FONDEMENTS DU MACHINE LEARNING

17 septembre 2024

Aujourd'hui

SVD et applications

SVD et applications

Rappel : Soit $X \in \mathbb{R}^{m \times n}$

Une SVD de X est de la forme

$$U \Sigma V^T = \begin{bmatrix} u_1 \cdots u_m \end{bmatrix} \left[\begin{array}{ccc|c} \sigma_1 & & & \\ & \ddots & & 0 \\ & & \sigma_r & \\ \hline & 0 & & 0 \end{array}\right] \begin{bmatrix} v_1 \cdots v_n \end{bmatrix}^T$$

$U \in \mathbb{R}^{m \times m}$, $U^T U = I_m = U U^T$

$V \in \mathbb{R}^{n \times n}$, $V^T V = I_n = V V^T$

$$\Sigma = \left[\begin{array}{ccc|c} \sigma_1 & & 0 & \\ & \ddots & & 0 \\ 0 & & \sigma_{\min(m,n)} & \\ \hline & 0 & & 0 \end{array}\right] \begin{array}{l} \\ \\ \\ \updownarrow m-\min(m,n) \end{array}$$

$$\begin{array}{c} \overleftrightarrow{n-\min(m,n)} \end{array}$$

$m > n$ $\quad \Sigma = \left[\begin{array}{ccc} \sigma_1 & & 0 \\ 0 & \ddots & \sigma_n \\ \hline & 0 & \end{array}\right] \begin{array}{l} \updownarrow n \\ \updownarrow m-n \end{array}$ $\quad n \geq m \quad \Sigma = \left[\begin{array}{ccc|c} \sigma_1 & & 0 & \\ 0 & \ddots & \sigma_m & 0 \end{array}\right]$

$$\sigma_1 \geq \cdots \geq \sigma_{\min(m,n)} \geq 0$$

Remarque : La SVD permet naturellement (par construction) d'ordonner les valeurs singulières

$$\text{rang}(X) = r \iff \sigma_1 \geq \cdots \geq \sigma_r > 0 = \sigma_{r+1} = \cdots = \sigma_{\min(m,n)}$$

Si X est de rang r, alors la SVD s'écrit

$$\begin{bmatrix} u_1 & \cdots & u_m \end{bmatrix} \begin{array}{c} \\ \updownarrow r \\ \updownarrow m-r \end{array} \begin{array}{cc} \overset{r}{\longleftrightarrow} & \overset{n-r}{\longleftrightarrow} \\ \multicolumn{2}{c}{\left[\begin{array}{c|c} \begin{matrix}\sigma_1 & & \\ & \ddots & \\ & & \sigma_r\end{matrix} & 0 \\ \hline 0 & 0 \end{array}\right]} \end{array} \begin{bmatrix} v_1^T \\ \vdots \\ v_n^T \end{bmatrix}$$

$V = \begin{bmatrix} v_1 & \cdots & v_n \end{bmatrix}$

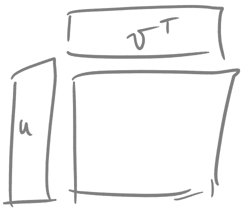
$V^T = \begin{bmatrix} v_1^T \\ \vdots \\ v_n^T \end{bmatrix}$

$u_i \in \mathbb{R}^m$
i-ème colonne de U

$v_i \in \mathbb{R}^n$
colonnes de V

$$U\Sigma V^T = U\left(\Sigma V^T\right) = \underbrace{U}_{m \times m} \begin{array}{c} \left[\begin{matrix} \sigma_1 v_1^T \\ \vdots \\ \sigma_r v_r^T \\ \hline 0 \end{matrix}\right] \end{array} \begin{array}{l} \updownarrow r \\ \\ \updownarrow m-r \end{array}$$

$u \in \mathbb{R}^m$
$v \in \mathbb{R}^n$

$uv^T =$ 

"Matrice de rang 1"

$$= \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \ldots + \sigma_r u_r v_r^T$$

Somme de matrices

$$= \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

$\sigma_i \in \mathbb{R}$ $\quad \underbrace{\underset{m \times 1}{u_i} \quad \underset{1 \times n}{v_i^T}}_{m \times n}$

↳ Toute matrice $X \in \mathbb{R}^{m \times n}$ de rang r s'écrit comme une somme de r matrices de rang 1 données par sa SVD

$$X = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

$$= \underbrace{\begin{bmatrix} u_1 & \cdots & u_r \end{bmatrix}}_{m \times r} \underbrace{\begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix}}_{r \times r} \underbrace{\begin{bmatrix} v_1^T \\ \vdots \\ v_r^T \end{bmatrix}}_{r \times n}$$

SVD réduite de X

Corollaire : Pour représenter X (de taille $m \times n$), on peut utiliser $(m+n+1)r$ coefficients

$mr : u_1, \dots, u_r$
$nr : v_1, \dots, v_r$
$r : \sigma_1, \dots, \sigma_r$

Si $r \ll \min(m,n)$, $(m+n+1)r \ll mn$

Matrice de rang faible (notion relative)

