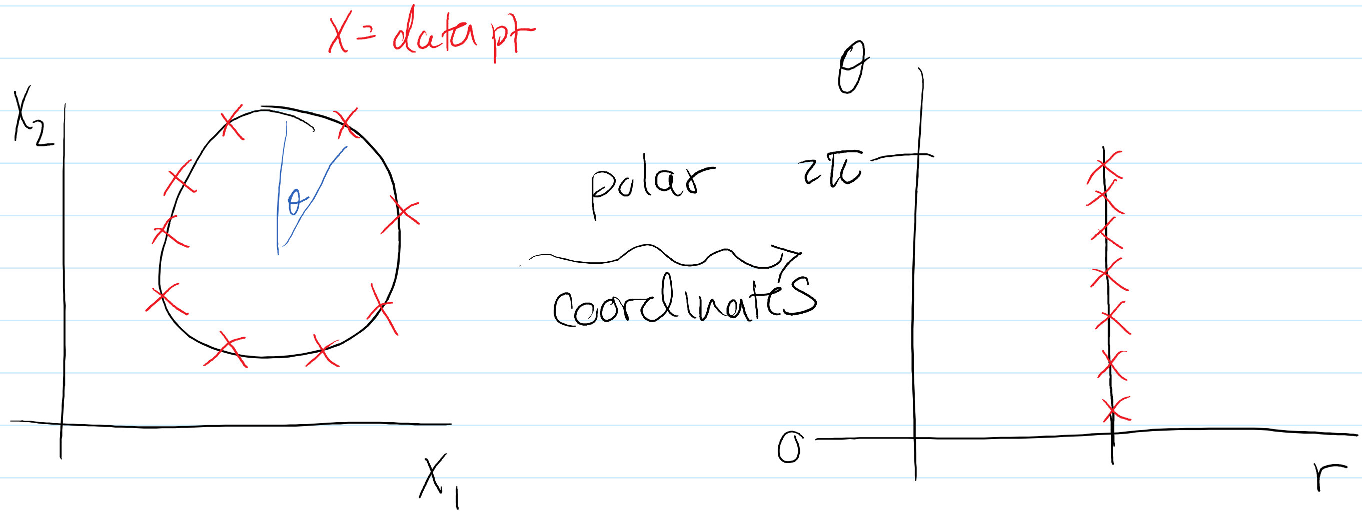
PCA: only can recover a linear subspace.

Ex.



technically 2D

Q: If I embed my data into some (potentially high-dimensional space) through some transformation - maybe(?) I can better summarize my data using PCA.

Non-linear PCA

$$\underset{N\times P}{X} = \begin{bmatrix} — & x_1 & — \\ — & x_2 & — \\ & \vdots & \\ — & x_N & — \end{bmatrix}, \quad x_n \in \mathbb{R}^P$$

x_n ↖ obs of P variables

embedding map: $\varphi : \mathbb{R}^P \to \mathbb{R}^D$

(typically $D \gg P$)

(typically $D \gg P$)

$$\Phi_{N\times D} = \begin{bmatrix} — \varphi(x_1) — \\ — \varphi(x_2) — \\ \vdots \\ — \varphi(x_N) — \end{bmatrix} = \text{embedded data Matrix}$$

NLPCA: do PCA on this embedding!

(0) Form Φ = embedded data mtx

(1) center data

$$\bar{\varphi} = \frac{1}{N}\sum_{n=1}^{N}\varphi(x_n) \in \mathbb{R}^D$$

(2) $$\tilde{\Phi} = \begin{bmatrix} — \varphi(x_1)-\bar{\varphi} — \\ — \varphi(x_2)-\bar{\varphi} — \\ \vdots \\ — \varphi(x_N)-\bar{\varphi} — \end{bmatrix}$$

$$\bar{\varphi} = \frac{1}{N}\mathbb{1}_N^T\Phi$$

$$\tilde{\Phi} = \Phi - \mathbb{1}_N\bar{\varphi}$$

$$= \Phi - \mathbb{1}_N\frac{\mathbb{1}_N^T}{N}\Phi$$

$$= \underbrace{\left(I - \frac{\mathbb{1}\mathbb{1}^T}{N}\right)}_{J_N \,=\, \text{centering mtx}}\Phi$$

