CENG 371 - Scientific Computing
Fall 2023
Homework 3

Adıgüzel, Gürhan İlhan
e2448025@metu.edu.tr

December 29, 2023

Answer 1

a) For given $Ax = b$ where $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^n$, and $m > n$ and the set $B = \{b' \mid Ax = b + b'\}$.

To determine the dimensionality of B, we can consider the equation $Ax = b + b'$. The solution x is fixed as it satisfies $Ax = b$. Now, we want to find all possible vectors b' that satisfy this equation.

The equation $Ax = b + b'$ implies that b' is the difference between two vectors: Ax and b. Since Ax is fixed (and it's unique solution to $Ax = b$), b' can vary freely. Therefore, the set B is essentially the set of all possible vectors that can be added to b.

Now, the dimensionality of B is the same as the dimensionality of the vector space of b'. Since b' can vary freely, it can take any value in $\mathbb{R}^n$. Therefore, the dimensionality of B is n. .

b) In "$hw3\_q1.m$" file I computed A and b as:

$$A = \begin{bmatrix} 1.4142 & 2.2361 & 3.1623 \\ 2.2361 & 2.8284 & 3.6056 \\ 3.1623 & 3.6056 & 4.2426 \\ 4.1231 & 4.4721 & 5.0000 \\ 5.0990 & 5.3852 & 5.8310 \end{bmatrix} \qquad b = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

The basis of B is the column space of A as we mentioned in part a). Basis for column space of A is all the columns of A as the $rank(A) = n$. So, the basis of B are:

$$\begin{bmatrix} 1.4142 \\ 2.2361 \\ 3.1623 \\ 4.1231 \\ 5.0990 \end{bmatrix} \quad \begin{bmatrix} 2.2361 \\ 2.8284 \\ 3.6056 \\ 4.4721 \\ 5.3852 \end{bmatrix} \quad \begin{bmatrix} 3.1623 \\ 3.6056 \\ 4.2426 \\ 5.0000 \\ 5.8310 \end{bmatrix}$$

Answer 2

a) I calculate the low-rank approximation for various values of k with the help of $[U, S, V] = svd(I)$ function. My r value is 379, so I compute the approximations when $k = 189.5$ and $k = 369$. We can observe that as we increase the k value in our implementation, we will take into account fewer singular values and retain less information from the entire image, leading to a less accurate approximation and poor quality image.

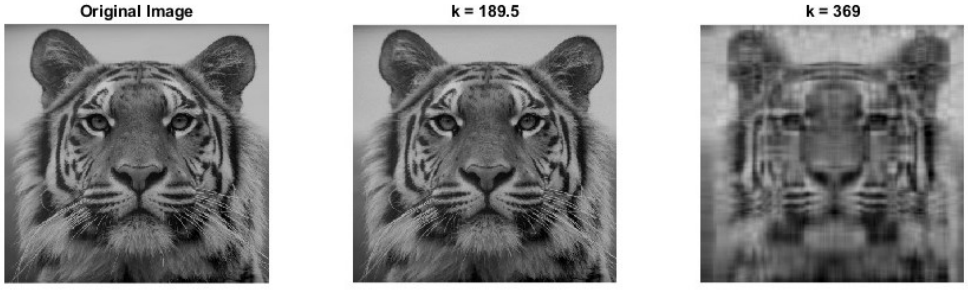

Figure 1: Images with different k

b) I obtain the singular values by extracting them from the diagonal elements of the S matrix and arrange them in ascending order. Subsequently, I calculate $S(k)$ and errors using $\|I - I_k\|_F$, where I_k represents the approximated images that have already been computed. As $S(k)$ increases, the error also rises, indicating a reduced capacity to capture information and yielding a less precise representation of the original matrix. Information loss during the approximation process is the primary contributor to the observed differences and errors between the original and approximated images. Consequently, the graphical representations of $S(k)$ and errors exhibit analogous patterns, reflecting the impact of diminishing information content on the accuracy of the approximation.

