Remember:

$N >> p$

Rank $r = p$

$\boxed{\text{SVD and PCA}}$

$$\underbrace{X_{N\times p} = Y^T_{N\times p}} = \begin{pmatrix} \vec{y}_1^{\,T} \\ \vdots \\ \vec{y}_N^{\,T} \end{pmatrix} \qquad \vec{y}_i \in \mathbb{R}^p$$

$$\tilde{Y}^T_{N\times p} = \left(I - \frac{1}{N} 1 1^T\right) Y^T$$

centered data matrix

$$\sum_{j=1}^{p} \sigma_j \vec{u}_j \vec{v}_j^{\,T} = \tilde{Y}^T_{N\times p} = U_{N\times p} \Sigma_{p\times p} V^T_{p\times p}$$

SVD

$$U = (\vec{u}_1, \dots, \vec{u}_p)_{N\times p}$$

SVD provides a new orthonormal basis $\{\vec{V}_j\}_{j=1}^{p}$

$$V = (\vec{v}_1, \dots, \vec{v}_p)_{p\times p}$$

$\boxed{\text{our new basis}}$

$\downarrow$ E-Y

Closest approximation of rank k

For a given k:

$$\hat{\tilde{Y}}^T_{N\times p} = \Big[\underbrace{\left[U_k \Sigma_k\right]}_{N\times k} V^T_{k\times p}\Big]$$

$$\left\{ \begin{array}{l} (Y_{PCA}) = \left[U_k \Sigma_k\right]_{N\times k} \\ \text{coordinates of the} \\ V_k = (\vec{v}_1, \dots, \vec{v}_k) \quad \vec{v}_i \in \mathbb{R}^p \end{array} \right\}$$

where $\Sigma_k = \mathrm{diag}\begin{pmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{pmatrix}$

$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_k$

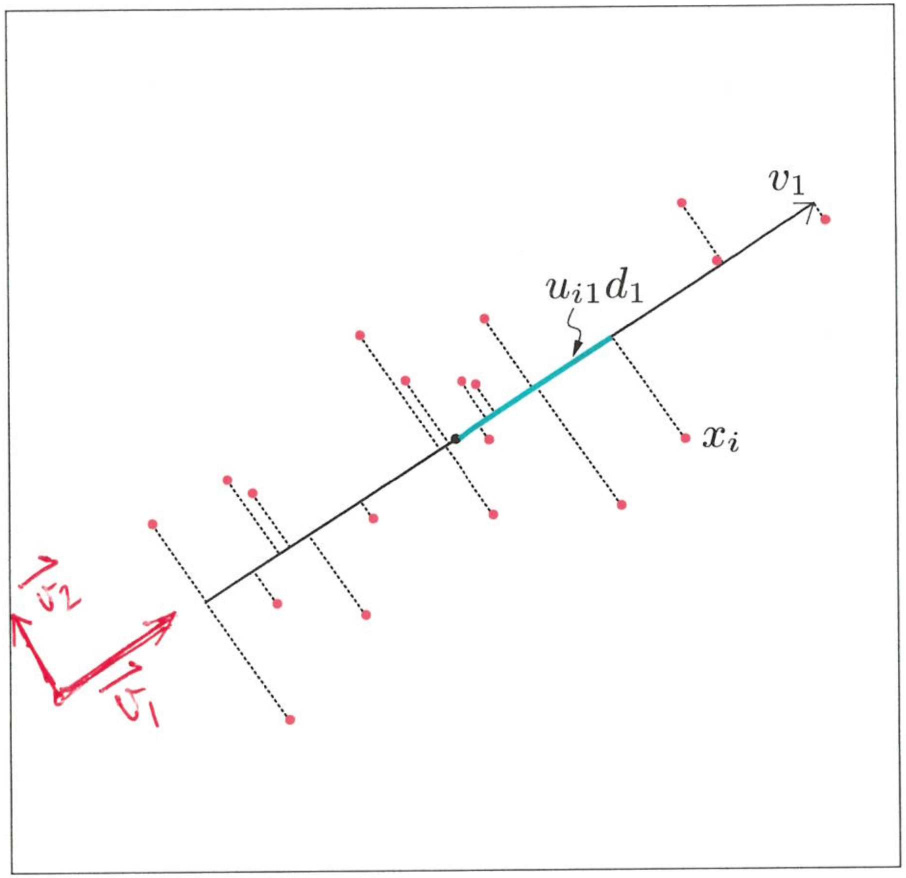
ordered singular values.

