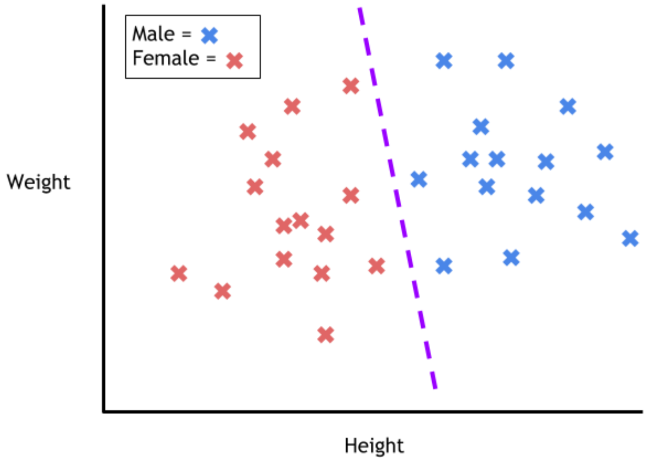
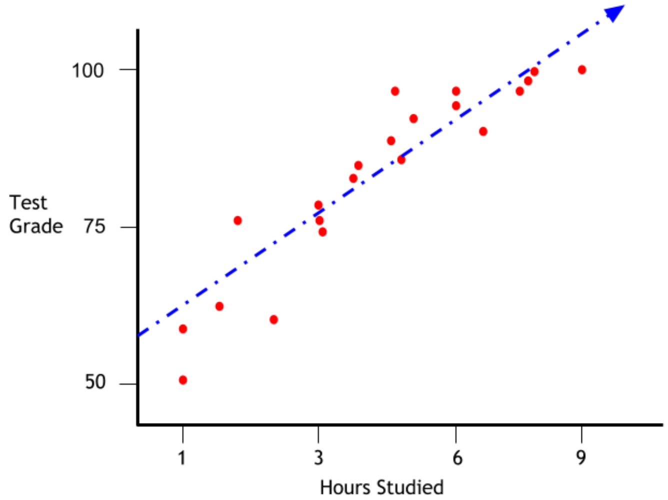
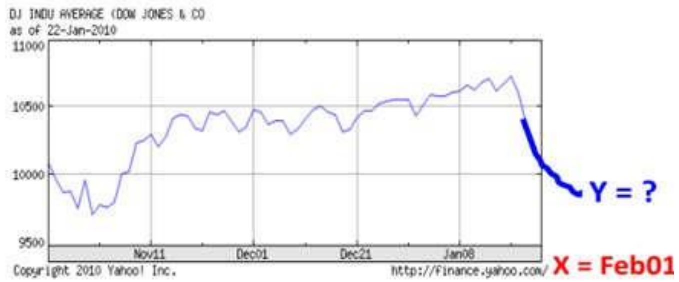
Logistic Regression

Prediction Problems: Discrete and Continuous Labels







Classification:

Regression:

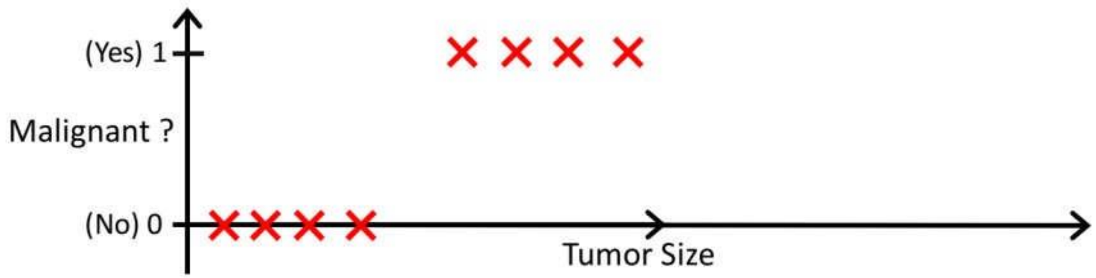
Example Classification Problem

Tumor: Malignant / Benign ?

