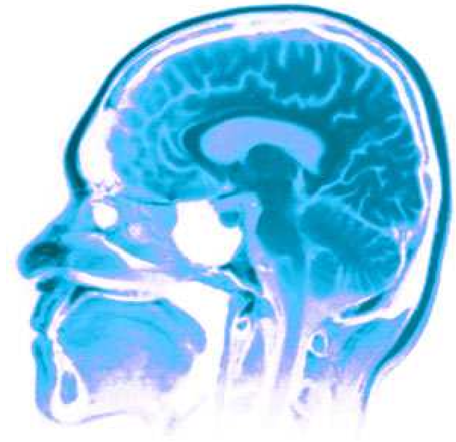
CPSC340



Dimensionality reduction with the SVD

Nando de Freitas
October, 2012
University of British Columbia

Outline of the lecture

This lecture introduces the singular value decomposition (SVD). The SVD is a matrix factorization that has many applications. The goal of the lecture is for you to learn:

- The definition of the SVD
- How to compute the SVD for a small matrix
- Low-rank approximation
- Application to lossy image compression

SVD decomposition

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

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

$\mathbf{\Sigma} \in \mathbb{R}^{n \times n}$ is diagonal with positive entries (singular values in the diagonal).
$\mathbf{U} \in \mathbb{R}^{m \times n}$ has orthonormal columns.
$\mathbf{V} \in \mathbb{R}^{n \times n}$ has orthonormal columns and rows.
That is, $\mathbf{V}$ is an orthogonal matrix, so $\mathbf{V}^{-1} = \mathbf{V}^T$.

$\boxed{U^T U = I}$

In some code U is m x m

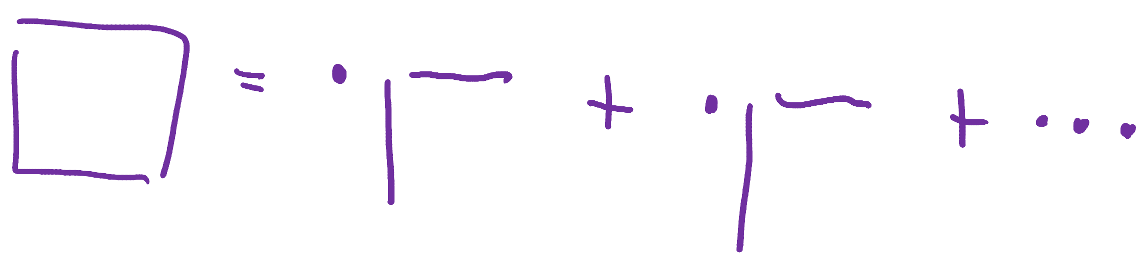
$$A = \begin{bmatrix} U & | & | & | \end{bmatrix} \begin{bmatrix} \Sigma \\ 0 \; 0 \; 0 \; 0 \\ 0 \quad 0 \quad 0 \\ 0 \quad 0 \quad 0 \end{bmatrix} \begin{bmatrix} V^T \end{bmatrix}$$

Equivalent ways of writing the SVD (thin)

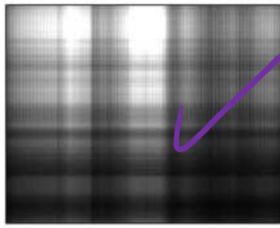
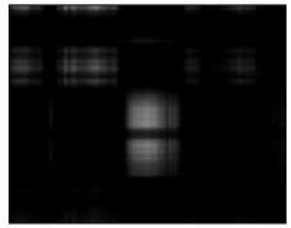
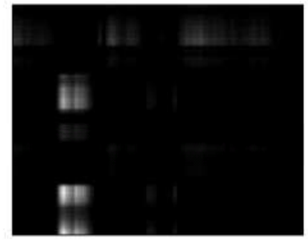
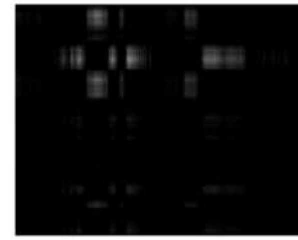
$$\overset{m\times n}{\underline{\underline{A}}} = \overset{m\times n}{\underline{\underline{U}}}\;\overset{n\times n}{\underline{\underline{\Sigma}}}\;\overset{n\times n}{\underline{\underline{V}}^T}$$

