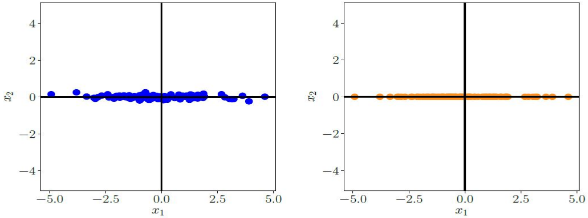
CALCUL NUMERIC
Analiza Componentelor Principale

Paul Irofti
Cristian Rusu
Andrei Pătraşcu

Departament Informatică
Universitatea din Bucureşti

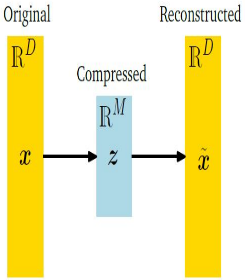
- **Reducţie Dimensională**
- Algoritm PCA
- Aplicaţie

- Utilizarea şi prelucrarea datelor de dimensiuni mari impune adesea mari dificultăţi privind analiza, vizualizarea şi stocarea acestora
- Cu toate acestea, dimensiunile mari atrag redundanţe ascunse în date, e.g. anumite componente pot fi exprimate ca şi combinaţii liniare de alte componente.



Analiza componentelor principale (PCA) reprezintă o metodă simplă de reducţie dimensională liniară, aparută pentru prima oară în lucrările lui Pearson (1901)

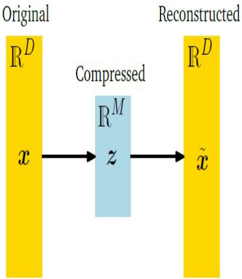
- Datele: $X = [x_1, x_2, \cdots, x_N] \in \mathbb{R}^{d \times N}$ cu media 0
- PCA determină o bază subspaţiului optim de dimensiune $m << d$, i.e. $B \in \mathbb{R}^{d \times m}$, care reţine maximum de informaţie conţinută în X
- Astfel, reduce dimensiunea de la d la m, $X \to Z$



Matricea bazei $B = [b_1 \; b_2 \; \cdots \; b_m]$, unde $m < d$, este ortonormală:

- $b_i^T b_j = 0$ pentru orice $i \neq j$
- $b_i^T b_i = 1$ pentru orice $1 \leq i \leq N$

Coordonatele vectorilor în noua bază: $z_i = B^T x_i$
Proiecţiile vectorilor în noua bază: $\hat{x}_i = Bz_i$



Subspaţiul optim corespunde bazei B are minimizează distanţa $\|\hat{x}_i - x_i\|$.
Echivalent, baza care maximizează varianţa noilor coduri $\{z_i\}_{i=1}^{N}$ (Hotteling)

$$V_z[z] = V_x[B^T(x - \bar{x})] = V_x[B^T x - B^T \bar{x}] = V_x[B^T x],$$

unde am folosit media $\bar{x} = 0$.

Reamintim: varianţa unei variabile aleatoare discrete y (cu N valori) este

$$V_y[y] = \frac{1}{N} \sum_i (y_i - \bar{y})^2$$

unde $\bar{y}$ este media lui y.

- Reducţie Dimensională
- **Algoritm PCA**
- Aplicaţie

Începem prin maximizarea varianţie pentru prima componentă din z:

$$\begin{aligned} V[z_i] &= \frac{1}{N}\sum_{i=1}^{N} z_{i1}^2 \\ &= \frac{1}{N}\sum_{i=1}^{N}(b_1^T x_i)^2 = \frac{1}{N}\sum_{i=1}^{N} b_1^T (x_i x_i^T) b_1 \\ &= b_1^T \underbrace{\left(\frac{1}{N}\sum_{i=1}^{N} x_i x_i^T\right)}_{S} b_1 \end{aligned}$$

Notăm: $S := \frac{1}{N}\sum_{i=1}^{N} x_i x_i^T$ matricea de covarianţă a datelor X

PCA - prima iteraţie

Prima iteraţie necesită rezolvarea:

$$\max \; b_1^T S b_1$$
$$\text{s.l. } \|b_1\| = 1$$

Condiţii de optimalitate: punctele de maxim satisfac setul de egalităţi

$$S b_1 = \lambda_1 b_1$$
$$\|b_1\| = 1$$

- Recunoaştem aceste relaţii din cursul de valori/vectori proprii
- Soluţia: (b_1, λ_1) vectorul propriu maximal al lui S
- b_1 reprezintă prima componentă principală
- λ_1 varianţa datelor proiectate pe subspaţiul definit de prima componentă principală

Considerăm că au fost executate $k-1$ iteraţii, care au produs $\{b_1, \cdots, b_{k-1}\}$
Mai departe, la iteraţia k, eliminăm "efectul" primelor $k-1$ componente

$$\hat{X}^{(k)} := X - \sum_{i=1}^{k-1} b_i b_i^T X \qquad \hat{S}^{(k)} := \frac{1}{N}\hat{X}^{(k)}(\hat{X}^{(k)})^T$$

