L17: 22.07.2022

5-17

5-18

5-19

6 Feature dimension reduction

6-1

6-2

6-3

6-4

6.1 Feature selection

Given : • $F_d = \{x_1, \dots, x_d\}$ → set of all d features

• desired number of features $\bar{d} < d$

Feature selection problem :

Find an optimum subset

$$F_{\bar{d},opt} = \{x_{i_1} \dots x_{i_{\bar{d}}}\} \subset F_d$$

Goal: min a certain cost function $J(F)$

$$J(F_{\bar{d},opt}) \leq J(F_\alpha) \quad \forall F_{\bar{d}}$$

- filter approach : J ? → not depending on task at hand
- wrapper approach : J = classification error rate

How to find $F_{\bar{d},opt}$?

(a) exhaustive search:

try all ${}^{d}C_{\bar{d}} = \frac{d!}{\bar{d}!\,(d-\bar{d})!}$ possible subsets

eg $d = 300$, $\bar{d} = 50$ ${}^{300}C_{50} \approx 3 \times 10^{57}$

impossible

(b) single feature ranking

- find the best single feature x_{b_1}

$J(x_{b_1}) \leq J(x_i) \quad \forall x_i \in F_d$

$\Rightarrow d$ single-feature classification

- find the 2nd best single feature x_{b_2}

$J(x_{b_2}) \leq J(x_i) \quad \forall x_i \in F_d \setminus \{x_{b_1}\}$

$\Rightarrow d-1$ single feature classification

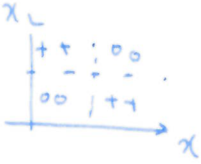
⋮

- find the $\bar{d}$-th best single feature $x_{b_{\bar{d}}}$

Does it work well?

- many features may be redundant, there may be dependencies between them, no new information is gained

- bad single features may combine to form a better multiple feature eg XOR



x_1 : bad feature
x_2 : " "
(x_1, x_2) : perfect

6-5

(c) heuristic sequential algorithms

6-6

6-7

6-8

Pro/cons :

+) Features keeping their physical meanings

+) only $\bar{d}$ features needs to be calculated

(−) a special case of feature transform, not optimum

6.2 Feature transform

$$\underline{x} \in \mathbb{R}^d \xrightarrow{\phi()} \bar{\underline{x}} = \underset{\underset{2o}{\uparrow}}{\underline{\phi}} (\underline{x}) \in R^{\bar{d}} \quad \bar{d} < d$$

How to choose $\underline{\phi}$?

Pro/cons :

(+) more general, better performance than purely feature selection for the same $\bar{d}$ because feature selection is a special case of $\underline{\phi}()$

eg $\bar{\underline{x}} = \begin{bmatrix} x_2 \\ x_4 \end{bmatrix} = \underbrace{\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \underline{x} \end{bmatrix}}_{\underline{\phi}(\underline{x})}$

(−) compute all d features

(−) loss of physical meaning of $\underline{x}$

optimum $\underline{\phi}(\,)$: difficult to find

a simple $\underline{\phi}(\,)$

- linear transform : $\underline{\bar{x}} = \underline{\underline{W}}^T \underline{x}$
- affine transform : $\underline{\bar{x}} = \underline{\underline{W}}^T(\underline{x} - \underline{\mu}) + \underline{\mu}$

$\Rightarrow$ only optimize $\underline{\underline{W}}$ (and $\underline{\mu}$)

Principal Component Analysis (PCA) :

N samples $\underline{x}_n \in R^d$ $\quad 1 \le n \le N$

no class labels , unsupervised

- $\underline{\underline{W}} = [\underline{V}_1 \ \dots \ \underline{V}_{\bar{d}}] \in \mathbb{R}^{d \times \bar{d}}$ $\quad \bar{d} < d$
- $\underline{\bar{x}} = \underline{\underline{W}}^T \underline{x} \quad \in \mathbb{R}^{\bar{d}}$
- Reconstruction : $\underline{\hat{x}} = \underline{\underline{W}} \cdot \underline{\bar{x}} = \underline{\underline{W}}\,\underline{\underline{W}}^T \underline{x} \in R^d$

Optimization criterion :

$$\min \underbrace{\sum_{n=1}^{N} \| \underline{x}_n - \overbrace{\underline{\underline{W}} \ \underbrace{\underline{\underline{W}}^T \underline{x}_n}_{\text{compression} : R^d \to R^{\bar{d}}}}^{\text{reconstruction} : R^{\bar{d}} \to R^d} \|^2}_{\text{reconstruction error}}$$

s. t.

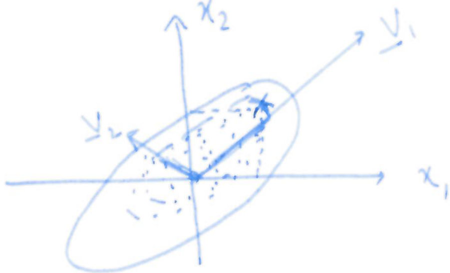
$\underline{\underline{W}}^T \underline{\underline{W}} = I$

i. e. orthonormal

6 - 9

6 - 11

Illustration of PCA:



project $(x_1, x_2) \rightarrow (\underline{v}_1, \underline{v}_2)$

PCA: keep $\vec{d}$ coordinates $\underline{v}_i^T \underline{x}$ of $\underline{x}$ along

the most dominant direction $\underline{v}_i$

$\Rightarrow$ $\vec{d}$ principal components of $\underline{x}$

Why DCT over PCA → ~~depends on data~~. DCT is independent of input whereas PCA is more computationally costly

Non-linear extension of PCA → auto encoder → compression done by encoder, decompression done by decoder.