Examples overleaf $\longrightarrow$

Imperial College London

Illustration of PCA and SVD

$\vec{v}_1$ captures maximum variability



From the SVD:

$V = (\vec{v}_1 \ \vec{v}_2)$

$\boxed{\text{New basis}}$

$\vec{y}_i \in \mathbb{R}^2$

$i = 1, \dots, N$

$\vec{y}_i = \alpha_1^{(i)} \vec{v}_1 + \alpha_2^{(i)} \vec{v}_2$

$= \sigma_1 u_{i1} \vec{v}_1 + \sigma_2 u_{i2} \vec{v}_2$

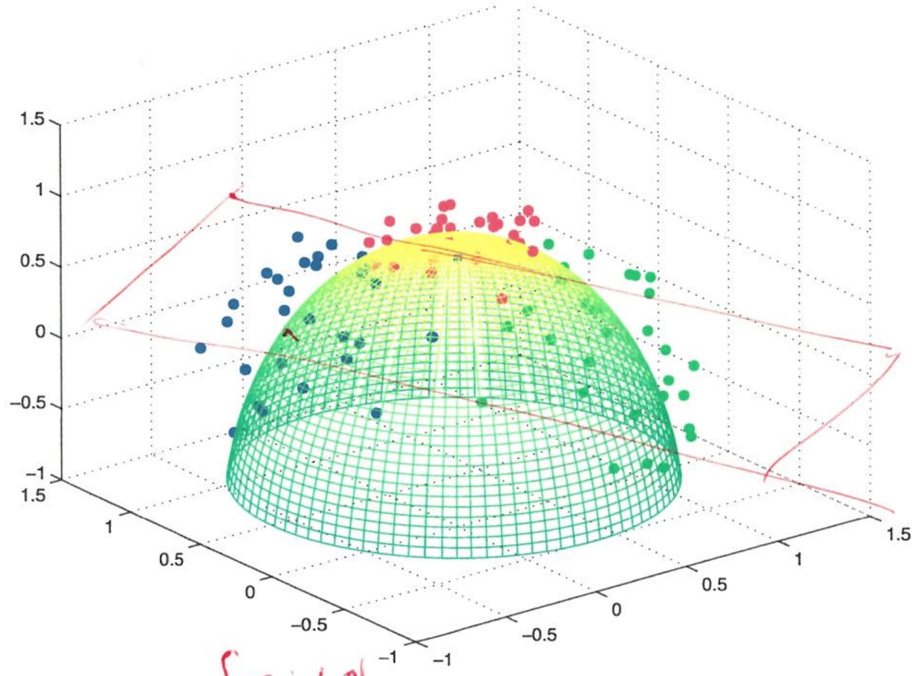
$= (\vec{v}_1^T \cdot \vec{y}_i)\, \vec{v}_1 + (\vec{v}_2^T \cdot \vec{y}_i)\, \vec{v}_2$

If we approximate by first component, we minimize the sum of perpendicular errors.

