Abstract – Dimension reduction has long been associated with retinotopic mapping for understanding cortical maps and neural information processing. Multisensory information is perceived, propagated and mapped onto the 2-D cortex in a near-optimal information preserving manner. Data visualization, inspired by this mechanism, is playing an increasingly important role in many applications involving feature and data reduction, from biology, neuroscience, decision support, social science, to management science. The topic has also attracted a great deal of attention in computer vision and pattern recognition. Classic linear methods include principal component analysis (PCA), factor analysis, projection pursuit and independent component analysis. Recently there have been considerable efforts and advances in developing methodologies and techniques for nonlinear dimensionality reduction. A number of novel projection methods have been proposed from statistics, geometry theory and neural networks. Two fundamental approaches are multidimensional scaling and nonlinear PCA. This tutorial will provide an introduction to this challenging and demanding topic. Various methods along these lines such as, self-organising maps, kernel PCA, principal manifold, metric and non-metric scaling, isomap, local linear embedding, Laplacian eigenmap, as well as spectral clustering will be explained and discussed. It will also attempt to unify these methods under a constrained self-organising framework. Examples and applications will be shown to illustrate the usefulness and strengthen of various methods, as well their weakness and limitation.
Dimensionality Reduction and Data Visualisation

- PCA, MDS, Principal Curve/Surface
- LLE, ISOMAP, Kernel PCA, Eigenmap
- SOM & Data Visualisation
- ViSOM & MDS, Principal Curve/Surface
- SOMs & Mixture Model, Kernel Method
- Conclusions

1. PCA, MDS & Principal Curve/Surface

**PCA: A linear coordinate transformation**

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

\[
\min \sum_i \|X - \sum_j q_j^T X q_j\|^2 \\
\ max \{q_j^T C q_j = \sigma_j^2\}, \ q_j \perp q_j, i \neq j
\]

- \(X\): \(n\)-dimensional vector, zero-mean
- \(\{q_j\}\): orthogonal eigenvectors of covariance \(C=E[XX^T]\)
- \(m\leq n\)

\[
|C-\lambda I|=0 \\
(C-\lambda I)q_j=0
\]

**PCA decomposition**

\[
Q^T E[XX^T]Q = \Lambda
\]

- simple, direct visualisation
- stable (fast) solution
- linear mapping
- batch operation

\[
\begin{pmatrix}
\mathbf{x}_1 \\
\cdots \\
\mathbf{x}_n
\end{pmatrix}
\]

\[
\begin{pmatrix}
\mathbf{q}_1 \\
\cdots \\
\mathbf{q}_n
\end{pmatrix}
\]

\[
\begin{pmatrix}
\lambda_1 \\
\cdots \\
\lambda_n
\end{pmatrix}
\]

\[
\begin{pmatrix}
\mathbf{x}_1 \\
\cdots \\
\mathbf{x}_n
\end{pmatrix} = \begin{pmatrix}
\mathbf{q}_1 \\
\cdots \\
\mathbf{q}_n
\end{pmatrix} \begin{pmatrix}
\lambda_1 & 0 & \cdots \\
0 & \ddots & \vdots \\
0 & \cdots & \lambda_n
\end{pmatrix}
\]

- \(\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n\) eigenvalues or variances
1. PCA, MDS & Principal Curve/Surface

**PCA:** Example – Iris data

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

<table>
<thead>
<tr>
<th>Sepal Length</th>
<th>Sepal Width</th>
<th>Petal Length</th>
<th>Petal Width</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.9</td>
<td>1.0</td>
<td>1.4</td>
<td>0.2</td>
</tr>
<tr>
<td>4.7</td>
<td>1.2</td>
<td>1.3</td>
<td>0.2</td>
</tr>
<tr>
<td>4.6</td>
<td>1.1</td>
<td>1.5</td>
<td>0.2</td>
</tr>
<tr>
<td>5.0</td>
<td>1.6</td>
<td>1.4</td>
<td>0.2</td>
</tr>
<tr>
<td>5.4</td>
<td>1.9</td>
<td>1.7</td>
<td>0.4</td>
</tr>
<tr>
<td>4.6</td>
<td>1.4</td>
<td>1.4</td>
<td>0.3</td>
</tr>
<tr>
<td>7.0</td>
<td>1.2</td>
<td>1.4</td>
<td>1.4</td>
</tr>
<tr>
<td>6.4</td>
<td>1.2</td>
<td>1.4</td>
<td>1.5</td>
</tr>
<tr>
<td>6.9</td>
<td>1.1</td>
<td>1.4</td>
<td>1.5</td>
</tr>
<tr>
<td>5.5</td>
<td>1.2</td>
<td>1.4</td>
<td>1.5</td>
</tr>
<tr>
<td>6.5</td>
<td>1.2</td>
<td>1.4</td>
<td>1.5</td>
</tr>
<tr>
<td>5.7</td>
<td>1.2</td>
<td>1.4</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Projection onto the 1st×2nd eigenvectors

