Chapter 6

Multi-dimensional Scaling (MDS)

N points. Distances \( d_{ij} \), \( i, j = 1 \ldots N \).

MDS puts these points into low-dimensional space so that the distance between them in this space is as close as possible to \( d_{ij} \).

Sample: \( X = \left< X^t \right>_{t=1}^N, \quad X^t \in \mathbb{R}^d \).

Two points \( r, s \):

\[
\delta_{rs} = \frac{1}{2} \left( \sum_{t=1}^N (x_t^r - x_t^s)^2 + \sum_{t=1}^N (x_t^s - x_t^r)^2 \right) = \frac{1}{2} \left( \sum_{t=1}^N (x_t^r - x_t^s)^2 + \sum_{t=1}^N (x_t^s - x_t^r)^2 \right)
\]

Center data at origin: \( \sum_{t=1}^N x_t^t = 0, \forall j \).

Example: distances between cities.

Define \( \delta_{rs} = \frac{1}{2} \left( \delta_{rr} + \delta_{ss} - \delta_{rs} \right) \).

Can compute \( \delta_{rs} = \frac{1}{2} \left( \delta_{rr} + \delta_{ss} - \delta_{rs} \right) \).

\[
\Sigma = X X^T
\]

Look for an approximation. From spectral decomposition, we know that \( \Sigma = \Sigma \Sigma^T \) can be used to approximate, where \( \Sigma \) is the matrix whose columns are eigenvectors of \( \Sigma \).

\[
\Sigma = \Sigma \Sigma^T, \quad j = 1, k, t = 1, N
\]
MDS (and)

Eigenvectors of $X^T X$ are the same as $X^T X$

eigenvectors are related by simple linear

transformation.

PCA does some work as MDS, and does it

more cheaply.

More generally, we want a mapping $\mathbf{z} = g(x)$

classical(MDS) : $\mathbf{z} = g(x) = \frac{1}{\sqrt{n}} X$

In general, prefer a nonlinear mapping

Sammon mapping

Sammon Stress:

$$E(\theta(x)) = \frac{1}{n^2} \sum_{i < j} \frac{|| z_i - z_j ||^2}{|| x_i - x_j ||^2} \left( \frac{|| g(x_i(x)) - g(x_j(x)) ||^2}{|| x_i - x_j ||^2} \right)$$

Can use any regression method for $g(x)$. 


Linear Discriminant Analysis (LDA)

Start with 2 classes
\[ Z = \omega^T x. \]
\[ m_1, m_2 \text{ are the means of samples from } C_1 \text{ before and after projection.} \]
Sample \( X = \{ x^1, x^2 \} \quad r^1 = 1, \ldots, x^2 \in C_1 \quad = 0, \quad q \quad x^q \in C. \)

\[ m_1 = \frac{\Sigma \omega^T x^i r^i}{\Sigma r^i} = \omega^T m_1 \]

\[ m_2 = \frac{\Sigma \omega^T x^i (1-r^i)}{\Sigma (1-r^i)} = \omega^T m_2. \]

The scatter of samples from \( C \) after projection,
\[ s_1^2 = \frac{\Sigma}{r^i} (\omega^T x^i - m_1)^2 r^i \]
\[ s_2^2 = \frac{\Sigma}{r^i} (\omega^T x^i - m_2)^2 (1-r^i) \]

Fisher's Linear Discriminant maximizes
\[ J(\omega) = \frac{(m_1 - m_2)^2}{s_1^2 + s_2^2} \]

Result: \( (m_1 - m_2)^2 = (\omega^T m_1 - \omega^T m_2)^2 = \omega^T S_B \omega \)
where \( S_B = (m_1 - m_2)(m_1 - m_2)^T \)

between-class scatter matrix
(4) \( (LDA) \) (Goal)

\[ S_1^2 = \sum \left( \omega^T (x_t - m_i) (x_t - m_i)^T \omega \right)^2 r_t \]

\[ = \sum \frac{\omega^T (x_t - m_i) (x_t - m_i)^T \omega}{r_t} \]

where \( S_1 = \sum r_t (x_t - m_i) (x_t - m_i)^T \)

within-class scatter matrix for \( S \).

\[ S_1^2 + S_2^2 = \sum \frac{\omega^T S_{\omega} \omega}{S_{\omega}} = \sum S_{\omega} = S_1 + S_2 \]

\[ J(\omega) = \omega^T S_{\omega} \omega = \lambda \left( \omega^T S_{\omega} \omega - 1 \right) \]

Formulae as

\[ \frac{\partial}{\partial \omega} \sum S_{\omega} \omega = \lambda \sum S_{\omega} \omega \]

\[ S_{\omega} \omega = (m_i - m) (m_i - m)^T \omega \]

Hence solution \( \omega \propto S_{\omega}^{-1} (m_i - m) \)

magnitude of \( \omega \) is unchanged.

just need the direction.
(5) \[ \text{LDA, for } k > 2. \]

Task: find \( w \) st. \( Z = w^T X \)

where \( Z \) is \( k \)-dim and \( w \) is \( 1 \)-dim.

Within-class scatter for \( \mathcal{C}_i \)
\[ S_i = \sum_{x \in \mathcal{C}_i} (x - m_i)(x - m_i)^T \]
\[ r_i^c = \frac{1}{|\mathcal{C}_i|} \sum_{x \in \mathcal{C}_i} x \cdot x \]

Total within-class scatter is
\[ S_w = \sum_{i=1}^{k} S_i \]

Scatter of means
\[ m = \frac{1}{k} \sum_{i=1}^{k} m_i \]

Between-class scatter
\[ S_b = \sum_{i=1}^{k} N_i (m_i - m)(m_i - m)^T \]

After projection:

\[ w^T S_b w \]
\[ w^T S_w w \]

\[ J(w) = \frac{w^T S_b w}{w^T S_w w} \]

Solution \( \Rightarrow w^\top \) is column of the largest eigenvector of \( S_w^{-1} S_b \).

\( S_b \) has max-rank \( k-1 \).

Low-dim space dim \( k-1 \).