Přednáška 24 - SVD, PCA & data analysis

Motivace: máme dataset → uložený obrázek / zdravotní data / data preference hudby/filmů/... / data z burzy/...

Problém → máme hodně dat & nevíme jak je analyzovat & & nelze je prakticky dlouho skladovat

→ potřebujeme
- kompresi dat, která zachová důležité rysy
- analýzu dat → co z nich lze vyčíst

→ ve spoustě případů máme dataset v tabulce:

12	12	203	12	12
12	12	203	12	12
203	203	203	203	203
12	12	203	12	12
12	12	203	12	12

~ odpovídá černobílému obrázku → čísla na škále 0 až 255 odpovídají odstínům šedi (0 ~ černá, 255 ~ bílá)
⇒ tady máme „světlý kříž na tmavém poli"

	věk	váha	krevní tlak
pacient 1			
pacient 2			
pacient 3			
⋮			

	akcie A	akcie B	akcie C
simulace I			
simulace II			
simulace III			
⋮			

	věk	bydliště	příjem
uživatel 1			
uživatel 2			
uživatel 3			
⋮			

⇒ máme data v matici a chceme
- komprimovat velikost matice
- „analyzovat, co nám říká"

Hlavní nástroj: singulární rozklad (SVD) pro $A \in \mathbb{R}^{m \times n}$, $m > n$.

Lingebra 2: $\forall A \in \mathbb{R}^{m \times n}$ $\exists U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$, $\Sigma \in \mathbb{R}^{m \times n}$:
- U, V jsou unitární
- $\Sigma = \begin{bmatrix} \sigma_1 & & \\ & \sigma_2 & \\ & & \ddots \\ & & & \sigma_n \end{bmatrix}$ & $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$
- $A = U \Sigma V^T$

Komprese dat

$A = U \Sigma V^T$ & $U = [\vec{u}_1, \dots, \vec{u}_m]$, $V = [\vec{v}_1, \dots, \vec{v}_n]$ $\Rightarrow$

$\Rightarrow A = \sigma_1 \cdot \vec{u}_1 \vec{v}_1^T + \sigma_2 \cdot \vec{u}_2 \vec{v}_2^T + \dots + \sigma_n \cdot \vec{u}_n \vec{v}_n^T$ (tzv. dyadický rozvoj A)

kde
- σ_i jsou nezáporné a nerostoucí
- $\| \vec{u}_i \vec{v}_i^T \| = 1$ ... unitární matice

$\Rightarrow$ idea aproximace A: $A \approx \underbrace{\sigma_1 \vec{u}_1 \vec{v}_1^T + \dots + \sigma_r \cdot \vec{u}_r \vec{v}_r^T}_{=: A_r}$

$A_r = U_{:,1:r} \Sigma_{1:r,1:r} \left(V_{:,1:r}\right)^T$

memory cost $A \sim m \cdot n$

memory cost $A_r \sim r \cdot (m+n+1)$ ... (v praxi $\tilde{\vec{v}}_k := \sigma_k \vec{v}_k$ $\rightsquigarrow$ $r \cdot (m+n)$)

Věta (Eckhart-Young-Mirsky)

Mějme $A \in \mathbb{R}^{m \times n}$, $r \le n \le m$. Nechť $A = U \Sigma V^T$ je singulární rozklad. Pak nejpřesnější aproximace A maticí hodnosti r je právě matice A_r dána prvními r členy dyadického rozvoje A.

Můžeme zapsat jako $\forall X \in \mathbb{R}^{m \times r}, Y \in \mathbb{R}^{n \times r}$:

$$\| A - X Y^T \| \ge \| A - A_r \|$$

matice hodnosti r jsou právě matice tvaru $X \cdot Y^T$ pro nějaké $X \in \mathbb{R}^{m \times r}$, $Y \in \mathbb{R}^{n \times r}$ $\rightarrow$ tedy právě matice s memory cost $r \cdot (m+n)$

kde $A_r = U_{:,1:r} \Sigma_{1:r,1:r} \left(V_{:,1:r}\right)^T = \sigma_1 \vec{u}_1 \vec{v}_1^T + \dots + \sigma_r \vec{u}_r \vec{v}_r^T$.

$\Rightarrow$ spočtením singulárního rozkladu (≡ singular value decomposition ≡ SVD) jsme schopni najít nejpřesnější kompresi dat v Euklidovské normě z $m \cdot n$ dat na $r \cdot (m+n)$ dat.

pythonDemo: image compression

Analýza dat

data ≡ řádky matice ≡ body $\vec{a}_i \in \mathbb{R}^n$ & máme jich m → $i = 1, \dots, m$

motivace pro $n = 3$:

- když budou $\vec{a}_1, \dots, \vec{a}_m$ všechny ležet na jedné přímce
  ⇒ směrový vektor $\vec{s}$ té přímky nám o těch datech hodně vypovídá
  ⇝ $\vec{a}_i = \alpha_i \cdot \vec{s}$ ⇒ $\begin{bmatrix} — \vec{a}_1^T — \\ \vdots \\ — \vec{a}_m^T — \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_m \end{bmatrix} \cdot \vec{s}^T$ ... rank-1
  ⇝ slovy: všechna data mají stejný „poměr" mezi hodnotami „features" (≡ratio) ⇝ liší se jen škálováním
  neboli rozptyl v datech lze „vysvětlit" pozorováním (≡variance) pouze tohoto škálování
- když $\vec{a}_1, \dots, \vec{a}_m$ leží „okolo přímky" ⇝ všechny rovnosti výše lze nahradit „$\approx$" a dostaneme
  ⇝ $\begin{bmatrix} — \vec{a}_1^T — \\ \vdots \\ — \vec{a}_m^T — \end{bmatrix} \approx \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_m \end{bmatrix} \cdot \vec{s}^T$
  ⇝ slovy: data mají podobný „poměr" mezi hodnotami „features" a tedy rozptyl/rozdíly v datech odpovídají pouze škálování tohoto poměru

Pozorování 1: Pokud by data ležela přibližně v rovině dané vektory $\vec{s}_1, \vec{s}_2$ pak máme $\begin{bmatrix} — \vec{a}_1^T — \\ \vdots \\ — \vec{a}_m^T — \end{bmatrix} \approx \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \vdots & \vdots \\ \alpha_{m1} & \alpha_{m2} \end{bmatrix} \cdot \begin{bmatrix} — \vec{s}_1^T — \\ — \vec{s}_2^T — \end{bmatrix}$ ... rank-2 ... a opět platí, že rozptyl v datech lze „přibližně vysvětlit" pouze škálováním poměrů hodnot prvků vektorů $\vec{s}_1, \vec{s}_2$, tj. poměrů konkrétních hodnot našich měřených „features"

Pozorování 2: V $\mathbb{R}^n$ funguje stejná idea, jen „nejde vizualizovat"

Pozorování 3: Analyzovat data ≡ najít co nejmenší počet $\vec{s}_1, \dots, \vec{s}_r$, které nám takto „vysvětlí" ty data. Vektorům $\vec{s}_1, \dots, \vec{s}_r$ se říká „principal components (≡of the dataset)".
Jejich nalezení odpovídá nalezení nejlepší rank-r aproximaci matice A ⇝ stačí spočítat SVD! (tzv. PCA = principal component analysis)

Python demo: Iris dataset