Data Approximation

Anouk Allenspach, Rodolphe Farrando

December 15, 2017

1 Goal

The goal of this project is to find a polynomial approximation of some data. Several approximation can be made. The goal of the user can be either to find a trend for the or to interpolate the data. Several possibilities are available to interpolate, the first one is a simple polynomial interpolation, where the degree is equal to the number of points minus 1. Then, a piecewise interpolation can be made, this interpolation can be spline or not.

2 Requirements

There is three requirements:

- A gcc compiler (We used Apple machine to run the program, it uses Clang instead of gcc).
- The CMake software (the version 3.10 was used to run the tests) to link the files are necessary to compile the program
- An external library is used to solve linear systems, that is the Eigen library.

3 Method

To approximate the data, the main method that have been used is the least squares method. The method aim to minimize the error between the true value and the estimated one, that is:

$$min \sum_i (y_i - p(x_i))^2 \tag{1}$$

$p(x_i) = a_0 + a_1 \cdot x_i + \ldots + a_m \cdot x_i^m$ being the interpolant polynomial at x_i of degree m. By deriving this equation with respect with all the coefficient of the polynomial, the following linear system is obtained:

$$\begin{bmatrix} \sum_i x_i^0 & \sum_i x_i^1 & \cdots & \sum_i x_i^m \\ \sum_i x_i^1 & \sum_i x_i^2 & \cdots & \sum_i x_i^{m+1} \\ \vdots & \ddots & \ddots & \vdots \\ \sum_i x_i^m & \cdots & \sum_i x_i^{2m-1} & \sum_i x_i^{2m} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_m \end{bmatrix} = \begin{bmatrix} \sum_i y_i \\ \sum_i y_i x_i \\ \vdots \\ \sum_i y_i x_i^m \end{bmatrix} \tag{2}$$

The linear system $Xa = b$ can be solved by computing $a = X^{-1}b$.
For the spline interpolation, `https://www.math.uh.edu/ jingqiu/math4364/spline.pdf` the following paper has been used.

4 Usage of the program

4.1 Input file

The program developed allows the user to give a .csv file with some input inside. Depending on what is in the file, the program will compute what the user has asked. The .csv should have the following form:

data.csv	Approximation Degree	Type

The *data.csv* file contains the points coordinate. The type is the approximation wanted by the user; it can be either *Fitting*, *Interpolation*, *Piecewise* or *PiecewiseContinuous* - the Piecewise continous is a spline interpolation. Finally, the approximation degree is the degree of the polynomial which approximate the data.

4.2 Interactive main

The interactive main allows the user to interact with a small program that gives the results wanted. Just compile the program and enter the name file in the console and the program will give the coefficient, the function(s) and the error.

4.3 Create a new main

The user just have to construct an object Points then he can construct the interpolation object that he needs and finally, he can call all the method that are in the approximation class.

5 Features description

Points (class): This class creates an object that instantiate the class Approximation. There is two ways to create an object Points, either with a .csv file ore directly with vector and number.
Approximation (class): The approximation class has three derived class: Fitting, Interpolation and PieceWiseInterpolation. It will give the coefficients of the polynomial, the display of the function(s) and the error associated.
Function Approximation (class):

6 Tests

A series of main are testing all the possible interpolation. All this tests returns the same output: the interpolation function(s) and the error compared to the points given. For the 4 first test we generate a .csv file using the FunctionApprox class. We define a function, an interval, the number of points and the degree of interpolation.
Test 1 $\rightarrow$ Fitting Method: $f(x) = sin(x/2)$, number of points = 20 , degree = 5, interval = [0 2π]
Test 2 $\rightarrow$ Interpolation Method: $f(x) = log(x)$, number of points = 6, degree = 5, interval [1 6]
Test 3 $\rightarrow$ Piece-wise Method: $f(x) = cos(x) * sin(x/2)$, number of points = 16, degree = 5, interval = [0 1]
Test 4 $\rightarrow$ Piece-wise ContinuousMethod: $f(x) = e^{(x/2)} - x^3$number of points = 20, degree = 3, interval = [3 10]
Test 5 $\rightarrow$ Function approximation class: Small test which shows how this class works

7 Remarks

When the degree of interpolation is high ($\geq$ 7 approximatively), the polynomial is badly condition and the points are not exactly interpolated. Indeed, the condition number of the matrix X in the linear system $Xa = b$ is big. It results difficulties to inverse the matrix and the solution is biased. The user needs to interpolate few points to have a good result. The same happen for fitting the data, the degree shouldn't be too high otherwise same problems could happen.