01/17/2023 SVD

Singular Value Decomposition

- rank of a matrix
  - number of independent columns/rows in a matrix
    - example:

$$\begin{bmatrix} 1 & 1 & 2 \\ 2 & 1 & 3 \\ 3 & 1 & 4 \\ 4 & 1 & 5 \end{bmatrix}$$

$1 \cdot col_1 + 1 \cdot col_2 - 1 col_3 = 0$

$[1, 1, -1]$

rank is 2

    - a column is not independent if you can find scalar multiples of the other columns that add to that column
    - a matrix is not full rank if there are scalar multiples of the columns that will result in the 0 vector when the columns are added
    - example:

$$\begin{bmatrix} 1 & 2 \\ 1 & 1 \\ 1 & 1 \end{bmatrix}$$

rank is 2 because there are 2 independent rows/columns

  - $Dim(A) \rightarrow m \times n$

    rank: r

    $r \leq \min(m, n)$

- SVD

  $A \in \mathbb{R}^{m \times n}$

  $A_{m \times n} = U \Sigma V^T$

$U = [u_1, u_2, \ldots, u_m] \in \mathbb{R}^{m \times m}$ (orthonormal matrix)

$u^T u = I_{m \times m}$

$V = [V_1, V_2, \ldots V_n] \in \mathbb{R}^{n \times n}$ (orthonormal matrix)

$V^T V = I_{n \times n}$

Σ is $m \times n$

if $m > n$

$$\Sigma = \begin{bmatrix} \sigma_{1,1} & 0 & \cdots & 0 \\ 0 & \sigma_{2,2} & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_{n,n} \\ 0 & \cdots & \cdots & 0 \end{bmatrix}_{m \times n}$$

if $m < n$

$$\Sigma = \begin{bmatrix} \sigma_{1,1} & 0 & \cdots & 0 & \cdots & 0 \\ 0 & \sigma_{2,2} & 0 & & 0 & \vdots \\ \vdots & 0 & \ddots & 0 & & \vdots \\ 0 & \cdots & 0 & \sigma_{m,m} & 0 \cdots & 0 \end{bmatrix}_{m \times n}$$

if $m = n$

$$\Sigma = \begin{bmatrix} \sigma_{1,1} & 0 & \cdots & 0 \\ 0 & \sigma_{2,1} & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \sigma_{m,m} \end{bmatrix}$$ $\leftarrow$ or $\sigma_{n,n}$ since $m=n$

$\{\sigma_{1,1} \geq \sigma_{2,2} \geq \sigma_{3,3} \geq \ldots \geq \sigma_{\min(m,n)} \geq 0\} \rightarrow$ singular value of matrix A

$u_1, u_2, \ldots u_m$: left singular values

$V_1, V_2, \ldots V_n$: right singular values

- generalized example

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} \\ a_{2,1} & a_{2,2} & a_{2,3} \\ a_{3,1} & a_{3,2} & a_{3,3} \\ a_{4,1} & a_{4,2} & a_{4,3} \end{bmatrix} \qquad A = U \Sigma V^T$$

$$A = \underbrace{\begin{bmatrix} | & | & | & | \\ u_1 & u_2 & u_3 & u_4 \\ | & | & | & | \end{bmatrix}}_{U_{4\times4}} \underbrace{\begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \\ 0 & 0 & 0 \end{bmatrix}}_{\Sigma_{4\times3}} \underbrace{\begin{bmatrix} — v_1^T — \\ — v_2^T — \\ — v_3^T — \end{bmatrix}}_{V^T_{3\times3}}$$

$A_1 = \sigma_1 \cdot \underbrace{\underbrace{u_1}_{4\times1} \cdot \underbrace{v_1^T}_{1\times3}}_{4\times3}$ → Rank 1 approximation

$A_2 = \sigma_1 u_1 v_1^T + \underbrace{\sigma_2 u_2 v_2^T}_{4\times3}$ → Rank 2 approximation

$A_3 = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T = A$ → Rank 3 approximation

- Why is A_1 a Rank 1 approx?
  - ex.) $u_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ $v_1^T = \begin{bmatrix} 3 & 4 \end{bmatrix}$

$$u_1 \cdot v_1^T = \begin{bmatrix} 3 & 4 \\ 6 & 8 \end{bmatrix} = \begin{bmatrix} 3\begin{bmatrix} 1 \\ 2 \end{bmatrix} & 4\begin{bmatrix} 1 \\ 2 \end{bmatrix} \end{bmatrix}$$

- Why is A_2 a Rank 2 approx?
  - because U is orthonormal
- Best k^{th} approximation of A

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

$A_{m \times n}$
rank: r $\{r \leq \min(m,n)\}$

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

- if $k \geq r$ $A_k = A_r = A$ because there is no improvement after A_r as there are no σ after σ_r
- furthermore, there is the most change in the smaller A approximations because they have the largest σ's

- Storage savings with SVD
  - $A_{m \times n}$ where m and n are both large
    - $m \times n$ entries need to be saved
  - Rank 1 good enough
    - save σ_1, u_1, v_1^T $\{m+n+1\}$
  - Rank 2 good enough
    - save $\sigma_1, \sigma_2, u_1, u_2, v_1^T, v_2^T$ $\{2(m+n+1)\}$
  - Rank k good enough
    - save $\sigma_1 \cdots \sigma_k, u_1 \cdots u_k, v_1^T \cdots v_k^T$ $\{k(m+n+1)\}$
  - example

    $m = 10{,}000$ $n = 10\,000$ rank $= 100$

    $m \times n = 10^8$

    $m + n + 1 = 20001$

    savings for rank 1 approx: $\frac{10^8}{20001} \approx 0.5 \times 10^4$

savings for rank k approximation

$$\frac{10^8}{k \times 20{,}000} = \frac{0.5 \times 10^4}{k}$$

$k = 100 \ldots$ savings 50 times while still
$r = 100$ obtaining full A matrix
$A_{100} = A$

$\therefore$ instead of saving 10^8 values,
you are saving $100 * (m+n+1) \approx 100 \times (20000)$
$\approx 2 \times 10^6$