**MDS (Multidimensional Scaling)**

\[
S = \sum_{i<j} \frac{1}{d_{ij}} \sum_{i<j} [d_{ij} - D_{ij}]^2
\]

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

- \(d_{ij}\): inter-point distance in original space
- \(\delta_{ij}\): dissimilarity
- \(D_{ij}\): inter-point distance in projected plot

- Classical MDS: \(D_{ij} \approx \delta_{ij}\)
- Metric MDS: \(D_{ij}(\delta_{ij}) = d_{ij}\)
- Nonmetric MDS: \(\delta_{ij} \leq D_{ij} \Rightarrow D_{ij} \leq D_{kl} \forall i,j,k,l\)

Nonlinear, direct visualisation

Stable solution

Point-point mapping (no function)

Computational intensive
1. PCA, MDS & Principal Curve/Surface

**MDS:** Sammon Mapping

<table>
<thead>
<tr>
<th>4.9</th>
<th>10</th>
<th>1.4</th>
<th>0.2</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.7</td>
<td>12</td>
<td>1.3</td>
<td>0.2</td>
</tr>
<tr>
<td>4.6</td>
<td>11</td>
<td>1.5</td>
<td>0.2</td>
</tr>
<tr>
<td>5.0</td>
<td>16</td>
<td>1.4</td>
<td>0.2</td>
</tr>
<tr>
<td>5.4</td>
<td>19</td>
<td>1.7</td>
<td>0.4</td>
</tr>
<tr>
<td>4.6</td>
<td>14</td>
<td>1.4</td>
<td>0.3</td>
</tr>
<tr>
<td>7.0</td>
<td>12</td>
<td>4.7</td>
<td>1.4</td>
</tr>
<tr>
<td>6.4</td>
<td>12</td>
<td>4.3</td>
<td>1.5</td>
</tr>
<tr>
<td>6.9</td>
<td>11</td>
<td>4.9</td>
<td>1.5</td>
</tr>
<tr>
<td>5.5</td>
<td>23</td>
<td>4.0</td>
<td>1.3</td>
</tr>
<tr>
<td>6.5</td>
<td>28</td>
<td>4.6</td>
<td>1.5</td>
</tr>
<tr>
<td>5.7</td>
<td>28</td>
<td>4.5</td>
<td>1.3</td>
</tr>
<tr>
<td>6.3</td>
<td>15</td>
<td>6.0</td>
<td>2.5</td>
</tr>
<tr>
<td>5.8</td>
<td>27</td>
<td>6.1</td>
<td>1.9</td>
</tr>
<tr>
<td>7.1</td>
<td>10</td>
<td>5.8</td>
<td>2.1</td>
</tr>
<tr>
<td>6.3</td>
<td>29</td>
<td>5.6</td>
<td>1.8</td>
</tr>
<tr>
<td>6.5</td>
<td>10</td>
<td>5.8</td>
<td>2.2</td>
</tr>
<tr>
<td>7.6</td>
<td>10</td>
<td>6.6</td>
<td>2.1</td>
</tr>
</tbody>
</table>

…...

**Principal Curve/Surface**

-Hastie and Stuetzle (1989)

A smooth and self-consistent curve passing through the “middle” of the data.