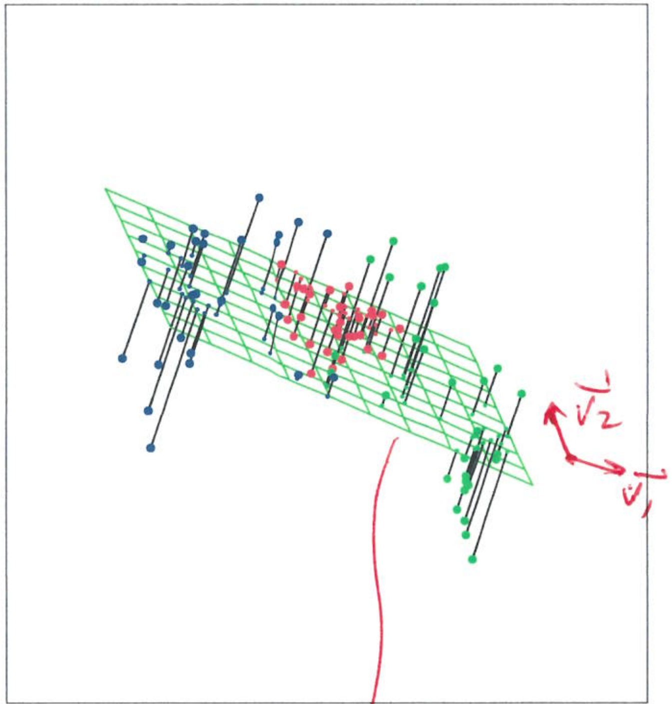
Imperial College London

Illustration of PCA and SVD

$\vec{y}_i \in \mathbb{R}^3$



$\{\vec{y}_i\}_{i=1}^{N}$

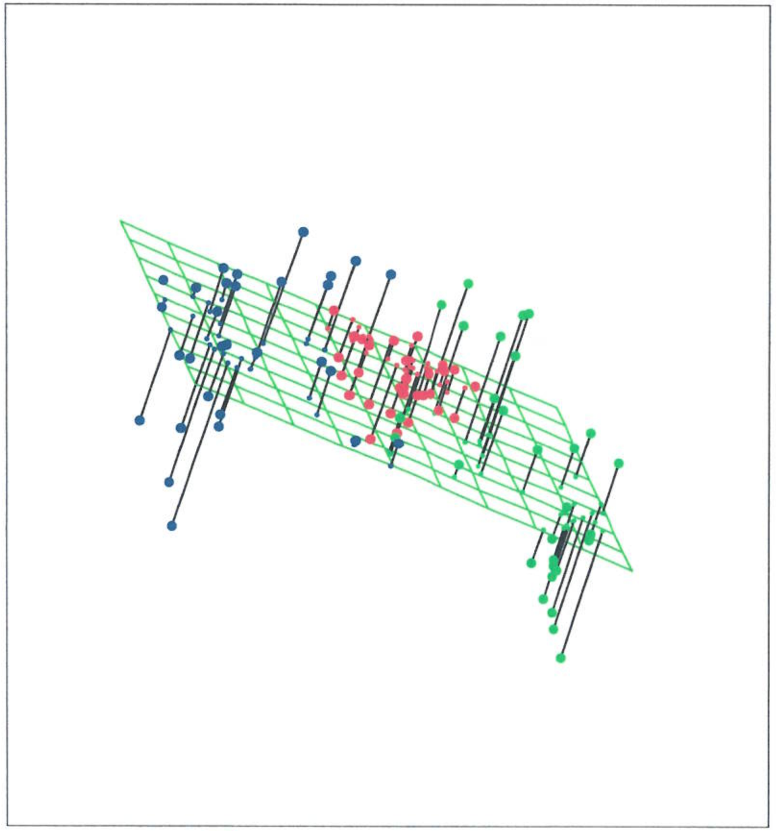


Best approximation given by hyperplane $(\vec{v}_1, \vec{v}_2)$

