ECE 9243: Optimal and Learning Control for Robotics
Homework 1

Mingyu Zhao (N19569671)

February 18, 2019

Exercise 1: Convex optimization with linear equalities

1.1 Write the Lagrangian of the optimization problem as well as the KKT conditions for optimality.

$$\mathcal{L}(\boldsymbol{x}, \boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{x}^\intercal \boldsymbol{Q}\boldsymbol{x} + \boldsymbol{\lambda}^\intercal(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b})$$

1.2 Solve the KKT system.

Take the partial derivative with respect to $\boldsymbol{x}$ and $\boldsymbol{\lambda}$ and set them to 0:

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{x}} = 0 \ \Rightarrow\ \frac{1}{2}(\boldsymbol{Q} + \boldsymbol{Q}^\intercal)\boldsymbol{x} + \boldsymbol{A}^\intercal \boldsymbol{\lambda} = 0 \ \Rightarrow\ \boldsymbol{x} = -2(\boldsymbol{Q} + \boldsymbol{Q}^\intercal)^{-1}\boldsymbol{A}^\intercal \boldsymbol{\lambda}$$

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{\lambda}} = 0 \ \Rightarrow\ \boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} = 0$$

Plug $\boldsymbol{x}$ into $\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b} = 0$:

$$-2\boldsymbol{A}(\boldsymbol{Q} + \boldsymbol{Q}^\intercal)^{-1}\boldsymbol{A}^\intercal \boldsymbol{\lambda} - \boldsymbol{b} = 0 \ \Rightarrow\ \boldsymbol{\lambda} = -\frac{1}{2}(\boldsymbol{A}(\boldsymbol{Q} + \boldsymbol{Q}^\intercal)^{-1}\boldsymbol{A}^\intercal)^{-1}\boldsymbol{b}$$

Plug $\boldsymbol{\lambda}$ back into the expression of $\boldsymbol{x}$:

$$\boldsymbol{x} = (\boldsymbol{Q} + \boldsymbol{Q}^\intercal)^{-1}\boldsymbol{A}^\intercal(\boldsymbol{A}(\boldsymbol{Q} + \boldsymbol{Q}^\intercal)^{-1}\boldsymbol{A}^\intercal)^{-1}\boldsymbol{b}$$

1.3 Use the above results to compute the minimum a function.

Here, we have:

$$\boldsymbol{Q} = \begin{bmatrix} 10 & 1 & 0 \\ 1 & 100 & 2 \\ 0 & 2 & 1 \end{bmatrix}$$

And the constraint:

$$\sum_{i=1}^{3} x_i = 1$$

That is:

$$\boldsymbol{A} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix} \quad b = 1$$

Plug the value of $\boldsymbol{Q}$, $\boldsymbol{A}$ and b into the expression of $\boldsymbol{\lambda}$:

$$\boldsymbol{\lambda} = -0.89878 \quad \boldsymbol{x} = \begin{bmatrix} 0.090909 & -0.010309 & 0.919400 \end{bmatrix}$$

Verify the constraint: $\sum_{i=1}^{3} x_i = 0.090909 + -0.010309 + 0.919400 = 1$

Exercise 2: Newton's Method

2.1 & 2.2 Implement the Newton's Method.

The Newton's method with backtracking line search is implemented in the attached Jupyter Notebook.

2.3 Compare the Convergence Result with the Gradient Descent Algorithm.

As it is shown in the Jupyter Notebook, the Gradient Descent Algorithm takes 573 steps to converge for the 1D function and 75 steps for the 2D function, while the Newton's method only takes 7 steps for the 1D function and 1 step for the 2D function. So just as proved by mathematics methods, the convergence of Newton's method is quadratic near the minimum, which is much faster than the linear convergence of Gradient Descent algorithm.

Exercise 3: Linear Least Squares

3.1 Polynomial Fitting

Polynomial with order 3 can best fit the data. According to the plot of the error and the order, we can find out that when the order is less than 3, both training error and testing error are high, while when the order is larger than 3, both training error and testing error do not change very much. So the polynomial with order 3 is good enough for fitting this dataset without using meaningless auguments.

