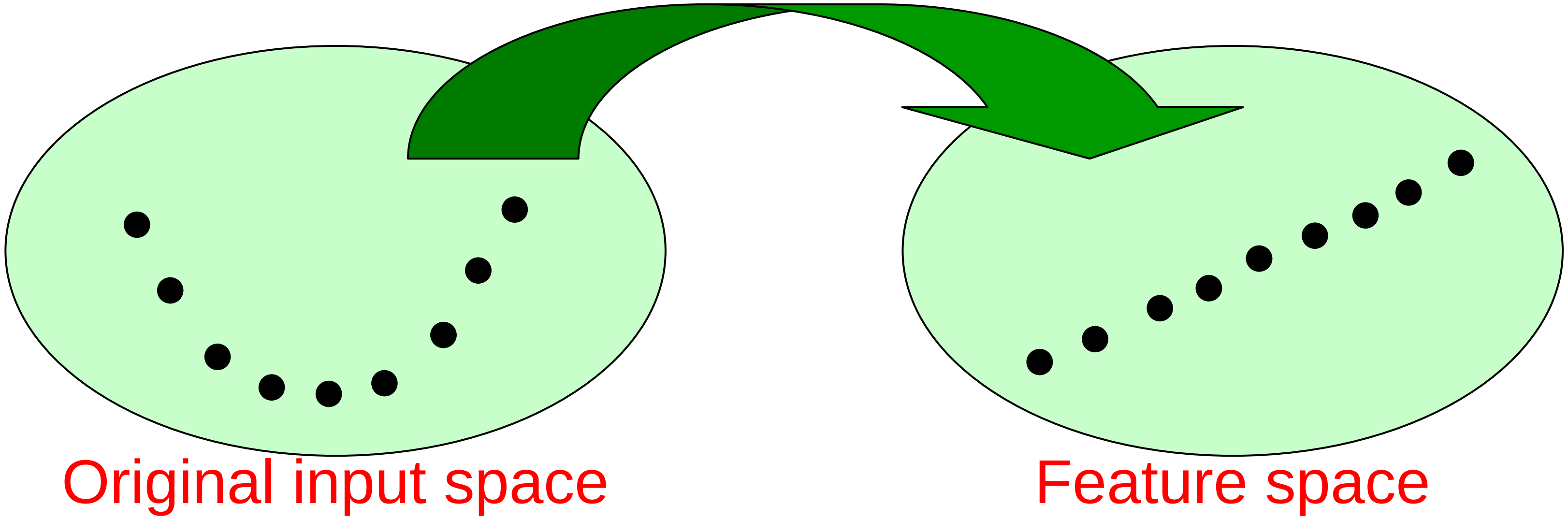
Non-Linearizing Linear Methods

- A simple non-linear extension of linear methods while keeping advantages of linear methods:
  - Map the original data to a feature space by a non-linear transformation
  - Run linear algorithm in the feature space



PCA in Feature Space

- Non-linear transformation: $\phi : \boldsymbol{x} \rightarrow \boldsymbol{f}$
- Feature vectors: $\{\boldsymbol{f}_i \mid \boldsymbol{f}_i = \phi(\boldsymbol{x}_i)\}_{i=1}^{n}$
- Centered feature vectors:

$$\boldsymbol{f}_i \longleftarrow \boldsymbol{f}_i - \frac{1}{n}\sum_{j=1}^{n}\boldsymbol{f}_j$$

- Eigenproblem: $\boldsymbol{C}_{\boldsymbol{f}}\boldsymbol{\psi} = \lambda\boldsymbol{\psi}$

$$\boldsymbol{C}_{\boldsymbol{f}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{f}_i\boldsymbol{f}_i^{\top}$$

