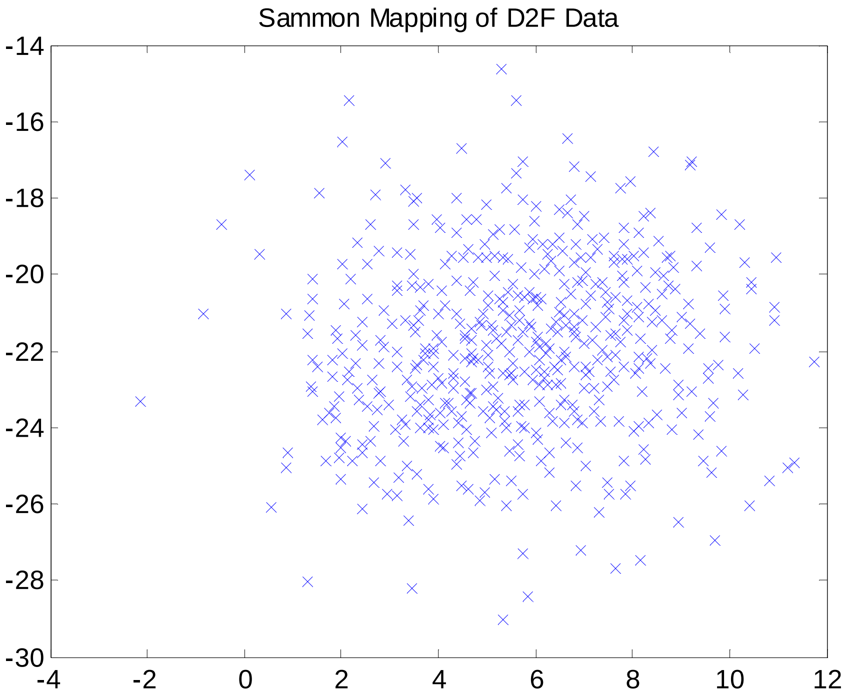


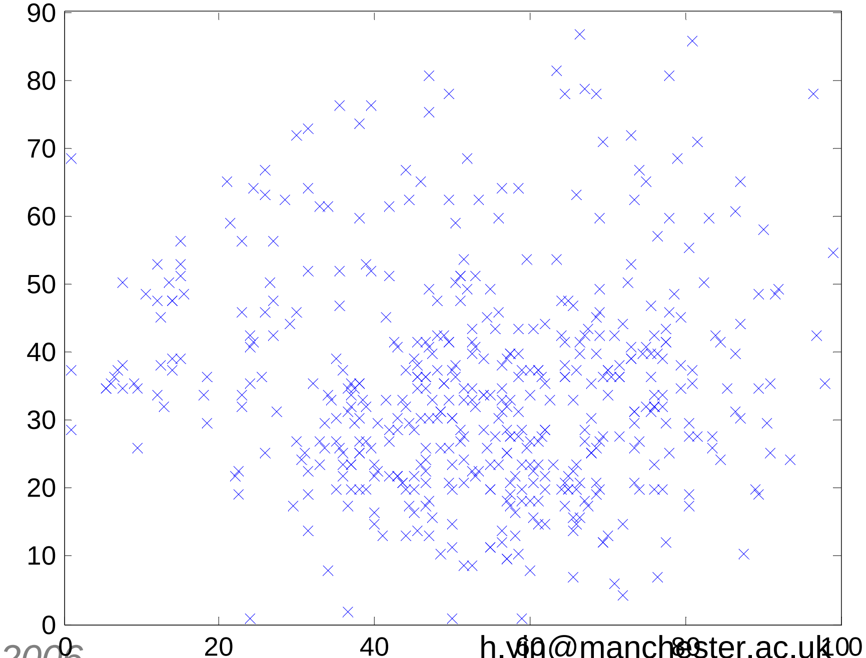
# Dataset II - samples: PCA



# Dataset II - 500 genes: PCA Sammon / ViSOM(100x100)



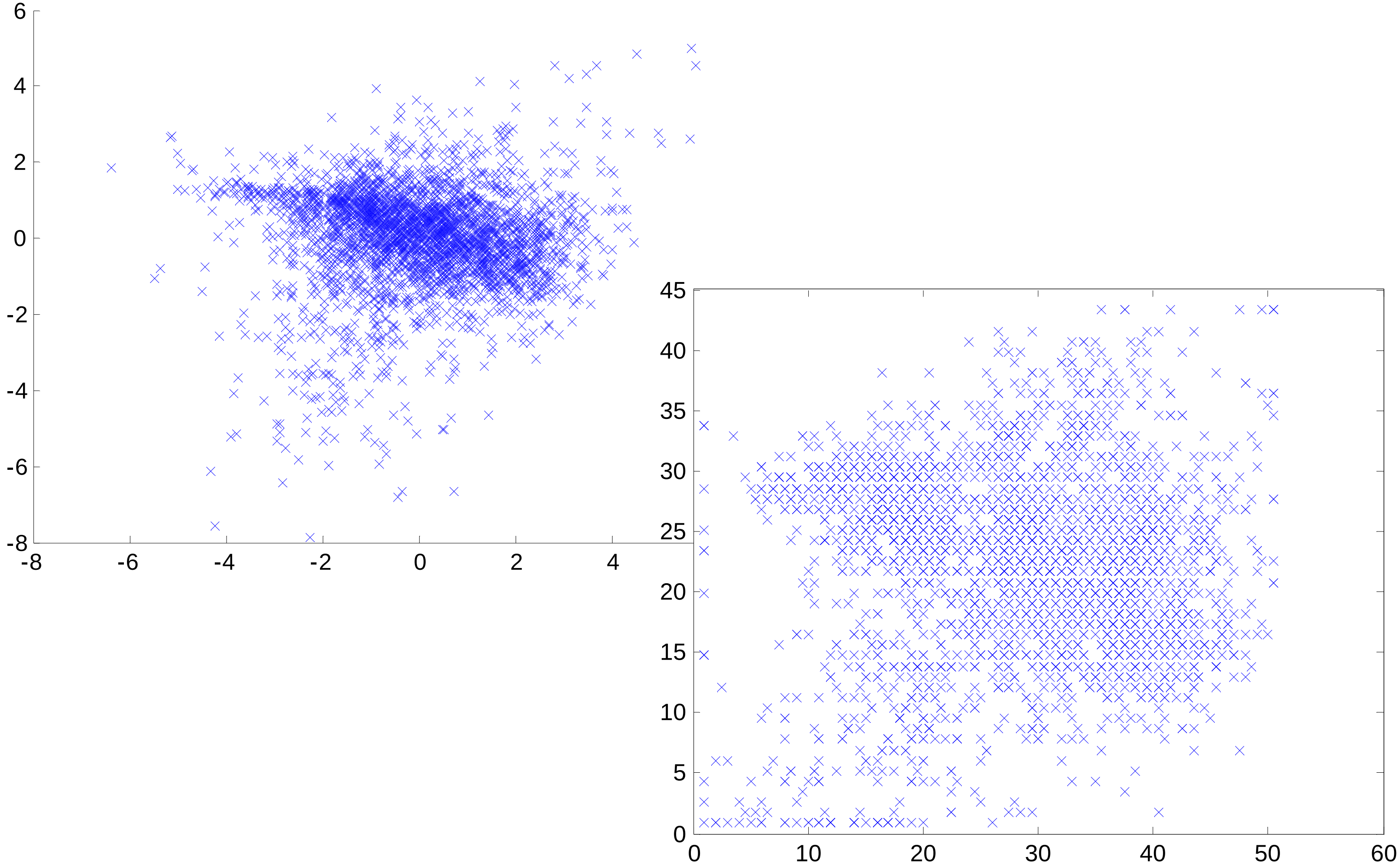


MANCHESTER 1824

The University of Manchester

# Dataset II - all genes: PCA/ ViSOM (50x50)



The University of Manchester

# Thank You!

Questions ?



h.yin@manchester.ac.uk