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

$$y \in \{0, 1\}$$

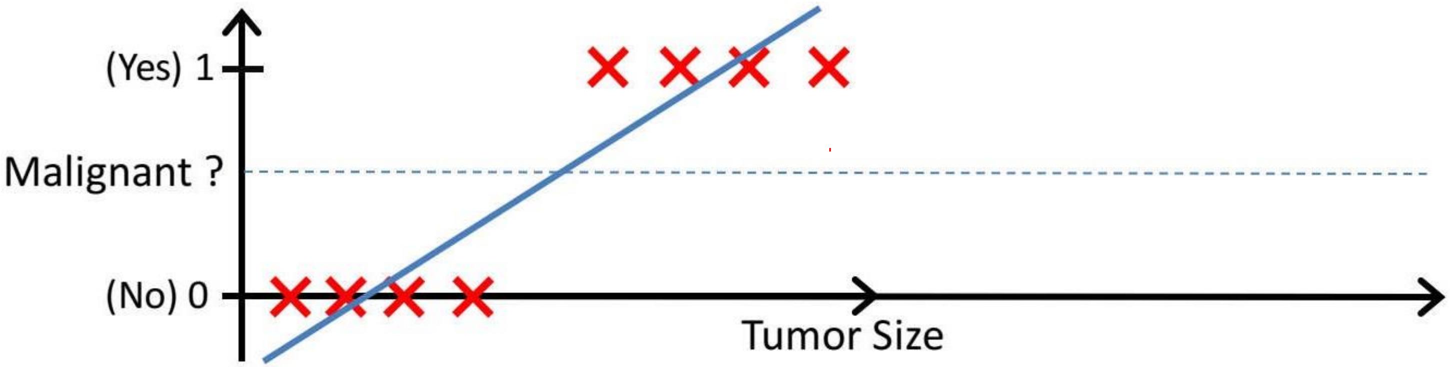
0: “Negative Class” (benign tumor)
1: “Positive Class” (malignant tumor)



slides taken from course materials of Andrew Ng

Can we solve using linear regression?

Fit a straight line and define a threshold



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

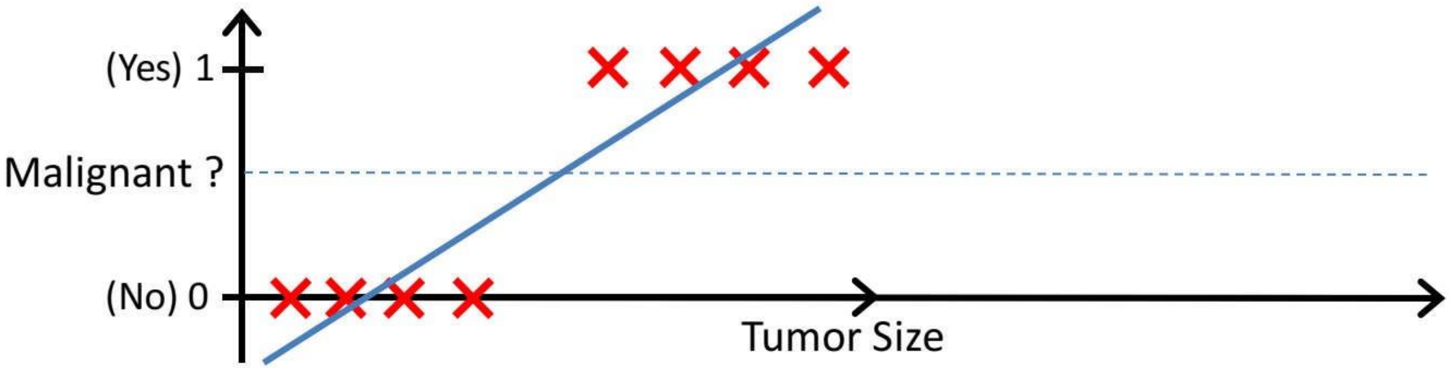
If $h_\theta(x) \geq 0.5$, predict "y = 1"

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

slides taken from course materials of Andrew Ng

Can we solve using linear regression?