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

Projection:

\[ \rho_j(x) = \sup_{\rho \in \Lambda} \{ \rho : \|x - f(\rho)\| = \inf_{\varphi} \|x - f(\varphi)\| \} \]

Expectation:

\[ f(\rho) = E[X \mid \rho_j(X) = \rho] \]

Kernel smoothing:

\[ F(\rho) = \frac{\sum_{i=1}^{S} x_i \kappa(\rho, \rho_i)}{\sum_{i=1}^{S} \kappa(\rho, \rho_i)} \]
2. Isomap, LLE, Eigenmap & Kernel PCA

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

- **LLE (Local Linear Embedding)**: Roweis and Saul (2000)
  for nonlinear 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 \( Y \):
    \[
    \Phi(Y) = \min \sum_i \| Y_i - \sum_j W_{ij} Y_j \|^2
    \]
2. Isomap, LLE, Eigenmap & Kernel PCA

- **Laplacian Eigenmap**: Belkin and Niyogi (2003) for dimensionality reduction.
  
  - Select neighbourhood graph: \( k \) nearest neighbours or \( \epsilon \) rule.
  
  - Construct the weightings (heat kernel):
    
    \[
    W_{ij} = e^{-\frac{||X_i - X_j||^2}{\epsilon}}
    \]

  - Eigenmap: compute the embedding
    
    \[
    Lf = \lambda Df
    \]
    
    \[
    D_{ii} = \sum_j W_{ji}, \text{ and } L = D - W
    \]
    
    \[
    X_i \rightarrow (f_1(i), f_2(i), \ldots, f_m(i))
    \]

  - Objective function:
    
    \[
    \min_{ij} \sum (Y_i - Y_j)^2 W_{ij}
    \]

- **Spectral Clustering**
  - Weiss 1999; Ng, Jordan and Weiss 2002

- **Laplacian Eigenmaps**
  - Belkin and Niyogi (2003)

- **Laplacian Eigenmaps**
  - Belkin and Niyogi (2003)

- **Kernel PCA**: Shölkopf, Smola and Müller (1998) for nonlinear PCA.
  
  - Kernel method has become popular.
    
    \[
    \phi: X \rightarrow F,
    \]
    
    \[
    \kappa: X \times X \in \mathbb{R},
    \]
    
    \[
    \kappa(x; y) = \langle \phi(x), \phi(y) \rangle
    \]

  - **PCA**
    
    \[
    Cq = \lambda q,
    \]
    
    \[
    C = \frac{1}{n} \sum_i x_i x_i^T, \quad q = \sum_i \alpha_i x_i,
    \]

  - **Kernel PCA**
    
    \[
    K \alpha = \lambda \alpha,
    \]
    
    \[
    K_{ij} := \langle \phi(x_i), \phi(x_j) \rangle, \quad \alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T,
    \]
    
    \[
    q = \sum_i \alpha_i \phi(x_i), \quad \langle \phi(x_i), q \rangle = \sum_i \alpha_i \kappa(x_i, x_i),
    \]
3. Self-Organising Maps

**SOM:** 
**Background–lateral inhibition**

It explains Mach-band effect and abstraction purpose.

Hartline, et al. 1960s

V. Bruce & P.R Green

Kohonen’s model (1982) is an abstraction of von der Malsburg and Willshaw’s model (1973, 1976)

Hebbian learning (Hebb 1949) 
\[ \Delta w = \alpha xy \]

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

\[
\begin{align*}
\frac{\partial y_j(t)}{\partial t} + cy_j(t) &= \sum_j w_{ij}(t)x_i(t) + \sum_k e_k y^*_k(t) - \sum_l h_{j+l}x_l(t) \\
\frac{\partial w_{ij}(t)}{\partial t} &= \alpha y_j(t)y^*_j(t), \quad \text{subject to } \sum w_{ij} = \text{constant}
\end{align*}
\]

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

\[
\begin{align*}
y_j(t + 1) &= \varphi \left( w_j^T x(t) + \sum_{l} h_{j+l}y_{j+l}(t) \right) \\
\frac{\partial w_{ij}(t)}{\partial t} &= \alpha y_j(t)x_i(t) - \beta y_j(t)w_{ij}(t) \\
&= \alpha[x_j(t) - w_{ij}(t)]y_j(t), \quad \text{if } j \in \eta(t) \\
&= 0, \quad \text{if } j \notin \eta(t)
\end{align*}
\]

Hujun Yin, University of Manchester  
WCCI 2008, Hong Kong  
h.yin@manchester.ac.uk
3. Self-Organising Maps

**SOM: Algorithm**

- At each time \( t \), present an input, \( x(t) \), select the winner.

\[ v = \text{arg min}_{c \in \Omega} \| x(t) - w_c \| \]

- Updating the weights of winner and its neighbours.

\[ \Delta w_k(t) = \alpha(t) \eta(v, k, t) \langle x(t) - w_k(t) \rangle \]

- Repeat until the map converges.

