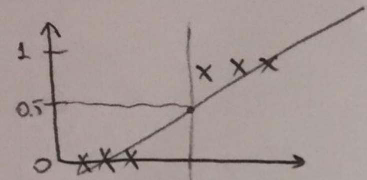


① CLASSIFICATION & REPRESENTATION.

1.1. Classificatn:

$y \in \{0,1\}$ $\quad y \in \{0,1,2,3\}$



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

Threshold classifier output $h_\theta(x)$ at 0.5

if $h_\theta(x) \geq 0.5$, predict "$y=1$"

if $h_\theta(x) < 0.5$, predict "$y=0$"

⇒ Applying linear regression to a classificatn problem often is not a great idea

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

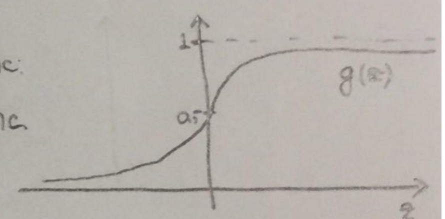
Logistic Regression: $0 \leq h_\theta(x) \leq 1$
(classificatn algorithm)

1.2. Hypothesis Representatn:

Logistic Regression Model

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

$h_\theta(x) = g(\theta^T x)$ $\qquad g(z) = \dfrac{1}{1+e^{-z}}$ $\qquad$ sigmoid func. / logistic func.



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

Interpretatn of hypothesis output:

$h_\theta(x)$ = estimated prob. that $\boxed{y=1}$ on input x

ex/ $h_\theta(x) = 0.7$ ⇒ 70% chance of tumor being malignant.

$h_\theta(x) = P(y=1 | x; \theta)$ "Prob. that $y=1$, given x, parameterized by θ".

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

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

1.3. Decision Boundary:

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

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

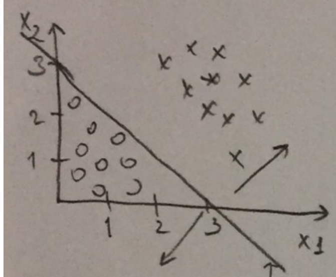
→ Try to understand better when this hypothesis will make predictns that $y=1$ vs. when it might make predictns that $y=0$.

→ Understand better what hypothesis fxn looks like particularly when we have more than one feature.

$g(z) \geq 0.5$ when $z \geq 0$ $(y=1)$ ⟶ $\theta^T x \geq 0$

$g(z) < 0.5$ when $z < 0$ $(y=0)$ ⟶ $\theta^T x < 0$

Decision Boundary: (2 features)



$x_1 + x_2 = 3$
(decision boundary)

$h_\theta(x) = g(\overset{-3}{\theta_0} + \overset{1}{\theta_1} x_1 + \overset{1}{\theta_2} x_2)$

Predict "$y=1$" if $-3 + x_1 + x_2 \geq 0$

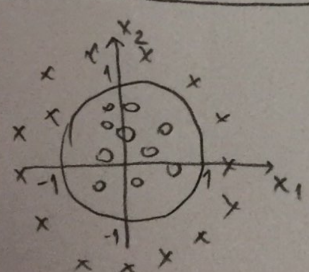
$x_1 + x_2 \geq 3 \rightarrow y=1$

$x_1 + x_2 < 3 \rightarrow y=0$

$x_1 + x_2 = 3 \rightarrow h_\theta(x) = 0.5$

⇒ Once we have particular values for the paramtrs $\theta_0, \theta_1, \theta_2$ ⇒ that completely defines the decision boundary and we do not need to plot a training set i.o.t. plot the decision boundary.

Non-Linear decision boundaries:



$\boxed{x_1^2 + x_2^2 = 1}$

circle eqn.
decision boundary

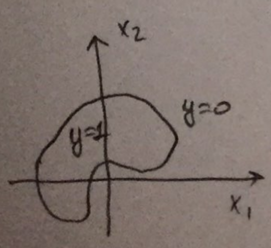
$h_\theta(x) = g(\overset{-1}{\theta_0} + \overset{0}{\theta_1} x_1 + \overset{0}{\theta_2} x_2 + \overset{1}{\theta_3} x_1^2 + \overset{1}{\theta_4} x_2^2)$

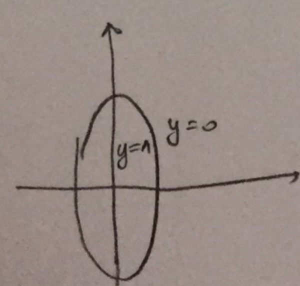
Predict "$y=1$" if $-1 + x_1^2 + x_2^2 \geq 0$

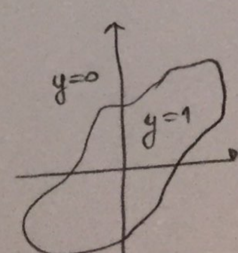
$x_1^2 + x_2^2 \geq 1$

⇒ Add higher order features like in polynomial regression.

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







2. LOGISTIC REGRESSION MODEL

2.1. Cost Function

Training set – m examples

$$x \in \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \quad x_0 = 1 \quad y \in \{0,1\} \qquad h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

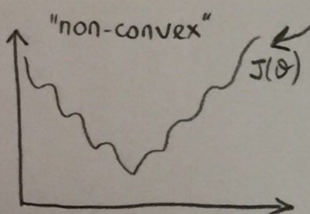
How to choose paramtrs θ?