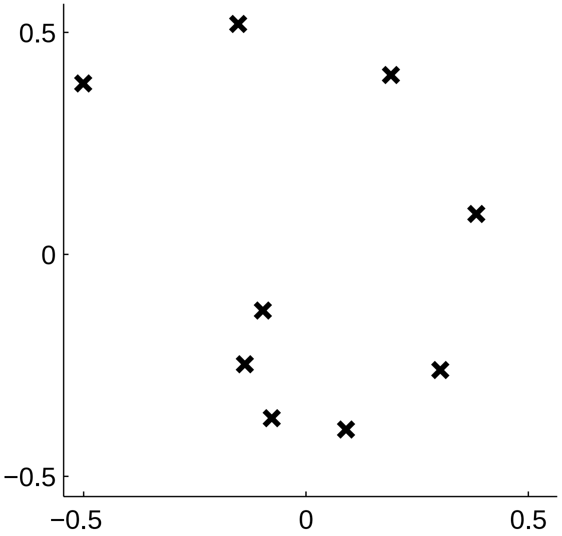
- PCA Projection: $\boldsymbol{g}_i = \boldsymbol{B}_{PCA}\boldsymbol{f}_i$

$$\boldsymbol{B}_{PCA} = (\boldsymbol{\psi}_1|\boldsymbol{\psi}_2|\cdots|\boldsymbol{\psi}_m)^{\top}$$

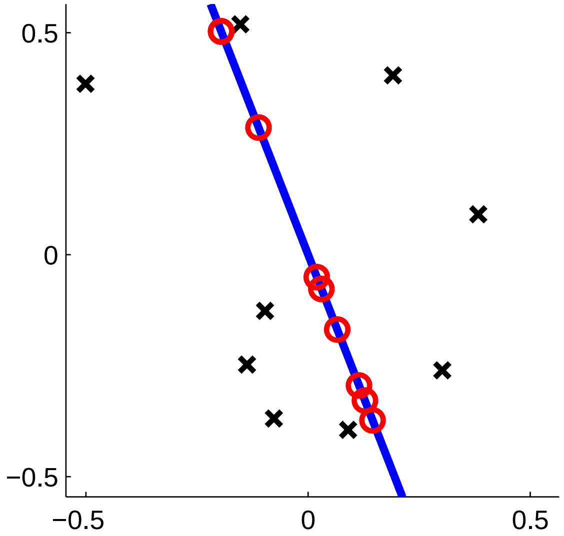
Example

- $d = 2$

Centered data
in input space



Linear PCA

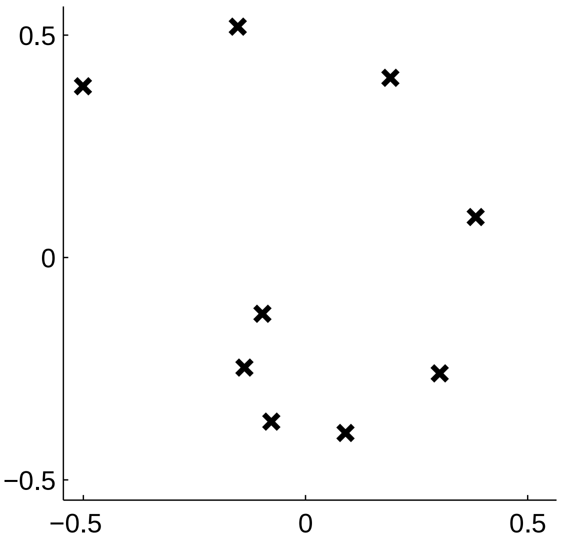


Example (cont.)

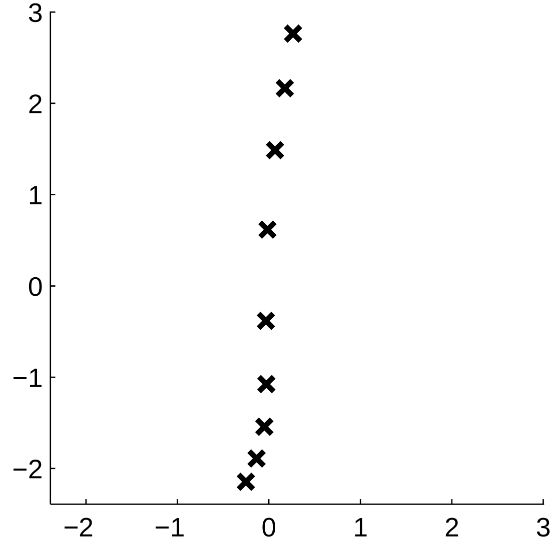
- $\boldsymbol{x} = (a, b)^\top$
- $\phi(\boldsymbol{x}) = (r, \theta)^\top$:Polar coordinate

