ML OL Course

Week 3

Logistic Regression

Classificiation

Email: Spam / Not Spam?

Online Transactions: Fraudulent (Yes / No)?

Tumor: Malignant / Benign?

Binary Class,

Linear Regression is not a good idea to apply in ~~Linear Regression~~ Classification.

also it seems strange that the output is larger than 1 or lower than 0.

Classification: $y = 0$ or 1

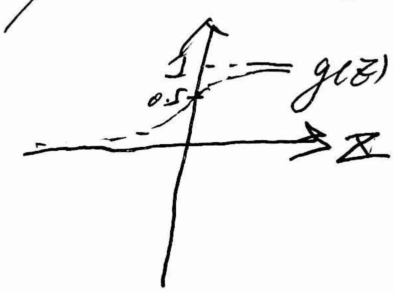
$h_\theta(x)$ can be > 1 or < 0

Logistic Regression: $0 \leq h_\theta(x) \leq 1$

Logistic Regression Model

where $0 \leq h_\theta(x) \leq 1$

$$\left.\begin{array}{l} h_\theta(x) = g(\theta^T x) \\ \\ g(z) = \dfrac{1}{1-e^{-z}} \end{array}\right\} \rightarrow h_\theta(x) = \frac{1}{1-e^{-\theta^T x}}$$



→ Sigmoid function

→ logistic function

Interpretation of Hypothesis Output

$h_\theta(x)$ = estimated probability that $y=1$ on input x

Example: If $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tumorsize} \end{bmatrix}$

$h_\theta(x) = 0.7$

Tell patient that 70% chance of tumor being malignant.

$h_\theta(x) = P(y=1 \mid x; \theta)$ "probability that $y=1$, given x parameterized by θ"

$P(y=0 \mid x;\theta) = 1 - P(y=1 \mid x;\theta)$

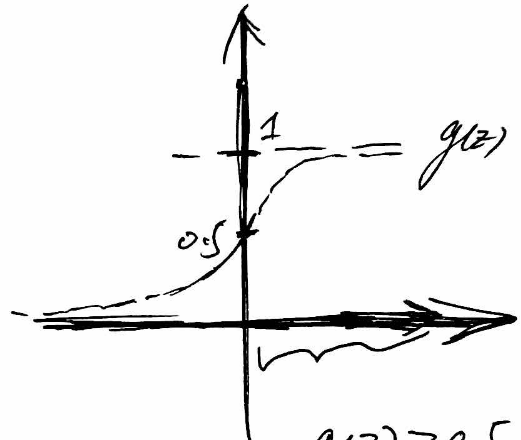
$h_\theta(x) = P(y=1|x;\theta) = 1 - P(y=0|x;\theta)$

$P(y=0|x;\theta) + P(y=1|x;\theta) = 1$

Decision Boundary

$h_\theta(x) = g(\theta^T x) = P(y=1|x;\theta)$

$g(z) = \dfrac{1}{1+e^{-z}}$



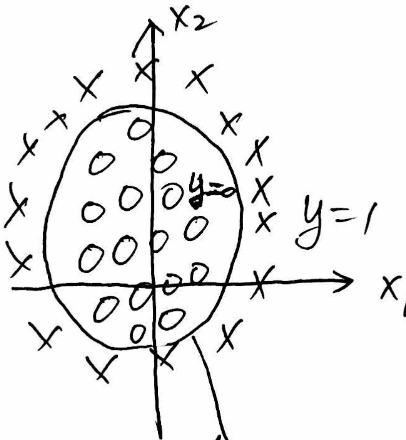
$g(z) \geqslant 0.5$
when $z \geqslant 0$

$h_\theta(x) = g(\theta^T x) \geqslant 0.5$
whenever $\underbrace{\theta^T x}_{z} \geqslant 0$

Suppose predict "$y=1$" if $h_\theta(x) \geqslant 0.5$ $\qquad \theta^T x \geqslant 0$

predict "$y=0$" if $h_\theta(x) < 0.5$ $\qquad \theta^T x < 0$

☆ Decision Boundary is a property of hypothesis function !!!



