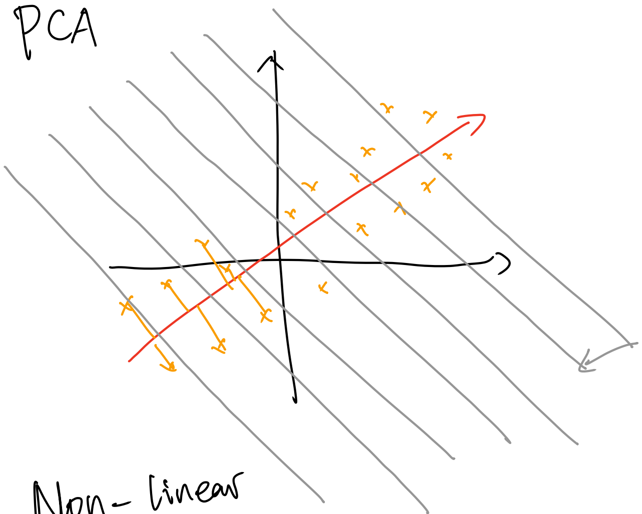
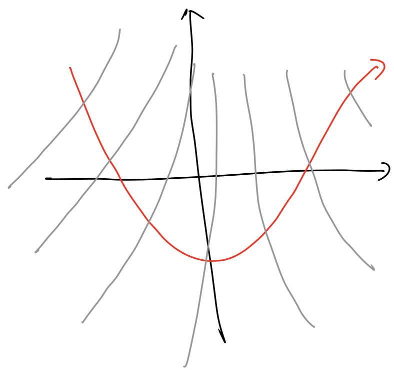
PCA



all point on this line have the same PCA coefficient

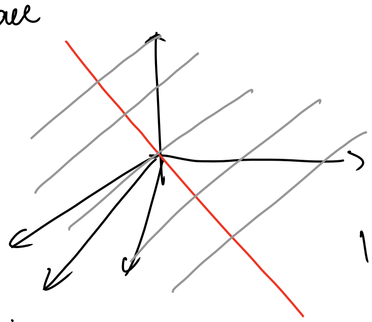
Non-linear



we want to poject onto non-linear function

non-linear function in original space

Φ : apply x transformation & kernel trick



linear transform in high-dim space

Kernel PCA:

Data $\{x_1, \cdots, x_N\}$

Apply feat transform: $x_i \rightarrow \Phi(x_i)$

$$[x_1 \cdots x_N] = X \rightarrow \Phi = [\Phi(x_1) \cdots \Phi(x_N)]$$

- Assume x-formed data is centered: $\sum_i \phi(x_i) = 0$

- covariance matrix in high-dim space

$$C = \frac{1}{N} \sum_{i=1}^{N} \phi(x_i)\phi(x_i)^T \quad \text{(Note: zero mean)}$$

- Find (λ_j, v_j) eigen value / eigen vector

$$C v_j = \lambda_j v_j$$

$$\underbrace{\frac{1}{N}}_{\text{scalar}} \sum_{m=1}^{N} \phi(x_m) \underbrace{\phi(x_m)^T v_j}_{a_{m,j} \to \text{scalar}} = \underbrace{\lambda_j}_{\text{scalar}} v_j$$

Hence, v_j has the form: $v_j = \sum_{i=1}^{N} a_{ij} \phi(x_i) = \Phi a_j$

where $a_j = \begin{bmatrix} a_{1j} \\ \vdots \\ a_{nj} \end{bmatrix}$

i.e. eigenvector is a linear combination of $\phi(x_i)$'s

$\Rightarrow$ we need to find a_j

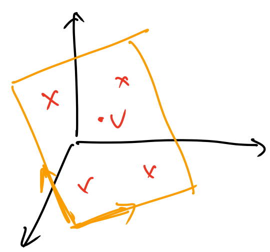
<u>Substitute for v_j</u>

$$\frac{1}{N} \underbrace{\sum_m \phi(x_m) \phi(x_m)^T}_{} \Phi a_j = \lambda_j \Phi a_j$$

$$\frac{1}{N} \Phi \Phi^T \Phi a_j = \lambda_j \underbrace{\Phi a_j}_{}$$

eigen vector is a linear combo of $\phi(x_i)$ = span $(\phi(x_1) \dots \phi(x_N))$

An equivalent set of equs is to project in to the coordinates of span of Φ and solve there.



$\Rightarrow$ premultiply by Φ^T

$$\Rightarrow \frac{1}{N} \underbrace{\Phi^T \Phi}_{K} \underbrace{\Phi^T \Phi}_{K} a_j = \lambda_j \underbrace{\Phi^T \Phi}_{K.} a_j$$

kernel trick.

$\Rightarrow K K a_j = N \lambda_j K a_j$

① if K is invertible, then solve $K a_j = N \lambda_j a_j$

② if K is not invertible the only difference is the eigenvectors w/ $\lambda_j = 0$ but those are not PC's because $\lambda = 0$

solve ① anyway.

verify: suppose (a_j, λ_j) is eigenvector/value of K.

s.t. $K a_j = N\lambda_j K a_j$, $\lambda_j \neq 0$, $a_j^T a_j = 1$

original eigen

$$K\underbrace{K a_j}_{N\lambda_j a_j} = N\lambda_j \underbrace{K a_j}_{N\lambda_j a_j}$$

$$N\lambda_j \underbrace{K a_j}_{N\lambda_j a_j} = \underbrace{N\lambda_j N\lambda_j a_j}_{N^2\lambda_j^2 a_j}$$

$$N^2\lambda_j^2 a_j = N^2\lambda_j^2 a_j \checkmark$$

PC should be normalized

$$v_j^T v_j = 1 = a_j^T \Phi^T \Phi a_j = a_j^T K a_j$$
$$= N\lambda_j a_j^T a_j$$

