DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

Algebra Linear Computacional

SVD e Aproximação de Matrizes

Erickson e Fabricio

Slides parcialmente baseados em J. Leskovec, A. Rajaraman, J. Ullman: Mining of Massive Data Sets

Objetivos de Aprendizagem

1. Entender fundamentos da aplicação do SVD para sistemas de recomendação
2. Calcular representação latente de usuários e filmes
3. Saber fazer a compressão de imagens usando SVD
4. Entender a relação entre qualidade da imagem comprimida e seu espectro (conjunto de valores singulares)
5. Saber encontrar aproximação de matrizes (ou de elementos específicos) usando SVD truncado
6. Conhecer e entender o Teorema de Eckart-Young

Referências Adicionais

- YouTube
  - Jure Leskovec: SVD for Recommender Systems
    https://youtu.be/K38wVcdNuFc
  - Steve Brunton: SVD for Image Compression (Python)
    https://youtu.be/H7qMMudo3e8

DCC

AV=UΣ

$$\underset{m\times n}{A}\underbrace{\begin{bmatrix} v_1 & v_2 & \dots & v_r \end{bmatrix}}_{\substack{V \\ n\times r}} = \underbrace{\begin{bmatrix} u_1 & u_2 & \dots & u_r \end{bmatrix}}_{\substack{U \\ m\times r}} \underbrace{\begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_r \end{bmatrix}}_{\substack{\Sigma \\ r\times r}}$$

$$Av_i = \sigma_i u_i$$

Nota: Estas dimensões correspondem ao SVD reduzido.
Retificando a explicação: precisamos transformar U e V em matrizes **quadradas**.

DCC

Tornando V quadrada

Se o espaço coluna C(A) tem dimensão r, existem r vetores $\sigma_i u_i$ independentes que podem ser escritos como Av_i.

Sabemos ainda que o espaço nulo N(A) tem dimensão n-r. Logo, existem vetores n-r vetores v_j independentes tais que $Av_j = 0$. Esses vetores são ortogonais a $v_1, \ldots, v_r$. Concatenando-os a V,

$$A \overbrace{\begin{bmatrix} | & \cdots & | & | & \cdots & | \\ v_1 & \cdots & v_r & v_{r+1} & \cdots & v_n \\ | & \cdots & | & | & \cdots & | \end{bmatrix}}^{V} = \underbrace{\begin{bmatrix} | & \cdots & | \\ u_1 & \cdots & u_r \\ | & \cdots & | \end{bmatrix}}_{m \times r} \underbrace{\begin{bmatrix} \sigma_1 & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \ldots & \sigma_r & 0 & \ldots & 0 \end{bmatrix}}_{r \times n}$$

m x r

n-r

Tornando U quadrada

Como os vetores u_i estão no R^m, existem m-r vetores independentes u_j ortogonais a $u_1, \ldots, u_r$. Logo, podemos concatená-los a U, desde que sejam adicionadas m-r linhas nulas em Σ:

$$A\underbrace{\begin{bmatrix} | & \cdots & | & | & \cdots & | \\ v_1 & \cdots & v_r & v_{r+1} & \cdots & v_n \\ | & \cdots & | & | & \cdots & | \end{bmatrix}}_{n\times n} = \overbrace{\underbrace{\begin{bmatrix} | & \cdots & | & | & \cdots & | \\ u_1 & \cdots & u_r & u_{r+1} & \cdots & u_m \\ | & \cdots & | & | & \cdots & | \end{bmatrix}}_{m\times m}}^{\substack{U \\ m \times r \qquad \qquad m-r}} \underbrace{\begin{array}{r} \\ \\ \\ m-r \left\{ \begin{array}{c} \\ \\ \\ \end{array}\right. \end{array} \begin{bmatrix} \sigma_1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \sigma_r & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \cdots & 0 \end{bmatrix}}_{m\times n}$$

Nota: Estas dimensões correspondem ao SVD completo.