decision Boundary

Non-linear decision boundaries

$h_\theta(x) = g(\underset{=-1}{\theta_0} + \underset{=0}{\theta_1} x_1 + \underset{=0}{\theta_2} x_2 + \underset{=1}{\theta_3} x_1^2 + \underset{=1}{\theta_4} x_2^2)$

predict "$y=1$" if $-1 + x_1^2 + x_2^2 \geqslant 0$

$x_1^2 + x_2^2 = 1 \qquad x_1^2 + x_2^2 \geqslant 1$

Cost function: Traing set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(m)}, y^{(m)})\}$

m examples $x \in \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$ $\mathbb{R}^{n+1}$ $x_0 = 1, y \in \{0, 1\}$

$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$

how to choose parameter θ ? $= \frac{1}{m} \sum_{i=1}^{m} \frac{1}{2} (h_\theta(x^{(i)}) - y^{(i)})^2$

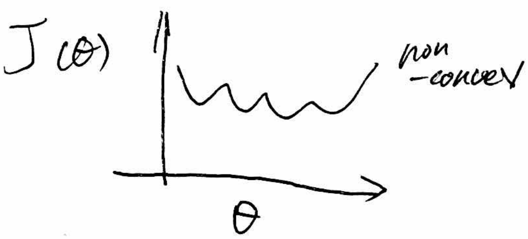
Cost Fun: linear regression: $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \text{cost}(h_\theta(x^{(i)}), y)$

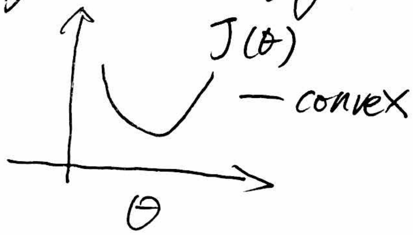
$\text{Cost}(h_\theta(x^{(i)}), y^{(i)}) = \frac{1}{2} (h_\theta(x^{(i)}) - y^{(i)})^2$

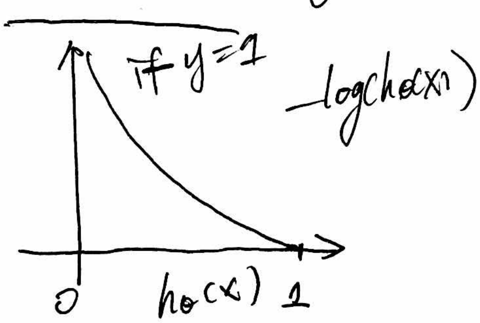
Convex or not Convex

Logistic regression cost function.

$$\text{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$



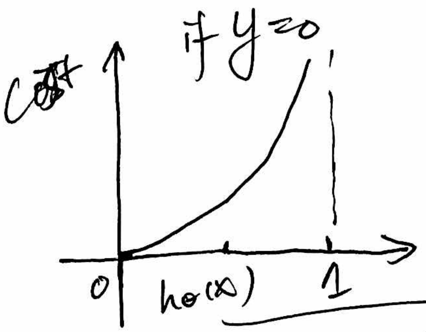




cost = 0 if $y = 1$, $h_\theta(x) = 1$

But as $h_\theta(x) \to 0$ cost $\to \infty$

Captures intuition that if $h_\theta(x) = 0$, (predict $P(y=1 | x; \theta) = 0$), but $y = 1$, we'll penalized learning algorithm by a very large cost.



cost = 0 if $y = 0$, $h_\theta(x) = 0$

But as $h_\theta(x) \to 1$, cost $\to \infty$

overall; $\text{Cost}(h_\theta(x), y) = 0$ if $h_\theta(x) = y$

$\text{Cost}(h_\theta(x), y) \to \infty$ if $y = 0$ and $h_\theta(x) \to 1$

$\text{Cost}(h_\theta(x), y) \to \infty$ if $y = 1$ and $h_\theta(x) \to 0$

Simplied Cost function and Gradient Decent

Logistic regression cost function: $J(\theta) = \frac{1}{m}\sum_{i=1}^{m} cost(h_\theta(x^{(i)}), y^{(i)})$

$$Cost(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y=1 \\ -\log(1-h_\theta(x)) & \text{if } y=0 \end{cases}$$

