Math 343 Lab 9: Application of the SVD

Objective

In this lab, we explore an image compression application of the Singular Value Decomposition (SVD).

Introduction

Recall that the SVD is a decomposition of the $m \times n$ matrix A of rank r into the product $A = U\Sigma V^H$, where U and V are unitary matrices having dimensions $m \times m$ and $n \times n$, respectively, and Σ is the $m \times n$ diagonal matrix

$$\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r, 0, \ldots, 0),$$

where $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r > 0$ are the singular values of A. Upon closer inspection, we can write

$$U = \begin{pmatrix} U_1 & U_2 \end{pmatrix}, \quad \Sigma = \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix}, \quad V = \begin{pmatrix} V_1 & V_2 \end{pmatrix},$$

where U_1 and V_1 have dimensions $m \times r$ and $n \times r$ respectively and Σ_r is the $r \times r$ diagonal matrix of (nonzero) singular values. Multiplying this out yields the reduced form of the SVD

$$A = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_1^H \\ V_2^H \end{pmatrix} = U_1 \Sigma_r V_1^H$$

Low rank data storage

If the rank of a given matrix is significantly smaller than its dimensions, the reduced form of the SVD offers a way to store the matrix with less memory. As it is, the $m \times n$ matrix requires the storage of mn numbers, whereas U_1, Σ_r and V_1 in the reduced form of the SVD, all together require $r(m+n+1)$ numbers (verify this). Thus if r is much smaller than both m and n, we can obtain considerable efficiency. As an example, suppose $m = 100$, $n = 200$ and $r = 20$. Then the original matrix requires 20,000 numbers for storage whereas the reduced form of the SVD only requires 6020 numbers.

Low rank approximation

The reduced form of the SVD also provides a way to approximate a matrix with one of lower rank. This idea hits many areas of applied mathematics, including signal processing, statistics, semantic indexing (search engines), and control theory. Given a matrix A, we can find an approximate matrix $\widehat{A}$ of rank r by taking the SVD of A and setting all of its singular values after σ_r to zero, that is,

$$\sigma_{r+1} = \sigma_{r+2} = \cdots = \sigma_n = 0,$$

and then multiplying the matrix back together again. To see an example, enter the following into Matlab:

```
>> A = [1 1 3 4; 5 4 3 7; 9 10 10 12; 13 14 15 16; 17 18 19 20]

>> rank(A)

>> [U,S,V] = svd(A)

>> Ahat = U(:,1:3)*S(1:3,1:3)*V(:,1:3)'

>> rank(Ahat)
```

Note that $\widehat{A}$ is "close" to the original matrix A, but that its rank is 3 instead of 4.

Application to Imaging

Enter the following into Matlab:

```
>> load('clown.mat');

>> image(X);

>> colormap(gray); axis off;
```

The image `X` is a 200×320 matrix (type `size(X)` into Matlab's command line). The numbers range from 1 to 81 and correspond to different shades of gray. We compute the SVD of our image `X` by executing

```
>> [U,S,V] = svd(X);
```

Note that the rank of `X` is 200. We can reduce our clown image to a rank of 50 by executing the following:

```
>> n=50;

>> Xhat = U(:,1:n)*S(1:n,1:n)*V(:,1:n)';

>> image(Xhat);
```

Note that the clown's left cheek is a little blurry, but it otherwise looks ok. How low can you take the rank and still have a decent looking image? What happens when you take the rank too low?

Assignment

Problem 1. *Show that the reduced form of the SVD requires knowledge of only* $r(m+n+1)$ *numbers.*

Problem 2. *Explore the clown picture for several different values of rank. Conduct the experiments described above. Note that the original image takes 64,000 integers to store. Compare this with the storage needs for various lower-rank SVD approximations. What conclusions can you draw?*