Typical neighbourhood function:

\[ \eta(v, k, t) \propto \exp\left[-\frac{\| v - k \|^2}{2\sigma(t)^2}\right] \]

**SOM: Quantisation, topology & cost function**

- Topologically “ordered” map
- “Error tolerant” coding - HVQ (Luttrell, NC 1994)
- “Minimum wiring” (Mitchison, NC 1995), (Durbin & Mitchson, Nature 1990)

\[ E(w_1, ..., w_Q) = \sum_i \sum_k h_{i,k} \| x - w_i \| \rho(x) dx \] (Heskes, 1999)
3. Self-Organising Maps

SOM: Variants & extensions
- HSOM (Miikkulainen 1990), DISLEX (1990, 1997)
- PSOM (Ritter 1993), Hyperbolic SOM (1999), H²SOM
- Temporal Kohonen Map (Chappell & Taylor 1993)
- Neural Gas (Martinetz et al. 1991), Growing Grid (Fritzke 1995)
- ASSOM (Kohonen 1997)
- Recurrent SOM (Koskela, 1997), RecursiveSOM (Voegtlin 2001)
- SOAR (Lampinen & Oja 1989), SOMAR (Ni & Yin, 2007)
- Bayesian SOM & SOMN (Yin & Allinson 1995, 1997; Utsgui 1997)
- GTM (Bishop et al. 1998)
- GHSM (Merkel et al. 2000), TOC (TreeSOM) (Freeman & Yin 2004)
- PicSOM (Laaksonen, Oja, et al., 2000)
- ViSOM (Yin 2001, 2002), gViSOM (Yin 2007)
3. Self-Organising Maps

SOM: Applications -snapshots

A Temporal Shape Metric:

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

Foreign exchange modelling:
SOM+local SVM (H. Ni & Yin, 2006), SOMAR, 2007

Spike Trains:
Compression

SOM: Data visualisation & dimensionality reduction

- topology preserving mapping
- (discrete) mapping function
- non distance preserving
- boundary problems
3. Self-Organising Maps

**SOM**: Data visualisation & knowledge management


**Tree-View SOM**
(Freeman and Yin 2004)

4. ViSOM & Principal Curve/Surface, MDS

**ViSOM**: Visualisation induced SOM
(Yin, 2001, 2002)

- To preserve distance/metric (**locally**) on the map
- To extrapolate smoothly

**Principle**

SOM update:

\[ w_k(t+1) = w_k(t) + \alpha(t) \eta(v,k,t)[x(t) - w_k(t)] \]
4. ViSOM & Principal Curve/Surface, MDS

**ViSOM: Algorithm**

- **Grid structure and winner selection same to SOM**

- **Updating**

\[
\Delta w_{k}(t) = \alpha(t)\eta(\nu, k, t)\left( [x(t) - w_{\nu}(t)] + [w_{\nu}(t) - w_{k}(t)]\frac{d_{vk} - \Delta_{vk}\lambda}{\Delta_{vk}\lambda} \right)
\]

- **Refreshing**

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

\[
\Delta w_{k} = w_{k}(t) + \alpha(t)\eta(\nu, k, t)\left( [x(t) - w_{\nu}(t)] + [\xi + (1 - \xi)\frac{d_{vk} - \Delta_{vk}\lambda}{\Delta_{vk}\lambda} - 1][w_{\nu}(t) - w_{k}(t)] \right)
\]

