Prof. Dr. Jens Honore Walther
Dr. Julija Zavadlav
ETH Zentrum, CLT
CH-8092 Zürich

Set 2

Issued: 01.03.2019; Due: 10.03.2019

In this exercise, you will learn about problems with the normal equation to solve the least square problem and the solution in terms of a Singular Value Decomposition (SVD). In the second part you will learn about the Newton's method to solve nonlinear systems of equations.

Question 1: Normal Equation versus SVD

The matrix A for M'th order polynomial regression over N measurements is given as a Vandermonde matrix

$$A = \begin{bmatrix} 1 & t_0 & \dots & t_0^{M-1} \\ 1 & t_1 & \ddots & t_1^{M-1} \\ \vdots & \ddots & \ddots & \vdots \\ 1 & t_{N-1} & \dots & t_{N-1}^{M-1} \end{bmatrix} \tag{1}$$

We assume that $M = 15$ and $N = 50$. The measurements where taken at times $t_i = i \cdot \frac{1}{49} \in [0, 1]$ with $i \in \{0, \dots, 49\}$. Further we assume that we know that the exact coefficients are given by $\vec{x} = [1, 2, \dots, 15]^\top$. Using this we can compute the perfect measurement values as $\vec{b} = A\vec{x}$. On the so defined artificial data-set we will compare the applicability of the normal equation and SVD by computing the respective values $\bar{x}$. For the comparison we regard the normalized residual

$$\rho(\bar{x}) = \frac{\|A\bar{x} - \vec{b}\|_2}{\|A\|_2 \|b\|_2} \tag{2}$$

and the absolute error

$$\text{error} = \|\vec{x} - \bar{x}\|_2 \tag{3}$$

a) Compute the condition number of A and explain what your observation means.

b) Compute the coefficients $\bar{x}$ using the normal equation

$$\bar{x} = (A^\top A)^{-1} A^\top \vec{b} \tag{4}$$

and report the obtained values for the residual (equation 2) and the error (equation 3). Discuss whether the so obtained result are expected.

c) From linear algebra we know that we can decompose every matrix $A \in \mathbb{R}^{N \times M}$ into two orthogonal matrices $U \in \mathbb{R}^{N \times N}$, $V \in \mathbb{R}^{M \times M}$ and a diagonal matrix $\Sigma \in \mathbb{R}^{N \times M}$

$$A = U \Sigma V^\top \tag{5}$$

Using this decomposition and the Moore-Penrose pseudo-inverse we can write the solution to our linear least squares problem as

$$\bar{x} = A^+ b = V \Sigma^+ U^\top \vec{b} \tag{6}$$

Compute the coefficients according to SVD and report the obtained values of the residual (equation 2) and the error (equation 3). Compare the obtained results to the ones from the previous exercise and discuss the applicability of the two methods. Which one would you chose if you have to solve a linear least squares problem?

Question 2: Newton's Method

a) [Pen and paper] Provide Newton's Method in pseudo-code. Therefore write down how you would in principle implement the ideas presented in the lecture. In order to get the idea regard the following pseudo-code computing the Fibonacci Series $F_n = F_{n-1} + F_{n-2}$ with $F_0 = 0$ and $F_1 = 1$ up the N'th term, while aborting if $F_n > n_{max}$.

Algorithm 1 Fibonacci Series

Input:

N, {number of elements to compute}
$n_{\max}$, {threshold to stop computation}

Output:

$\vec{F}$, {vector containing Fibonacci numbers}

Steps:

$F[0] \leftarrow 0$
$F[1] \leftarrow 1$
$n \leftarrow 2$
while $n < N + 2$ **do**
  $F[n] \leftarrow F[n-1] + F[n-2]$
  if $F[n] > n_{\max}$ **or** $n > N$ **then**
    break
  end if
  $n \leftarrow n + 1$
end while

b) [Pen and paper] Estimate $\sqrt{34}$ up to the accuracy of 2 decimals using Newton's method. As initial guess use $x^{(0)} = \sqrt{36} = 6$. To estimate the error of your approximation, assume that the exact solution is *not* known. How many iterations are needed?

c) Implement the Newton's Method on the computer and repeat the above computation. Design your program such that you can change the objective function easily. In order to do so you should implement the objective function and it's derivative and call them in your implementation of the Newton algorithm. In order to check whether everything is right, compare the output of your code to the result computed in the previous exercise. If this validation passes, run the code until the accuracy reaches 10^{-15} and plot the proceeding of the algorithm.