Imperial College London

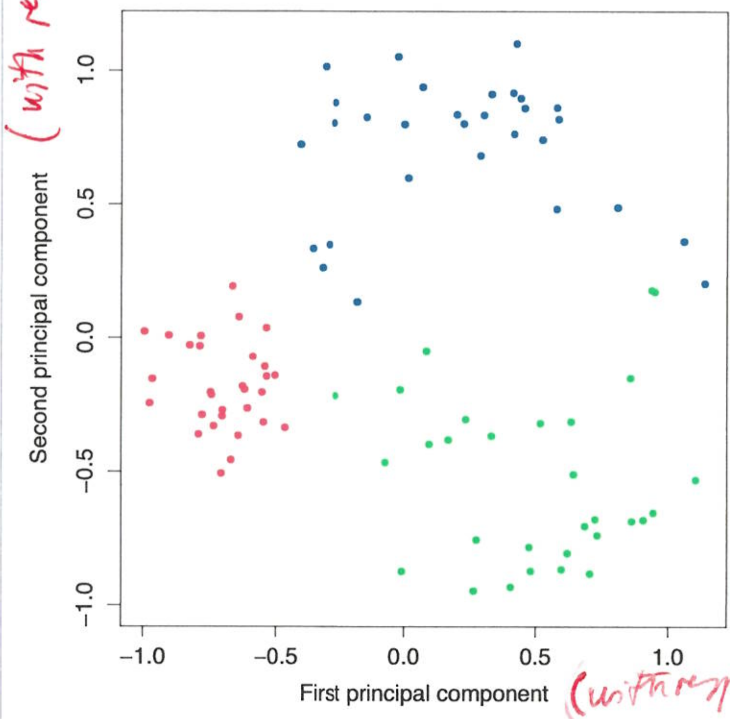
Illustration of PCA and SVD

$p=3$



$\tilde{y}^T_{N\times p}$

mapped to



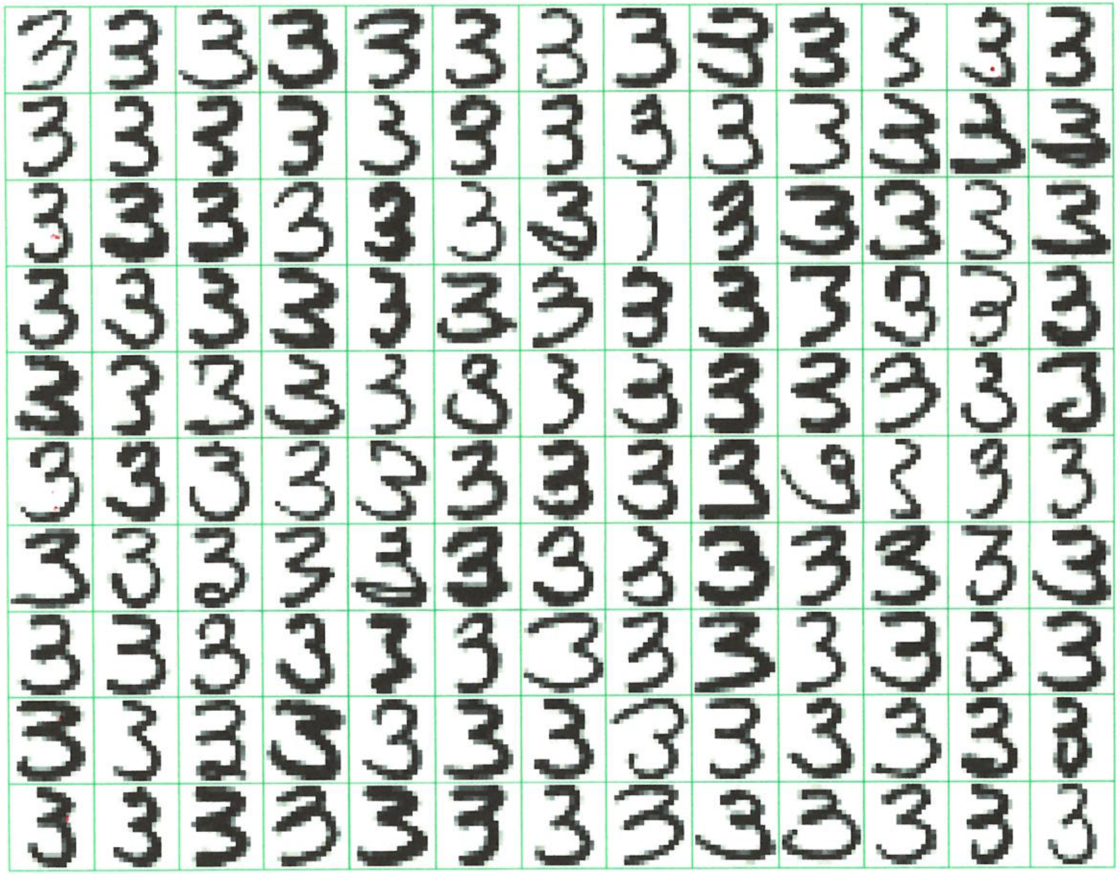
(with respect to $\vec{v}_2$)