**ViSOM: Examples**

[Diagrams showing SOM and ViSOM]
4. ViSOM & Principal Curve/Surface, MDS

**ViSOM: Examples**

**PCA**

**Sammon**

**SOM**

**ViSOM**

---

**Ranking table of UK universities**

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

<table>
<thead>
<tr>
<th>Ranking</th>
<th>University</th>
<th>F1</th>
<th>F2</th>
<th>F3</th>
<th>F4</th>
<th>F5</th>
<th>F6</th>
<th>F7</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cambridge</td>
<td>244</td>
<td>182</td>
<td>247</td>
<td>97</td>
<td>88</td>
<td>100</td>
<td>50</td>
<td>1504</td>
</tr>
<tr>
<td>2</td>
<td>Oxford</td>
<td>214</td>
<td>175</td>
<td>244</td>
<td>97</td>
<td>81</td>
<td>100</td>
<td>50</td>
<td>1441</td>
</tr>
<tr>
<td>3</td>
<td>LSE</td>
<td>200</td>
<td>175</td>
<td>253</td>
<td>97</td>
<td>68</td>
<td>100</td>
<td>50</td>
<td>1323</td>
</tr>
<tr>
<td>4</td>
<td>Imperial</td>
<td>205</td>
<td>154</td>
<td>232</td>
<td>98</td>
<td>67</td>
<td>100</td>
<td>50</td>
<td>1264</td>
</tr>
<tr>
<td>5</td>
<td>York</td>
<td>206</td>
<td>143</td>
<td>208</td>
<td>94</td>
<td>63</td>
<td>76</td>
<td>60</td>
<td>1010</td>
</tr>
<tr>
<td>6</td>
<td>UCL</td>
<td>172</td>
<td>152</td>
<td>210</td>
<td>95</td>
<td>71</td>
<td>100</td>
<td>30</td>
<td>830</td>
</tr>
<tr>
<td>7</td>
<td>St Andrews</td>
<td>139</td>
<td>131</td>
<td>194</td>
<td>96</td>
<td>73</td>
<td>91</td>
<td>100</td>
<td>824</td>
</tr>
<tr>
<td>8</td>
<td>Warwick</td>
<td>155</td>
<td>155</td>
<td>215</td>
<td>97</td>
<td>69</td>
<td>86</td>
<td>20</td>
<td>795</td>
</tr>
<tr>
<td>9</td>
<td>Bath</td>
<td>132</td>
<td>142</td>
<td>211</td>
<td>97</td>
<td>66</td>
<td>83</td>
<td>60</td>
<td>791</td>
</tr>
<tr>
<td>10</td>
<td>Nottingham</td>
<td>176</td>
<td>125</td>
<td>218</td>
<td>96</td>
<td>74</td>
<td>72</td>
<td>30</td>
<td>791</td>
</tr>
<tr>
<td>11</td>
<td>Bristol</td>
<td>145</td>
<td>131</td>
<td>218</td>
<td>96</td>
<td>75</td>
<td>94</td>
<td>20</td>
<td>770</td>
</tr>
<tr>
<td>12</td>
<td>Durham</td>
<td>163</td>
<td>132</td>
<td>207</td>
<td>91</td>
<td>64</td>
<td>72</td>
<td>50</td>
<td>779</td>
</tr>
<tr>
<td>13</td>
<td>Edinburgh</td>
<td>106</td>
<td>145</td>
<td>218</td>
<td>96</td>
<td>74</td>
<td>100</td>
<td>40</td>
<td>779</td>
</tr>
<tr>
<td>14</td>
<td>Lancaster</td>
<td>156</td>
<td>144</td>
<td>186</td>
<td>95</td>
<td>62</td>
<td>63</td>
<td>50</td>
<td>756</td>
</tr>
<tr>
<td>15</td>
<td>UMIST</td>
<td>135</td>
<td>144</td>
<td>188</td>
<td>97</td>
<td>58</td>
<td>100</td>
<td>30</td>
<td>752</td>
</tr>
<tr>
<td>16</td>
<td>Birmingham</td>
<td>146</td>
<td>127</td>
<td>204</td>
<td>96</td>
<td>87</td>
<td>87</td>
<td>20</td>
<td>747</td>
</tr>
<tr>
<td>17</td>
<td>Loughborough</td>
<td>142</td>
<td>115</td>
<td>177</td>
<td>95</td>
<td>57</td>
<td>66</td>
<td>60</td>
<td>732</td>
</tr>
<tr>
<td>18</td>
<td>Southampton</td>
<td>143</td>
<td>124</td>
<td>180</td>
<td>93</td>
<td>55</td>
<td>71</td>
<td>50</td>
<td>726</td>
</tr>
<tr>
<td>19</td>
<td>King's College</td>
<td>139</td>
<td>126</td>
<td>204</td>
<td>96</td>
<td>63</td>
<td>100</td>
<td>-10</td>
<td>714</td>
</tr>
<tr>
<td>20</td>
<td>Newcastle</td>
<td>134</td>
<td>117</td>
<td>193</td>
<td>97</td>
<td>60</td>
<td>87</td>
<td>20</td>
<td>708</td>
</tr>
<tr>
<td>21</td>
<td>Manchester</td>
<td>125</td>
<td>134</td>
<td>199</td>
<td>96</td>
<td>66</td>
<td>98</td>
<td>-10</td>
<td>707</td>
</tr>
<tr>
<td>22</td>
<td>Leeds</td>
<td>122</td>
<td>127</td>
<td>199</td>
<td>97</td>
<td>61</td>
<td>74</td>
<td>20</td>
<td>700</td>
</tr>
<tr>
<td>23</td>
<td>Sheffield</td>
<td>145</td>
<td>128</td>
<td>213</td>
<td>97</td>
<td>61</td>
<td>72</td>
<td>-20</td>
<td>689</td>
</tr>
<tr>
<td>24</td>
<td>East Anglia</td>
<td>125</td>
<td>127</td>
<td>176</td>
<td>96</td>
<td>63</td>
<td>60</td>
<td>40</td>
<td>687</td>
</tr>
<tr>
<td>25</td>
<td>Leicester</td>
<td>125</td>
<td>120</td>
<td>183</td>
<td>94</td>
<td>52</td>
<td>82</td>
<td>20</td>
<td>687</td>
</tr>
</tbody>
</table>

