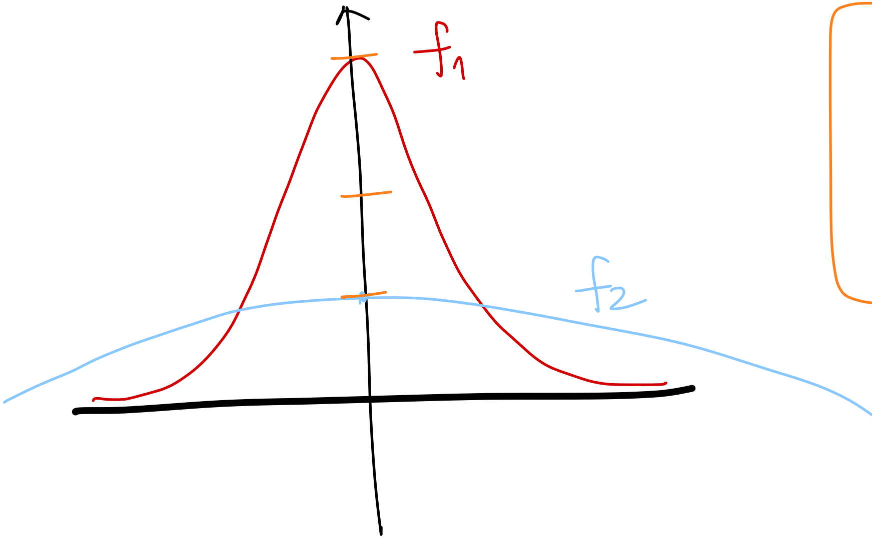


Lecture 21

Do Well!
20 min

PCA
CV



$$\boxed{\begin{array}{c}\sigma_1 = 1 \\[1em] \sigma_2 = 3\end{array}}$$

$$f_1(0) = \frac{1}{\sqrt{2\pi}}$$

$$f_3(0) = \frac{1}{3}\,\frac{1}{\sqrt{2\pi}}$$

on Q3 Pb 4

Predictors

Bias - Var

Training

Neural Architectures

Unsupervised

Statistical

CV

Regularisation

ML & People

Lecture Notes VII – Principal Component Analysis

Marina Meilă
mmp@uwaterloo.ca

With Thanks to Pascal Poupart & Gautam Kamath
Cheriton School of Computer Science
University of Waterloo

March 19, 2026

Eigendecompositions of Variance

1. $Var(x) \equiv \Sigma = \frac{1}{n} X^T X = \frac{1}{n} V \tilde{\Sigma} V^T$

2. $G = X X^T = U \tilde{\Sigma} U^T$

3. $X = U \tilde{\Sigma} V^T \leftarrow \boxed{\text{SVD}}$

$u^i_j = u_{ij} \tilde{\sigma}_j$

$\Rightarrow \tilde{x}^i = \underbrace{u^i_1 v_1 + \cdots u^i_d v_d}$ optimal approximation

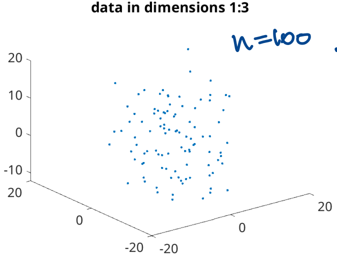
new basis vectors

$\tilde{x} \in \mathbb{R}^d$

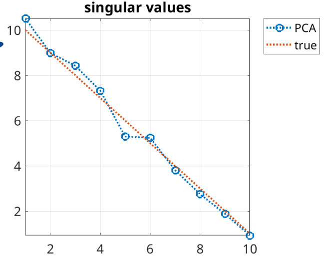
$$x^i = \begin{bmatrix} x^i_1 \\ \vdots \\ x^i_D \end{bmatrix} \rightarrow \tilde{x}^i = \begin{bmatrix} u^i_1 \\ \vdots \\ u^i_d \end{bmatrix}$$

$$\underbrace{\tilde{\sigma}_1 \geq \tilde{\sigma}_2 \geq \cdots \tilde{\sigma}_d}_{\text{Principal } d \text{ e-values, e-vectors}} \geq \cdots$$

