Computing for Scientists - Lab 6

CS 1340 — Dr. Mihail
Department of Computer Science
Valdosta State University

October 24, 2018

1 Introduction

In this lab, you will learn how to use the MATLAB optimization toolbox to fit functions to data, a process called curve fitting. In many real world situations you are given some data (usually from measurements) and are asked to find a mathematical "model" (an underlying generating function) that best fits the observed data points. The data is usually contaminated by noise. Once the measurements are taken, the scientist analyzes the data and gets some insight into what mathematical model could best fit it. In Figure 1, the generating function is a quadratic. On the left, one can see the best fit (solid green line) and on the right one can see a poor fit. The curve fitting process involves minimizing a cost function. The input to the function is the model's parameters and the data points. The value of this function is low when the model fits the data well, and high otherwise. In this lab we will use the sum of squares cost function. Intuitively, this cost function sums up the squared differences between the measurements and the model function.

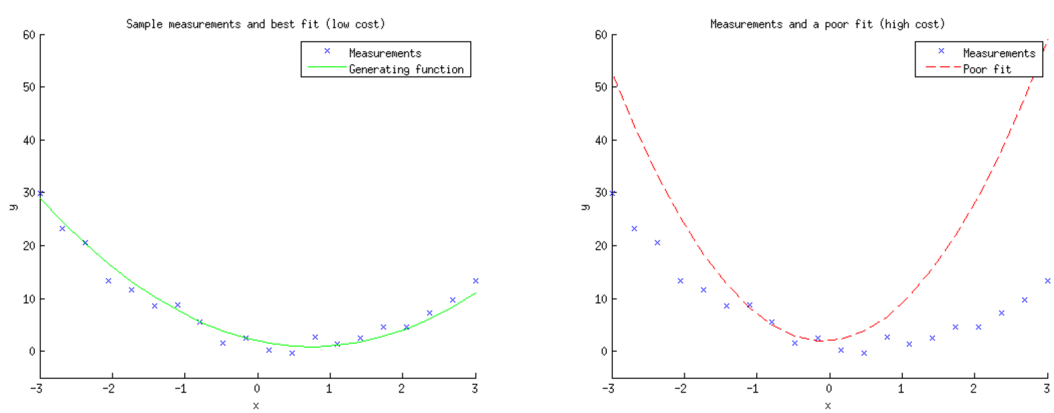

Figure 1: Sample measurements and fit

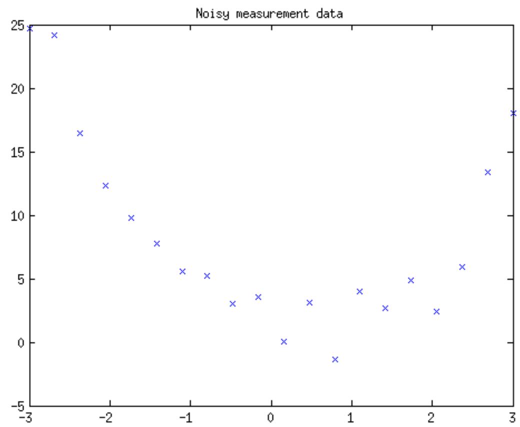

Figure 2: Noisy input data

2 Quadratics revisited

The family of quadratic functions has the following form:

$$f(x,p) = p_1 x^2 + p_2 x + p_3 \tag{1}$$

where p is a vector of parameters, so p_1, p_2 and p_3 are scalars.

3 Cost

Given some data, X and Y (vectors of measured data points) our goal is to find the **best** parameter vector p, such that the cost is at a minimum. The sum of squares cost is a function of **parameters** (the parameters are what we will optimize), and has the following form:

$$C(p) = \sum_i (f(X_i, p) - Y_i)^2 \tag{2}$$

4 Part 1 (in-class) 30 points

4.1 Step 1

Download the data file from the course web page, called **measurements.mat**. You can load it in MATLAB using the *load* command, and the filename as an argument:`load('measurements.mat');`. This will load the two vectors X and Y in the workspace. You can verify they exist in the workspace using the `whos` command. Also verify that their size is 1 by 20.