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ \vdots & \ddots & & \vdots \\ a_{m1} & & \ddots & a_{mn} \end{bmatrix} = \begin{bmatrix} \overset{m\times 1}{\underline{u}_1} & \underline{u}_2 & \cdots & \underline{u}_n \end{bmatrix} \begin{bmatrix} \sigma_1 & & & \\ & \sigma_2 & & \\ & & \ddots & \\ & & & \sigma_n \end{bmatrix} \begin{bmatrix} \overset{1\times n}{\underline{v}_1^T} \\ \vdots \\ \underline{v}_n^T \end{bmatrix}$$

$$A = \sigma_1 \underline{u}_1 \underline{v}_1^T + \sigma_2 \underline{u}_2 \underline{v}_2^T + \sigma_3 \underline{u}_3 \underline{v}_3^T + \cdots + \sigma_n \underline{u}_n \underline{v}_n^T$$



Equivalent ways of writing the SVD

 =  +  +  +  + ...

$$A = \sigma_1 \underline{u}_1 \underline{v}_1^T + \sigma_2 \underline{u}_2 \underline{v}_2^T + \sigma_3 \underline{u}_3 \underline{v}_3^T + \sigma_4 \underline{u}_4 \underline{v}_4^T + \dots$$

$$\begin{bmatrix} 3 & 6 & 255 \\ 1 & 20 & 60 \\ 5 & 3 & 40 \end{bmatrix}$$

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \dots \geq 0$$

$$\begin{bmatrix} 1 \\ 2 \end{bmatrix} \begin{bmatrix} 3 & 4 \end{bmatrix} = \begin{bmatrix} 3 & 4 \\ 6 & 8 \end{bmatrix}$$

Computing the SVD

The nonzero singular values of $\mathbf{A}^{m \times n}$ *are the (positive) square roots of the nonzero eigenvalues of* $\mathbf{A^TA}$ *or* $\mathbf{AA^T}$

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

$$(A^TA)^T = A^T A^{T^T} = A^TA$$

$$A^TA = (U\Sigma V^T)^T U\Sigma V^T = (V^{T^T}\Sigma^T U^T) U\Sigma V^T$$

$$= V\Sigma U^T U\Sigma V^T = V\Sigma^2 V^T = X\Lambda X^T$$

$$AA^T = U\Sigma V^T (V\Sigma U^T) = U\Sigma^2 U^T$$

Computing the SVD

<u>Example:</u> Compute the SVD of the following matrix.

$m = 3$

$n = 1$

$$A = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$$A^TA = \begin{bmatrix} 1 & 2 & 3 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} = \underline{14} = \underbrace{\begin{bmatrix} 1 \end{bmatrix}}_{V}\underbrace{\begin{bmatrix} \sqrt{14} \end{bmatrix}^2}_{\Sigma^2}\underbrace{\begin{bmatrix} 1 \end{bmatrix}}_{V^T}$$

$$A^TA = V\Sigma^2V^T$$

$$AA^T = U\Sigma^2U^T$$

$$AA^T = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}\begin{bmatrix} 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}$$