Do SVD completo ao reduzido

SVD Completo

$$\underbrace{A}_{m\times n} = \underbrace{U}_{m\times m}\underbrace{\Sigma}_{m\times n}\underbrace{V^\top}_{n\times n}$$

$$\underbrace{A}_{m\times n}\underbrace{V}_{n\times n} = \underbrace{U}_{m\times m}\underbrace{\Sigma}_{m\times n}$$

$$\underbrace{A}_{m\times n} = \underbrace{U}_{m\times r}\underbrace{\Sigma}_{r\times r}\underbrace{V^T}_{r\times n}$$

Como apenas os r primeiros elementos da diagonal de Σ são não-nulos, podemos tomar esta submatriz, descartando as colunas r+1,...,m de U e as linhas r+1,...,n de V^T. O resultado, denotado pelas mesmas letras, é

SVD Reduzido

$$\underbrace{A}_{m\times n} = \underbrace{U}_{m\times r}\underbrace{\Sigma}_{r\times r}\underbrace{V^\top}_{r\times n}$$

Exemplo de uma matriz tall-thin

$$\begin{array}{c c l}
 & \begin{array}{ccccc}\text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie}\end{array} & \\
\begin{array}{c} \uparrow \\ \text{SciFi} \\ \downarrow \\ \\ \uparrow \\ \text{Romance} \\ \downarrow \end{array} &
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
3 & 3 & 3 & 0 & 0 \\
4 & 4 & 4 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 0 & 0 & 2 & 2
\end{bmatrix} &
\begin{array}{l} \text{Ana} \\ \text{Bruno} \\ \dots \\ \\ \\ \\ \\ \end{array}
\end{array}$$

Aprendendo sobre espaço latente

$$
\begin{array}{cc}
 & \begin{array}{ccccc}\text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie}\end{array} \\
\begin{array}{c}\\ \text{SciFi} \\ \\ \\ \\ \text{Romance} \\ \\ \end{array} & \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ 5 & 5 & 5 & 0 & 0 \\ 0 & 0 & 0 & 4 & 4 \\ 0 & 0 & 0 & 5 & 5 \\ 0 & 0 & 0 & 2 & 2 \end{bmatrix}
\end{array}
=
\begin{bmatrix} 0.14 & 0.00 \\ 0.42 & 0.00 \\ 0.56 & 0.00 \\ 0.70 & 0.00 \\ 0.00 & 0.60 \\ 0.00 & 0.75 \\ 0.00 & 0.30 \end{bmatrix}
\times
\begin{bmatrix} 12.4 & 0 \\ 0 & 9.5 \end{bmatrix}
\times
\begin{bmatrix} 0.58 & 0.58 & 0.58 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.71 & 0.71 \end{bmatrix}
$$

Usuário e conceito

$$
\begin{array}{cc}
 & \begin{array}{ccccc}\text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie}\end{array} \\
\begin{array}{c}\uparrow \\ \text{SciFi} \\ \downarrow \\ \uparrow \\ \text{Romance} \\ \downarrow\end{array} &
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
3 & 3 & 3 & 0 & 0 \\
4 & 4 & 4 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 0 & 0 & 2 & 2
\end{bmatrix}
\end{array}
=
\begin{array}{c}
\begin{array}{cc}\text{SciFi-concept} & \text{Romance-concept}\end{array} \\
\begin{bmatrix}
0.14 & 0.00 \\
0.42 & 0.00 \\
0.56 & 0.00 \\
0.70 & 0.00 \\
0.00 & 0.60 \\
0.00 & 0.75 \\
0.00 & 0.30
\end{bmatrix}
\end{array}
\times
\begin{bmatrix}
12.4 & 0 \\
0 & 9.5
\end{bmatrix}
\times
\begin{bmatrix}
0.58 & 0.58 & 0.58 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.71 & 0.71
\end{bmatrix}
$$

U is “user-to-concept” similarity matrix

Filme e conceito

SciFi-concept

Romance-concept

$$
\begin{array}{c|c}
 & \begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \end{array} \\
\begin{array}{c} \text{SciFi} \\ \\ \\ \\ \text{Romance} \\ \\ \\ \end{array} &
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
3 & 3 & 3 & 0 & 0 \\
4 & 4 & 4 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 0 & 0 & 2 & 2
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
0.14 & 0.00 \\
0.42 & 0.00 \\
0.56 & 0.00 \\
0.70 & 0.00 \\
0.00 & 0.60 \\
0.00 & 0.75 \\
0.00 & 0.30
\end{bmatrix}
\times
\begin{bmatrix}
12.4 & 0 \\
0 & 9.5
\end{bmatrix}
\times
\begin{bmatrix}
0.58 & 0.58 & 0.58 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.71 & 0.71
\end{bmatrix}
$$