**F1**: Research  
**F2**: Teaching  
**F3**: A-levels  
**F4**: Employment  
**F5**: S/S ratio  
**F6**: 1st/2:1s  
**F7**: Dropout rate
4. ViSOM & Principal Curve/Surface, MDS

ViSOM: A discrete principal curve/surface (Yin, 2002)

Projection:
\[ \rho_f(x) = \sup_{\rho \in A} \{ \rho : \| x - f(\rho) \| = \inf_{\rho} \| x - f(\theta) \| \} \]

Expectation:
\[ f(\rho) = E[X \mid \rho_f(X) = \rho] \]

Kernel smoothing:
\[ F(\rho) = \frac{\sum S_i \kappa(\rho, \rho_i)}{\sum S_i} \]

SOM/VisOM smoothing:
\[ w_k = \frac{\sum S_i h(k,i)}{\sum h(k,i)} \]

SOM: \[ \| k-i \| \| w_k - w_i \| \]

ViSOM: \[ \| k-i \| \| w_k - w_i \| \]
4. ViSOM & Principal Curve/Surface, MDS

ViSOM Extensions:

° Resolution Enhancement by Local Linear Projection - LLP (Yin, 2003)

\[
LLP_{x,y} = \max_{v_x,v_y \in \mathbb{R}, x \neq y} \left\{ \frac{(x - w_y) \cdot (w_x - w_y)}{\|w_y - w_v\|^2}, 0 \right\}
\]

\[
x' = \arg \max_{e} \{ (x - w_e) \} + LLP(x, y)
\]

° Incremental or growing ViSOM - gViSOM (Yin, 2007)

Start with a small map (rectangular or hexagonal); add rows/columns or polygons to the sides or vertices where activities are high; etc.
4. ViSOM & Principal Curve/Surface, MDS

Other PC/S algorithms:

- **SOM** has been related to PC/S and termed discrete PC/S by *Ritter, Martineitz & 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

\[
\begin{align*}
\sum_i x_i h(k,i) &= \sum_i x_i \delta(k,i) \\
\sum_i h(k,i) &= x(k,i) \\
V_i &= \sum_i x_i \rho(k,i) \\
SOM: |k-i| = |w_i-w_i| = |\rho-i| \\
ViSOM: |k-i| = |w_i-w_i| = |\rho-i|
\end{align*}
\]

