

Přednáška 24 - SVD, PCA & data analysis

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

Problém \rightarrow máme hodně dat & nevíme jak je analyzovat &
& může je prakticky dlouho skladovat

\rightarrow potřebujeme

- kompresi dat, která zachová důležité rysy
- analýzu dat \rightarrow co z nich lze vyčíst

\rightarrow ve spoustě případech 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 \rightarrow čísla na škále
 \sim 0 až 255 odpovídají odstínům šedi (0 ~ černá)
 \Rightarrow tedy máme „světlý kříž na tmavém poli“

	věk	váha	krevní tlak
pacient1			
pacient2			
pacient3			
\vdots			

	akcie A	akcie B	akcie C
simulace I			
simulace II			
simulace III			
\vdots			

	věk	budíště	příjem
uživatel 1			
uživatel 2			
uživatel 3			
\vdots			

\Rightarrow máme data v matici a chceme

- komprimovat velikost matice
- „analyzovat, co nám říká“

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

Linegebra 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

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

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

kde

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

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

memory cost $A \sim m \cdot n$

$$A_r = U_{:,1:r} \sum_{i=1:r} (V_{:,i:r})^T$$

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

Věta (Eckhart-Young-Thirsky)

Nějme $A \in \mathbb{R}^{m \times n}$, $r \leq n \leq m$. Necht $A = U \Sigma V^T$ je singulární rozklad.

Pak nejpresnější aproximace A maticí hodnosti r je právě matice A_r dána prvními r členy dyadického rozvoje A .

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

$$\|A - XY^T\|_{2,F} \geq \|A - A_r\|_{2,F}$$

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

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

$$\text{Navíc: } \|A - A_r\|_2 = \sigma_{r+1} \quad \& \quad \|A - A_r\|_F = \sqrt{\sigma_{r+1}^2 + \sigma_{r+2}^2 + \dots + \sigma_{\min(m,n)}^2}$$

\Rightarrow spočtením singulárního rozkladu (\equiv singular value decomposition \equiv SVD)

jsme schopni najít nejpresnější kompresi dat v

Eukleidovské normě z $m \cdot n$ dat na $r \cdot (m+n)$ dat.

Python Demo: image compression

Analýza dat

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

motivace pro $n=3$: • když budeme $\vec{a}_1, \dots, \vec{a}_m$ všechny ležet na jedné přímce

\Rightarrow směrový vektor \vec{s} té přímky mi o těch datech hodně vypovídá

$$m \rightarrow \vec{a}_i = \alpha_i \cdot \vec{s} \Rightarrow \begin{bmatrix} -\vec{a}_1- \\ \vdots \\ -\vec{a}_m- \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_m \end{bmatrix} \cdot \vec{s}^T \quad \dots \text{rank} = 1$$

$m \rightarrow$ slovy: všechna data mají stejný „poměr“ mezi hodnotami „features“
(=ratio) $m \rightarrow$ liší se jen škálováním

nebo-li rozptyl v datech lze „vysvětlit“ pozorováním
(=variance) pouze tohoto škálování

• když $\vec{a}_1, \dots, \vec{a}_m$ leží „okolo přímky“ $m \rightarrow$ všechny rovnosti výše lze nahradit „ \approx “ a dostaneme

$$m \rightarrow \begin{bmatrix} -\vec{a}_1- \\ \vdots \\ -\vec{a}_m- \end{bmatrix} \approx \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_m \end{bmatrix} \cdot \vec{s}^T$$

$m \rightarrow$ 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 $\begin{bmatrix} -\vec{a}_1- \\ \vdots \\ -\vec{a}_m- \end{bmatrix} \approx \begin{bmatrix} \alpha_{11} & \alpha_{12} \\ \vdots & \vdots \\ \alpha_{m1} & \alpha_{m2} \end{bmatrix} \cdot \begin{bmatrix} -\vec{s}_1- \\ -\vec{s}_2- \end{bmatrix} \quad \dots \text{rank} = 2 \dots$ 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 měření „features“

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

Pozorování 3: Analyzovat data \equiv 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)“.

Jeich nalezení odpovídá nalezení nejlepší rank- r aproximace matice $A \rightarrow$ stačí spočítat SVD! (tj. PCA \equiv principal component analysis)

Python demo: Iris dataset

Motivace „ze statistiky“:

ve smyslu paměti, tj. low-rank
tj. $\tilde{A} \approx XY^T$

Matice $\tilde{A} \in \mathbb{R}^{m \times n}$ obsahuje naše data $m \rightarrow$ chtěl bych „levnou aproximaci“, která zachytí „co nejvíce“ z našich dat.

Protože každý sloupec (\equiv feature) může mít úplně jiné velikosti a typy dat (věk vs. váha vs. příjem) jako první krok data vycentrujeme a normalizujeme

$$\vec{\mu} := \frac{1}{n} \sum_{i=1}^n \tilde{A}_{:,i} \quad \& \quad A^{\text{centr}} := \left[\tilde{A}_{:,1} - \vec{\mu}, \dots, \tilde{A}_{:,n} - \vec{\mu} \right]$$

centrování dat je vždy užitečné

$$\vec{\sigma} := \begin{bmatrix} \|A_{:,1}^{\text{centr}}\|_2 \\ \vdots \\ \|A_{:,m}^{\text{centr}}\|_2 \end{bmatrix} \quad \& \quad A := \begin{bmatrix} -A_{1,:}^{\text{centr}} / \sigma_1 & \\ \vdots & \\ -A_{m,:}^{\text{centr}} / \sigma_m & \end{bmatrix}$$

normalizace závisí na aplikaci \rightarrow často vynecháváme nebo volíme jinou normu

Pak „zachytit co nejvíce z dat \tilde{A} “ lze pochopit jako „zachytit co nejvíce z variability data-setu \tilde{A} “

Variabilita $\tilde{A} \approx$ kovarianční matice $= \frac{1}{N-1} \cdot (\tilde{A} - \vec{\mu})(\tilde{A} - \vec{\mu})^T = \frac{1}{N-1} A^{\text{centr}} (A^{\text{centr}})^T$

Zachytit co nejlépe variability \tilde{A}

tedy odpovídá

$$\min_{X, Y \in \mathbb{R}^{m \times r}} \left\| \frac{1}{N-1} A^{\text{centr}} (A^{\text{centr}})^T - XY^T \right\|_F$$

Protože matice $\frac{1}{N-1} A^{\text{centr}} (A^{\text{centr}})^T$ je zjevně symetrická a pozitivně semi-definitní

\Rightarrow řešením jsou X, Y dané dominantními vl. páry $\frac{1}{N-1} A^{\text{centr}} (A^{\text{centr}})^T$

\rightsquigarrow tedy dominantními členy SVD rozvoje matice A^{centr}

Tomuto přístupu odvození se říká PCA (principle component analysis)