(with respect to $\vec{v}_1$)

$(y_{PCA})_{N\times 2}$ = coordinates in $(\vec{v}_1, \vec{v}_2)$

Imperial College London

Illustration of PCA and SVD



Sample:

$\{\vec{y}_i\}_{i=1}^{130}$

$N = 130$ images

vectors:

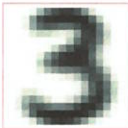
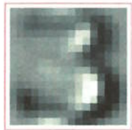
$\vec{y}_i \in \mathbb{R}^{16 \times 16}$

Each $\vec{y}_i$ contains the grayscale of the images with 256 pixels.

Illustration of PCA and SVD

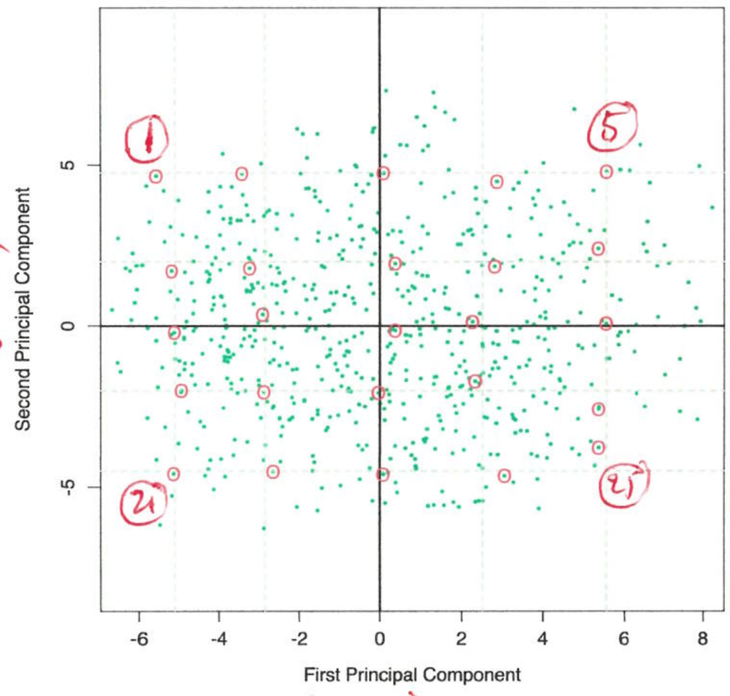
vectors expressed in terms of the first two $\vec{v}_i$

$$\hat{f}(\lambda) = \bar{x} + \lambda_1 \vec{v}_1 + \lambda_2 \vec{v}_2$$

=  $+ \lambda_1 \cdot$  $+ \lambda_2 \cdot$  .

