Linear Least Squares and Lagrangian

— Joaquín Gómez

— February 14, 2025

(Example taken from Deep Learning by Ian Goodfellow, Yoshua Bengio and Aaron Courville)

Suppose we want to find the value of x that minimizes

$$f(\mathbf{x}) = \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|^2$$

To do so, we first obtain the gradient:

$$\nabla_{\mathbf{x}} f(\mathbf{x}) = \mathbf{A}^T(\mathbf{A}\mathbf{x} - \mathbf{b}) = \mathbf{A}^T\mathbf{A}\mathbf{x} - \mathbf{A}^T\mathbf{b}$$

The derivative in the direction of a unit vector $\mathbf{u}$ is given by $\nabla_{\mathbf{x}} f(\mathbf{x}) \cdot \mathbf{u} = \|\nabla_{\mathbf{x}} f(\mathbf{x})\| \cos\theta$, and this suggests that if the angle in between the vectors is zero, then the directional derivative is maximum, if the angle is $\theta = \pi$, the value is minimum, this suggests that the gradient always points upwards. The method of **gradient descent** uses this principle to set a formula:

$$\mathbf{x}' = \mathbf{x} - \epsilon\nabla_{\mathbf{x}} f(\mathbf{x})$$

where ϵ is the learning rate, that controls the size of the step.

We can code an algorithm in python that performs a basic gradient descent on f

```
import numpy as np

A = np.array(...) # Matrix of mxn
x = np.array(...) # Vector of nx1
b = np.array(...) # Vector of mx1
learning_rate = 0.01
tolerance = 1e-6

def gradient(x):
    return A.T@(A@x - b)
```

```
while np.linalg.norm(gradient(x)) > tolerance:
    x = x - learning_rate * gradient(x)
```

Now, suppose we wish to minimize the same function, subject to a constraint $\mathbf{x}^T\mathbf{x} = 1$. To do so, we will use the Lagrangian (see Karush-Kuhn-Tucker approach):

$$L(\mathbf{x}, \lambda) = f(\mathbf{x}) + \lambda(\mathbf{x}^T\mathbf{x} - 1)$$

here $g^{(1)}(\mathbf{x}) = \mathbf{x}^T\mathbf{x} - 1 = 0$. Then we take the derivative with respect to $\mathbf{x}$

$$\frac{\partial L}{\partial \mathbf{x}} = \mathbf{A}^T\mathbf{A}\mathbf{x} - \mathbf{A}^T\mathbf{b} + 2\lambda\mathbf{x}$$

Minimizing this expression requires setting $\frac{\partial L}{\partial \mathbf{x}} = 0$, thus

$$\mathbf{A}^T\mathbf{A}\mathbf{x} - \mathbf{A}^T\mathbf{b} + 2\lambda\mathbf{x} = 0$$

Solving for $\mathbf{x}$

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A} + 2\lambda\mathbf{I})^{-1}\mathbf{A}^T\mathbf{b}$$

Now, we want to find the optimal value for λ, observe that

$$\frac{\partial L}{\partial \lambda} = \mathbf{x}^T\mathbf{x} - 1$$

and that if you increase the value of λ, the value of $\mathbf{x}$ decreases

$$\begin{aligned}\lambda \to \infty &: \mathbf{x} \to \mathbf{0} \\ \lambda \to 0 &: \mathbf{x} \to \boldsymbol{\infty}\end{aligned}$$

Suppose we start with $\lambda = 0$, increasing its value will cause the penalty $\lambda(\mathbf{x}^T\mathbf{x} - 1)$ to be stronger, but as $\mathbf{x}$ depends more on λ, $\mathbf{x}$ becomes smaller, making the difference $\mathbf{x}^T\mathbf{x} - 1 \to 0$.

— END — February 14, 2025