Add a new data point



$_{\theta}(x)$ at 0.5:

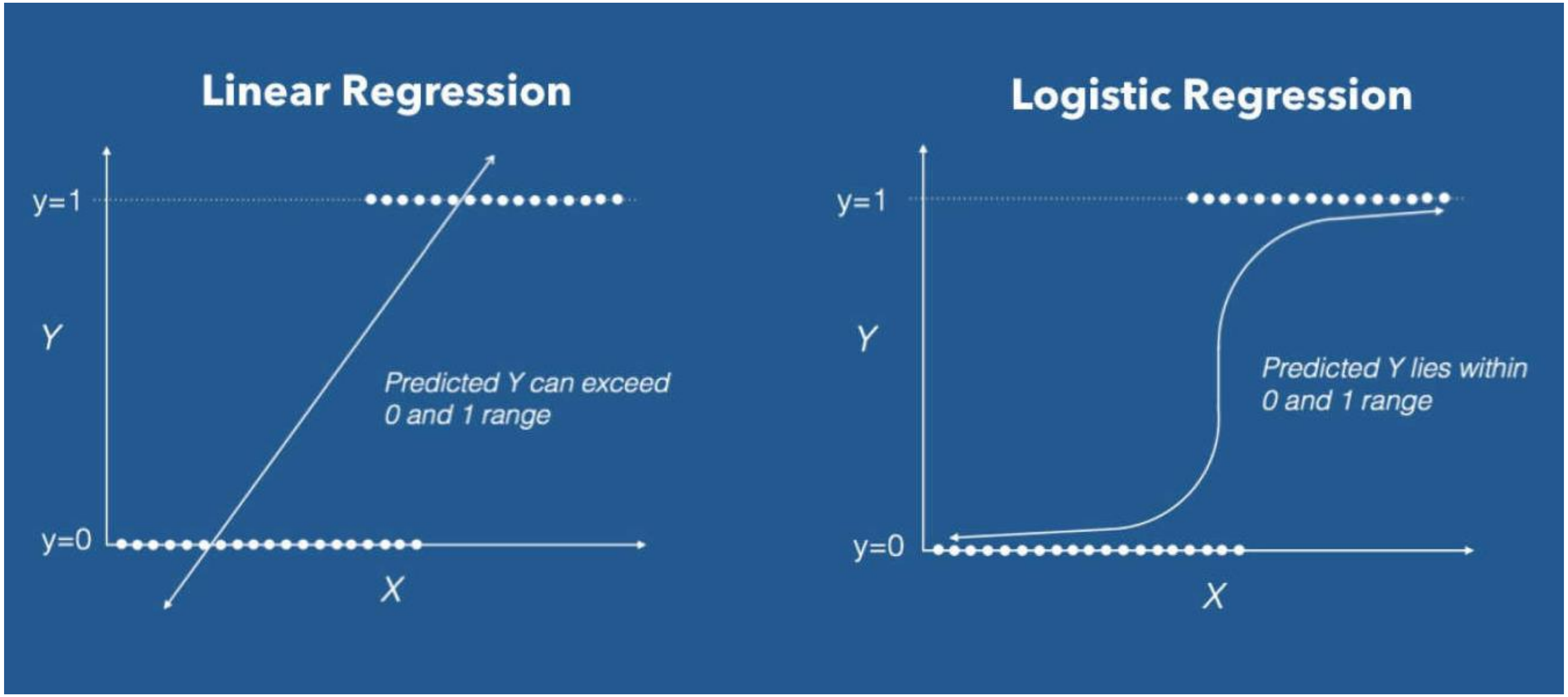
If $h_\theta(x) \geq 0.5$, predict "y = 1"

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

H(x) can be > 1 or < 0

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

How to limit h(x)?



