HW #1 - due 10/17/2017

The goal of this assignment is to illustrate applications of the singular value decomposition (SVD) to low-rank representation of information from images. You are encouraged to use any software available in your favorite programming language. In particular, Matlab provides a friendly syntax for implementation.

Background: The singular value decomposition

Every matrix $A \in \mathbb{R}^{m\times n}$ can be decomposed as the product of 3 matrices

$$A = USV^T \in \mathbb{R}^{m\times n} \tag{1}$$

where $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{n\times n}$ are orthogonal matrices, $U^TU = I_{m\times m}, V^TV = I_{n\times n}$; $S \in \mathbb{R}^{m\times n}$ is a matrix with entries $S_{ii} = \sigma_i, i = 1:p$ where $p = \min\{m,n\}$ and zero entries elsewhere. The values σ_i are ordered such that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$ and are called the *singular values of* A. If we consider the columnwise representation $U = [u_1\ u_2 \ldots u_m]$ and $V = [v_1\ v_2 \ldots v_n]$ it follows from (1) that

$$AV = US \Rightarrow Av_i = \sigma_i u_i, \quad i = 1,2,\ldots,p$$

$$A^TU = VS^T \Rightarrow A^Tu_i = \sigma_i v_i, \quad i = 1,2,\ldots,p$$

and

$$(A^TA)v_i = \sigma_i^2 v_i, \quad i = 1,2,\ldots,p$$

such that σ_i^2 are the eigenvalues of A^TA and v_i are the associated eigenvectors; u_i are called *left singular vectors* of A and v_i are called *right singular vectors* of A. The *dyadic decomposition of* A is

$$A = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \ldots + \sigma_p u_p v_p^T \tag{2}$$

The singular value decomposition provides the solution to the following problem: Given $A \in \mathbb{R}^{m\times n}$ find $X \in \mathbb{R}^{m\times n}$, $rank(X) = k < rank(A)$, such that $\|A - X\|_2$ is minimized

$$\min_{rank(X)\leq k} \|A - X\|_2$$

The solution is obtained by truncating (2) to the k^{th} term

$$X_k = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \ldots + \sigma_k u_k v_k^T \tag{3}$$

If $\sigma_k > \sigma_{k+1}$, the absolute and relative errors in the approximation, measured in the matrix 2-norm, are

$$\|A - X_k\|_2 = \sigma_{k+1}, \qquad \frac{\|A - X_k\|_2}{\|A\|_2} = \frac{\sigma_{k+1}}{\sigma_1} \tag{4}$$

In addition, (3) also provides an *optimal rank k approximation in the Frobenius matrix norm*,

$$\min_{rank(X)\leq k} \|A - X\|_F = \|A - X_k\|_F$$

and the absolute and relative errors in the approximation measured in the Frobenius matrix norm are

$$\|A - X_k\|_F = \sqrt{\sigma_{k+1}^2 + \sigma_{k+2}^2 + \ldots + \sigma_p^2}, \qquad \frac{\|A - X_k\|_F}{\|A\|_F} = \frac{\sqrt{\sigma_{k+1}^2 + \sigma_{k+2}^2 + \ldots + \sigma_p^2}}{\sqrt{\sigma_1^2 + \sigma_2^2 + \ldots + \sigma_p^2}} \tag{5}$$

Homework content: optimal low-rank image approximation

We are interested to generate low rank approximations (3) to a given image represented by a matrix $A \in \mathbb{R}^{m\times n}$ where each entry (i, j) of the matrix specifies the value of the pixel (i, j) in the image. A measure of the amount of information captured by the first k pairs of singular vectors is given by the relative error in the approximation,

$$err(k) = \frac{\|A - X_k\|}{\|A\|} \tag{6}$$

that is evaluated in the 2-norm and Frobenius norm according to (4) and (5), respectively. For example, if $err(10) = 0.1$ then 90% of the information contained in A is captured by the first 10 pairs of singular vectors.

The data file clown1.m contains a 200×320 matrix A that represents the picture clown1.jpg. Alternatively, in Matlab use "load clown" since "clown.mat" should be available in the Matlab built in data files. The image is loaded by default in a matrix X. Then use "A=X; imagesc(A)", "colormap(gray)".

To find the SVD of a matrix A, in Matlab you may simply use

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

and the approximation (3) is obtained as

$$Xk = U(:, 1:k) * S(1:k, 1:k) * V(:, 1:k)'$$

Task 1 (50 points) Given a number $x, 0 < x < 1$, implement a function $[k2, kf] = svdinfo(A, x)$ that takes as input the matrix A and the number x and returns the smallest numbers $k2$ and kF for which $err(k) \leq x$ in the matrix 2-norm and Frobenius norm, respectively (that means: what is the minimum number of singular vectors that will capture x relative information from A?).

How many singular vectors are required to capture $90\%(x = 0.1), 99\%(x = 0.01), 99.9\%(x = 0.001)$ of information, respectively? Use (3) to construct the corresponding images that approximate the original.

Task 2 (10 points) For $x = 0.01 : 0.005 : 0.1$ (i.e., $x(1) = 0.01, x(2) = 0.015, \ldots x(19) = 0.1$) and $[k2, kf]$ =svdinfo(A, x) provide the graph k vs. x ($plot(x, k)$).

Task 3 (10 points) Consider $memA = m \times n$ as an indication of the memory space required for storage of the matrix A, and $memX(k) = k(n + m + 1)$ as an indication of the memory space required for storage of k pairs of singular vectors and the k singular values used to define the approximation (3). The ratio

$$r(k) = [memA - memX(k)]/memA$$

is an indication of the relative storage savings when A is replaced by $X(k)$. For

$$x = 0.01 : 0.01 : 0.5, \qquad [k2, kf] = svdinfo(A, x)$$

provide the graph of r versus x for the 2-norm and the Frobenius norm.

What to hand in

- Task 1: Listing of "svdinfo", number of singular vectors necessary to capture the required information.
- Task 2: The graph k vs. x
- Task 3: The graph r vs. x