Example – Gaussian data

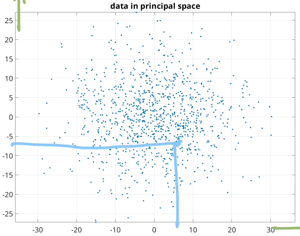


$n=100$

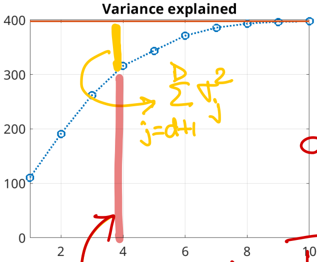


$\sigma_j = 11 - j$

$D = 10$



v_2 v_1 u_2^i u_1^i



$\sum_{j=d+1}^{D} \sigma_j^2$

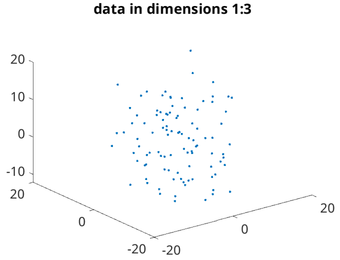
choosing d

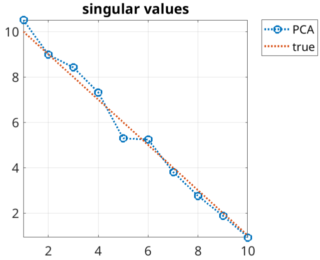
d

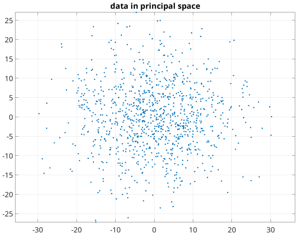
D

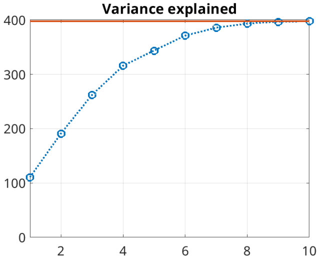
Var explained

Example – Gaussian data



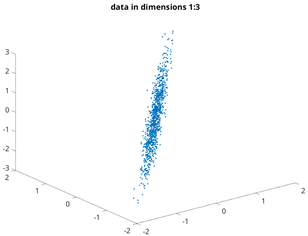


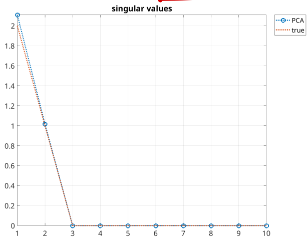


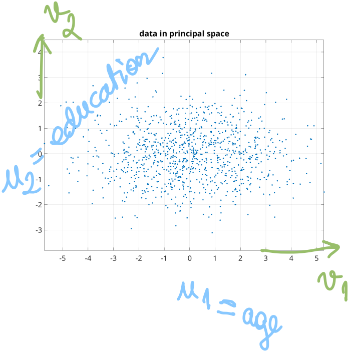


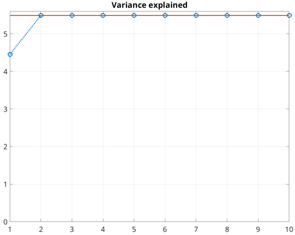
Example – Gaussian data 2D

PC Analysis

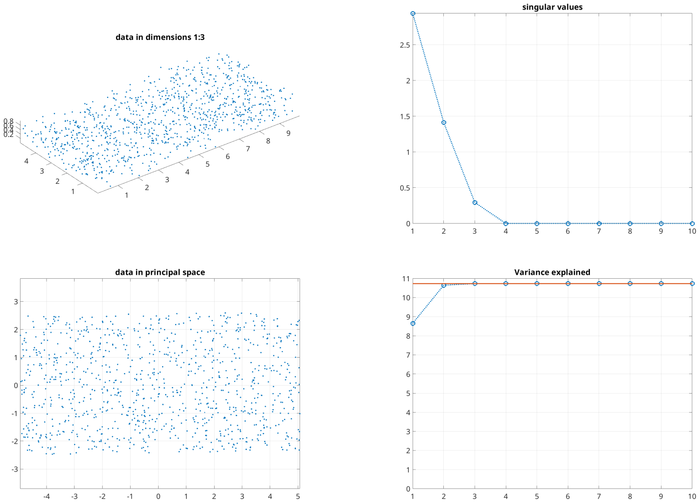








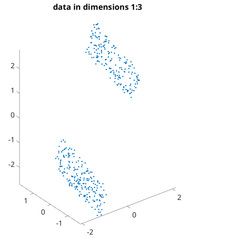
Example – Brick

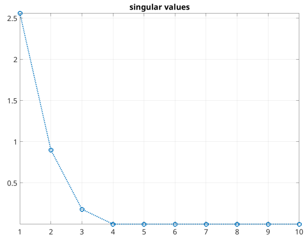


Example – clusters

In general K clusters PCA(K−1)

