Recall:

1) If A is $n \times n$ symmetric matrix then A can be written as a product

$$A = QDQ^T$$

where:

- $Q = \begin{bmatrix} \mathbf{u}_1 & \mathbf{u}_2 & \dots & \mathbf{u}_n \end{bmatrix}$ is an orthogonal matrix.
- D is diagonal matrix:

$$D = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & \lambda_n \end{bmatrix}$$

2) Spectral decomposition of a symmetric matrix:

$$A = \lambda_1(\mathbf{u}_1\mathbf{u}_1^T) + \lambda_2(\mathbf{u}_2\mathbf{u}_2^T) + \dots + \lambda_n(\mathbf{u}_n\mathbf{u}_n^T)$$

where $|\lambda_1| \geq |\lambda_2| \geq \dots \geq |\lambda_n|$.

Theorem

Any A an $m \times n$ matrix can be written as a product

$$A = U\Sigma V^T$$

where:

- $U = \begin{bmatrix} \mathbf{u}_1 & \dots & \mathbf{u}_m \end{bmatrix}$ is an $m \times m$ orthogonal matrix.
- $V = \begin{bmatrix} \mathbf{v}_1 & \dots & \mathbf{v}_n \end{bmatrix}$ is an $n \times n$ orthogonal matrix.
- Σ is an $m \times n$ matrix of the following form:

$$\begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & \sigma_n \\ 0 & 0 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & \sigma_m & 0 & \cdots & 0 \end{bmatrix}$$

(if $n \leq m$) (if $n \geq m$)

where $\sigma_1 \geq \sigma_2 \geq \ldots \geq 0$.

Note.

- The numbers $\sigma_1, \sigma_2, \ldots$ are called *singular values* of A.
- The vectors $\mathbf{u}_1, \ldots, \mathbf{u}_m$ are called *left singular vectors* of A.
- Then the vectors $\mathbf{v}_1, \ldots, \mathbf{v}_n$ are called *right singular vectors* of A.
- The formula $A = U\Sigma V^T$ is called a *singular value decomposition (SVD)* of A.
- The matrix Σ is uniquely determined, but U and V depend on some choices.

Theorem

Let A be a matrix with a singular value decomposition

$$A = U\Sigma V^T$$

If

$$U = \begin{bmatrix} \mathbf{u}_1 & \dots & \mathbf{u}_m \end{bmatrix} \qquad V = \begin{bmatrix} \mathbf{v}_1 & \dots & \mathbf{v}_n \end{bmatrix}$$

and $\sigma_1, \dots, \sigma_r$ are singular values of A then then

$$A = \sigma_1(\mathbf{u}_1\mathbf{v}_1^T) + \sigma_2(\mathbf{u}_2\mathbf{v}_2^T) + \dots + \sigma_r(\mathbf{u}_r\mathbf{v}_r^T)$$

<u>Example:</u>

$$\underset{3\times 2}{A} = \underset{3\times 3}{U} \cdot \underset{3\times 2}{\Sigma} \cdot \underset{2\times 2}{V^T} = \underbrace{\begin{bmatrix} u_1 & u_2 & u_3 \end{bmatrix}}_{U} \cdot \underbrace{\begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \\ 0 & 0 \end{bmatrix}}_{\Sigma} \cdot \underbrace{\begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix}}_{V^T}$$

$$= \begin{bmatrix} \sigma_1 u_1 & \sigma_2 u_2 \end{bmatrix} \cdot \begin{bmatrix} v_1^T \\ v_2^T \end{bmatrix}$$

$$= \sigma_1(u_1 v_1^T) + \sigma_2(u_2 v_2^T)$$

Application: **Image compression**

- The size of this image is 800×700 pixels.
- The color of each pixel is represented by an integer between 0 (black) and 255 (white).
- The whole image is described by a matrix A consisting of $800 \times 700 = 560,000$ numbers.
- Each number is stored in 1 byte, so the image file size is $560,000$ bytes (≈ 0.53 MB).