V is “movie to concept” similarity matrix

Peso de um conceito

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

$= \sigma_1 [\quad] + \sigma_2 [\quad]$

SciFi-concept

Romance-concept

Σ is the "concept strength" matrix

$$
\begin{array}{c}
 \\
\text{SciFi} \\
 \\
 \\
\text{Romance} \\
 \\
 \\
\end{array}
\begin{array}{c}
\begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \end{array} \\
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
3 & 3 & 3 & 0 & 0 \\
4 & 4 & 4 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 0 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 0 & 0 & 2 & 2
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
0.14 & 0.00 \\
0.42 & 0.00 \\
0.56 & 0.00 \\
0.70 & 0.00 \\
0.00 & 0.60 \\
0.00 & 0.75 \\
0.00 & 0.30
\end{bmatrix}
\times
\begin{bmatrix}
12.4 & 0 \\
0 & 9.5
\end{bmatrix}
\times
\begin{bmatrix}
0.58 & 0.58 & 0.58 & 0.00 & 0.00 \\
0.00 & 0.00 & 0.00 & 0.71 & 0.71
\end{bmatrix}
$$

Exemplo um pouco mais real

Note que existem usuários que vão assistir filmes de vários tipos (isso pode ser apenas ruído)

$$
\begin{array}{c}
\\ \\ \\ \text{SciFi} \\ \\ \\ \text{Romance} \\ \\
\end{array}
\begin{array}{c}
\begin{array}{ccccc} \text{Matrix} & \text{Alien} & \text{Serenity} & \text{Casablanca} & \text{Amelie} \end{array} \\
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
3 & 3 & 3 & 0 & 0 \\
4 & 4 & 4 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & \boxed{2} & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & \boxed{1} & 0 & 2 & 2
\end{bmatrix}
\end{array}
=
\begin{bmatrix}
0.13 & -0.02 & -0.01 \\
0.41 & -0.07 & -0.03 \\
0.55 & -0.09 & -0.04 \\
0.68 & -0.11 & -0.05 \\
0.15 & 0.59 & 0.65 \\
0.07 & 0.73 & -0.67 \\
0.07 & 0.29 & 0.32
\end{bmatrix}
\times
\begin{bmatrix}
12.4 & 0 & 0 \\
0 & 9.5 & 0 \\
0 & 0 & 1.3
\end{bmatrix}
\times
\begin{bmatrix}
0.56 & 0.59 & 0.56 & 0.09 & 0.09 \\
-0.12 & 0.02 & -0.12 & 0.69 & 0.69 \\
0.40 & -0.80 & 0.40 & 0.09 & 0.09
\end{bmatrix}
$$

Exemplo um pouco mais real

$$
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
3 & 3 & 3 & 0 & 0 \\
4 & 4 & 4 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 2 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 1 & 0 & 2 & 2
\end{bmatrix}
\approx
\begin{bmatrix}
\mathbf{0.13} & -0.02 & \cancel{-0.01} \\
\mathbf{0.41} & -0.07 & \cancel{-0.03} \\
\mathbf{0.55} & -0.09 & \cancel{-0.04} \\
\mathbf{0.68} & -0.11 & \cancel{-0.05} \\
0.15 & \mathbf{0.59} & \cancel{\mathbf{0.65}} \\
0.07 & \mathbf{0.73} & \cancel{\mathbf{-0.67}} \\
0.07 & \mathbf{0.29} & \cancel{\mathbf{0.32}}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{12.4} & 0 & 0 \\
0 & \mathbf{9.5} & 0 \\
0 & 0 & \cancel{\mathbf{1.3}}
\end{bmatrix}
\times
\begin{bmatrix}
\mathbf{0.56} & \mathbf{0.59} & \mathbf{0.56} & 0.09 & 0.09 \\
-0.12 & 0.02 & -0.12 & \mathbf{0.69} & \mathbf{0.69} \\
\cancel{0.40} & \cancel{\mathbf{-0.80}} & \cancel{0.40} & \cancel{0.09} & \cancel{0.09}
\end{bmatrix}
$$

Um sistema de recomendação simples

- Temos k perfis de usuários
  - Cada usuário será o resultado da combinação linear desses perfis (comédia, drama, ação)
  - A matrix "verdadeira" então terá posto k!
- Pergunta
  - Como recomendar um filme para um determinado usuário?