Remarque : L'écriture $\sum_{i=1}^{r} \sigma_i u_i v_i^T$ a beaucoup d'avantages pratiques (en plus du coût de stockage)

- Plus facile de calculer la somme de manière incrémentale
- plus facile de l'appliquer à un vecteur

$$\forall w \in \mathbb{R}^n, \quad Xw \Rightarrow \sum_{i=1}^{r} \sigma_i u_i \underbrace{v_i^T w}_{\text{produit scalaire}} = \sum_{i=1}^{r} \sigma_i (v_i^T w) u_i$$

SVD tronquée : $\forall k \in [\![1, r]\!]$ avec r rang de X

k-SVD / SVD tronquée de rang k : $X_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$

- $X_r = X$ mais $X_k \neq X$ pour $k < r$
- Coût de stockage de la k-SVD : $(m+n+1)k$
- $\text{rang}(X_k) = k$ (SVD réduite $X_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$)

Q) Est-ce que la k-SVD est une bonne approximation de la matrice d'origine ?

Théorème : (Eckhart-Young)

Soit $X \in \mathbb{R}^{m \times n}$ de rang r et soit X_k une k-SVD de X.
Alors, pour toute matrice $A \in \mathbb{R}^{m \times n}$ de rang k, on a :

$$\|X - A\| \geq \|X - X_k\|$$

$$\left(\forall M \in \mathbb{R}^{m \times n},\ \|M\| = \max_{x \neq 0} \frac{\|Mx\|}{\|x\|} \right)$$

La k-SVD fournit la meilleure approximation possible de X par une matrice de rang k.

Q) En pratique, est-il intéressant d'approcher une matrice de rang r par sa k-SVD de rang $k < r$?

→ Oui ! En pratique, on rencontre des matrices dont les valeurs singulières décroissent rapidement, pour lesquelles l'approximation par une k-SVD est une très bonne approximation, c-à-d que

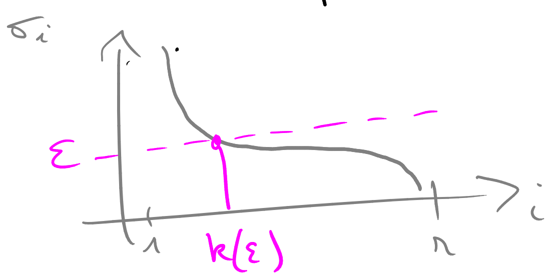
erreur relative d'approximation

$$\underbrace{k \ll r}_{\text{Gain mémoire}} \quad \text{et} \quad \underbrace{\frac{\|X - X_k\|}{\|X\|} \ll 1}_{\text{Approximation précise}}$$

→ En fonction de X, m, n, r et de la nature des données, on cherche à déterminer une bonne valeur de k pour que la k-SVD soit une bonne approximation

- Peut se faire | naïvement / itérativement ($k=1$, si pas suffisant $k=2$, etc)

- Peut se faire en fonction des valeurs singulières
  → Tronquer les valeurs singulières inférieures à une certaine précision/tolérance $\varepsilon > 0$



⇒ Le choix de ε ou de k est très dépendant du problème

Exemple :

- $X \in \mathbb{R}^{m \times m}$

$$X = \underbrace{X_{true}}_{\text{vraie matrice}} + \underbrace{X_{noise}}_{\text{bruit aléatoire}}$$

(qu'on cherche à mesurer) (bruit qui entache les observations)

But : Récupérer X_{true} à partir de X sans connaître X_{noise} de manière exacte (ou approcher X_{true})

Hypothèse : X_{true} est de rang faible, $< \sqrt{m}$; X est de rang plein à cause du bruit

X_{noise} bruit iid gaussien

$$\forall (i,j) \in \{1, \dots, m\}^2, \ [X_{noise}]_{ij} \sim \mathcal{N}(0,1)$$

Théorème (Gavish & Donoho, 2014)

Une k-SVD de X avec $\sigma_k \geq \frac{4}{\sqrt{3}}\sqrt{n}$ élimine le bruit.

Référence : S. Brunton et J.N. Kutz, Data-driven Science and engineering, 2021 (avec codes Python)

En bref :

→ Si une matrice est de rang faible, sa SVD réduite peut déjà être une façon moins coûteuse de stocker la matrice

→ On peut toujours construire une approximation de rang faible avec une k-SVD où k est suffisamment petit

→ En pratique, si on regarde l'ensemble des valeurs singulières d'une matrice, les premières valeurs singulières sont souvent bien plus grandes que les autres ⇒ incite à faire une approximation !

→ Déterminer un bon compromis entre approximation de qualité et approximation peu coûteuse est un processus qui dépend fortement des données

À suivre :

$$X = \begin{bmatrix} x_1^T \\ \vdots \\ x_m^T \end{bmatrix}$$ matrice d'individus $x_1, \dots, x_m$

$\frac{XX^T}{n-1}$: matrice de covariance empirique

SVD de X ⇒ ACP de XX^T