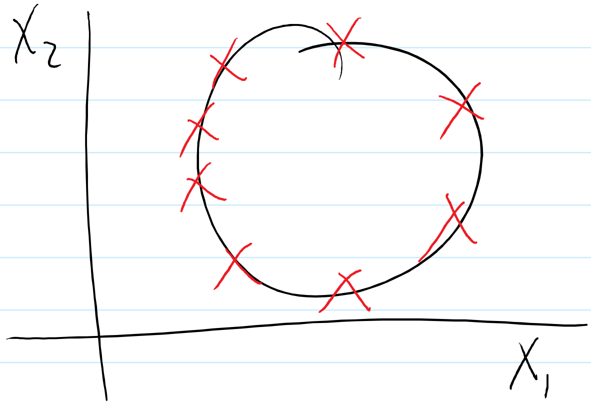
(3) SVD: $\tilde{\Phi} = U\Sigma V^T$

Σ = diag(σ_i)

(4) $Z = \tilde{\Phi}V_{1:q} \in \mathbb{R}^{N\times q}$

Ex. $P=2$ so $x_n = (x_{n1}, x_{n2})$; $D=3$

let $$\varphi(x_n) = \begin{pmatrix} x_{n1}^2 \\ \sqrt{2}\,x_{n1}x_{n2} \\ x_{n2}^2 \end{pmatrix} = \begin{pmatrix} z_1 \\ z_2 \\ z_3 \end{pmatrix}$$

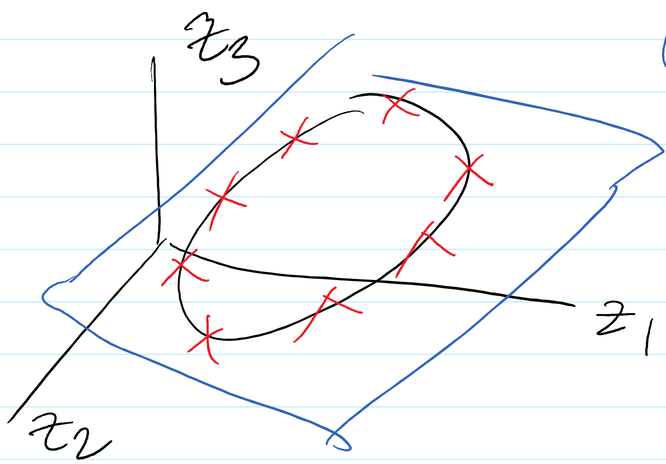


my data satisfies:

$$a x_{n1}^2 + b x_{n2}^2 + c x_{n1} x_{n2} + d = 0$$

general eqn for circle (conic)

φ



$$a z_1 + b z_2 + c z_3 + d = 0$$

eqn for a plane

So we can recover these as a <u>linear</u> subspace!

Calculate PCA using SVD.

$$\tilde{\bar{\Phi}}_{N\times D} = U \Sigma V^T \iff \boxed{\bar{\Phi} V = U \Sigma}$$

U: $\tilde{\bar{\Phi}}\tilde{\bar{\Phi}}^T_{N\times N}$ left gram mtx

Σ: e-vals of either, $\sigma_i = \sqrt{\lambda_i}$

λ_i = e-vals of either

V: $\tilde{\bar{\Phi}}^T\tilde{\bar{\Phi}}_{D\times D}$ right gram mtx

more attractive to use b/c $N \ll D$

So alt to $Z = \tilde{\Phi} V_{1:q}$ I could calc $Z = (U\Sigma)_{1:q}$

by e-decomp $\tilde{\Phi}\tilde{\Phi}^T = U \Lambda U^T \quad = U_{1:q} \Lambda_{1:q}^{1/2}$

Quick NLPCA:

0. Calc Φ
1. $\tilde{\Phi} = J_N \Phi$
2. $K = \tilde{\Phi}\tilde{\Phi}^T_{N \times N}$ ← Kernel matrix
3. $K = U \Lambda U^T$
4. $Z = U_{1:q} \Lambda^{1/2}_{q}$ ← $\text{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_q})$

Q! Can I avoid Φ all together?