Linear Regression VS Logistic Regression Graph| Image: Data Camp

Logit and Logistic Function

Odds - *ratio of success to ratio of failure*

Logit function

$$logit(p) = \log\left(\frac{p}{1-p}\right), \text{ for } 0 \leq p \leq 1$$

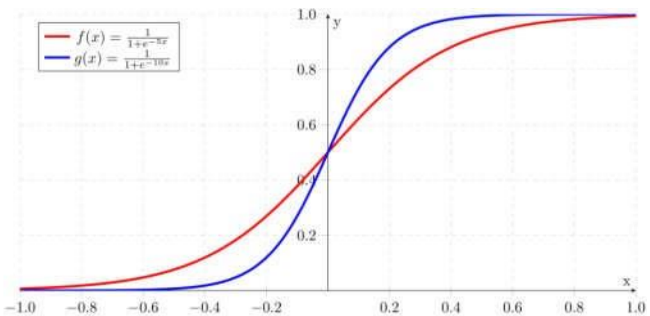
The logit function takes a value between 0 and 1 and maps it to a value between $-\infty$ and ∞

Inverse logit (logistic) function

$$g^{-1}(x) = \frac{\exp(x)}{1+\exp(x)} = \frac{1}{1+\exp(-x)}$$

Sigmoid function

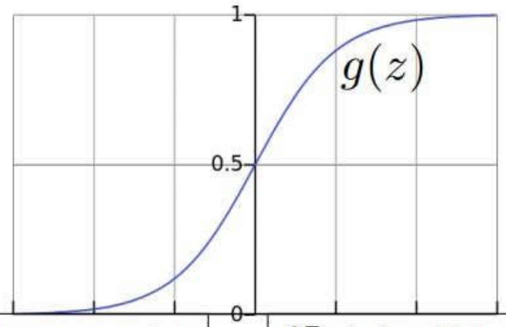
The inverse logit function takes a value between $-\infty$ and ∞ and maps it to a value between 0 and 1



Logistic Regression Model

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = g\left(\boldsymbol{\theta}^\intercal \boldsymbol{x}\right)$$

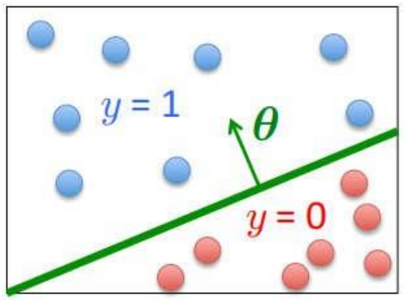
$$g(z) = \frac{1}{1+e^{-z}}$$



$\boldsymbol{\theta}^\intercal \boldsymbol{x}$ should be large negative values for negative instances

$\boldsymbol{\theta}^\intercal \boldsymbol{x}$ should be large positive values for positive instances

- Assume a threshold and...
  - Predict $y = 1$ if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) \geq 0.5$
  - Predict $y = 0$ if $h_{\boldsymbol{\theta}}(\boldsymbol{x}) < 0.5$



Interpretation of Hypothesis Output

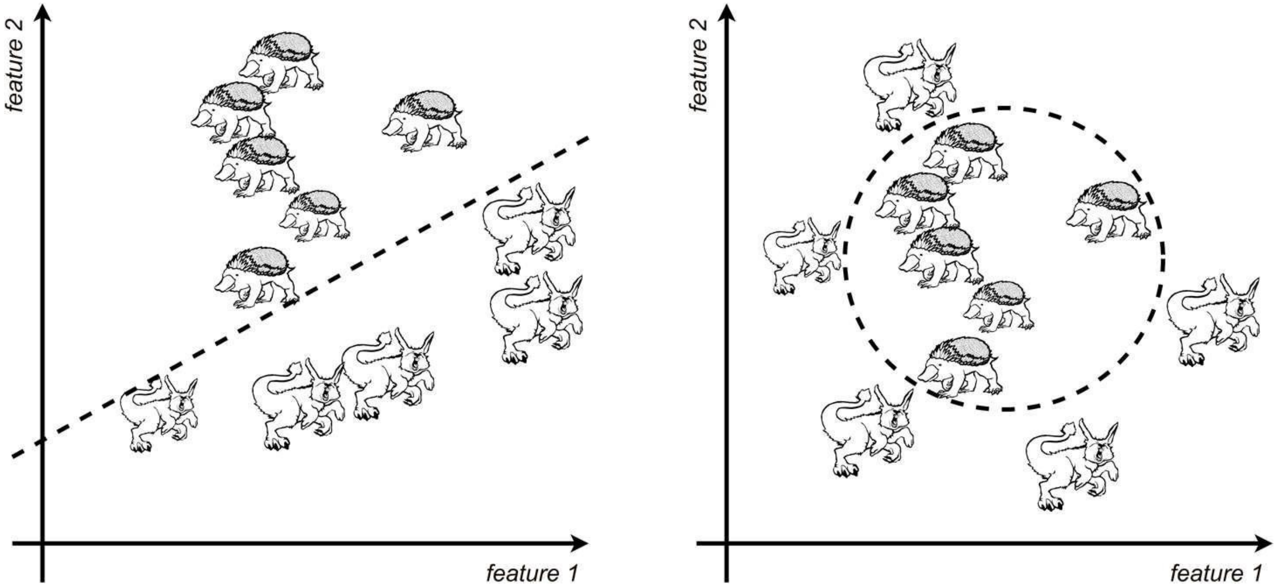
Tell patient that 70% chance of tumor being malignant