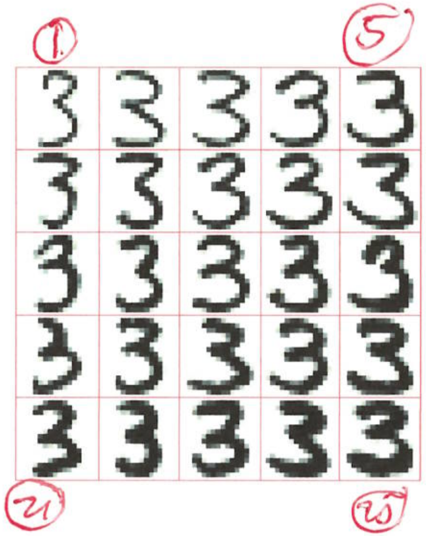
Imperial College
London

Illustration of PCA and SVD



(λ_2)

(λ_1)



PC2

The 3 becomes less thick

PC1

(The bottom part of the 3 becomes more curved)

First extension:

1. PCA is about linear projections that minimize a quadratic. How do we extend?

(1) "Nonlinearity": Kernel PCA

$$(\tilde{Y}^T\tilde{Y})_{ij} = (\vec{\tilde{y}}_i^T \cdot \vec{\tilde{y}}_j)$$

PCA was about eigenvalues of $\tilde{Y}^T\tilde{Y}$ and $\tilde{Y}\tilde{Y}^T$

which are "Kernel" matrices.

If we assume a kernel function

$$K_{ij} = K(\vec{\tilde{y}}_i, \vec{\tilde{y}}_j)$$

positive definite, etc. (see SVMs)

Then Eigendecomposition of K (kernel matrix)

e.g. $K(\vec{x}_i, \vec{x}_j) = e^{-\frac{\|\vec{x}_i - \vec{x}_j\|^2}{c}}$

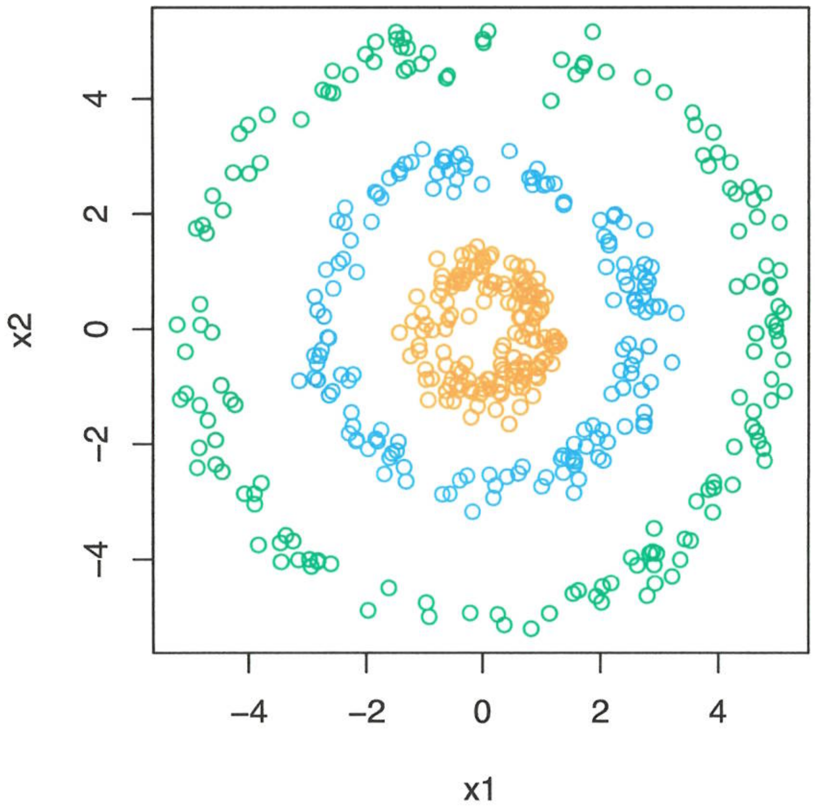
Radial Kernel

Example overleaf

We can do spectral analysis of the kernel (which is nonlinear in the original coordinates)