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.
4. ViSOM & Principal Curve/Surface, MDS

SOM, ViSOM & MDS:

• SOM is a qualitative (nonmetric) MDS.
• ViSOM is a metric MDS.

\[ \text{MDS cost function:} \]
\[ \min \sum_{i,j} (d_i - d_j)^2 = \min \sum_{i,j} (d_i^2 + D_i^2 - 2d_i D_i) = \min \sum_{i,j} (-2d_i D_i) \]

\[ \text{SOMs cost function (sample cost of Voronoi region } i): \]
\[ \min \sum_{j} \eta(i, j) \| \mathbf{x} - \mathbf{w}_j \|^2 = \min \sum_{j} f(||i - j||) \| \mathbf{x}_i - \overline{\mathbf{x}}_j \|^2 \]

\[ \min = \sum_{j} f(D_{ij})d_{ij}^2 \approx \min \sum_{j} -(D_{ij}d_{ij})^2 \]

- SOM is a qualitative (nonmetric) MDS.
- ViSOM is a metric MDS.

4. ViSOM Examples

**Iris:**

**Sammon metric MDS**

\[ C \text{ Measure (Goodhill & Sejnowski, 1997)} \]
\[ C = \sum_{i,j} F(i,j) G(M(i), M(j)) \]
4. ViSOM Examples

**S-curve:**

- **Yin (2007)**
- **LLE**
- **ViSOM embedding**
- **ViSOM output**
- **Isomap**
- **SOM**
- **PCA**

Two-dimensional Isomap embedding (with neighborhood graph).
4. ViSOM Examples

**Swissroll:**
Yin (2007)

Swissroll data & gViSOM embedding (final size 18x70, $\lambda=1.5$)

Isomap & LLE

**gViSOM with LLP**

5. Kernel SOM & Mixture Model

**Kernel SOM: Background**

- Kernel method has become popular.

  $\phi: X \rightarrow F$, $x \mapsto \phi(x)$

  $\kappa: X \times X \in \mathbb{R}$, $\kappa(x, y) = \langle \phi(x), \phi(y) \rangle$

- PCA

  $Cq = \lambda q, \quad C = \frac{1}{n} \sum_{i} x_i x_i^T, \quad q = \sum_{i} \alpha_i x_i$

- Kernel PCA

  $K \alpha = \lambda \alpha, \quad K_{ij} := \langle \phi(x_i), \phi(x_j) \rangle, \quad \alpha = [\alpha_1, \alpha_2, \ldots, \alpha_n]^T$

  $q = \sum_{i} \alpha_i \phi(x_i), \quad \langle \phi(x_i), q \rangle = \sum_{j} \alpha_i \kappa(x_i, x_j)$
5. Kernel SOM & Mixture Model

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

\[
\phi: x \rightarrow F \quad x \mapsto \phi(x), \quad m_i = \sum_n \alpha_{i,n} \phi(x^n),
\]

\[
||\phi(x) - m_j||^2 = ||\phi(x) - \sum_n \alpha_{i,n} \phi(x^n)||^2
\]

\[
= \kappa(x, x) - 2 \sum_n \alpha_{i,n} \kappa(x, x^n) + \sum_{n,m} \alpha_{i,n} \alpha_{j,m} \kappa(x^n, x_m)
\]

\[
m_j(t + 1) = m_j(t) + \Lambda[\phi(x) - m_j(t)], \quad \Lambda = \frac{\zeta_i(x_j)}{\sum_{i=1}^{n+1} \zeta_i(x_j)}
\]

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

---

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

\[
\nu = \arg \min \|x - m_j\|^2 \quad \nu = \arg \min \|\phi(x) - \phi(m_j)\|^2
\]

\[
m_j(t + 1) = m_j(t) + \alpha(t) h(\nu(x), i) \nabla J(x, m_j)
\]

\[
J(x, m_j) = ||\phi(x) - \phi(m_j)||^2 = \kappa(x, x) + \kappa(m_j, m_j) - 2\kappa(x, m_j)
\]

\[
\nabla J(x, m_j) = \frac{\partial \kappa(m_j, m_j)}{\partial m_j} - 2 \frac{\partial \kappa(x, m_j)}{\partial m_j}
\]

\[
v = \arg \min \nu J(x, m_j) = \arg \min [\nu \kappa(x, m_j)] = \arg \min \left[ -2 \kappa(x, m_j) \right]
\]

\[
m_j(t + 1) = m_j(t) + \alpha(t) h(\nu(x), i) \frac{1}{2\sigma^2} \exp(-\frac{||x - m_j||^2}{2\sigma^2})(x - m_j)
\]
5. Kernel SOM & Mixture Model