$K=2$





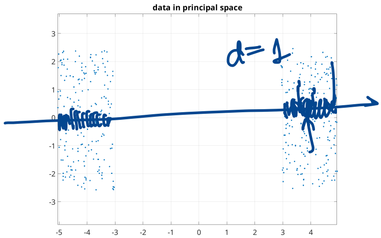
$d=K-1$

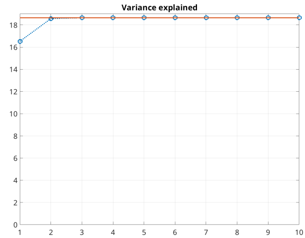
⇓

clustering easier after PCA

⇔

in high dim

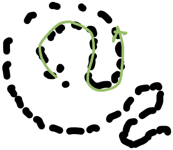




PCA Summary

- Reduces data dimensiont from D to d
- Linear operation (projection) → manifold learning, embedding, non-linear dim reduction
- "Optimal" linear method to reduce dimension
- Can discover if data is low-dimensional
- For clustering – recommended pre-processing: PCA in $K-1$ dimensions
- Limitation: fails to discover non-linear low dimensional structure

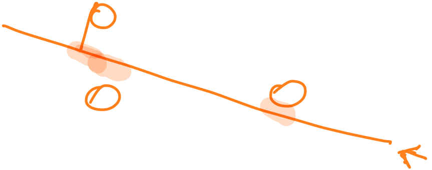
- Low Rank approx
- Intepretation



$d=1$

$D=2$

PCA: $d=1$



V_1

Torus

$D=2$
$d=2$

Model Selection

- Selecting K for clustering
- —"— K in K-nearest n.
- comparing 2 neural networks
- comparing different types of predictors $\{DT, K\text{-}nn, \text{logistic } r, \ldots\}$

Given $\mathcal{D}$, Loss $L_{01}, L_{LS}, \ldots$

1. Train predictors $f_1, f_2, \ldots f_M$
2. Do Model Selection $(= \text{compare } f_{1:M})$

$$f^* = \underset{\min}{\operatorname{argmax}} \{ \text{score}(f_m) \; m=1:M \}$$

- BIC – only for ML estimation, cheap → score
- CV – any predictors, expensive → score L^{valid} = validation loss

Cross Validation

Assume $\mathcal{D}'$ available $|\mathcal{D}'| = n'$ (from same distribution)

1. for $m = 1:M$ train on $\mathcal{D}$ f_m
2. —"— calculate $L(f_m; \mathcal{D}') = L^V_m$
3. choose $f^* = \arg\min L^V_{1:M}$

n' = how large? ~ 1000

n

If $n \not\gg 1000$ K-fold CV

1. Partition $\mathcal{D} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \dots \cup \mathcal{D}_K$ (disjoint)

$$|\mathcal{D}_k| \approx \frac{n}{K}$$

$\mathcal{D}_{-k} = \mathcal{D} \setminus \mathcal{D}_k$

train: $\mathcal{D}_{-k}$

validation: $\mathcal{D}_k$

2. for $k = 1:K$
   - train $f_{1:M}$ on $\mathcal{D}_{-k} \Rightarrow f_m^{(k)}$
   - calculate loss $L_m^{(k)} = \text{Loss}\left(f_m^{(k)}; \mathcal{D}_k\right)$

- $\bar{L}^V(f_m) = \frac{1}{K} \sum_{k=1}^{K} L_m^{(k)}$
- $m^* = \underset{m=1:M}{\operatorname{argmin}} \; \bar{L}_m^V$
- retrain f_{m^*} on $\mathcal{D}$

$K \nearrow$ $|\mathcal{D}_{-k}| = \frac{K-1}{K} n$ more data for training

more computation

very small $n \lessapprox 100 \rightarrow K = n$ Leave-One-Out CV