Thus, rescale

$$a_j \leftarrow \frac{1}{\sqrt{N\lambda_j}} a_j$$

Kernel centering

How to center the feature space?

centered feature

$$\tilde{\phi}(x_i) = \phi(x_i) - \underbrace{\frac{1}{N}\sum_{k=1}^{N}\phi(x_k)}_{\text{means in f.s}}$$

$$= \phi(x_i) - \frac{1}{N}\Phi\mathbf{1} \leftarrow \mathbf{1} \text{ vector} = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

Centered data

$$\tilde{\Phi} = \Phi - \left[\frac{1}{N}\Phi\mathbf{1}\right]\mathbf{1}^T = \Phi - \frac{1}{N}\Phi\mathbf{1}\mathbf{1}^T = \Phi\left(I - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)$$

Centered kernel

$$\tilde{K} = \tilde{\Phi}^T\tilde{\Phi} = \underbrace{\left(I - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)^T}_{\text{symmetric}}\Phi^T\Phi\left(I - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)$$

$$\tilde{K} = \left(I - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)K\left(I - \frac{1}{N}\mathbf{1}\mathbf{1}^T\right)$$

test kernel:

$$\tilde{k}(x_*, x_j) = k(x_*, x_j) - \frac{1}{N}k_*^T\mathbf{1} - \frac{1}{N}k_j^T\mathbf{1} + \frac{1}{N^2}\mathbf{1}^T K\mathbf{1}$$

← jth row of K.

Summary: KPCA

1) calculate kernel matrix: $K = [k(x_i, x_j)]_{ij}$

2) center the kernel: $\tilde{K} = (I - \frac{1}{N}\mathbf{1}\mathbf{1}^T) K (I - \frac{1}{N}\mathbf{1}\mathbf{1}^T)$

3) Find the top D eigenvectors: $\tilde{K} a_j = \lambda_j a_j$, $j = 1 \dots D$

4) Scale $a_j \leftarrow \frac{1}{\sqrt{\lambda_j}} a_j$ (N可以并入λj中，所以这里和上面略有不同，本质一样)

5) project data x_*: $z_{*,j} = \tilde{k}_*^T a_j$

$$\boxed{z_{*j} = \sum_{i=1}^{N} a_{ij} \tilde{k}(x_*, x_i)}$$

Note: original problem needs $\underline{d\text{-dim}}$ eigenvector

kernel problem needs $\underline{N\text{-dim}}$ eigenvector

pick problem that is more efficient.

PS 10-15: kernelized FLD → kernel discriminant analysis.

Pre-image Problem

Given PCA coeff z, we can reconstruct $\hat{x}$, e.g. denoisy x

$$\hat{x} = \sum_j v_j z_j$$

What about KPCA?

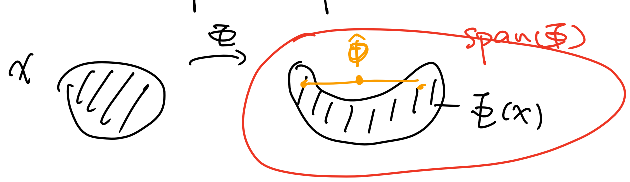
- Given z, what is the high dimension feature?

$$\hat{\phi} = \sum_j v_j z_j = \sum_j \Phi a_j z_j = \Phi\Big(\sum_j a_j z_j\Big)$$

linear combo of $\Phi(x_i)$

$\hat{\phi} \in \mathrm{span}(\Phi)$

But, not all points span(Φ) have a corresponding x!



Example : let $\phi(x) = \begin{bmatrix} x^2 \\ x \end{bmatrix}$

$x_1 = 1 \Rightarrow \Phi(x_1) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

$x_2 = 2 \Rightarrow \Phi(x_2) = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$

let $\hat{\phi} = \phi(x_1) + \phi(x_2) = \begin{bmatrix} 5 \\ 3 \end{bmatrix}$, what is $\hat{x}$ s.t $\hat{\phi} = \phi(\hat{x})$?

there is no $\hat{x}$ possible

Approximate Pre-Image

general problem $\hat{\phi} = \Phi\alpha = \sum_i \alpha_i \phi(x_i)$ ← given

Find $\hat{x} = \arg\min_x \|\phi(x) - \hat{\phi}\|^2$ ← find the x that gives the closest ϕ in the f.s. (feature space)

$$\hat{x} = \arg\min_x k(x,x) - 2\sum_i \alpha_i k(x, x_i) + \underbrace{\sum_i \sum_j \alpha_i \alpha_j k(x_i, x_j)}_{\text{constant}}$$

$$= \arg\min_x k(x,x) - 2\sum_i \alpha_i k(x, x_i)$$

Soln 1 : nearest neighbour.

select $x \in X$ (traing data)

Soln 2: solve optimization problem numerically w/ package

Soln 3 : suppose $k(x,x) = 1$, $k(x_i, x_j) \geq 0$ (e.g Gaussian)

$$\hat{x} = \arg\min_x - \sum_i \alpha_i k(x, x_i)$$

$$= \arg\max_x \underbrace{\sum_i \alpha_i k(x, x_i)}_{\langle \Phi(x), \hat{\Phi} \rangle}$$

$$= \arg\max \left(\sum_i \alpha_i k(x, x_i)\right)^2$$

assume homogenous kernel , $k(\|x - x'\|^2)$

iteractive algorithm

$$\boxed{\hat{x} \leftarrow \frac{\sum_i \alpha_i k'(\|x_i - \hat{x}\|^2) x_i}{\sum_i \alpha_i k'(\|x_i - \hat{x}\|^2)}}$$

similiar to mean-shift w/ weights α_i on each point x_i