Aplicăm iteraţia k pe noile date proiectate $\hat{X}^{(k)}$:

$$\max \; b_k^T \hat{S}^{(k)} b_k$$
$$\text{s.l. } \|b_k\| = 1$$

Vectorii proprii ai lui $\hat{S}^{(k)}$ sunt identici cu cei ai lui S

Vectorii proprii ai lui $\hat{S}^{(k)}$ sunt identici cu cei ai lui S

$$\hat{S}^{(k)} b_i = \frac{1}{N} \hat{X}^{(k)} (\hat{X}^{(k)})^T b_i = \frac{1}{N} \left(X - \sum_{i=1}^{k-1} b_i b_i^T X \right) \left(X - \sum_{i=1}^{k-1} b_i b_i^T X \right)^T b_i$$

$$= \begin{cases} 0, & \text{dacă } i < k \\ \lambda_i b_i, & \text{dacă } i \geq k \end{cases}$$

Verificaţi!

Din relaţiile precedente, avem $\lambda_i(\hat{S}^{(k)}) = \begin{cases} 0, & \text{dacă } i < k \\ \lambda_i(S), & \text{dacă } i \geq k \end{cases}$

De aceea, subproblema asociată iteraţiei k:

$$\max\ b_k^T \hat{S}^{(k)} b_k$$
$$\text{s.l. } \|b_k\| = 1$$

are soluţia: *vectorul propriu asociat valorii proprii maxime a matricii* $\hat{S}^{(k)}$

Concluzie: Baza optimă PCA, i.e. $B = [b_1\ b_2 \cdots\ b_m]$, este formată din vectorii proprii ai matricii S asociaţi celor mai mari m valori proprii.

Informaţia conservată prin reconstrucţie pe baza lui B este data de:

$$V_m := \sum_{i=1}^{m} \lambda_i(S)$$

Informaţia pierdută (eroarea) la reconstrucţie:

$$J_m := \sum_{i=m+1}^{d} \lambda_i(S)$$

Paşi PCA:

1. Se compune $S = \frac{1}{N}XX^T$, unde $X = [x_1 \ \cdots \ x_N]$
2. Calculează DVP (Descompunerea Valorilor Proprii) lui $S = U\Lambda U^T$

2' Calculează DVS (Descompunerea Valorilor Singulare) lui $X = U\Sigma V^T$

3. Calcul subspaţiu: $B = [u_1 \cdots u_m]$
4. Reconstrucţie: $Z = B^T X$; $\tilde{X} = BZ$

Pentru pasul al doilea se execută: 2 SAU 2'

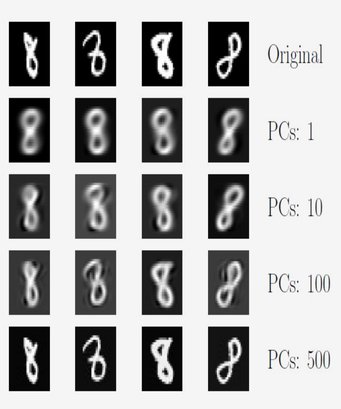
- Coloanele lui U reprezintă vectorii proprii ai matricii XX^T
- Între valorile proprii λ_i (pas 2) şi valorile singulare (pas 2') are loc relaţia:

$$\lambda_i = \frac{\sigma_i^2}{N}$$

- Reducţie Dimensională
- Algoritm PCA
- **Aplicaţie**

Aplicaţie MNIST:

- Considerăm setul de date MNIST format din imagini ale cifrelor 0-9 scrise de mână
- Se echivalează fiecare pixel cu un număr egal cu nivelul de gri şi se vectorizează
- După transformarea imaginilor în vectori numerici avem: $N = 5389, d = 784, m \in \{1, 10, 100, 500\}$
- Numărul maxim de componente principale: 784


Original
PCs: 1
PCs: 10
PCs: 100
PCs: 500

Concluzii aplicaţie MNIST:

- Pe prima linie apar 4 imagini originale ale cifrei 8
- Pe fiecare dintre liniile următoare transformările în imagini ai vectorilor reconstruiţi folosind un număr fixat de componente principale
- La final de linie este indicat numărul de componente principale folosit
- Observăm că reconstrucţia pe baza primelor 500 de componente este aproape perfectă
- **Compresie**: stocăm doar primele 500 CP (< 784) + coordonatele asociate imaginilor $z_i = B^T x_i$
- Necesar memorie: $\underbrace{500 \cdot 784}_{Mem.\ CP} + \underbrace{500 \cdot N}_{Mem.\ coduri\ z} < \underbrace{784 \cdot N}_{Mem.\ imagini\ originale}$

Referinţe

- Deisenroth, M. P., Faisal, A. A., & Ong, C. S. (2020). Mathematics for machine learning. Cambridge University Press.