4.2 Step 2

Create Figure 1. Plot the data points using blue x's. It should look like Figure 2.

4.3 Step 3

Create an anonymous function called `model`, with two parameters x and p, exactly as Equation 1. Keep in mind that x is a vector, so element-wise squaring is necessary. Because we are modeling a quadratic, p will be a three element vector. In MATLAB p_1 is `p(1)`, p_2 is `p(2)` and p_3 is `p(3)`. Refer to this link for more information and syntax on MATLAB anonymous functions. The optimization slides on the course web page can also be used.

4.4 Step 4

Create an anonymous function called `cost`, with one parameter p, defined exactly as Equation 2. You have to use MATLAB's `sum` function for summation and do element-wise squaring.

4.5 Step 5

Create a row vector p with values $[1, 1, 1]$. Test your cost function in the command window as follows:

```
>> cost(p)
```

It should return 1557.292. p is your starting point, or an initial guess of what the generating function's parameters might be. Plot this function over the measurement data.

4.6 Step 6

Use MATLAB's unconstrained minimization function `fminunc` to find the optimal p as follows:

```
bestP = fminunc(C, p);
```

As you can see above, `fminunc` takes the cost function as the first argument and a starting point for the optimization. After this line is executed, $bestP$ will contain the quadratic function parameters that **best fit** the data. Verify that the cost of $bestP$ is significantly lower than the initial guess.

4.7 Step 7

Plot your function of best fit over the data points in the same figure. You can create new Y values by calling `model` using the X vector provided and $bestP$. It should look like Figure 3.

4.8 Step 8

Submit the script using the electronic submission link on the course web page.

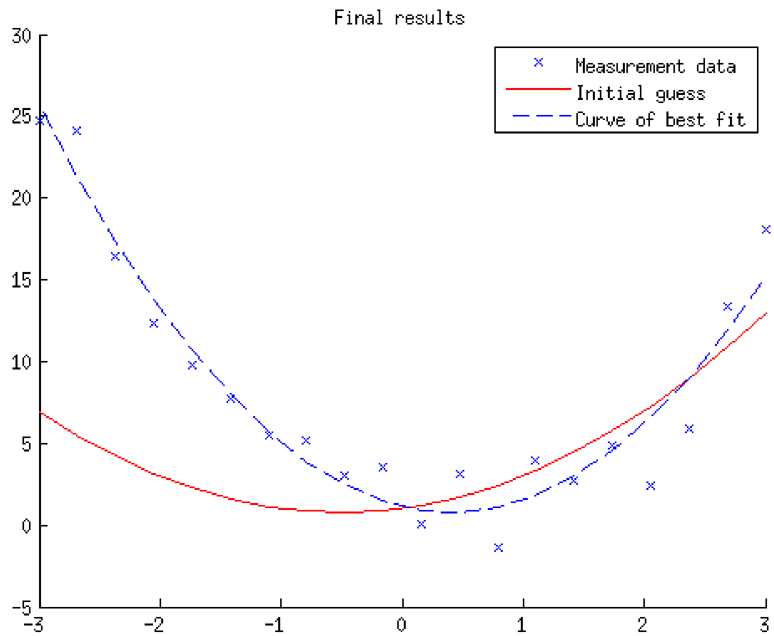

Figure 3: Curve of best fit.

5 Part 2 (homework) 70 points

5.1 Problem Description

A professor computed the probability density function of the grades assigned in her Rocket Science course. She recorded the data as pairs of X and Y values, (available to download from the course webpage as homework.mat). She plotted the data in MATLAB and obtained the results seen in Figure 4.

The professor seems to believe that the generating function is a Gaussian (or a Normal distribution), which has the following form:

$$Gaussian(x,p) = \frac{1}{p_2\sqrt{2\pi}} exp(-\frac{(x-p_1)^2}{2p_2^2}) \tag{3}$$

Use the curve fitting technique from Part 1 to solve for p_1 (μ, mu, the mean) and p_2 (σ, sigma, the standard deviation). The cost function should be the same. Your script needs to plot the data points and the Gaussian of best fit on top of the data, exactly as seen in Figure 5. It should also output the mean (p_1) and the standard deviation (p_2). `exp` is the MATLAB exponential function $exp(x) = e^x$, where e is the base of the natural logarithm.