Um sistema de recomendação simples

$$
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
0 & 3 & 3 & 0 & 0 \\
4 & 4 & 0 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 2 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 1 & 0 & 2 & 2
\end{bmatrix}
=
\overbrace{\begin{bmatrix}
0.14 & -0.06 & -0.04 \\
0.30 & -0.11 & -0.61 \\
0.43 & -0.16 & 0.76 \\
0.74 & -0.31 & -0.18 \\
0.15 & 0.53 & 0.02 \\
0.07 & 0.70 & -0.03 \\
0.07 & 0.27 & 0.01
\end{bmatrix}}^{U}
\times
\overbrace{\begin{bmatrix}
12.4 & 0 & 0 \\
0 & 9.5 & 0 \\
0 & 0 & 1.3
\end{bmatrix}}^{\Sigma}
\times
\underbrace{\begin{bmatrix}
0.51 & 0.66 & 0.44 & 0.23 & 0.23 \\
-0.24 & -0.13 & -0.21 & 0.66 & 0.66 \\
0.59 & 0.08 & -0.80 & 0.01 & 0.01
\end{bmatrix}}_{V^T}
$$

Um sistema de recomendação simples

Como estamos supondo termos k=2 perfis

$$
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 \\
0 & 3 & 3 & 0 & 0 \\
4 & 4 & 0 & 0 & 0 \\
5 & 5 & 5 & 0 & 0 \\
0 & 2 & 0 & 4 & 4 \\
0 & 0 & 0 & 5 & 5 \\
0 & 1 & 0 & 2 & 2
\end{bmatrix}
\cong
\begin{bmatrix}
0.14 & -0.06 & \cancel{-0.04} \\
0.30 & -0.11 & \cancel{-0.61} \\
0.43 & -0.16 & \cancel{0.76} \\
0.74 & -0.31 & \cancel{-0.18} \\
0.15 & 0.53 & \cancel{0.02} \\
0.07 & 0.70 & \cancel{-0.03} \\
0.07 & 0.27 & \cancel{0.01}
\end{bmatrix}
\times
\begin{bmatrix}
12.4 & 0 & \cancel{0} \\
0 & 9.5 & \cancel{0} \\
\cancel{0} & \cancel{0} & \cancel{1.3}
\end{bmatrix}
\times
\begin{bmatrix}
0.51 & 0.66 & 0.44 & 0.23 & 0.23 \\
-0.24 & -0.13 & -0.21 & 0.66 & 0.66 \\
\cancel{0.59} & \cancel{0.08} & \cancel{-0.80} & \cancel{0.01} & \cancel{0.01}
\end{bmatrix}
$$

Um sistema de recomendação simples

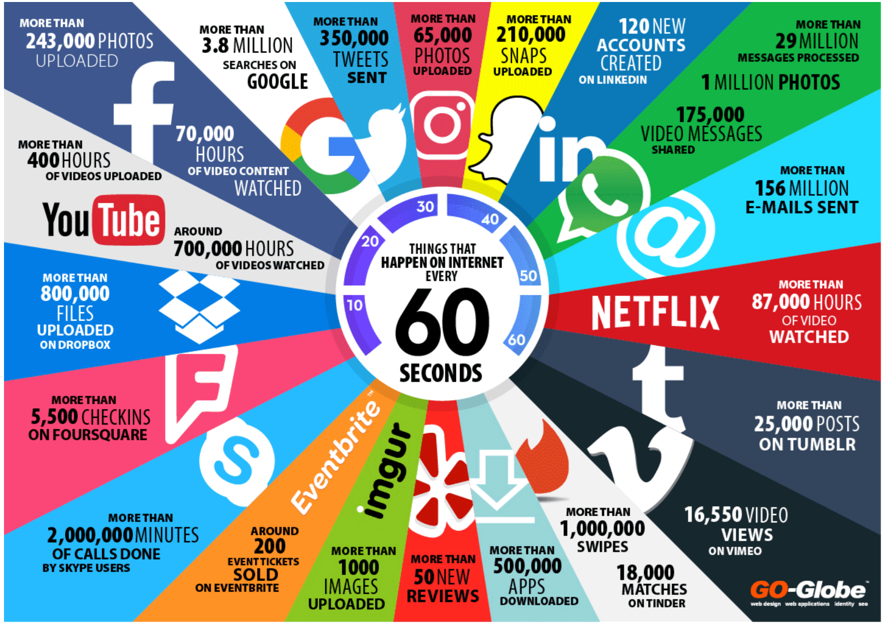
Ao realizar a multiplicação (após remover o menor valor singular)