Note: $y=0$ or 1 always

More convenient way $\Rightarrow$ $Cost(h_\theta(x), y) = -y\log(h_\theta(x)) - (1-y)\log(1-h_\theta(x))$

If $y=1$: $cost(h_\theta(x), y) = -\log h_\theta(x)$

If $y=0$: $Cost(h_\theta(x), y) = -\log(1-h_\theta(x))$

$J(\theta) = \frac{1}{m}(-y^T\log(h) - (1-y)^T\log(1-h))$ where $h = g(X\theta)$

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)})$$

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

principle of maximum likelihood estimation

To fit parameters θ:

$$\min_\theta J(\theta)$$

To make a prediction given new x:

Output $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$ $\quad P(y=1 \mid x; \theta)$

GD: Repeat $\{ \theta_j := \theta_j - \alpha \frac{\partial}{\partial\theta_j} J(\theta) \}$ simultaneously update all θ_j

$\Leftrightarrow$ Repeat $\{ \theta_j := \theta_j - \frac{\alpha}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} \}$

Algorithm looks identical to linear regression!

Linear Regression: $h_\theta(x) = \theta^T x$

$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$

a vectorized implementation is

$$\theta := \theta - \frac{\alpha}{m} X^T(g(X\theta) - y)$$

Advanced Optimization:

Opti algorithm: cost $J(\theta)$, $\min_\theta J(\theta)$

Given θ; we can get

- $J(\theta)$
- $\frac{\partial}{\partial \theta_j} J(\theta)$ (for $j = 0, 1, \dots, n$)

GD: Repeat $\{ \theta_j = \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta) \}$

Opt algorithms:
- GD
- Conjugate gradient
- BFGS
- L-BFGS

Advantages:
- No need to manually pick α
- Often faster than gradient descent

Disadvantages:
- more complex

Example: $\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$

$$J(\theta) = (\theta_1 - 5)^2 + (\theta_2 - 5)^2$$

$$\frac{\partial}{\partial \theta_1} J(\theta) = 2(\theta_1 - 5)$$

$$\frac{\partial}{\partial \theta_2} J(\theta) = 2(\theta_2 - 5)$$

octave functions

fminunc(@costFuntion, initialTheta, options)

$\theta \in \mathbb{R}^d$ $d \geq 2$

Multiclass classification:

$\longrightarrow$ one vs all classifecation

$h_\theta^{(i)}(x) = P(y=i \mid x;\theta)$ $(i=1,2,3)$

One vs all classification

Train a logistic regression classifier $h_\theta^{(i)}(x)$ for each class i to predict the probability that $y=i$. More details see below. ①

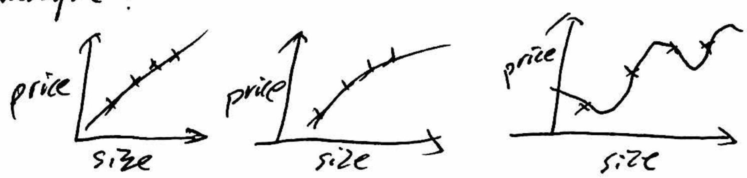
On a new input x, to make a prediction, pick the class i that maximizes

$\max_i h_\theta^{(i)}(x)$

① We are basically choosing one class and then lumping all the others into a single second class. We do this repeatedly, applying binary logistic regression to each case, and then use the hypothesis that returned the highest value as our prediction.