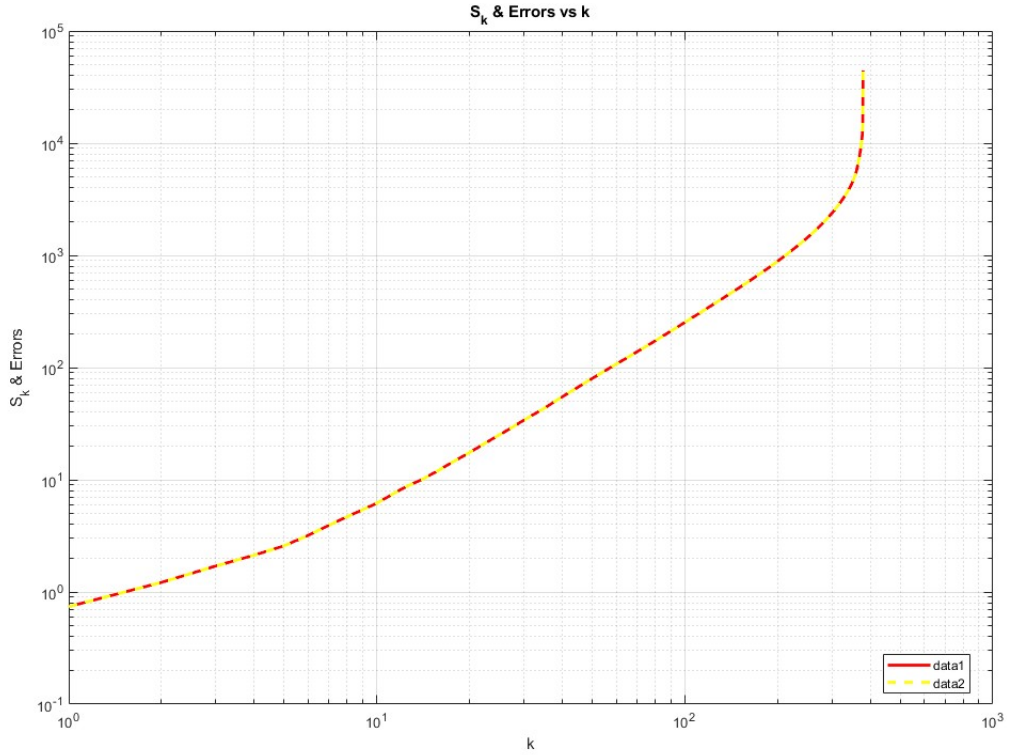

Figure 2: S_k & Errors vs k

c) The low-rank approximation scheme, as observed in parts a) and b), involves reducing the dimensionality of a matrix while retaining the most significant information. This technique is particularly useful in various applications where a trade-off between computational efficiency and precision in representing data can be acceptable.

(a) **Image Compression:**

- **Use Case:** When storing or transmitting images, especially in scenarios with limited bandwidth or storage capacity, low-rank approximation can be employed to compress images. This allows for reduced file sizes while maintaining a visually acceptable level of quality.

(b) **Signal Processing:**

- **Use Case:** In signal processing applications, such as audio or video signal analysis, low-rank approximation can be used to reduce the complexity of the data representation. This is particularly valuable in real-time processing or applications with limited computational resources.

(c) **Reducing Noise in Data:**

- **Use Case:** In scenarios where data may be corrupted by noise, low-rank approximation can help in denoising. By capturing the dominant patterns in the data, the technique can filter out noise and provide a cleaner representation.

In these applications, low-rank approximation trades off some information loss for gains in efficiency, making it suitable for various practical scenarios.

Answer 3

a) To construct matrix A_n, define its size as $n \times (n+1)$, create a vector t_n representing $\frac{i}{n+1}$, and fill A_n with powers of t_i.

$$A_n = \begin{bmatrix} t_1^0 & t_1^1 & \dots & t_1^n \\ t_2^0 & t_2^1 & \dots & t_2^n \\ t_3^0 & t_3^1 & \dots & t_3^n \\ \vdots & \vdots & \vdots & \vdots \\ t_n^0 & t_n^1 & \dots & t_n^n \end{bmatrix} \qquad A_n = \begin{bmatrix} \left(\frac{1}{n+1}\right)^0 & \left(\frac{1}{n+1}\right)^1 & \dots & \left(\frac{1}{n+1}\right)^n \\ \left(\frac{2}{n+1}\right)^0 & \left(\frac{2}{n+1}\right)^1 & \dots & \left(\frac{2}{n+1}\right)^n \\ \left(\frac{3}{n+1}\right)^0 & \left(\frac{3}{n+1}\right)^1 & \dots & \left(\frac{3}{n+1}\right)^n \\ \vdots & \vdots & \vdots & \vdots \\ \left(\frac{n}{n+1}\right)^0 & \left(\frac{n}{n+1}\right)^1 & \dots & \left(\frac{n}{n+1}\right)^n \end{bmatrix}$$

To create A_{noised}, generate random noise ϵ, and for each element in A_n, replace t_i with $t_i + \epsilon$.

