SVD Image Compression

01/19/2023

- review

$$A_{m\times n} = U_{m\times m} \Sigma_{m\times n} V^T_{n\times n}$$

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{bmatrix} = \begin{bmatrix} | & | & & | \\ u_1 & u_2 & \cdots & u_m \\ | & | & & | \end{bmatrix} \begin{bmatrix} \sigma_1 & & 0 \\ & \sigma_2 & \\ 0 & & \ddots \end{bmatrix} \begin{bmatrix} - & v_1^T & - \\ - & v_2^T & - \\ & \vdots & \\ - & v_N^T & - \end{bmatrix}$$

- approximations

$$A_1 = \sigma_1 u_1 v_1^T$$

$$A_2 = \sum_{i=1}^{2} \sigma_i u_i v_i^T$$

$$\vdots$$

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

- python execution example
  - used Sklearn's load_sample_images for the image (matrix)
  - images are converted to black and white or grayscale to make 2D arrays
  - np.linalg.svd is used for SVD
    - Σ comes out as a 1D array, use np.diag to make square matrix with σ_i on diagonal
    - since $A_k = A$ when k is rank of A, the extra rows/columns Σ should be will not add any info to the A approximations

- you can print the rank of A to determine what k^{th} approximation you need to obtain A
- ~~** Note **~~: np.linalg.svd returns V^T instead of V
-

- example

$$A = \begin{bmatrix} | & | & | & | \\ a_1 & a_2 & a_3 & a_4 \\ | & | & | & | \end{bmatrix}_{4\times4} = \begin{bmatrix} | & | & | & | \\ u_1 & u_2 & u_3 & u_4 \\ | & | & | & | \end{bmatrix}_{4\times4} \begin{bmatrix} \sigma_1 & 0 & 0 & 0 \\ 0 & \sigma_2 & 0 & 0 \\ 0 & 0 & \sigma_3 & 0 \\ 0 & 0 & 0 & \sigma_4 \end{bmatrix}_{4\times4} \begin{bmatrix} - v_1^T - \\ - v_2^T - \\ - v_3^T - \\ - v_4^T - \end{bmatrix}_{4\times4}$$

$$= \begin{bmatrix} | & | & | & | \\ u_1 & u_2 & u_3 & u_4 \\ | & | & | & | \end{bmatrix} \begin{bmatrix} - \sigma_1 v_1^T - \\ - \sigma_2 v_2^T - \\ - \sigma_3 v_3^T - \\ - \sigma_4 v_4^T - \end{bmatrix}$$

$$= \begin{bmatrix} | & | & | & | \\ u_1 & u_2 & u_3 & u_4 \\ | & | & | & | \end{bmatrix} \begin{bmatrix} \sigma_1 v_{11} & \sigma_1 v_{12} & \sigma_1 v_{13} & \sigma_1 v_{14} \\ \sigma_2 v_{21} & \sigma_2 v_{22} & \sigma_2 v_{23} & \sigma_2 v_{24} \\ \sigma_3 v_{31} & \sigma_3 v_{32} & \sigma_3 v_{33} & \sigma_3 v_{34} \\ \sigma_4 v_{41} & \sigma_4 v_{42} & \sigma_4 v_{43} & \sigma_4 v_{44} \end{bmatrix}$$

$$a_1 = \sigma_1 v_{11} u_1 + \sigma_2 v_{12} u_2 + \sigma_3 v_{13} u_3 + \sigma_4 v_{14} u_4$$

$$a_j = \sum_{i=1}^{4} \sigma_i v_{ji} u_i$$

$$a_j = \sum_{i=1}^{4} \omega_{ji} u_i$$

- Can we construct a new sample using

the U from the original 4 columns of A?

- sometimes