$$\begin{bmatrix} 0.96 & 1.14 & 0.82 & -0.01 & -0.01 \\ 1.94 & 2.32 & 1.66 & 0.07 & 0.07 \\ 2.77 & 3.32 & 2.37 & 0.08 & 0.08 \\ 4.84 & 5.74 & 4.14 & -0.08 & 0.08 \\ 0.40 & 1.42 & 0.33 & 4.06 & 4.06 \\ -0.42 & 0.63 & -0.38 & 4.92 & 4.92 \\ 0.20 & 0.71 & 0.16 & 2.03 & 2.03 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 2 & 2 & 2 & 0 & 0 \\ 3 & 3 & 3 & 0 & 0 \\ 4 & 4 & 4 & 0 & 0 \\ & & \vdots & & \end{bmatrix}$$

Compressão de Imagens

~80% da internet é imagem!



Compressão de Imagens

Dimensão: 3024 x 4032

Número de bytes: 12.192.768 x 3 (considerando cor)

Compressão de Imagens

Podemos tentar aproximar a matriz A que representa a imagem por A_k

$$A_k = \sigma_1 u_1 v_1^T + \cdots + \sigma_k u_k v_k^T$$

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

$\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_k \geq \sigma_{k+1} \geq \ldots \geq \sigma_r$

Compressão de Imagens

(3024,)

U, sigma, Vt = np.linalg.svd(img_gray)

elementos da diagonal de Σ

3024 x 4032

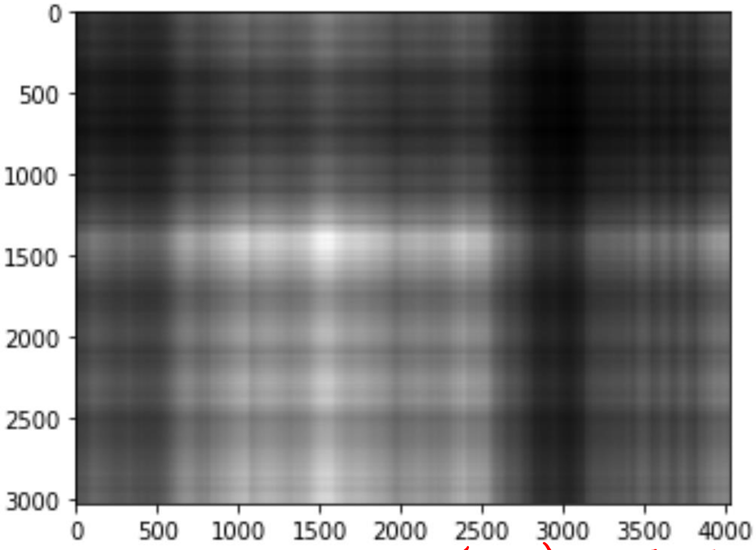
- U.shape: 3024 x 3024
- Vt.shape: 4032 x 4032
- sigma.shape: 3024 x 3024 (sendo apenas 3024 valores diferentes de zero)

sigma = np.diag(sigma)
3024x3024

$$\underset{m \times n}{A} = \underset{m\times m}{U}\,\underset{m\times n}{\Sigma}\,\underset{n\times n}{V^T}$$

Compressão de Imagens

A primeira aproximação seria: $A_1 = \sigma_1 u_1 v_1^T$



sigma[0] * U[:,0].reshape((-1,1)) @ Vt[0,:].reshape((1,-1))

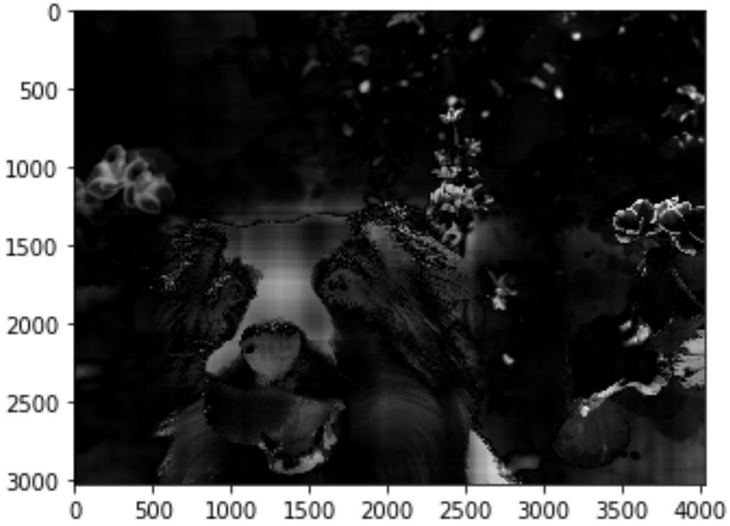
reconstimg = np.matrix(U[:, :1]) * np.diag(sigma[:1]) * np.matrix(V[:1, :])

Compressão de Imagens

Ficou bom?

