Indian Institute of Science

DS284 - Numerical Linear Algebra

Exercise 3

September 27, 2018

1. As you know SVD can be used to compress data (lossy compression). Let us put that to the test. Take any bitmap image (close to 500*500) and try to find how many singular values are required to make the approximated image indistinguishable from the original image. Let this value be 'K'.

   Now for this value of 'K', find the number of matrix entries required to store the imaage. Compare that to the matrix entries required for the bitmap image (for 500*500, we have 500*500*3 entries). What is the compression ratio? compare that to compression techniques like jpeg and png.

   What is the L2 and frobenius-norm error obtained for various values of K. Check if the following theorems hold for those errors:

   For a matrix 'A' of rank 'r' with singular values $\sigma_1, \sigma_2, ..., \sigma_r$, A_v is the 'v' rank approximation ($A_v = \sum_{i=1}^{v} \sigma_i u_i v_i^T$) such that $1 < v < r$. Then:

   $$||A - A_v||_2 = \sigma_{v+1}$$

   $$||A - A_v||_F = \sqrt{\sigma_{v+1}^2 + \sigma_{v+2}^2 + ... + \sigma_r^2}$$

   you can find starter code and a sample image here: `https://github.com/dugarab/ds284/tree/master/SVD`

2. Given below is the matrix A:

   $$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 1 & 2 & 3 \\ 2 & 8 & 2 \end{bmatrix}$$

b = $[1, 2, 3, 4]^T$. Find the least squares solution for the above. Now do the same for the below matrix also:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 1 & 2 & 3 \\ 5 & 7 & 9 \end{bmatrix}$$

Were you able to find a least squares solution? If not explain why? Must there be a restriction on A for a least squares solution to exist or will it always exist?

3. Given link (least_squares_link) contains code to generate a rectangular matrix A and a vector b. Find the least squared solution of $Ax = b$ using psuedo-inverse method and the QR factorization method. Use the norm of the residue $(b - Ax)$ as an error estimate and plot the average error (over multiple problems) with increasing values of N (50,51,52,..,100).

   What do you observe?

4. The Wilkinson's polynomial is given by:

$$w(x) = \prod_{i=1}^{20}(x - i) = (x - 1)(x - 2)\cdots(x - 20)$$

   We know the roots are 1,2,3,..,20 because of the form presented, but something very peculiar happens when you write $w(x)$ as

$$w(x) = a_0 + a_1 x + a_2 x^2 + ... + a_{20} x^{20}$$

   and try to solve for roots. Provided link(wilkinsons_poly_link) is the MATLAB code containing the polynomial and its coefficients. Perturb the coefficients minimally and observe the effect on the roots. Can you explain this?