Unit 2 - lecture 5: Linear Regression

Objectives:

- Write the training error as least squares criterion for linear regression
- Use stochastic gradient descent for fitting linear regression models.
- solve closed-form linear regression solution
- Indentify regularization term and how it changes the solution, generalization

Introduction:

Classification: $S_m = \{ (x^{(t)}, y^{(t)}) / t = 1, \dots, m \}$

$$x^{(t)} \in \mathbb{R}^d, \quad y^{(t)} \in \{-1, +1\}$$

In many problems, it is not enough to say yes or no kind of questions.

Regression: $f(x, \theta, \theta_0) = \sum_{i=1}^{d} \theta_i x_i + \theta_0 = \theta \cdot x + \theta_0$ and $y^{(t)} \in \mathbb{R}$

Empirical Risk:

$$R_m(\theta) = \frac{1}{m} \sum_{t=1}^{m} \frac{(y^{(t)} - \theta x^{(t)})^2}{2}$$

m training examples

$\frac{1}{m}$?: We want to for every single point in our training data and compute this extent of this deviation, compute some kind of loss. We can talk about how to define loss, sum them up, and then average

$\frac{(y^t - \theta x^t)^2}{2}$?: If the deviation is large, we're really penalized, and this is the behavior you are getting from the squared function, that the bigger difference would actually result in much higher loss.

Structural mistakes: Maybe the mapping between your training vectors and y's is actually highly non-linear. You would be incurring a high mistake.

Estimation mistakes: Even if we know that the mapping itself is linear, but you have very limited training data, you cannot estimate it correctly.

Empirical Risk: Hinge loss vs Squared Error loss

$(x^{(1)}, y^{(1)})$ $\quad x^{(1)} = [1, 0, 1]^T \quad y^{(1)} = 2$

$(x^{(2)}, y^{(2)})$ $\quad x^{(2)} = [1, 1, 1]^T \quad y^{(2)} = 2.7$ $\qquad$ and $\quad \theta = [0, 1, 2]^T$

$(x^{(3)}, y^{(3)})$ $\quad x^{(3)} = [1, 1, -1]^T \quad y^{(3)} = -0.7$

$(x^{(4)}, y^{(4)})$ $\quad x^{(4)} = [-1, 1, 1]^T \quad y^{(4)} = 2$

$$R_m(\theta) = \frac{1}{m} \sum_{t=1}^{m} \mathrm{loss}\big(\underbrace{y^{(t)} - \theta \cdot x^{(t)}}_{z}\big)$$

empirical risk

Hinge loss : $\mathrm{loss}_h(z) = \begin{cases} 0 & \text{if } z \geq 1 \\ 1-z & \text{otherwise} \end{cases}$

Squared Error loss : $\mathrm{Loss}_h(z) = \frac{z^2}{2}$

$R_m(\theta)$ with Hinge loss:

$$R_m(\theta) = \frac{1}{4} \sum_{t=1}^{4} \mathrm{Loss}_h\left(y^{(t)} - \theta \cdot x^{(t)}\right) \quad \text{with } \mathrm{loss}(z) = \begin{cases} 0 & \text{if } z \geq 1 \\ 1-z & \text{otherwise} \end{cases}$$

$z^{(1)}$: $y^{(1)} - \theta \cdot x^{(1)} = 2 - [0,1,2]^T \cdot [1,0,1]^T = 2 - (0.1 + 1.0 + 2.1) = 0 \quad < 1$

$z^{(2)}$: $y^{(2)} - \theta \cdot x^{(2)} = 2.7 - [0,1,2]^T \cdot [1,1,1]^T = 2.7 - (0.1 + 1.1 + 2.1) = -0.3 \quad < 1$

$z^{(3)}$: $y^{(3)} - \theta \cdot x^{(3)} = -0.7 - [0,1,2]^T \cdot [1,1,-1]^T = -0.7 - (0.1 + 1.1 + 2.(-1)) = 0.3 \quad < 1$

$z^{(4)}$: $y^{(4)} - \theta \cdot x^{(4)} = 2 - [0,1,2]^T \cdot [-1,1,1]^T = 2 - (0.(-1) + 1.1 + 2.1) = -1 \quad < 1$

$$R_m(\theta) = \frac{1}{4}\left((1-0) + (1-(-0{,}3)) + (1-0{,}3) + (1-(-1)\right) = 1.25$$

$R_m(\theta)$ with squared Error loss:

$$R_m(\theta) = \frac{1}{4} \sum_{t=1}^{4} \mathrm{loss}_h\left(y^{(t)} - \theta \cdot x^{(t)}\right) \quad \text{with } \mathrm{Loss}(z) = \frac{z^2}{2}$$

$$R_m(\theta) = \frac{1}{4}\left(\frac{0^2}{2} + \frac{(-0.3)^2}{2} + \frac{(0.3)^2}{2} + \frac{(-1)^2}{2}\right) = 0.1475$$

Gradient-Based Approach:

Let's compute GBA for one example:

$$\nabla_\theta \left(y^{(t)} - \theta . x^{(t)}\right)^2 / 2 = \frac{2}{2}\left(y^{(t)} - \theta . x^{(t)}\right)\left(-x^{(t)}\right) = -\left(y^{(t)} - \theta . x^{(t)}\right)\left(x^{(t)}\right)$$

Algorithm:

