Lab Session 7

MA423 : Matrix Computations July-November 2021 S. Bora

---

1. The purpose of this exercise is to solve the Least-Squares Problem (in short, LSP) $Ax = b$ by different methods and compare the solutions. Here $A \in \mathbb{R}^{n\times m}$ and $b \in \mathbb{R}^n$, and usually n is much bigger than m.

   Origin: Suppose that we have a data set (t_i, b_i), for $i = 1 : m$, that have been obtained from some experiment. These data are governed by some unknown laws. So, the task is to come up with a model that best fits these data. A model is generated by a few functions, called model functions, $\phi_1, \ldots, \phi_n$. Therefore once a model is chosen, the task is to find a function p from the span of the model functions that best fits the data.

   Suppose that the model functions $\phi_1, \ldots, \phi_n$ are given. For $p \in \text{span}(\phi_1, \ldots, \phi_n)$, we have $p = x_1\phi_1 + \cdots + x_n\phi_n$ for some $x_j \in \mathbb{R}$. Now, forcing p to pass through the data (t_i, b_i) for $i = 1 : m$, we have $p(t_i) = b_i + r_i$, where r_i is the error. We want to choose that p for which the sum of the squares of the errors r_i is the smallest, that is, $\sum_{i=1}^m |r_i|^2$ is minimized.

   Now $p(t_i) = b_i + r_i$ gives $x_1\phi_1(t_i) + \cdots + x_n\phi_n(t_i) = b_i + r_i$. Thus in matrix notation,

$$\begin{bmatrix} \phi_1(t_1) & \cdots & \phi_n(t_1) \\ \phi_1(t_2) & \cdots & \phi_n(t_2) \\ \vdots & \cdots & \vdots \\ \phi_1(t_m) & \cdots & \phi_n(t_m) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{bmatrix} + \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_m \end{bmatrix}.$$

   This is of the form $Ax = b + r$ and we have to choose $x \in \mathbb{R}^n$ for which the 2-norm of the residual vector $\|r\|_2$, is minimized. We write this as LSP $Ax = b$.

   Your task is to find the polynomial of degree 17 that best fits the function $f(t) = \sin(\frac{\pi}{5}t) + \frac{t}{5}$ for $t_1 = -5, t_2 = -4.5, \ldots, t_{23} = 6$. Set up the LSP $Ax = b$ and determine the polynomial p whose coefficients are determined by x in three different ways:

   (a) By using the Matlab command
   `>> A \ b`
   This uses QR factorization to solve the LSP $Ax = b$. Call this polynomial p_1.

   (b) By setting up the normal equation $A^TAx = A^Tb$ and solving them for x. You will need to use the Cholesky method for this. Call this polynomial p_2.

   (c) By solving the argumented system $\begin{bmatrix} I_n & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} -r \\ x \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix}$. Call this polynomial p_3.

   Set the formatting to `format long e` and compute the condition number of the coefficient matrix associated with each of the systems that you are solving. Which one is the most ill conditioned?

   Compute the $\|r\|_2$ which gives the value of $\sqrt{\Sigma_{i=1}^{23}|p_j(t_i) - f(t_i)|^2}\, j = 1, 2, 3$ for each of these methods (again in `format long e.`). This is a measure of the goodness of the fit in each case. Which of the methods provide the best fit?

2. This is a demonstration of image compression techniques using SVD. The following commands will first load a built-in 320×200 matrix X that represents the pixel image of a clown, computes its SVD $X = U\Sigma V^T$ and then displays the image when X is approximated by its best rank k approximation $X_k = \sum_{i=1}^{k} \sigma_i u_i v_i^T$ for a chosen value of k.

```
load clown.mat; [U, S, V] = svd(X); colormap('gray');
image(U(:, 1:k)*S(1:k, 1:k)*V(:,1:k)')
```

The storage required for A_k is $k(m+n) = 520k$ words whereas the storage required for the full image is $n \times m = 6400$ words in this case. Therefore, $\frac{520k}{6400}$ gives the compression ratio for the compressed image. Also the error in the representation is $\frac{\sigma_{k+1}}{\sigma_1}$. Run the above commands for various choices of k and make a table that records the relative errors and compression ratios for each choice.

3. The aim of the experiment in Exercise 4.2.21 of *Fundamentals of Matrix Computations* is to show that the Rank Revealing QR Decomposition is less efficient than the SVD method when detecting numerical rank deficiency. You will find it on page 273 of the second edition and pages 272 of third edition. Write a small description of your experiment.

Make a report of your experiments.