$$r\cos\theta = a \qquad r\sin\theta = b$$

Centered data in input space



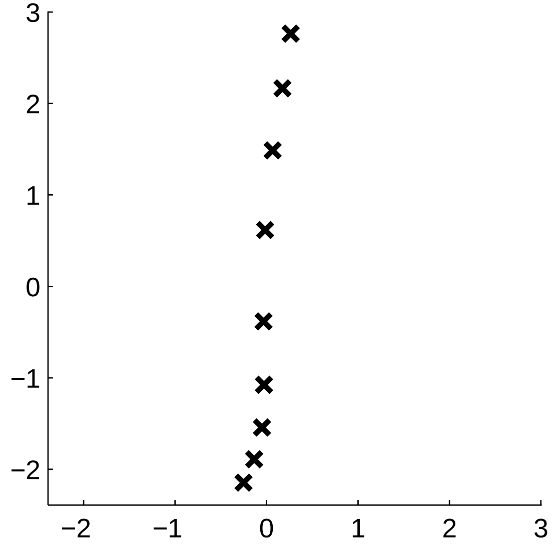
Centered data in feature space



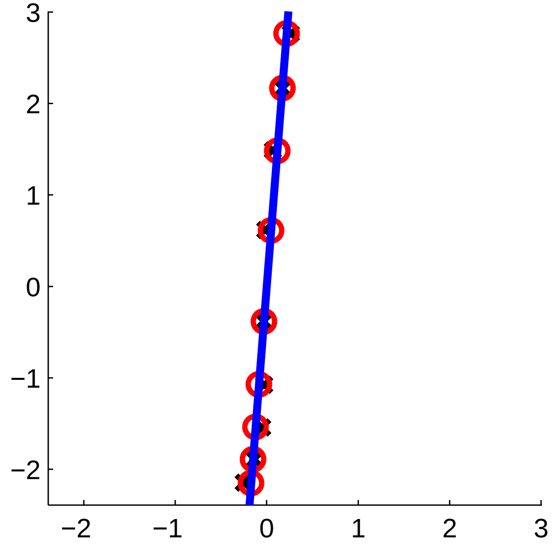
Example (cont.)

- Run PCA in feature space.

Centered data
in feature space

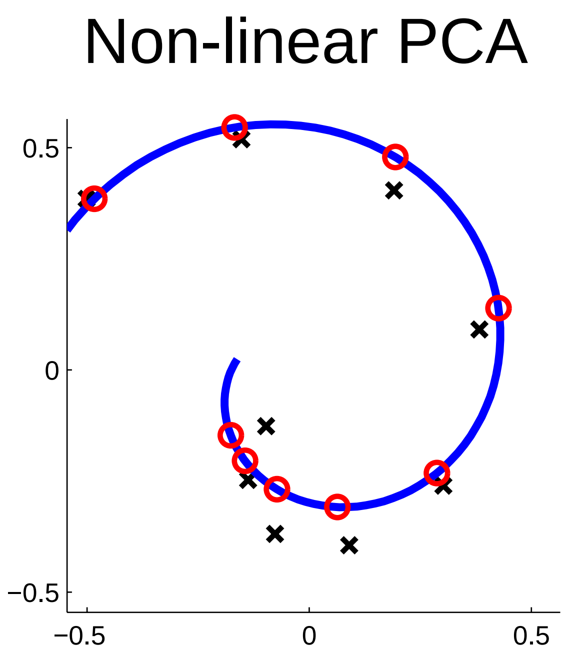


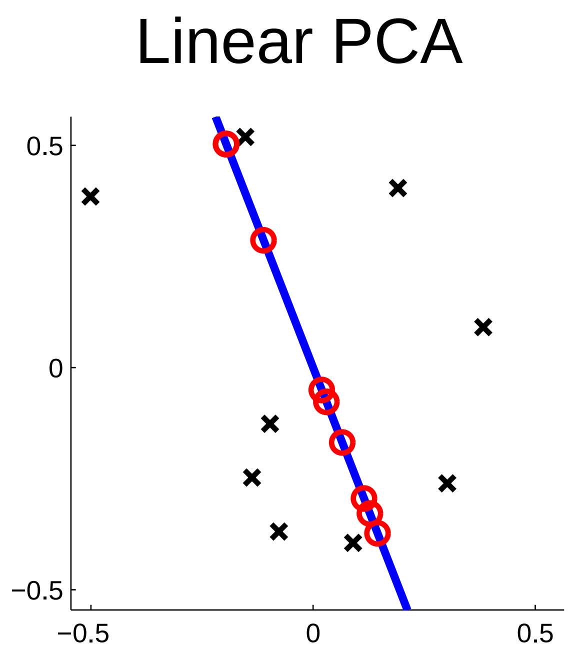
PCA projection
in feature space



Example (cont.)

