Regression Models for this week and next week:

- Regression models
  - Linear Regression
  - Polynomial Regression
    - (Linear and Polynomial Regression) → One variable / Multiple variable
      - L_2 Regularization ⇒ Ridge Regression
      - L_1 Regularization ⇒ LASSO Regression
  - Logistic Regression

Four Components for Regression Models:

1. Parameters : θ
2. Hypothesis : $h_\theta(X)$
3. Cost Function : $J(\theta)$
4. Goal : $\underset{\theta}{\text{minize}}\{J(\theta)\}$

Hierarchical structure

Something important about Gradient Descent:

① Updat simultaneously

② Learning Rate α

- Too small: slow
- Too large: Diverge
- No need to decrease over time ⇒ Auto take small steps
- Batch Gradient Descent:
- Debugging ⇒ Make sure that $J(\theta)$ decrease on every iteration.

How to choose: 0.003
0.03
0.3

③ Normalize Features

- Feature Scaling ($-1 \le x \le 1$)
- Mean Normalization ($-0.5 \le x \le 0.5$)

④ In specific cases: Normal equation.

$$\theta = (X^T \cdot X)^{-1} \cdot X^T \cdot y$$

In Octave/Matlab: $\boxed{\theta = \text{pinv}(X' * X) * X' * y}$

When $(X^T \cdot X)$ is not invertible:

How to solve: ① Delete some features $\Rightarrow$ avoid (A. linearly dependent) or (B. $n > m$ case)

$\Downarrow$ known as feature redundancy

② Use Regularization $\Rightarrow$

$$\theta = \left[X^T X + \lambda \cdot \begin{bmatrix} 0 & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{bmatrix}_{(n+1)\times(n+1)} \right]^{-1} \cdot X^T y, \text{ where } \lambda > 0$$

Comparison with Gradient Descent:

Advantage:

① No need to choose α

② No need to iterate

Disadvantage:

① Slow when n is large $\Rightarrow$ $O(n^3)$ while GD is $O(n^2)$

② $X^T X$ may non-invertible.

Linear Regressions:

Cost function:

$$\frac{1}{2m}\cdot\left[\sum_{i=1}^{m}\Big(\underbrace{h_\theta(x^{(i)})}_{\theta_0+\theta_1\cdot x_1^{(i)}+\theta_2\cdot x_2^{(i)}-\!-\,\theta_n\cdot x_n^{(i)}}-y^{(i)}\Big)^2+\underbrace{\lambda\cdot\sum_{i=1}^{n}}_{\text{不从0开始}}\cdot\theta_i^2\right]$$

Error function:

$$\frac{1}{2m_{test}}\cdot\sum_{i=1}^{m_{test}}\Big(h_\theta(x^{(i)})-y^{(i)}\Big)^2$$

Logistic Regressions:

Cost function: $\Rightarrow$ comes from cross entropy

$$-\frac{1}{m}\cdot\sum_{i=1}^{m}\left[y^{(i)}\cdot\log h_\theta(x^{(i)})+\left[1-y^{(i)}\right]\cdot\log\Big(1-h_\theta(x^{(i)})\Big)\right]$$

Error function:

$$\frac{1}{m_{test}}\cdot\sum_{i=1}^{m_{test}}\underbrace{\text{error}\big(h_\theta(x^{(i)}),\ y^{(i)}\big)}$$

$h_\theta(x^{(i)})>0.5\ \&\&\ y^{(i)}=0$

$\|\ h_\theta(x^{(i)})<0.5\ \&\&\ y^{(i)}=1$

$\Rightarrow\ 1$

0, otherwise

Cost :

$$L(y, x, p_\theta) = -\left[y \cdot \log(h_\theta(x)) + (1-y) \cdot \log(1-h_\theta(x))\right]$$

$$= -\left[y \cdot \log(\sigma(\theta^T \cdot x)) + (1-y) \cdot \log(1-\sigma(\theta^T \cdot x))\right]$$

Gradient:

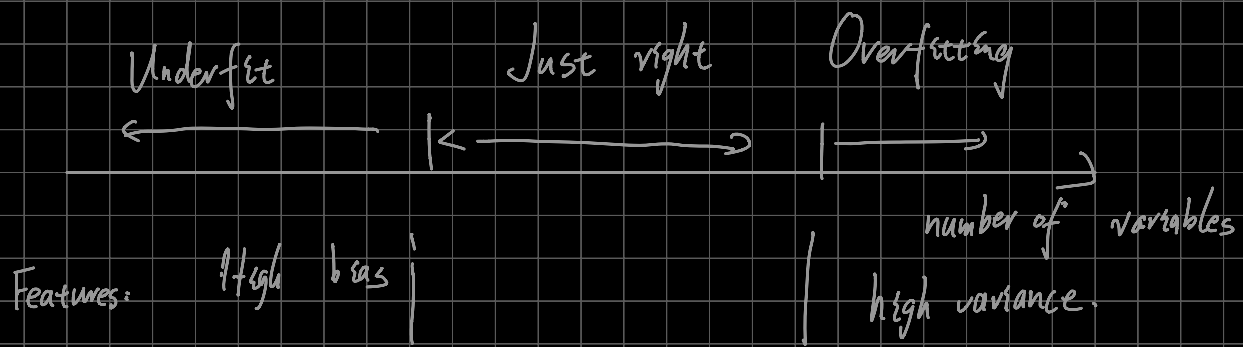
$$\frac{\partial L(y \cdot x; p\theta)}{\partial \theta} = -y \cdot \frac{1}{\cancel{\sigma(\theta^T x)}} \cdot \cancel{\sigma(\theta^T \cdot x)} \cdot (1-\sigma(\theta^T \cdot x)) \cdot x$$

$$- (1-y) \cdot \frac{1}{\cancel{1-\sigma(\theta^T \cdot x)}} \cdot -1 \cdot \sigma(\theta^T \cdot x)\cancel{(1-\sigma(\theta^T \cdot x))}\, x$$

$$= (\sigma(\theta^T \cdot x) - y) \cdot x$$

Overfitting:

Causes: When the samples is not enough while the features are too much, it will cause overfitting. The curve try too hard to wiggly through all the training examples.



Addressing Overfitting:

Method:

A. Reduce features
- Manually pick features
- Model selection Algorithms.

B. Regularization

$\Rightarrow$ Keep all the features, but reduce magnitude of parameters θ_i. This method is more suitable for handling multi-feature problems.

each feature is slightly useful.

- Weight regularization
  - L_1 norm
  - L_2 norm
- Drop out regularization
  - Normal Dropout
  - Data Augmentation
  - Early stopping