Due dates

MATLAB script files due before midnight on Sunday, October 28th.

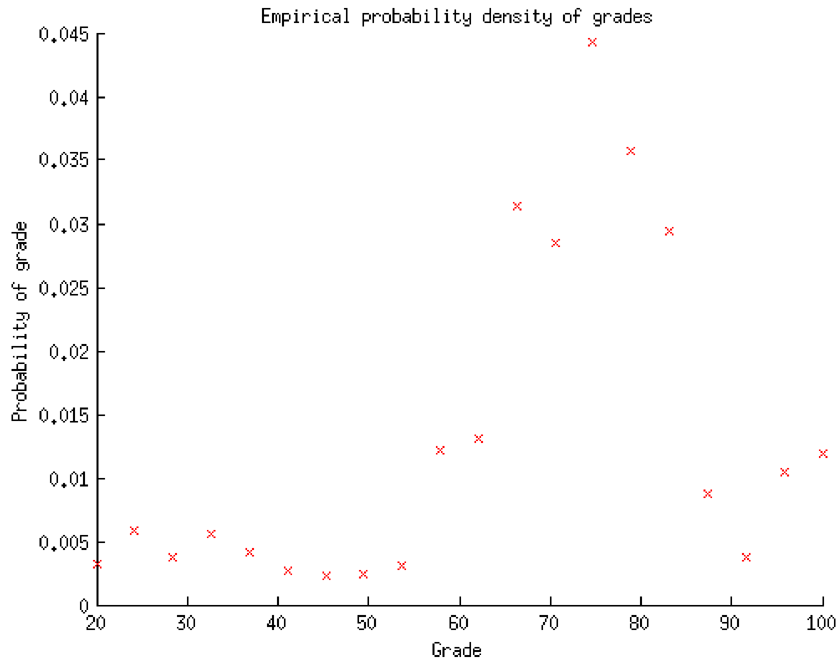

Figure 4: Visualizing the data.

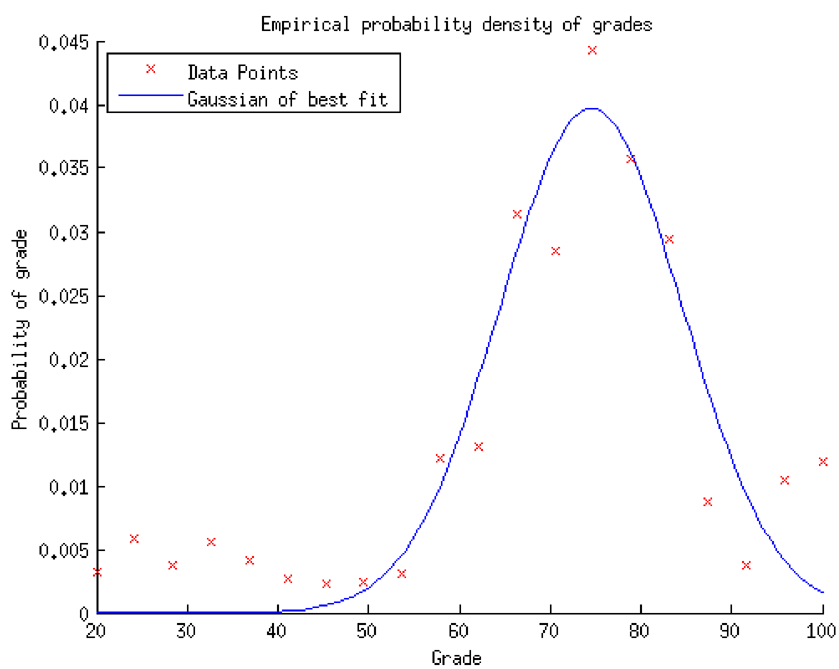

Figure 5: Final results for Part 2.