How to make the image file smaller:

1) Compute SVD of the matrix A:

$$A = U\Sigma V^T$$

where

$$U = \begin{bmatrix} \mathbf{u}_1 & \dots & \mathbf{u}_m \end{bmatrix} \qquad V = \begin{bmatrix} \mathbf{v}_1 & \dots & \mathbf{v}_n \end{bmatrix}$$

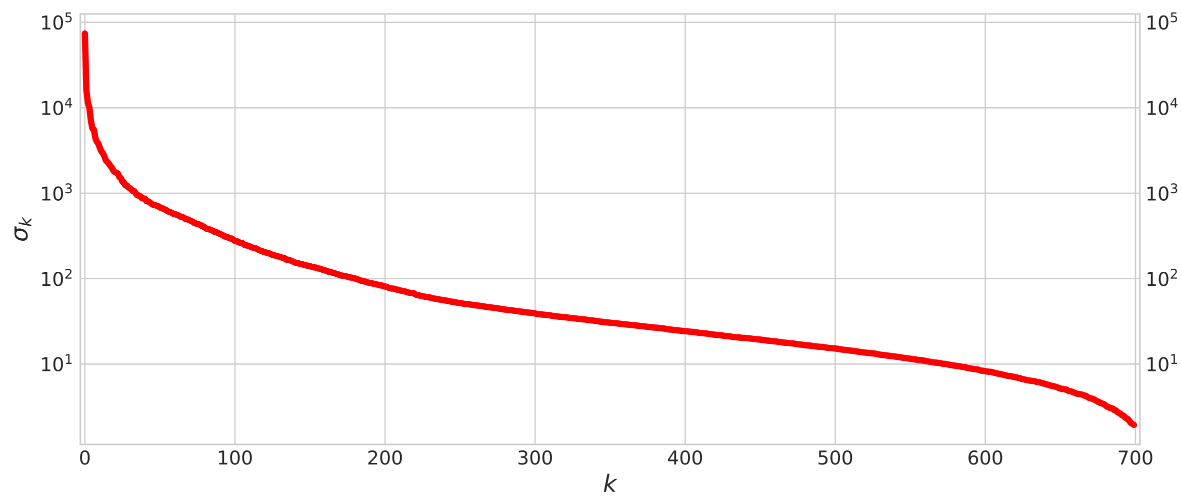
and $\sigma_1, \dots, \sigma_r$ are singular values of A.

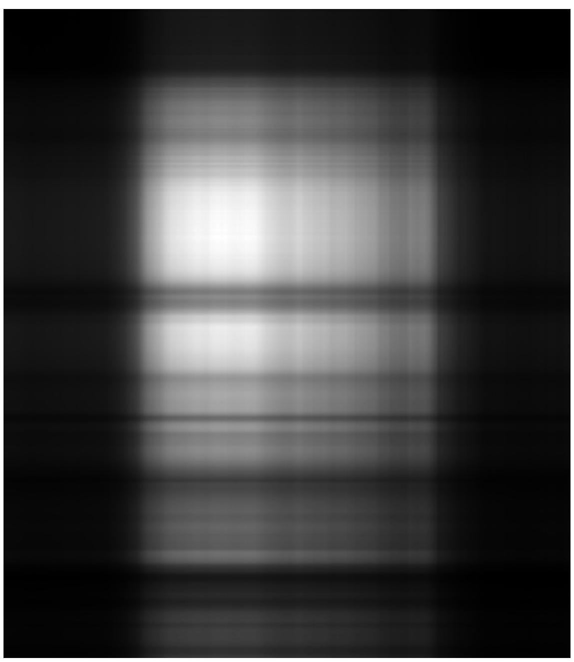
2) Replace A by the matrix

$$B_k = \sigma_1(\mathbf{u}_1\mathbf{v}_1^T) + \dots + \sigma_k(\mathbf{u}_k\mathbf{v}_k^T)$$

for some $1 \leq k \leq 700$. This matrix can be stored using $k \cdot (800 + 700 + 1)$ numbers.

Singular values of the matrix A





matrix B_1
1501 bytes
compression 374:1

matrix B_5
7905 bytes
compression 75:1

matrix B_{10}
15,010 bytes
compression 37:1

matrix B_{20}
30,020 bytes
compression 18:1

matrix B_{50}
75,050 bytes
compression 7:1

matrix B_{100}
150,100 bytes
compression 4:1