- Pull the results back to input space.





- Non-linear PCA describes the original data much better than linear PCA.

PCA in High-Dimensional Feature Space

- If $\dim(\mathcal{H})$ is high, description ability of non-linear PCA would be better.
- However, when $\dim(\mathcal{H})$ is very large, PCA in feature space is computationally demanding.
- We need a trick to reduce computational burden.

Dual Eigenproblem

$$\boldsymbol{C}_{\boldsymbol{f}}\boldsymbol{\psi} = \lambda\boldsymbol{\psi}$$

- Since $\boldsymbol{C}_{\boldsymbol{f}} = \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{f}_i\boldsymbol{f}_i^\top$,

$$^{\exists}\boldsymbol{\alpha},\ \ \boldsymbol{\psi} = \sum_{j=1}^{n}\alpha_j\boldsymbol{f}_j$$

- Eigenproblem is

$$\frac{1}{n}\sum_{i,j=1}^{n}\alpha_j\langle\boldsymbol{f}_i,\boldsymbol{f}_j\rangle\boldsymbol{f}_i = \lambda\sum_{i=1}^{n}\alpha_i\boldsymbol{f}_i$$

Dual Eigenproblem (cont.)

$$\frac{1}{n}\sum_{i,j=1}^{n}\alpha_j\langle \boldsymbol{f}_i,\boldsymbol{f}_j\rangle \boldsymbol{f}_i=\lambda\sum_{i=1}^{n}\alpha_i\boldsymbol{f}_i$$

- In matrix form, $\boldsymbol{F}\boldsymbol{K}\boldsymbol{\alpha}=\lambda'\boldsymbol{F}\boldsymbol{\alpha}$

$$\boldsymbol{F}\equiv(\boldsymbol{f}_1|\boldsymbol{f}_2|\cdots|\boldsymbol{f}_n)$$

$$\boldsymbol{K}_{i,j}\equiv\langle \boldsymbol{f}_i,\boldsymbol{f}_j\rangle \quad \lambda'\equiv n\lambda$$

- Solution of the above eigenproblem is given by $\boldsymbol{K}\boldsymbol{\alpha}=\lambda'\boldsymbol{\alpha}$

$$(\lambda'_1\geq\lambda'_2\geq\cdots\geq\lambda'_n) \qquad (\langle\boldsymbol{\alpha}^{(i)},\boldsymbol{\alpha}^{(j)}\rangle=\delta_{i,j})$$

Kernel Trick

$$\boldsymbol{K}\boldsymbol{\alpha} = \lambda' \boldsymbol{\alpha}$$

- Given $\boldsymbol{K}$, solving dual is faster than primal for $\dim(\mathcal{H}) > n$.
- However, calculating $\boldsymbol{K}$ is still painful.
- "Kernel trick": For some transformation $\phi(\boldsymbol{x})$,

$$^{\exists}K(\boldsymbol{x}, \boldsymbol{x}') \;\; s.t. \;\; K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \boldsymbol{f}_i, \boldsymbol{f}_j \rangle$$

$$\boldsymbol{f}_i = \phi(\boldsymbol{x}_i)$$

- $K(\boldsymbol{x}, \boldsymbol{x}')$:Kernel function

Kernels

$$K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \boldsymbol{f}_i, \boldsymbol{f}_j \rangle = \langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle$$

- Rather than directly defining $\phi(\boldsymbol{x})$, we implicitly specify $\phi(\boldsymbol{x})$ by $K(\boldsymbol{x}, \boldsymbol{x}')$.
- Kernel matrix: $\boldsymbol{K}_{i,j} \equiv K(\boldsymbol{x}_i, \boldsymbol{x}_j)$
- Implicit mapping $\phi(\boldsymbol{x})$ exists if $K(\boldsymbol{x}, \boldsymbol{x}')$ is s.t.
  - $\boldsymbol{K}$ is symmetric: $\boldsymbol{K}^\top = \boldsymbol{K}$
  - $\boldsymbol{K}$ is positive semi-definite: $\forall \boldsymbol{y}, \ \langle \boldsymbol{K}\boldsymbol{y}, \boldsymbol{y} \rangle \geq 0$
- Such $K(\boldsymbol{x}, \boldsymbol{x}')$ is often called the Mercer kernel or reproducing kernel.