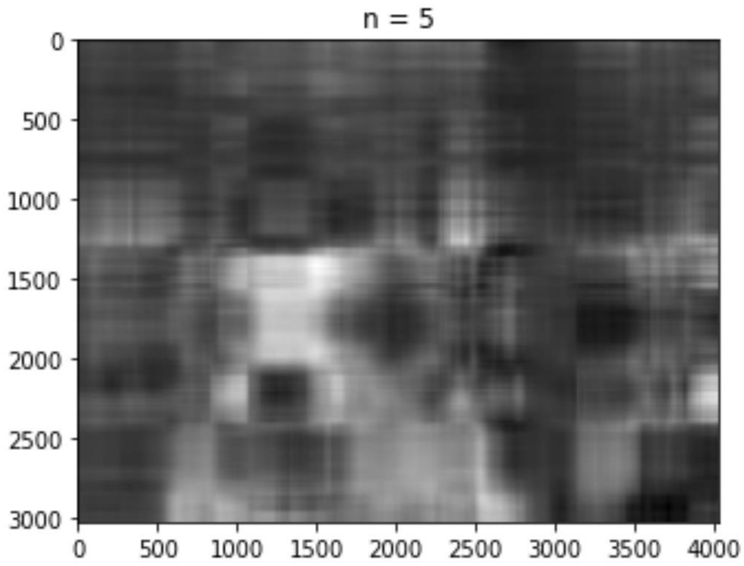
$$\underbrace{RMSE}_{M \times N} = \sqrt{\frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\left(\underbrace{I1}_{\text{original}}(m,n) - \underbrace{I2}_{\text{aproximada}}(m,n)\right)^2}\,.$$

