Machine Learning HW1

Xiaohe He 48655395

Due on 23:59 Mar 15, 2018

I. Preliminaries

1). Ex1: JD Goods Recommendations. From searching history to train the prediction function of each user, then link the goods data set and each user. Make a prediction before one serch a similar goods.

Ex2: Email spam filtering . By using users' setting and common dirt words to train the prediction funtion of spam eamil. One email will be marked as a spam email if fits the function well.

2). *Proof.* For sufficiently small stepsize α, we consider the next x_α can be expressed as:

$$x_\alpha = x + \alpha \cdot d$$

By first-order Taylor series expansion around x, we have

$$f(x_\alpha) = f(x) + \alpha \nabla f(x)^\top d + o(\alpha) \approx f(x) + \alpha \nabla f(x)^\top d$$

Since direction d is a descent direction, i.e., $\nabla f(x)^\top d < 0$, then

$$f(x_\alpha) < f(x)$$

Hence we can decrease f by moving along such a direction. □

II. Understanding Convex function and First-order necessary condition

1). *Proof.*

$$f(x) = f(x') + \nabla f(x')^T(x - x') + \frac{1}{2}(x - x')^T H(x' + \lambda(x - x'))(x - x')$$

Since Hessian of f is positive semidefinite, $\forall x, y \in dom(f)$, we have

$$f(x) \geq f(x') + \nabla f(x')^T(x - x')$$

Take any $x, y \in dom(f)$ and let $z = \lambda x + (1 - \lambda)y$, we have:

$$f(x) \geq f(z) + \nabla f(z)^T(x - z) \tag{1}$$

$$f(y) \geq f(z) + \nabla f(z)^T(y - z) \tag{2}$$

Multiplaying (1) by λ, (2) by $(1 - \lambda)$, adding we get:

$$\lambda f(x) + (1 - \lambda)f(y) \geq f(z) + \nabla f(z)^T(\lambda x + (1 - \lambda)y - z) = f(\lambda x + (1 - \lambda)y)$$

Hence, f is convex. □

2). *Proof.* Suppose x^* is the global minimizer of f.
Since function f is convex, for all $d \in \mathbb{R}^n, \lambda \in \mathbb{R}$, we have

$$f(x^*) \leq f(x^* + \lambda d)$$

Pick $\lambda > 0$,

$$\frac{f(x^* + \lambda d) - f(x^*)}{\lambda} \geq 0$$

Take limits as $\lambda \to 0$,

$$\nabla f(x^*)^T d \geq 0, \forall d \in \mathbb{R}^n$$

Since d arbitraty, replace with $-d$, $\Rightarrow \nabla f(x^*) = 0$ □

III. Linear Regression via Gradient Descent Method

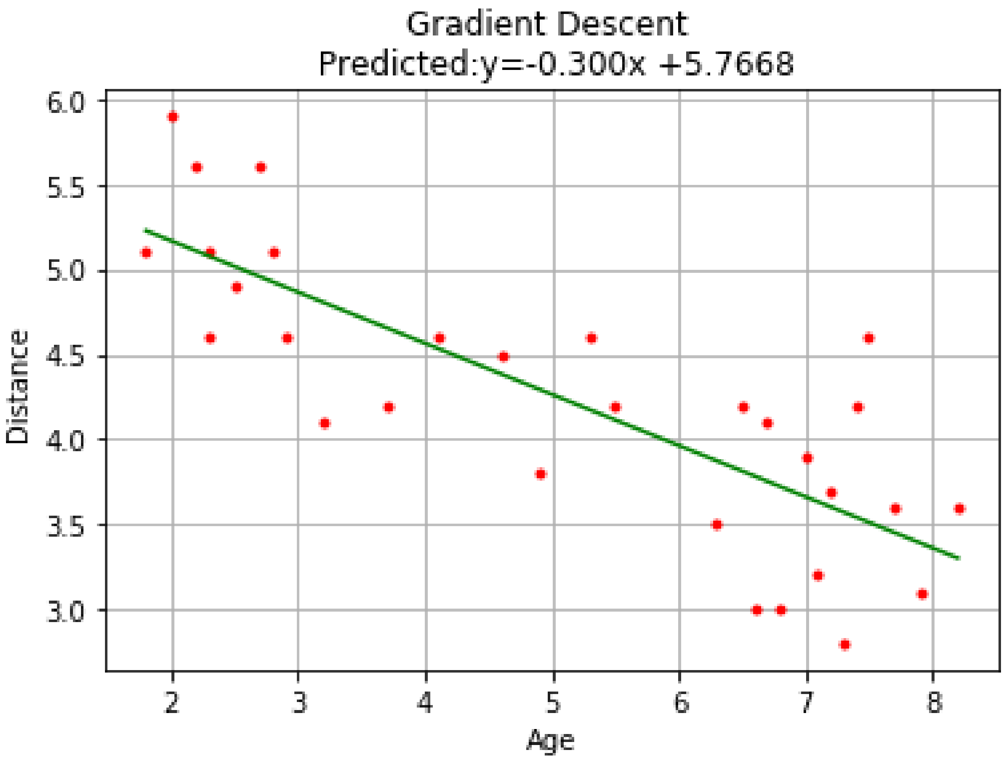
1). Linear regression formula:

$$y = k\mathbf{x} + b$$

Cost function:

$$J(\theta) = \frac{1}{2m}\sum_{i=0}^{m}(h(x^i) - y^i)^2$$

2). Termination criterion is number of iteration, which I set 5000. And learning rate is 0.01. Attachment *GradientDescent.ipynb* is the codes, please review.
Result as follow:



IV. How to Deal with Outliers

The function that I used here is Huber Loss Function, as it is less sensitive to outliers in data than the squared error loss:

$$L_\delta(y, f(x)) = \begin{cases} \frac{1}{2}(y - f(x))^2 & \text{for } |y - f(x)| \le \delta \\ \delta(|y - f(x)| - \frac{1}{2}\delta) & \text{otherwise} \end{cases}$$

However, I don't know how to implement it :(