— Regularization : The problem of overfitting

~~Example~~ Example :



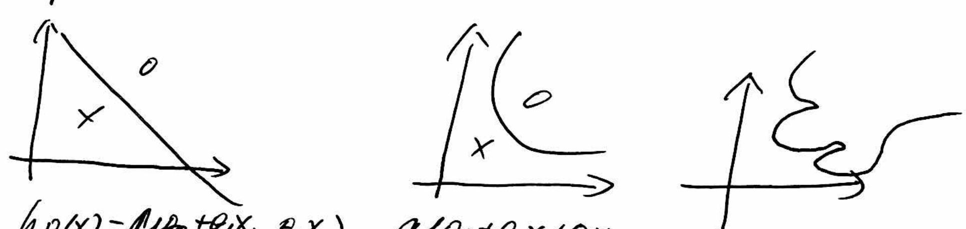
$\theta_0+\theta_1 x$ "underfit" "High bias"

$\theta_0+\theta_1 x+\theta_2 x^2$ "just right"

$\theta_0+\theta_1 x+\theta_2 x^2+\theta_3 x^3+\theta_4 x^4$ "overfit" "high variance"

Overfitting: If we have too many features, the learned hypothesis may fit the training set very well. ($J(\theta)=\frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})^2\approx 0$), but fail to generalize to new examples (predict prices on new examples).

Example



$h_\theta(x)=g(\theta_0+\theta_1 x_1, \theta_2 x_2)$

$g(\theta_0+\theta_1 x_1+\theta_2 x_2+\theta_3 x_1^2+\theta_4 x_2^2+\theta_5 x_1 x_2)$

$g(\theta_0+\theta_1 x_1+\theta_2 x_1^2+\theta_3 x_1^2 x_2+\theta_4 x_1^2 x_2^2+\theta_5 x_1^2 x_2^3+\theta_6 x_1^3 x_2+\dots)$

Overfit

1. reduce number of features.
   - Manually select which features to keep.
   - Model selection algorithm ~~(later)~~
2. Regularization.
   - Keep all the features, but reduce magnitude/values of parameters θ_j.
   - Works well when we have a lot of features, each of which contributes a bit to predicting y.

— Cost Function

Regularization.

- Small value for parameters $\theta_0, \theta_1, \dots, \theta_n$
- "Simpler" hypothesis
- Less prone ~~to~~ overfitting

regularization parameter

$$J(\theta)=\frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)})-y^{(i)})^2+\lambda\sum_{j=1}^{n}\theta_j^2\right]$$

If lambda is chosen to be too large, it may smooth out the function too much and cause underfitting.

Regularized linear regression:

$$J(\theta) = \frac{1}{2m}\left[\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 + \sum_{j=1}^{n}\theta_j^2\right]$$

Gradient decent:

Repeat {

$$\theta_0 := \theta_0 - \alpha\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha\left[\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} + \frac{\lambda}{m}\theta_j\right]$$

}

$$\Downarrow$$

$$\theta_j := \theta_j(1 - \alpha\frac{\lambda}{m}) - \frac{\alpha}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}$$

$1 - \alpha\frac{\lambda}{m} < 1$ since $\alpha, \lambda, m > 1$

Normal equation

$$X = \begin{bmatrix} (x^{(1)})^T \\ \vdots \\ (x^{(m)})^T \end{bmatrix} \qquad y = \begin{bmatrix} y^{(1)} \\ \vdots \\ y^{(m)} \end{bmatrix} \quad \mathbb{R}^m$$

$m \times (n+1)$

$$\min_\theta J(\theta)$$

$$\rightarrow \theta = \left(X^T X + \lambda \underbrace{\begin{bmatrix} 0 & & \\ & 1 & \\ & & \ddots & \\ & & & 1 \end{bmatrix}}_{(n+1)\times(n+1)}\right)^{-1} X^T y$$

E.g. $n=2$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Non-invertibility

Suppose $m \le n$,
(#examples) (#features)

$$\theta = \underbrace{(X^T X)^{-1}}_{\text{non-invertible/sigular}} X^T y$$

X is non-invertible if $m < n$,
and may be non-invertible if $m = n$.