1. Initialize $\theta = 0$
2. Randomly pick $t = \{1, \dots, m\}$
3. update $\theta = \theta - \left[-\left(y^{(t)} - \theta . x^{(t)}\right) . \left(x^{(t)}\right)\right] = \theta + \underbrace{\eta}\left(y^{(t)} - \theta x^{(t)}\right) x^{(t)}$

minus because we are going against the direction of our gradient to minimize the expression.

learning rate (how much to go in the opposite direction)

$\eta_k = \frac{1}{1+k}$: With k number of iteration

Closed form solution:

$$R_m(\theta) = \frac{1}{m} \sum_{t=1}^{m} \left(y^{(t)} - \theta . x^{(t)}\right)^2 / 2$$

$$\nabla_\theta R_m(\theta)_{|\theta = \hat{\theta}} = \frac{1}{m} \sum_{t=1}^{m} \left[\nabla_\theta \left(y^{(t)} - \theta x^{(t)}\right)^2 / 2\right]_{|\theta = \hat{\theta}}$$

$\hat{\theta}$ is where this expression is $= 0$

$$= -\frac{1}{m} \sum_{t=1}^{m} \left(y^{(t)} - \theta . x^{(t)}\right) x^{(t)}$$

dot product

$$= \underbrace{-\frac{1}{m} \sum_{t=1}^{m} y^{(t)} x^{(t)}}_{\text{vector } b} + \frac{1}{m} \sum_{t=1}^{m} \hat{\theta} . x^{(t)} x^{(t)}$$

$$= -b + \frac{1}{m} \sum_{t=1}^{m} x^{(t)} \hat{\theta} . x^{(t)} \qquad \hat{\theta} . x^{(t)} = \left(x^{(t)}\right)^T \hat{\theta}$$

$$= -b + \underbrace{\frac{1}{m} \sum_{t=1}^{m} x^{(t)} \left(x^{(t)}\right)^T}_{A} \hat{\theta}$$

$= -b + A\hat{\theta} \Rightarrow$ We are trying to solve $A\hat{\theta} = b$ $\left(-b + A\hat{\theta} = 0\right)$

$\Longrightarrow \hat{\theta} = A^{-1} b$ A is not always reversible, A is reversible when $m \gg d$

numb of training examples

dimension

Cost of reversibility of A has a huge cost $\theta(d^3)$

Regularization:

We completed the discussion of 2 ways of doing the linear algorithm for the linear Regression

What happens if we don't have enough training examples?

What happens if our training examples contain some noise?

We will solve these questions with a mechanism called Regularization.

What Regularization will do?

It will push you away from trying to perfectly fit your training examples.

Reach Regression: How well we are fitting the training examples like the Empirical Risk

$$J_{\lambda,m}(\theta) = \underbrace{\frac{\lambda}{2}\|\theta\|^2}_{\text{keep all the } \theta \text{ very close to zero, because it's norm}} + \underbrace{R_m(\theta)}_{\text{"Empirical Risk}}$$

Keep all the θ very close to zero, because it's norm

"Empirical Risk: Controls how much loss we're incurring in our training data set.

We want to keep θ's grounded in some area and only push them when we have enough evidence.

$$J_{\lambda,m}(\theta) = \frac{\lambda}{2}\|\theta\|^2 + \frac{1}{m}\sum_{t=1}^{m}\left(y^{(t)} - \theta . x^{(t)}\right)^2/2$$

$$\nabla_\theta J_{\lambda,m}(\theta) = \nabla_\theta \left(\frac{\lambda}{2}\|\theta\|^2 + \left(y^{(t)} - \theta x^{(t)}\right)^2/2\right)$$

$$= \lambda\theta - \left(y^{(t)} - \theta x^{(t)}\right)x^{(t)}$$

1. Initialize: $\theta = 0$
2. Randomly pick $t = \{1, \dots m\}$
3. $\theta = \theta - \eta\left(\lambda\theta - \left(y^{(t)} - \theta x^{(t)}\right)x^{(t)}\right)$

$$\theta = \theta - \eta\left(\lambda\theta - \left(y^{(t)} - \theta x^{(t)}\right)x^{(t)}\right)$$

$$\theta = (1 - \eta\lambda)\theta + \eta\left(y^{(t)} - \theta . x^{(t)}\right)x^{(t)}$$

Goals of this expression:

1. It's self-correcting it pushes the parameters in the right direction to minimize the loss.
2. It also pushes our thetas down

Regularization: Extreme case 1

$$J_{m,\lambda}(\theta,\theta_0) = \frac{1}{m}\sum_{t=1}^{m}\frac{(y^{(t)}-\theta\cdot x^{(t)}-\theta_0)^2}{2}+\frac{\lambda}{2}\|\theta\|^2$$

* If we increase λ to ∞, minimizing J is equivalent to minimizing $\|\theta\|$.
  Thus θ will have to be a zero vector $\Longrightarrow f(x)=\theta\cdot x+\theta_0$ becomes $f(x)=\theta_0$, a horizontal line

* If we decrease λ to zero, minimizing J is equivalent to minimizing $\frac{1}{m}\sum_{t=1}^{m}\frac{(y^t-\theta\cdot x^t-\theta_0)^2}{2}$
  which is the "fit" $\Longrightarrow$ fitting the data

$\lambda \nearrow \iff$ more mistakes, we are pushing $\theta \longrightarrow 0$

$\lambda \searrow \iff$ Fitting training data