MSE: 2486.98



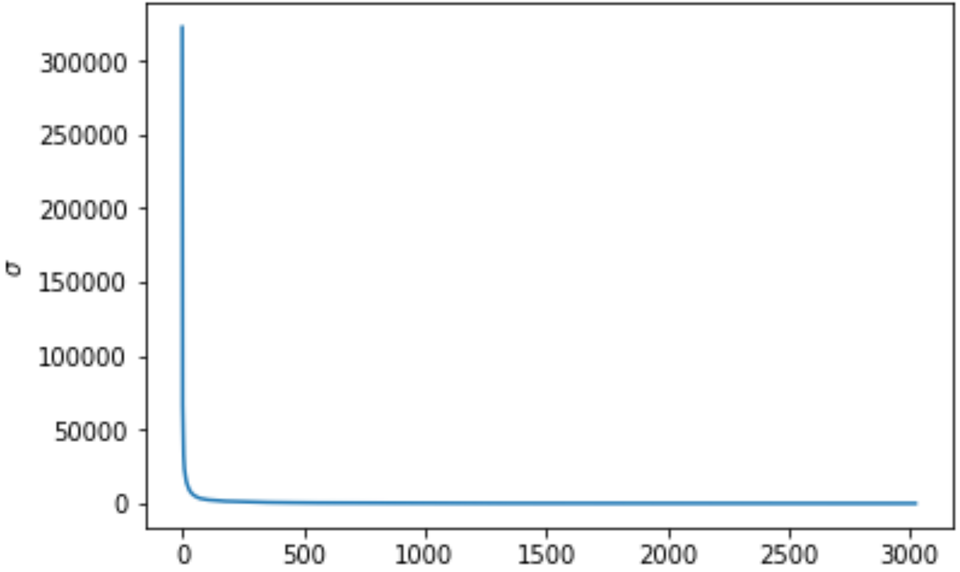
Compressão de Imagens

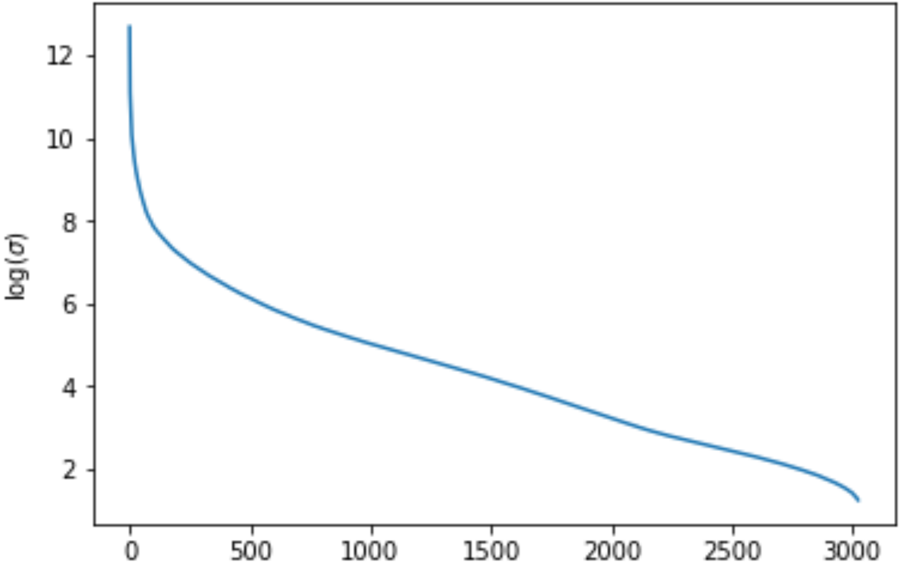
Uma aproximação seria usar mais valores singulares: A_5



reconstimg = ~~np.matrix~~(U[:, :5]) @ np.diag(sigma[:5]) @ ~~np.matrix~~(V[:5, :])

Até quando precisamos ir?

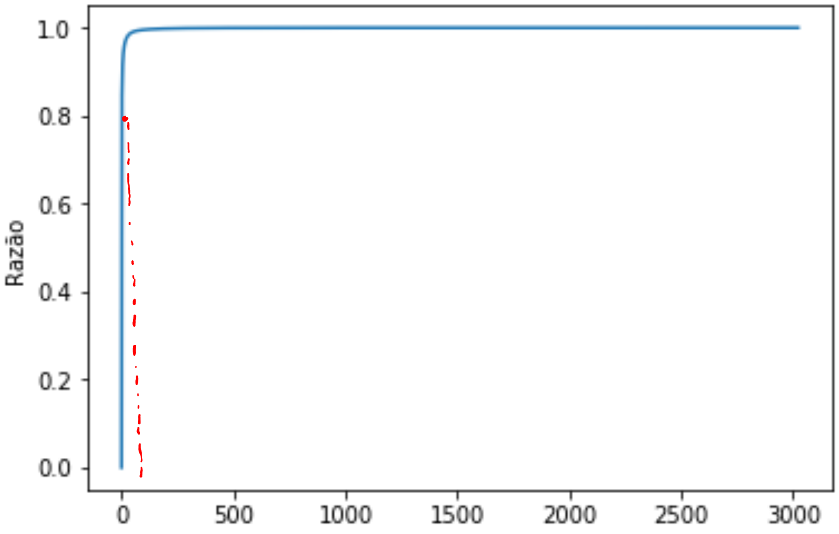




Até quando precisamos ir?