Examples of Kernels

- Polynomial kernel:

$$K(\boldsymbol{x}, \boldsymbol{x}') = \langle \boldsymbol{x}, \boldsymbol{x}' \rangle^c \qquad c \in \mathbb{N}$$

  - $d = 2 \quad c = 2$

$$\boldsymbol{x} = (s, t)^\top$$

$$\langle \boldsymbol{x}, \boldsymbol{x}' \rangle^2 = (ss' + tt')^2$$
$$= sss's' + 2ss'tt' + ttt't'$$

$$\Rightarrow \quad \boldsymbol{f} = \phi(\boldsymbol{x}) = (s^2, \sqrt{2}st, t^2)^\top$$
$$\dim(\mathcal{H}) = 3$$

  - In general,

$$\dim(\mathcal{H}) = \binom{c+d-1}{c}$$

Examples of Kernels (cont.)

- Gaussian kernel:

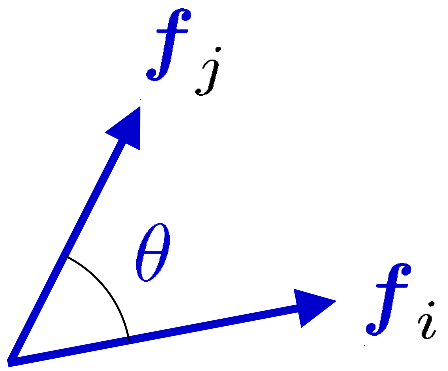
$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\|\boldsymbol{x} - \boldsymbol{x}'\|^2 / c\right) \qquad c > 0$$

$$\dim(\mathcal{H}) = \infty$$

Playing with Kernel Trick

$$\langle \boldsymbol{f}_i, \boldsymbol{f}_j \rangle = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

- Exercise: Prove the following
  - $\|\boldsymbol{f}_i - \boldsymbol{f}_j\|^2 = K(\boldsymbol{x}_i, \boldsymbol{x}_i) - 2K(\boldsymbol{x}_i, \boldsymbol{x}_j) + K(\boldsymbol{x}_j, \boldsymbol{x}_j)$
  - $\cos\theta = \dfrac{K(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sqrt{K(\boldsymbol{x}_i, \boldsymbol{x}_i)K(\boldsymbol{x}_j, \boldsymbol{x}_j)}}$



- For Gaussian kernel
  - $\|\boldsymbol{f}_i - \boldsymbol{f}_j\|^2 = 2 - 2K(\boldsymbol{x}_i, \boldsymbol{x}_j)$
  - $\cos\theta = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$

Centering in Feature Space

- For implicit feature map, how do we center $\{\boldsymbol{f}_i\}_{i=1}^n$?

$$\boldsymbol{f}_i \longleftarrow \boldsymbol{f}_i - \frac{1}{n}\sum_{j=1}^{n}\boldsymbol{f}_j$$

- We only need centered kernel matrix.
- Centered kernel matrix is given by

$$\boldsymbol{K} \longleftarrow \boldsymbol{K} - \boldsymbol{N}\boldsymbol{K} - \boldsymbol{K}\boldsymbol{N} + \boldsymbol{N}\boldsymbol{K}\boldsymbol{N}$$

$$\boldsymbol{N}_{i,j} = 1/n$$

Homework

- Prove that the centered kernel matrix is given by

$$\boldsymbol{K} \longleftarrow \boldsymbol{K} - \boldsymbol{N}\boldsymbol{K} - \boldsymbol{K}\boldsymbol{N} + \boldsymbol{N}\boldsymbol{K}\boldsymbol{N}$$

$$\boldsymbol{N}_{i,j} = 1/n$$