Lecture 3:
SVD, LSI and PCA

Nando de Freitas
www.cs.ubc.ca/~nando/340-2008/
September 2008

Outline

This lecture will cover applications of the SVD to:

- image compression,
- dimensionality reduction & visualization
- information retrieval and latent semantic analysis.

The truncated SVD

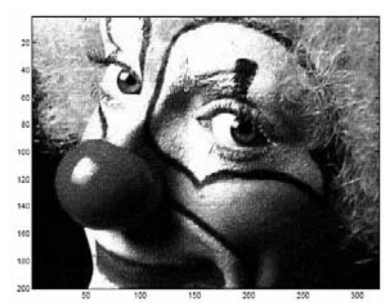
Image compression example in python

```
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()
```



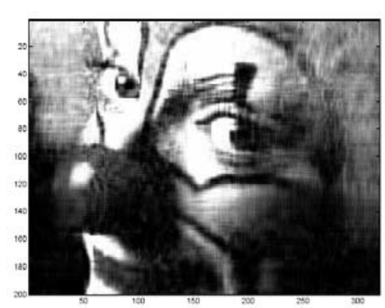


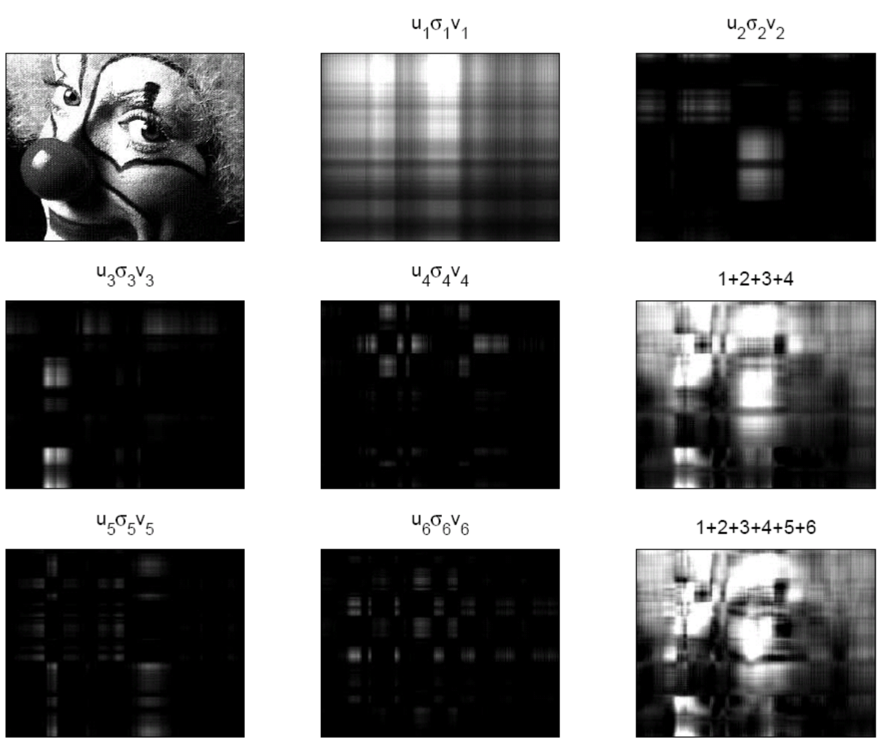
Image compression example

The code:

• loads a clown image into a 200 by 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:

The compressed representation requires:



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

Text retrieval: Latent semantic indexing (LSI)

The SVD can be used to cluster documents and carry out information retrieval by using concepts as opposed to exact word-matching.

This enables us to surmount the problems of synonymy (car, auto) and polysemy (money bank, river bank).

The data is available in a term-frequency (TF) matrix:

LSI example

Truncated SVD for LSI

If we truncate the approximation to the k-largest singular values, we have

$$\mathbf{A} = \mathbf{U}_k \mathbf{\Sigma}_k \mathbf{V}_k^T$$

So

$$\mathbf{V}_k^T = \mathbf{\Sigma}_k^{-1} \mathbf{U}_k^T \mathbf{A}$$

In English, $\mathbf{A}$ is projected to a lower-dimensional space spanned by the k singular vectors $\mathbf{U}_k$ (eigenvectors of $\mathbf{A}\mathbf{A}^T$).

Part I: Building the search engine

Part II: Querying the search engine

To carry out **retrieval**, a **query** $\mathbf{q} \in \mathbb{R}^n$ is first projected to the low-dimensional space:

$$\widehat{\mathbf{q}}_k = \mathbf{\Sigma}_k^{-1}\mathbf{U}_k^T\mathbf{q}$$

And then we measure the angle between $\widehat{\mathbf{q}}_k$ and the $\mathbf{v}_k$.

Part II: Querying the search engine

Final remark on LSI: TF - IDF

Principal component analysis

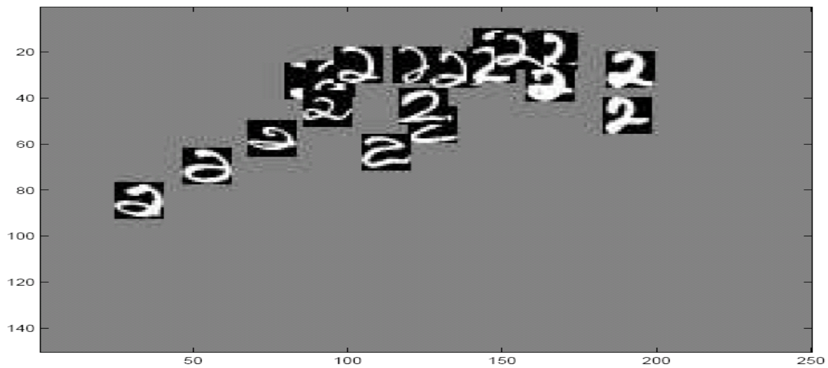
The columns of $\mathbf{U\Sigma}$ are called the **principal components** of $\mathbf{A}$. We can project high-dimensional data to these components in order to be able to visualize it. This idea is also useful for cleaning data as discussed in the previous text retrieval example.

PCA derivation: 2D to 1D

PCA derivation: 2D to 1D

PCA visualization example

For example, we can take several 16×16 images of the digit 2 and project them to 2D. The images can be written as vectors with 256 entries. We then from the matrix $\mathbf{A} \in \mathbb{R}^{n \times 256}$, carry out the SVD and truncate it to $k = 2$. Then the components $\mathbf{U}_k \mathbf{\Sigma}_k$ are 2 vectors with n data entries. We can plot these 2D points on the screen to visualize the data.



PCA visualization example