Math 156: Lab assignment #1

Due: Tuesday October 10th

The goal of this assignment is to learn the basic tools of linear regression and to get used to programming in MATLAB. The most useful MATLAB function is the help function. You can get information on the syntax and use of any function by simply typing help *your function*, where *your function* is the name of the function for which you want information. In addition the Mathworks website has extensive documentation on MATLAB functions.

For this first assignment you do not need to turn in code, but you will be asked to print and turn in certain figures.

1 Introduction

In this lab we will try to reconstruct a function $f : \mathbb{R} \to \mathbb{R}$ based on a small set of data points $\mathcal{D} = \{(x_1, t_1), \ldots, (x_N, t_N)\}$, here N represents the number of data points and $f(x_n) = t_n$ for all i.

The goal is to be able to predict the value of f at new points x outside of the data set. This is known as a *regression* problem. Here we will approach the problem using a polynomial model. Thus, we will try to approximate f using a function of the form:

$$y(x, \boldsymbol{w}) = w_0 + w_1 x + \ldots + w_M x^M = \sum_{j=0}^{M} w_j x^j \tag{1}$$

where M is the degree of the polynomial and $\boldsymbol{w} = (w_0, w_1, \ldots, w_M) \in \mathbb{R}^{M+1}$ are the polynomial coefficients.

2 Theory

In order to find the best coefficients for a given degree of our polynomial, we are going to solve an optimization problem. The first thing to do is to define an appropriate cost function. The cost function J should reflect how good we think a solution to the problem is.

The only information we have is the data points, and we want our model to fit those data points well. So it seems natural that our cost function will depend on the distance between each value of our data points t_n and the prediction by our model $y(x_n, w)$:

$$J(\boldsymbol{w}) = \frac{1}{2} \sum_{n=1}^{N} (y(x_n, \boldsymbol{w}) - t_n)^2 = \frac{1}{2} \sum_{n=1}^{N} (\boldsymbol{z}_n^T \boldsymbol{w} - t_n)^2 \tag{2}$$

where we define $\boldsymbol{z}_n = (1, x_n, x_n^2, \ldots, x_n^M) \in \mathbb{R}^{M+1}$. We will choose our parameters by finding the minimum of the cost function.

Task 1: Compute the gradient $\nabla J(\boldsymbol{w})$ and show that the equation $\nabla J(\boldsymbol{w}) = \boldsymbol{0}$ may be written in the form $A\boldsymbol{w} = \boldsymbol{b}$ where A is a matrix and $\boldsymbol{b}$ is a vector.

Task 2: Does the equation $\nabla J(\boldsymbol{w}) = \boldsymbol{0}$ always have a solution? If there is a solution is it unique? Explain how these two questions are related to the number of data points N and the degree M of the polynomial model.

3 Experiments

Assume that the function f is given by

$$f(x) = \cos(2x).$$

Task 4: Generate the data set $\mathcal{D}_1$ by evaluating f at $N = 12$ equally spaced points along $[-\pi, \pi]$. Plot the data points and print out the figure.

Task 5: For M between 1 and $N - 1$, write a function which takes as an input the data set $\mathcal{D}_1$ and returns the parameter $\boldsymbol{w}^*$ which solves $\nabla J(\boldsymbol{w}^*) = 0$. You don't need to turn anything in for this task.

Task 6: Using your function from task 5, plot the curve $y(x, \boldsymbol{w}^*)$ on the domain $[-3, 3]$ for each value of M between 1 and $N - 1$. Compare your curves to the actual function $f(x)$ and the data set $\mathcal{D}_1$. What value of M appears to give the best approximation to f?

Task 7: In practice, data measurements will always include some noise. Generate a new data set $\mathcal{D}_2$ by evaluating f at $N = 12$ equally spaced points along $[-\pi, \pi]$ and add random noise between -0.1 and 0.1. In other words your data points (x_n, t_n) should satisfy

$$t_n = f(x_n) + u_n$$

where u_n is a uniform random variable sampled from $(-0.1, 0.1)$. Plot the data points and print out the figure.

Task 8: Using your function from task 5, and the new dataset $\mathcal{D}_2$ plot the curve $y(x, \boldsymbol{w}^*)$ on the domain $[-3, 3]$ for each value of M between 1 and $N - 1$. Compare your curves to the actual function $f(x)$ and the data set $\mathcal{D}_2$. What value of M gives the best approximation to f? What happens when M is close to N? Print out and turn in a plot of $y(x, \boldsymbol{w}^*)$ for $M = 11$.