Linear Separability

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

Decision Boundary

Finding a good decision boundary => learn appropriate values for the parameters



Linear separable (on the left) and nonlinear separable (on the right) data. The decision boundary as shown in the dashed line. Source: Mykola Sosnovshchenko

Summary

- Logistic Regression is a classification algorithm
- Hypothesis use a sigmoid function to map output to 0-1 range
- Output is the estimated as a probability

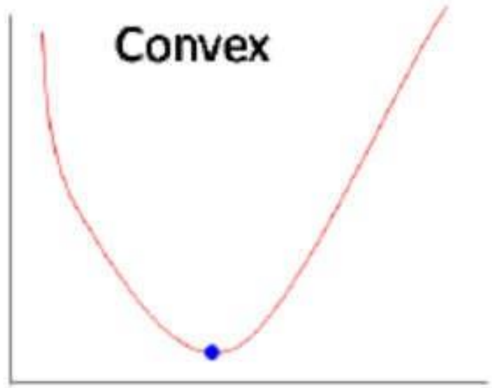
Cost Function

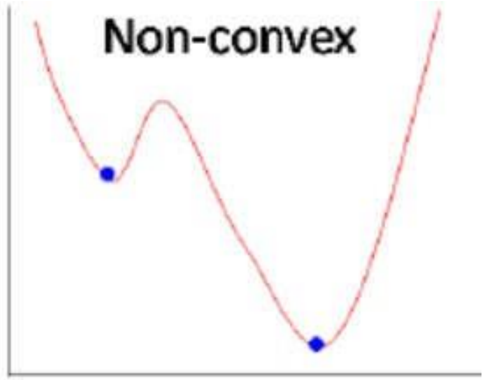
Linear Regression Squared error cost function $$J(\boldsymbol{\theta}) = \frac{1}{2n}\sum_{i=1}^{n}\left(h_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}\right) - y^{(i)}\right)^2$$

Nonlinear function in logistic regression $$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{\theta}^\mathsf{T}\boldsymbol{x}}}$$

results in a non-convex optimization

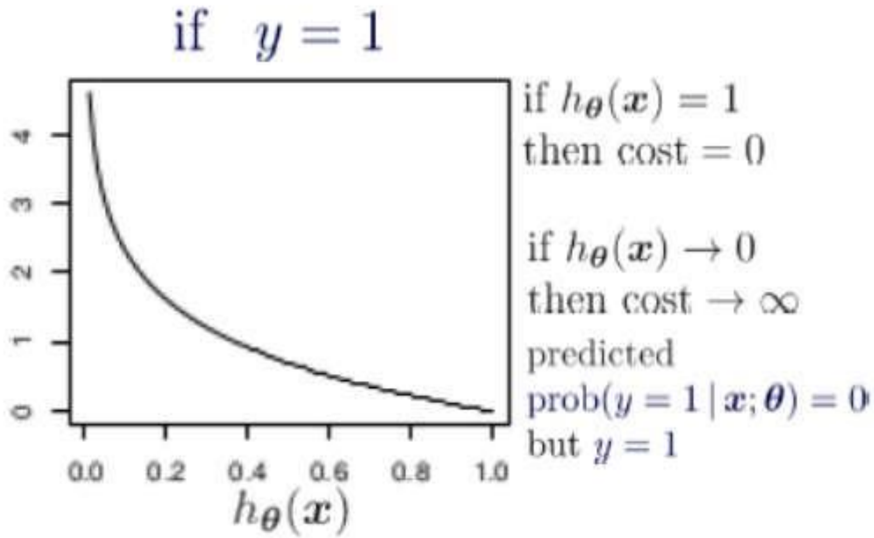
Assume that
$$\begin{aligned} P(y=1 \mid x;\theta) &= h_\theta(x) \\ P(y=0 \mid x;\theta) &= 1 - h_\theta(x) \end{aligned}$$
$$p(y \mid x;\theta) = (h_\theta(x))^y \,(1-h_\theta(x))^{1-y}$$