$$A_{\text{noised}} = \begin{bmatrix} (t_1+\epsilon)^0 & (t_1+\epsilon)^1 & \dots & (t_1+\epsilon)^n \\ (t_2+\epsilon)^0 & (t_2+\epsilon)^1 & \dots & (t_2+\epsilon)^n \\ (t_3+\epsilon)^0 & (t_3+\epsilon)^1 & \dots & (t_3+\epsilon)^n \\ \vdots & \vdots & \vdots & \vdots \\ (t_n+\epsilon)^0 & (t_n+\epsilon)^1 & \dots & (t_n+\epsilon)^n \end{bmatrix} \qquad A_{\text{noised}} = \begin{bmatrix} \left(\frac{1}{n+1}+\epsilon\right)^0 & \left(\frac{1}{n+1}+\epsilon\right)^1 & \dots & \left(\frac{1}{n+1}+\epsilon\right)^n \\ \left(\frac{2}{n+1}+\epsilon\right)^0 & \left(\frac{2}{n+1}+\epsilon\right)^1 & \dots & \left(\frac{2}{n+1}+\epsilon\right)^n \\ \left(\frac{3}{n+1}+\epsilon\right)^0 & \left(\frac{3}{n+1}+\epsilon\right)^1 & \dots & \left(\frac{3}{n+1}+\epsilon\right)^n \\ \vdots & \vdots & \vdots & \vdots \\ \left(\frac{n}{n+1}+\epsilon\right)^0 & \left(\frac{n}{n+1}+\epsilon\right)^1 & \dots & \left(\frac{n}{n+1}+\epsilon\right)^n \end{bmatrix}$$

Where $A_n \cdot c_k = b_n$, b_n represents the $N_{obs}(t)$. So,b_n can be calculated as $A_{noised} \cdot c_k$. Where c_k is

$$c_k = \begin{bmatrix} 0.3 \\ 2 \\ -1.2 \\ 0.5 \end{bmatrix}$$

O_n is constructed as $(t, N_{obs}(t))$, so $O_n = (t, b_n)$
As an example for n=5, Calculated values are:

$$A_n = \begin{bmatrix} 1 & 0.1667 & 0.0278 & 0.0046 \\ 1 & 0.3333 & 0.1111 & 0.0370 \\ 1 & 0.5000 & 0.2500 & 0.1250 \\ 1 & 0.6667 & 0.4444 & 0.2963 \\ 1 & 0.8333 & 0.6944 & 0.5787 \end{bmatrix} \qquad A_{\text{noised}} = \begin{bmatrix} 1.0000 & 0.1769 & 0.0380 & 0.0149 \\ 1.0000 & 0.3420 & 0.1198 & 0.0457 \\ 1.0000 & 0.4962 & 0.2462 & 0.1212 \\ 1.0000 & 0.6710 & 0.4487 & 0.3006 \\ 1.0000 & 0.8303 & 0.6915 & 0.5757 \end{bmatrix}$$

$$b_n = \begin{bmatrix} 0.6157 \\ 0.8632 \\ 1.0575 \\ 1.2537 \\ 1.4188 \end{bmatrix} \qquad O_n = \begin{bmatrix} 0.1667 & 0.6157 \\ 0.3333 & 0.8632 \\ 0.5000 & 1.0575 \\ 0.6667 & 1.2537 \\ 0.8333 & 1.4188 \end{bmatrix}$$

b) I have used the "Linear Least Squares Method with QR factorization" in "Lecture Notes Part9" to approximate c_k. We must solve the overdetermined system $A_n \cdot c_k = b_n$. In our case, the system is overdetermined with n taking values of 5, 10, and 100, and k having 4 components.

To find an approximation for c_k that minimizes the error

$$Q = min\|b_n - A_n \cdot c_k\|_2^2$$

$$r = Rank(A_n)$$

$$A = Q * R$$

$$Q1 = Q(:, 1:r)$$

we employ the Linear Least Squares method. Utilizing the QR factorization, we can express c_k as:

$$c_k = R(1:r, 1:r)^{-1}(Q1^T \cdot b_n)$$

c) The error in c_k can be determined by the expression:

$$\|c_k - \hat{c}_k\|_2^2$$

After 100 runs, the average error in c_k is computed as follows:

For $n = 5$, the average error is 1.0155
For $n = 10$, the average error is 0.7623
For $n = 100$, the average error is 0.1282

We observe that as n increases, the average error decreases. It can be concluded that increasing the number of observations leads to more accurate approximations.