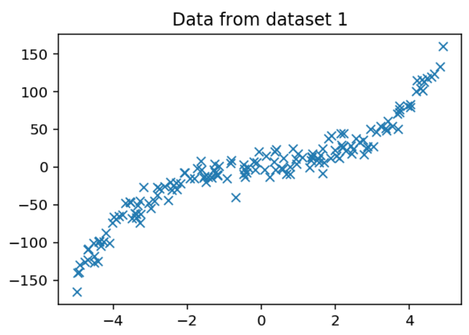
And the polynomial coefficient is:

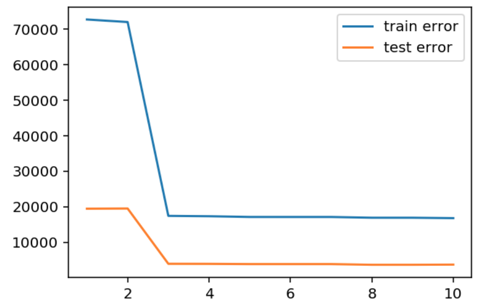
$$a = \begin{bmatrix} 0.72025061 & 4.67805198 & 0.05507879 & 1.0137015 \end{bmatrix}$$

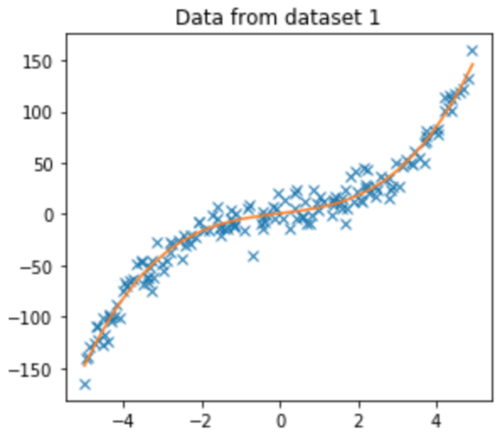
The function is:

$$f(x) = 0.72025061 + 4.67805198x + 0.05507879x^2 + 1.0137015x^3$$

Plot this function and the data from the dataset, and it looks good:







3.2 Fit with Periodic Functions

3.2.1 Write down the optimization problem that allows to find the coefficients.

First, write down the optimization problem, suppose we have dataset which contains X and Y:

$$X = \begin{bmatrix} x_{0,1} & x_{0,2} \\ x_{1,1} & x_{1,2} \\ \vdots & \vdots \\ x_{N-1,1} & x_{N-1,2} \end{bmatrix} \quad Y = \begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{N-1} \end{bmatrix}$$

Then, in order to find a_k and b_k, we need optimize:

$$\min_{a_0,\ldots,a_K\ b_1,\ldots,b_K} \sum_{i=0}^{N-1} \left(a_0 + \sum_{k=1}^{K} \Big(a_k \cos(kT2\pi X_{i,k}) + b_k \sin(kT2\pi X_{i,k}) \Big) - Y_i \right)^2$$

Let

$$W = \begin{bmatrix} 1 & \cos(T2\pi X_{0,1}) & \cdots & \cos(KT2\pi X_{0,K}) & \sin(T2\pi X_{0,1}) \cdots & \sin(KT2\pi X_{0,K}) \\ 1 & \cos(T2\pi X_{1,1}) & \cdots & \cos(KT2\pi X_{1,K}) & \sin(T2\pi X_{1,1}) \cdots & \sin(KT2\pi X_{1,K}) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \cos(T2\pi X_{N-1,1}) & \cdots & \cos(KT2\pi X_{N-1,K}) & \sin(T2\pi X_{N-1,1}) \cdots & \sin(KT2\pi X_{N-1,K}) \end{bmatrix}$$

As well as the parameters:

$$\boldsymbol{a} = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_K \\ b_1 \\ \vdots \\ b_K \end{bmatrix}$$

So that we can simplified the lose function:

$$L(\boldsymbol{a}) = (W\boldsymbol{a} - Y)^\mathsf{T}(W\boldsymbol{a} - Y)$$

The original problem can be rewrite as:

$$\min_{\boldsymbol{a}} (W\boldsymbol{a} - Y)^\mathsf{T}(W\boldsymbol{a} - Y)$$

which is equal to

$$\min_{\boldsymbol{a}} \boldsymbol{a}^\mathsf{T} W^\mathsf{T} W\boldsymbol{a} - 2Y^\mathsf{T} W\boldsymbol{a} + Y^2$$

which is a convex function, so the necessary and sufficient conditions to find the minimum of the function is to find the value of $\boldsymbol{a}$ that makes the gradient 0:

$$\frac{\partial L}{\partial \boldsymbol{a}} = \left(\boldsymbol{a}^\mathsf{T} X^\mathsf{T} W\boldsymbol{a} - 2Y^\mathsf{T} W\boldsymbol{a} + Y^2\right) = 2W^\mathsf{T} W a - 2W^\mathsf{T} Y = 0$$