Cost fxn:

Linear Regression: $J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \underbrace{\frac{1}{2} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2}_{cost(h_\theta(x^{(i)}), y^{(i)})}$

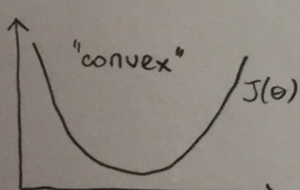
$cost(h_\theta(x), y) = \frac{1}{2}\left(h_\theta(x) - y\right)^2$

– for logistic regression , $\frac{1}{1+e^{-\theta^T x}}$, pretty complicated, nonlinear fxn



→ if you run grad. desc. on this fxn, it is not guaranteed to converge to the global minimum.

"LOGISTIC REGRESSION" — $J(\theta)$ – non convex fxn if we define a square cost fxn.



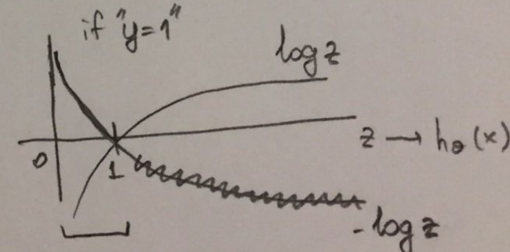
→ we would be guaranteed that grad. desc. would converge to the global minimum.

"LINEAR REGRESSION"

⇒ We want $J(\theta)$ to be convex fxn (for logistic regression), so we can apply a great algorithm (grad. desc.)

Logistic Regression Cost fxn:

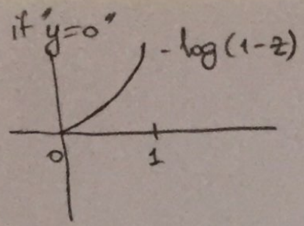
$$Cost(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y=1 \\ -\log(1-h_\theta(x)) & \text{if } y=0 \end{cases}$$



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

as $h_\theta(x) \to 0$, Cost $\to \infty$

if $h_\theta(x) = 0$ $(P(y=1|x;\theta)=0)$, but $y=1$, we will penalize learning algorithm by a very large cost.



2.2. Simplified Cost Fxn & Gradient descent.

Logistic Regression cost fxn:

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

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

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

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

→ This cost fxn can be derived from statistics using the principle of max. likelihood estimatn. Which is an idea in statistics for how to efficiently find paramtrs' data for different models.

To fit paramtrs θ: $\min_\theta J(\theta)$

To make a predictn given new x: Output $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$

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

Repeat {

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

} [simultaneously update $\forall\ \theta_j$]!

looks like identical to linear regression!

linear regression: $h_\theta(x) = \theta^T x$

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

⇒ feature scaling can help grad. desc. converge faster for both linear regr. & logis. regr.

2.3. Advanced Optimizatn:

Optimizatn algorithm:

Cost fxn $J(\theta)$. Want $\min_\theta J(\theta)$

Given θ, we have code that can compute
- $J(\theta)$
- $\frac{\partial}{\partial \theta_j} J(\theta)$

Grad. descent:

Repeat {

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

}

Optimizatn algorithms:

→ Gradient Descent

→ Conjugate Gradient
→ BFGS
→ L-BFGS

Pros:
- No need to manually pick α
- has clever inner-loop: line-search algorithm automatically tries out different α & pics α.
- Often faster than grad.desc. (converge much faster).

Cons:
- more complex

ex/ $\theta = \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}$

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

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

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

```
function [jVal, gradient] = costFunction (theta)
    jVal = (θ1-5)^2 + (θ2-5)^2          → J(θ)
    gradient = zeros (2, 1)              →
    gradient (1) = 2(θ1-5)               → ∂/∂θ1 J(θ)
    gradient (2) = 2(θ2-5)               → ∂/∂θ2 J(θ)

options = optimset ('GradObj', 'on', 'MaxIter', '100');
initialTheta = zeros (2,1);
[optTheta, functionVal, exitFlag, ...] =
    = fminunc (@costFunction, initialTheta, options);
```

[fminunc = func. minimizatn unconstrained]

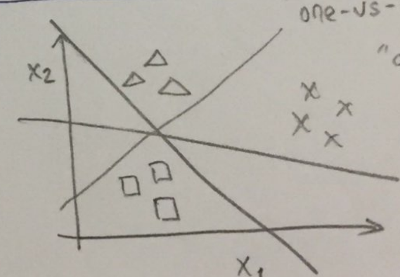
3. MULTICLASS CLASSIFICATION.

3.1. Multiclass Classificatn: One-vs-All:

"one-vs-rest"
"one-vs-all"

Email foldering: Work, Friends, Family, Hobby
$y=1, \quad y=2, \quad y=3, \quad y=4$

Weather: Sunny, Cloudy, Rain, Snow



⇓

3 seperate classificatn problem.

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

⇒ Train a logistic regression classifier $h_\theta^{(i)}(x)$ for each class i to predict the probability that $y=i$

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

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

4. QUIZ (100% ✓)