If $\lambda > 0$,

$$\theta = \left(X^T X + \lambda \begin{bmatrix} 0 & & & \\ & 1 & & \\ & & 1 & \\ & & & \ddots & \\ & & & & 1 \end{bmatrix}\right)^{-1} X^T y$$

不算上0, $n \times n$ identity matrix

Regularized logistic Regression

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1-y^{(i)}) \log(1-h_\theta(x^{(i)}))\right] + \frac{\lambda}{2m}\sum_{j=1}^{n} \theta_j^2$$

Gradient descent

Repeat {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

$$\theta_j := \theta_j - \alpha \left[\frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} + \frac{\lambda}{m}\theta_j\right]$$

}

$$\sigma(x)' = \left(\frac{1}{1+e^{-x}}\right)' = \frac{-(1+e^{-x})'}{(1+e^{-x})^2} = \frac{-1' - (e^{-x})'}{(1+e^{-x})^2} = \frac{+e^{x}}{(1+e^{-x})^2}$$

$$= \left(\frac{1}{1+e^{-x}}\right)\left(\frac{e^{-x}}{1+e^{-x}}\right) = \sigma(x)\left(\frac{1+1-1+e^{-x}}{1+e^{-x}}\right) = \sigma(x)\left(\frac{1+e^{-x}}{1+e^{-x}} - \frac{1}{1+e^{-x}}\right)$$

$$= \sigma(x)\left(1-\sigma(x)\right)$$

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

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

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

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[\frac{y^{(i)}\sigma(\theta^T x^{(i)})(1-\sigma(\theta^T x^{(i)}))\frac{\partial}{\partial\theta_j}\theta^T x^{(i)}}{h_\theta(x^{(i)})} + \frac{-(1-y^{(i)})\sigma(\theta^T x^{(i)})(1-\sigma(\theta^T x^{(i)}))\frac{\partial}{\partial\theta_j}\theta^T x^{(i)}}{1-h_\theta(x^{(i)})}\right]$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}(1-\sigma(\theta^T x^{(i)}))\,\cancel{\theta^T x^{(i)}}\,x_j + \frac{-(1-y^{(i)})\,\sigma(\theta^T x^{(i)})\,\cancel{(1-\sigma(\theta^T x^{(i)}))}\,x_j}{\cancel{h_\theta(x^{(i)})}}\right]$$

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

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

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

Logistic Regression : classification.

Binary Classification

$h_\theta(x) = g(\theta^T x)$

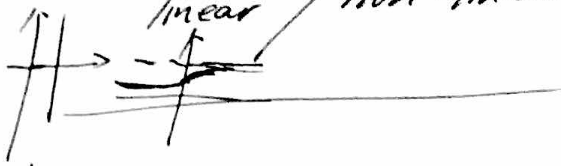
$z = \theta^T x \qquad g(z) = \frac{1}{1+e^{-z}}$

$h_\theta(x) = P(y=1 \mid x;\theta) = 1 - P(y=0 \mid x;\theta)$

$P(y=0 \mid x;\theta) + P(y=1 \mid x;\theta) = 1$

Decision Boundary

linear / non-linear



Cost Function:

$J(\theta) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{cost}(h_\theta(x^{(i)}), y^{(i)})$

$\mathrm{Cost}(h_\theta(x), y) = -\log(h_\theta(x))$ if $y=1$

$\mathrm{Cost}(h_\theta(x), y) = -\log(1-h_\theta(x))$ if $y=0$

combined:

$\mathrm{Cost}(h_\theta(x), y) = -y\log(h_\theta(x))$

$- (1-y)\log(1-h_\theta(x))$

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

Vectorization: $h = g(X\theta)$

$J(\theta) = \frac{1}{m}\cdot(-y^T\log(h) - (1-y^T)\log(1-h))$

GD:

$\theta = \theta - \frac{\alpha}{m}X^T(h-y) \qquad h = g(X\theta)$

$\frac{\partial}{\partial\theta_j}J(\theta) = \frac{1}{m}\sum_{i=1}^{m}[h_\theta(x^{(i)}) - y^{(i)}]x_j^{(i)}$

$\nabla J(\theta) = \frac{1}{m}X^T(h-y) \qquad h = g(X\theta)$

---

multiclass classification:

One vs. all $\quad \text{prediction} = \max_i(h_\theta^{(i)}(x))$