Imperial College London

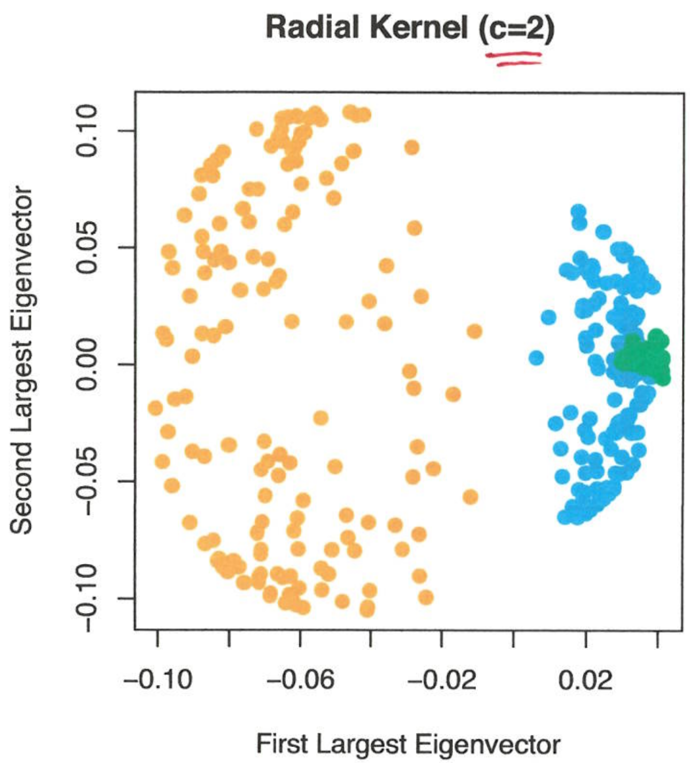
Representing these data in terms of standard PCA does not lead to a good mapping.

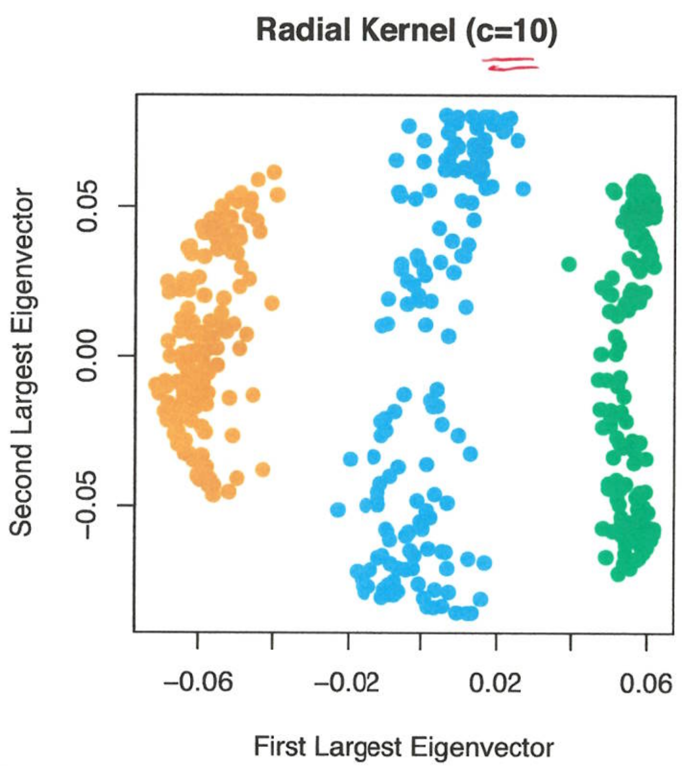


Imperial College London

Kernel PCA

Increasing the hyperparameter c





Second extension: Interpretability through sparsity.

