Master Degree in Artificial Intelligence
Statistical and Mathematical Methods for Artificial Intelligence

2021-2022

Homework 1: Linear Algebra and Floating Point Arithmetic

Direct Methods for the solution of Linear Systems.

1. Given a matrix $A \in \mathbb{R}^{n\times n}$ and the vector $x_{true} = (1, 1, \dots, 1)^T \in \mathbb{R}^n$, write a script that:
   - Computes the right-hand side of the linear system $b = Ax_{true}$.
   - Computes the condition number in 2-norm of the matrix A. It is ill-conditioned? What if we use the ∞-norm instead of the 2-norm?
   - Solves the linear system $Ax = b$ with the function `np.linalg.solve()`.
   - Computes the relative error between the solution computed before and the true solution x_{true}. Remember that the relative error between x_{true} and x in $\mathbb{R}^n$ can be computed as

   $$E(x_{true}, x) = \frac{||x - x_{true}||_2}{||x_{true}||_2}$$

   - Plot a graph (using `matplotlib.pyplot`) with the relative errors as a function of n and (in a new window) the condition number in 2-norm $K_2(A)$ and in ∞-norm, as a function of n.

2. Test the program above with the following choices of $A \in \mathbb{R}^{n\times n}$:
   - A random matrix (created with the function `np.random.rand()`) with size varying with $n = \{10, 20, 30, \dots, 100\}$.
   - The Vandermonde matrix (`np.vander`) of dimension $n = \{5, 10, 15, 20, 25, 30\}$ with respect to the vector $x = \{1, 2, 3, \dots, n\}$.
   - The Hilbert matrix (`scipy.linalg.hilbert`) of dimension $n = \{4, 5, 6, \dots, 12\}$.

Floating Point Arithmetic.

1. The Machine epsilon ϵ is the distance between 1 and the next floating point number. Compute ϵ, which is defined as the smallest floating point number such that it holds:

   $$fl(1 + \epsilon) > 1$$

   Tips: use a `while` structure.

2. Let's consider the sequence $a_n = (1 + \frac{1}{n})^n$. It is well known that:

   $$\lim_{n\to\infty} a_n = e$$

   where e is the Euler costant. Choose different values for n, compute a_n and compare it to the real value of the Euler costant. What happens if you choose a large value of n? Guess the reason.

3. Let's consider the matrices:

$$A = \begin{pmatrix} 4 & 2 \\ 1 & 3 \end{pmatrix}, B = \begin{pmatrix} 4 & 2 \\ 2 & 1 \end{pmatrix}$$

Compute the rank of A and B and their eigenvalues. Are A and B full-rank matrices? Can you infer some relationship between the values of the eigenvalues and the full-rank condition? Please, corroborate your deduction with other examples.

Tips: Please, have a look at `np.linalg`.