Kernel SOM: An application

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. (Lau, Yin & Hubbard 2006)

<table>
<thead>
<tr>
<th>Kernel</th>
<th>Type I Kernel SOM</th>
<th>Type II Kernel SOM</th>
<th>SOM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>M</td>
<td>A</td>
<td>V</td>
</tr>
<tr>
<td>Gaussian</td>
<td>5.6</td>
<td>5.8</td>
<td>5.6</td>
</tr>
<tr>
<td>Cauchy</td>
<td>5.5</td>
<td>5.6</td>
<td>5.5</td>
</tr>
<tr>
<td>Log</td>
<td>4.6</td>
<td>4.6</td>
<td>4.6</td>
</tr>
</tbody>
</table>

Mixture Model:

\[ p(x | \Theta) = \sum_{i=1}^{K} p_i(x | \Theta_i) \theta_i \]

Kullback-Leibner divergence:

\[ I = -\int \log \frac{\hat{p}(x)}{p(x)} p(x) dx \]

\[ \frac{\partial I}{\partial \theta_i} = -\int \left[ \frac{1}{\hat{p}(x | \Theta)} \frac{\partial \hat{p}(x | \Theta)}{\partial \theta_i} \right] p(x) dx \]
5. Kernel SOM & Mixture Model

**Self Organising Mixture Network:** (Yin & Allinson, 2001)

\[
\dot{\theta}_i(t + 1) = \dot{\theta}_i(t) + \alpha(t) h(v(x), i) \left[ \frac{1}{\hat{p}(x | \Theta)} \cdotp \partial_{\theta} \hat{p}(x | \Theta) \right] \\
= \dot{\theta}_i(t) + \alpha(t) h(v(x), i) \sum_j p_j(t) \cdotp \frac{\hat{p}_j(x | \Theta)}{\partial_{\theta} \hat{p}_j(x | \Theta)}
\]

\[
\hat{P}(t + 1) = \hat{P}(t) + \alpha(t) \left[ \frac{\hat{P}(x | \Theta) \cdotp \hat{P}_j(t)}{\hat{p}(x | \Theta)} - \hat{P}_j(t) \right] = \hat{P}(t) - \alpha(t) h(v(x), i) [\hat{P}(i | x) - \hat{P}_j(t)]
\]

\[
v = \arg \max_i \{ \hat{P}(i | x) \} = \frac{\hat{P}_j(x | \hat{\theta})}{\hat{p}(x | \Theta)}
\]

- **Homoscedastic case**

\[
v = \arg \max_i \frac{\hat{p}_j(x | \theta)}{\sum_j \hat{p}_j(x | \theta)}
\]

\[
m_j(t + 1) = m_j(t) + \alpha(t) h(v(x), i) \sum_j p_j(x | \theta) \cdotp \frac{1}{c_m} \partial_{\theta} \hat{p}_j(x | \Theta)
\]
5. Kernel SOM & Mixture Model

Self Organising Mixture Network:

Homoscedastic and Gaussian case

$$v = \arg \max_i \exp(-\frac{||x - m_i||^2}{2\sigma^2})$$

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

The same as those of Kernel SOM !!

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

Summary

- Objective functions of many mappings, such as MDS, LLE, Isomap, eigenmap and ViSOM are closely linked, but a good algorithm or implementation is the key.
- Converting problems (e.g. in eigenmap or spectral clustering) into a linear one is an advantage, in terms of uniqueness and stability of the solution.
- SOM is a useful tool for data clustering, relational visualisation (nonmetric scaling) and management. ViSOM is particularly suited for direct visualisation and is a metric preserving nonlinear manifold. gViSOM is an effective algorithm.
- SOM is linked to mixture model and kernel method.