$$\therefore \Sigma = \begin{bmatrix} \sqrt{14} \end{bmatrix}$$

$$V = \begin{bmatrix} 1 \end{bmatrix}$$

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

$$AV = U\Sigma \cancel{V^TV}$$

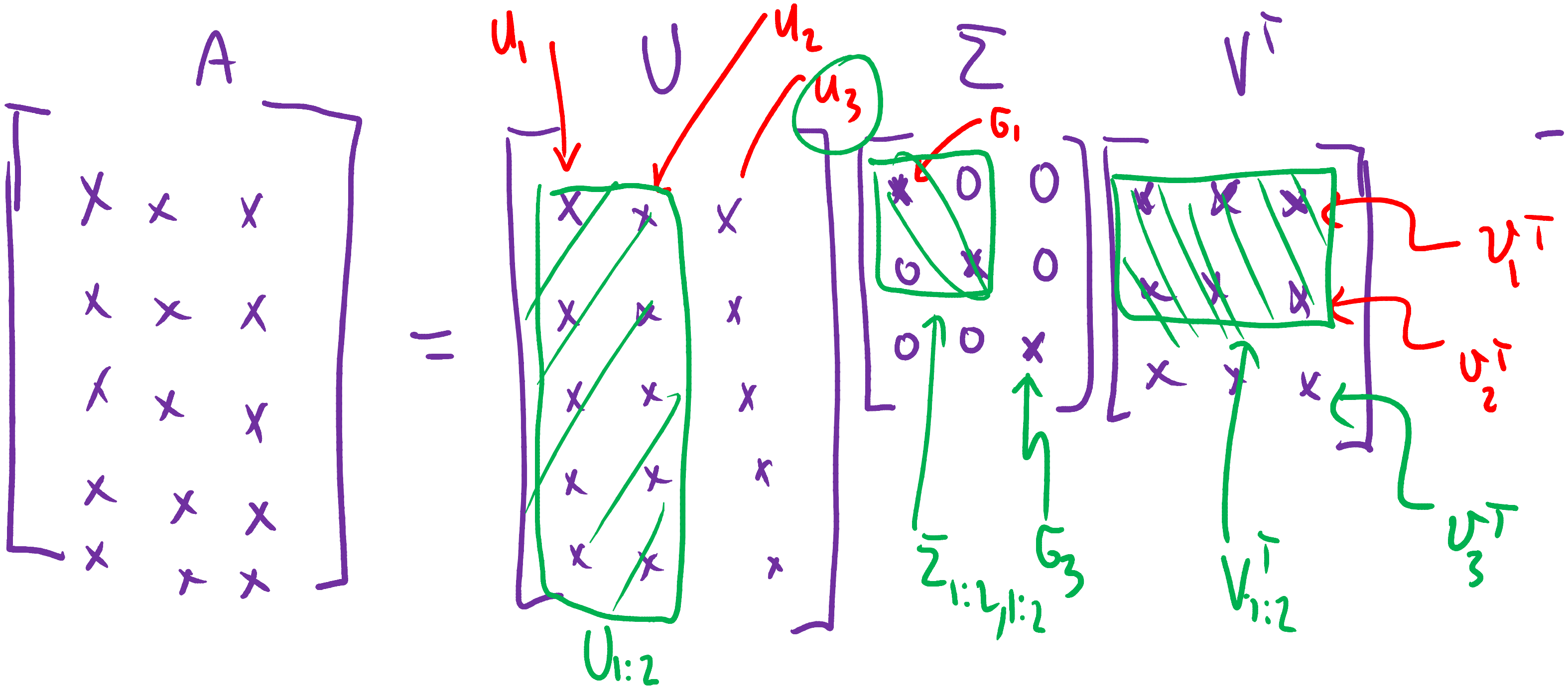
$$AV\Sigma^{-1} = U = \frac{1}{\sqrt{14}}\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq 0$