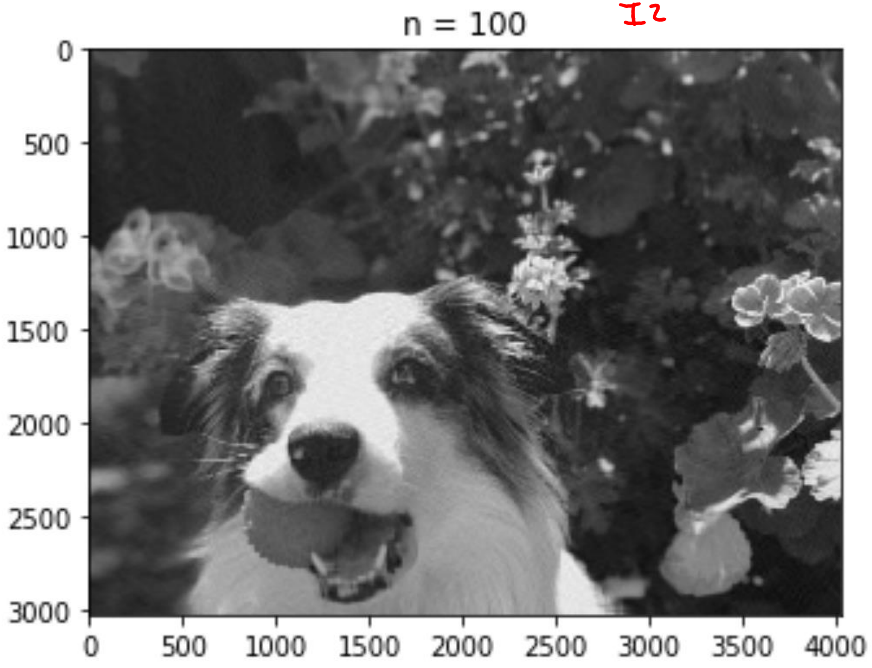
Regra do dedão: cortar de maneira a mantar entre 80-90% da “energia”

$$t = \frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{n} \sigma_i^2} \simeq 80\%$$

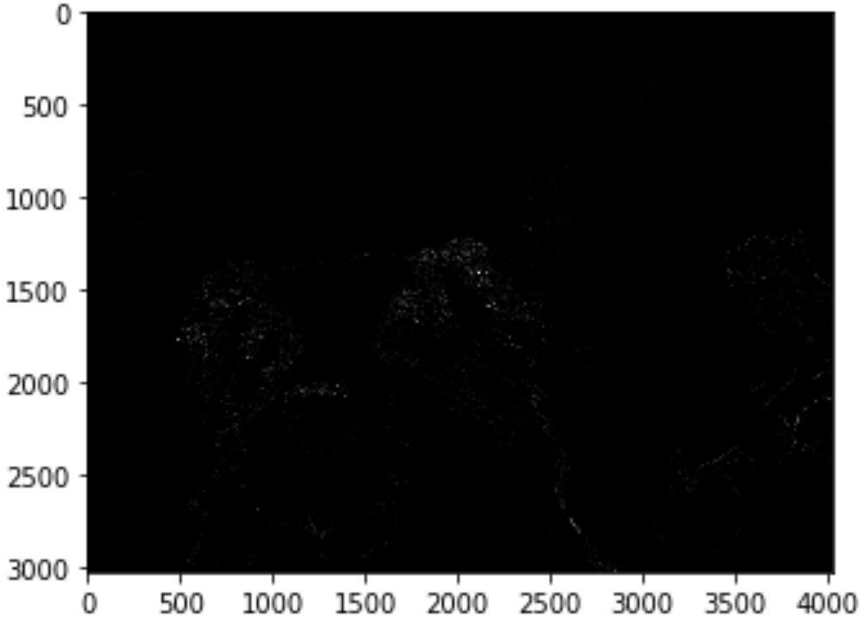


n=100 já temos uma ótima aproximação

I2



I1 - I2



O quanto estamos usando de espaço?

Tx = (100*3024 + 100 + 100*4032)/(3024*4032)

Tx = 5.7%

100 colunas de U 3024x3024

100 elementos de Σ 3024

100 linhas de V^T 4032x4032

Razão de compressão:

(3024*4032)/(100*3024 + 100 + 100*4032) = 17 vezes!

Aproximação da matriz A

É possível encontrar uma matriz B com posto k que seja mais próxima de A e que não seja A_k?

Aproximação de matriz

Eckart-Young: Suponha B uma matriz de posto k. Então B será uma aproximação no máximo tão boa quanto A_k

$\|A - A_k\|_2 \leq \|A - B\|_2$

onde B tem posto k.

$$A_k = \sigma_1 u_1 v_1^T + \cdots + \sigma_k u_k v_k^T$$

Aproximação de matriz

$\|C\|_2$ é o maior valor singular.

Se B tem posto k, então

$$\|A - B\|_2 \geq \underbrace{\|A - A_k\|_2}_{\sigma_{k+1} u_{k+1} v_{k+1}^T + \ldots + \sigma_p u_p v_p^T} = \sigma_{k+1}$$

Norma de Frobenius:

$$\|A\|_F = \sqrt{|a_{11}|^2 + |a_{12}|^2 + \cdots + |a_{mn}|^2}$$

$\|A\|_F \geq \|A\|_2$