If the matrix $W^\mathsf{T} W$ is invertable, then the solution to the problem is

$$\boldsymbol{a} = (W^\mathsf{T} W)^{-1} W^\mathsf{T} Y$$

3.2.2 Implement a function that solves the problem for periodic functions.

This function is implemented in Jupyter Notebook as **do_regression_with_periodic ()**.

3.2.3 Find the best fit for the data stored in the file regression_dataset2.

In this particular case that $K = T = 1$, so $f(x) = a_0 + a_1 \cos(2\pi x) + b_1 \sin(2\pi x)$. We need to minimize:

$$L(a_0, a_1, b_0) = \sum_{i=0}^{N-1} (a_0 + a_1 \cos(2\pi X_i) + b_1 \sin(2\pi X_i) - Y_i)^2$$

Let

$$\boldsymbol{a} = \begin{bmatrix} a_0 \\ a_1 \\ b_1 \end{bmatrix} \quad W = \begin{bmatrix} 1 & w_{0,1} & w_{0,2} \\ 1 & w_{1,1} & w_{1,2} \\ \vdots & \vdots & \vdots \\ 1 & w_{N-1,1} & w_{N-1,2} \end{bmatrix} = \begin{bmatrix} 1 & \cos(2\pi X_0) & \sin(2\pi X_0) \\ 1 & \cos(2\pi X_1) & \sin(2\pi X_1) \\ \vdots & \vdots & \vdots \\ 1 & \cos(2\pi X_{N-1}) & \sin(2\pi X_{N-1}) \end{bmatrix}$$

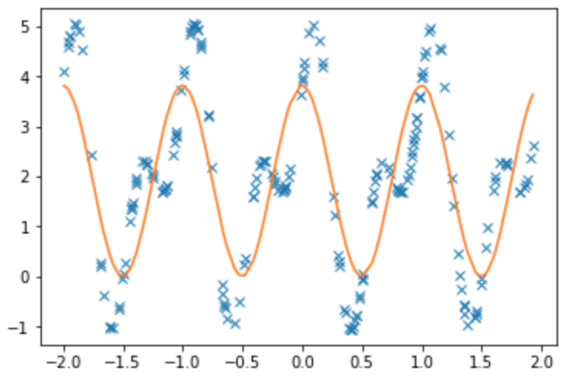
So we can derive the closed-form solution:

$$\boldsymbol{a} = (W^{\mathsf{T}}W)^{-1}W^{\mathsf{T}}Y$$

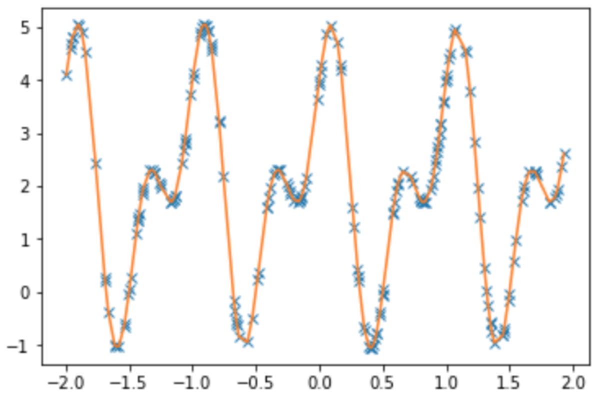
And I can provide more details of implementation in the Jupyter Notebook.

3.2.4 Plot the fitting function and the data from the dataset.

Plot this function and the data from the dataset, and it looks not good enough but it is limited by the setting of K and T. In this case, the loss function is already minimized:



As a result, I tried to change K into 2, while using the same x for each x_k to reconstruct the $\boldsymbol{W}$. Here is the fitting result, and it looks good:


5
4
3
2
1
0
−1
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
2.0