The truncated SVD

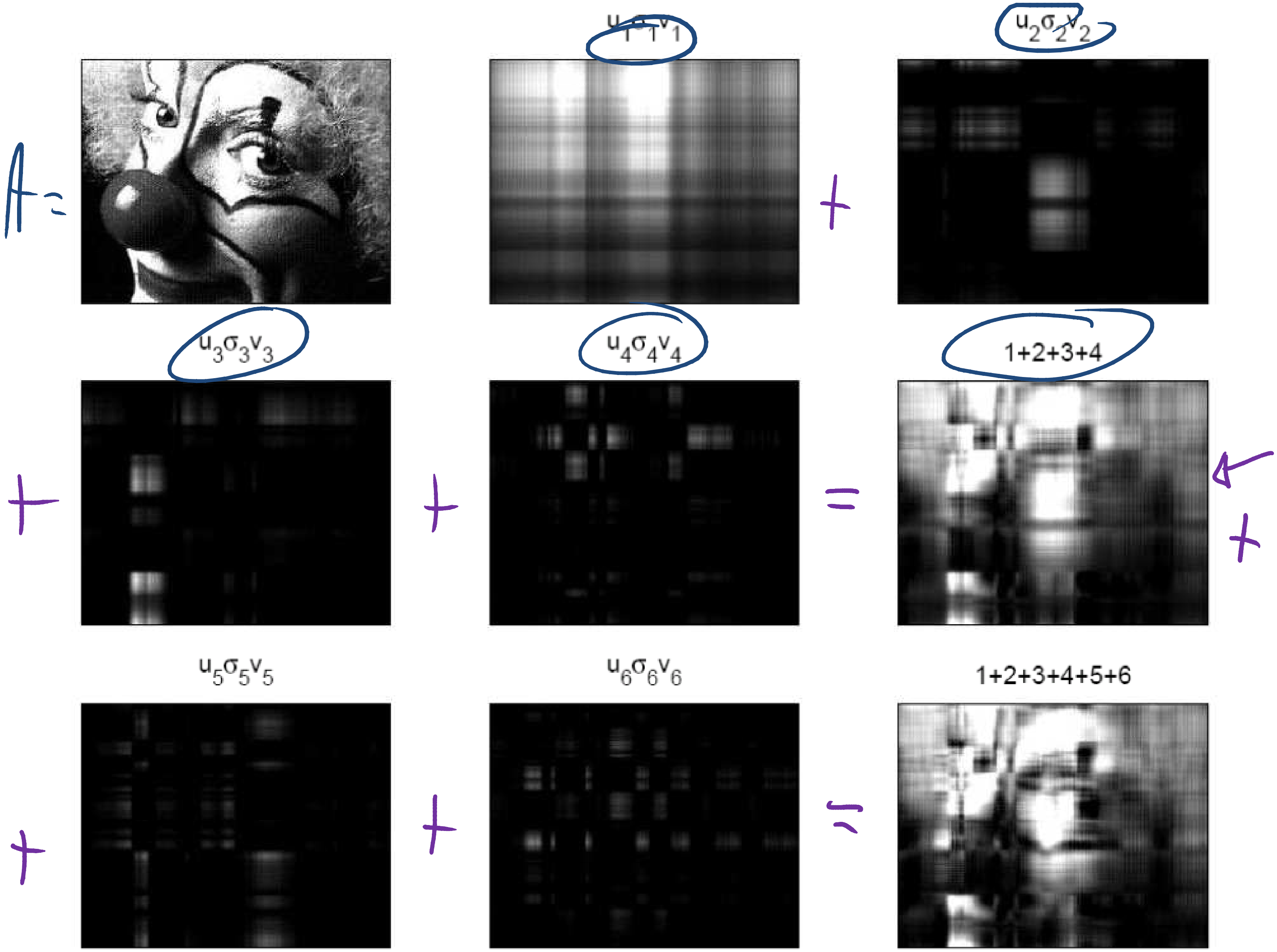
$m=5, \; n=3$

We can ***compress data*** *by using truncation of the eigen-components.*



$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \underline{\sigma_3 u_3 v_3^T}$$

$$A \approx \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T \leftarrow \text{compression}$$



Smaller eigenvectors capture high frequency variations (small brush-strokes).

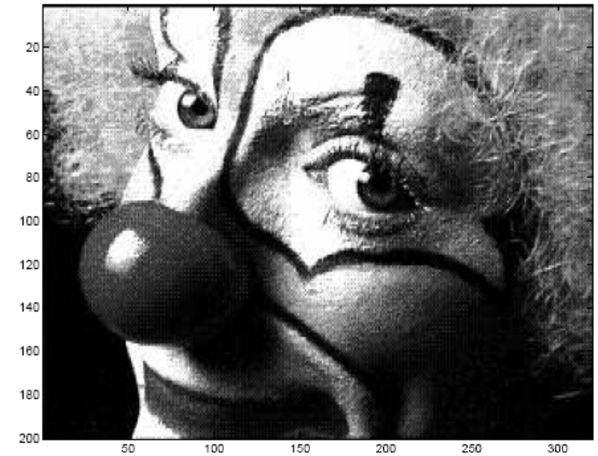
Image compression example in python

```
from scipy import *
from pylab import *

img = imread("clown.png")[:,:,0]
gray()
figure(1)
imshow(img)

m,n = img.shape
U,S,Vt = svd(img)
S = resize(S,[m,1])*eye(m,n)

k = 20
figure(2)
imshow(dot(U[:,1:k],dot(S[1:k,1:k],Vt[1:k,:])))
show()
```



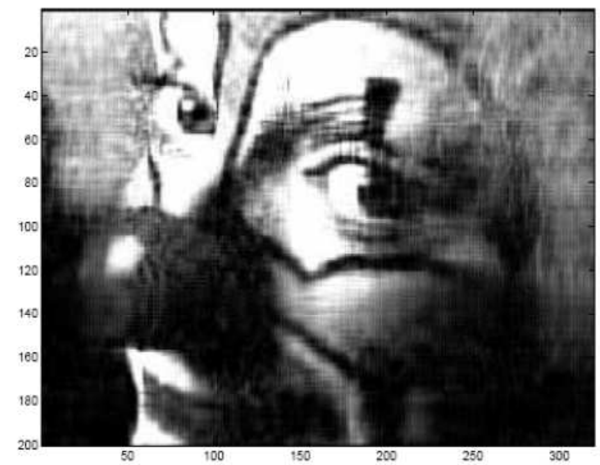


Image compression savings

The code:

- loads a clown image into a $\overset{m}{200}$ by $\overset{n}{320}$ array A,
- displays the image in one figure,
- performs a singular value decomposition on A,
- displays the image obtained from a rank-20 SVD approximation of A in another figure.

The original storage requirements for A are:

$$200 \times 320 = 64000 \text{ numbers}$$

The compressed representation requires: $(k=20)$

$$20 \times 200 + 20 + 20 \times 320 = 10420 \text{ numbers}$$

Next lecture

In the next lecture, we introduce PCA. PCA uses the SVD to project data to low dimensions. The projections are useful to:

- ❑ Eliminate redundant details

- ❑ Visualize data in 2D