Sparse PCA

Our description from standard PCA was of the form:

$$(y_{PCA})_{N\times K} = (U_K \Sigma_K)_{N\times K}$$

coordinates in terms of

$$V_K = (\vec{v}_1 \dots \vec{v}_K)_{p\times K}$$

The basis V is Non sparse in terms of our original descriptors

i.e. V is not a sparse matrix.

This leads to diminished interpretability

Original PCA:

$\vec{v} \in \mathbb{R}^p$

This is an alternative description of PCA

$$\left\{ \begin{array}{l} \max\limits_{\vec{v}} \; \vec{v}^T (\tilde{Y}\tilde{Y}^T)\vec{v} \\ \text{such that } \vec{v}^T\vec{v} = 1 \end{array} \right\}$$

$\boxed{\text{Sparser}}$ by adding extra constraints.

$$\sum_{j=1}^{p} |v_j| \leq t$$

Similar to LASSO

$$\underbrace{\{\vec{v}_{j,L}\}_{j=1}^{p}}_{\text{SVD LASSO basis}}$$

Find a basis that tries to minimize the quadratic error under sparsity constraints.

If we find a description in terms of a few vectors that are sparse then the description is closer to identifying the important combinations of descriptors.

$$\hat{\vec{y}} = \hat{f}(\lambda) = \sum_{j=1}^{k} \lambda_j \underbrace{\vec{v}_{j,L}}$$

$k <<$ from LASSO.

Then the matrix V will be sparser and the $\vec{v}_j$ will contain more zeros

$$\vec{v}_{j,L} = \begin{pmatrix} \blacksquare \\ 0 \\ 0 \\ 0 \\ \blacksquare \\ 0 \\ \vdots \\ 0 \end{pmatrix} \begin{array}{l} \to \text{coordinate for } x_1 \\ \\ \\ \\ \to \text{coordinate for } x_j \\ \\ \\ \\ \end{array}$$

Third extension: Interpretability through non-negativity

Non-negative matrix factorisation (NNM)

Standard PCA/SVD

$$\underset{N\times p}{\hat{\tilde{y}}^T} = \underset{N\times K}{(U_K \Sigma_K)} \underset{K\times p}{V_{K:}^T}$$

↓ the same matrix

$$\cancel{X} = \underbrace{(U_K \Sigma_K)}_{N\times K} \underbrace{V_{K:}^T}_{K\times p}$$

NNM:

Generalise to a factorisation:

$$X = \underset{N\times K}{W} \; H^T_{K\times p}$$

where we require positivity

$$w_{ij} \geq 0 \quad \forall \; i,j$$

$$h_{ij} \geq 0 \quad \forall \; i,j$$

It can be shown to be equivalent to:

$$\max_{W,H} \; L(W,H) = \sum_{i=1}^{N} \sum_{j=1}^{K} \left[X_{ij} \log (WH)_{ij} - (WH)_{ij} \right]$$

$X_{ij} \sim \text{Poi}$ with mean $(WH)_{ij}$

What does ~~it~~ NNM give?

See over →

for an example with images

Non negative matrix factorisation

$\{\vec{y}_i\}_{i=1}^{N}$ $\vec{y}_i \in \mathbb{R}^p$

Setup:

- Each $\vec{y}_i$ is an image 19x19 pixels, which is vectorized into a 361-dimensional vector.
- Number of images in sample $N=2429$

N=2429

p=381=19x19
k=49

$H^T_{49\text{x}381}$ picks the basis "features" of images

Each of the squares is a 381-dimensional vector

$\vec{W}_i^T$ (original) 1x49

all positive

VQ = K-means

The 49 faces closest to "Original"

This whole set of squares is the $(V_K)_{p \times k}$ matrix

Each of the squares is a $\vec{v}_j \in \mathbb{R}^{381}$

$\vec{u}_i$ (original) 1x49