Start at step 3 using K that I could calc w/o need to even know Φ.

Ex.

$$\varphi(x_n) = \begin{pmatrix} x_{n1}^2 \\ \sqrt{2}\, x_{n1} x_{n2} \\ x_{n2}^2 \end{pmatrix} \leftarrow$$

(assume Φ centered)

$$K_{nn'} = (\Phi\Phi^T)_{nn'}$$

$$= \text{row}(n,\Phi)^T \text{col}(n',\Phi^T)$$

$$= \text{row}(n,\Phi)^T \text{row}(n',\Phi)$$

$$\boxed{= \varphi(x_n)^T \varphi(x_{n'})}$$ = dot prod btw embeddings of data

$$= (x_{n1}^2, \sqrt{2}\, x_{n1}x_{n2}, x_{n2}^2)^T (x_{n'1}^2, x_{n'1}x_{n'2}\sqrt{2}, x_{n'2}^2)$$

$$= x_{n1}^2 x_{n'1}^2 + 2 x_{n1}x_{n2}x_{n'1}x_{n'2} + x_{n2}^2 x_{n'2}^2$$

$$(x_{n1}x_{n'1} + x_{n2}x_{n'2})^2$$

$$\boxed{= (x_n^T x_{n'})^2}$$

$k(x_n, x_{n'})$

only need orignal data to calc K (no φ req)

Don't need φ, just need $k(x,y) = \varphi(x)^T\varphi(y)$

Ex. 1

(1) $\varphi(a) = \begin{pmatrix} a_1^2 \\ \sqrt{2}\, a_1 a_2 \\ a_2^2 \end{pmatrix} \longrightarrow k(x,y) = (x^T y)^2$

(1) $\varphi(a) = \left(a_2^2 \right) \longrightarrow k(x,y) = (x^T y)$

$P = 2 \,;\, D = 3$

(2) $\varphi(a) = \begin{pmatrix} a_1^2 \\ a_1 a_2 \\ a_1 a_3 \\ \vdots \\ a_2^2 \\ a_2 a_3 \\ \vdots \\ a_3^2 \\ \vdots \end{pmatrix} \longrightarrow \boxed{k(x,y) = (x^T y)^2}$

P larger

all degree 2 monomials

$$D = \binom{P+1}{2}$$

(3) $\varphi(a) = \begin{pmatrix} \text{all monomials of} \\ \text{degree } M \end{pmatrix} \longrightarrow \boxed{k(x,y) = (x^T y)^M}$

$$D = \binom{M+P-1}{M}$$

Follow-on from Mercer's Theorem.

We can generally do this.

If $k : \mathbb{R}^2 \longrightarrow \mathbb{R}$ and k is

(PSD kernel fn)

(1) <u>symmetric</u>: $k(x,y) = k(y,x)$

(2) <u>Pos. semi-Def</u>: $\iint K(x,y) f(x) f(y) \, dx \, dy \geq 0$

ker (fn)

(2) Pos. Semi-Def: $\iint K(x,y) f(x) f(y)\, dx\, dy \geq 0$

(3) Banded

then $\exists\ \varphi: \mathbb{R}^P \to \mathbb{R}^M$

so that

$$k(x,y) = \varphi(x)^T \varphi(y)$$

Called Kernel Trick: implicitly working in high dim'l space w/o needing to actually calc. the embedding.

e.g. $k(x,y) = \exp(-\gamma \|x-y\|^2)$ [Gaussian]

$$D = \infty$$

Centering:

$$\tilde{\Phi} = J_N \Phi$$

want to work w/ $\tilde{K} = \tilde{\Phi}\tilde{\Phi}^T$

$$= J_N \Phi \Phi^T J_N^T$$

$$= J_N K J_N$$

KPCA: pick q, k (implict φ)

1. Calc K

$$K_{nn'} = k(x_n, x_{n'})$$

2. $\tilde{K} = J_N K J_N$

3. $\tilde{K} = U \Lambda U^T$

4. $Z = U_{1:q} \Lambda_q^{1/2}$