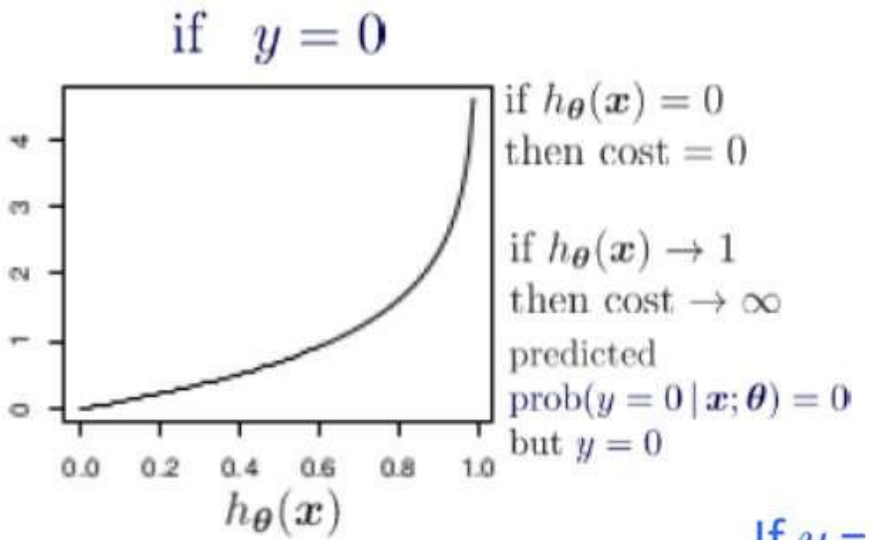


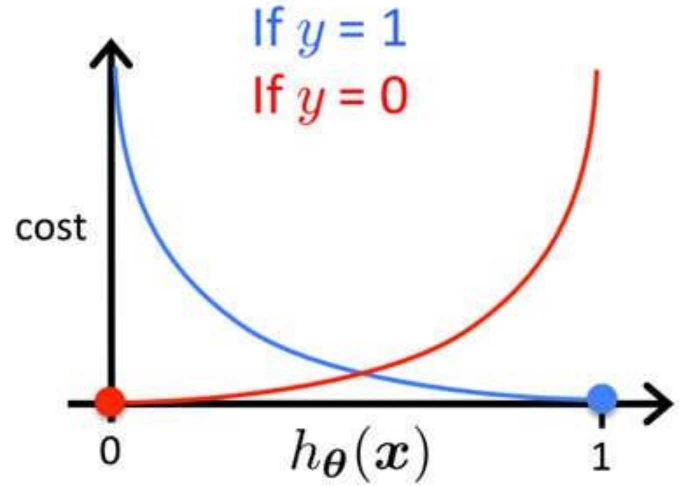


Cost Function

log loss error function







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

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

Maximum Likelihood Estimation (MLE)

- To chose values for the parameters of logistic regression
- (1) write the log-likelihood function
- (2) find the values of θ that optimize the log-likelihood function

1. log-likelihood function

likelihood

$$
\begin{aligned}
L(\theta) &= p(\vec{y} \mid X; \theta) \\
&= \prod_{i=1}^{m} p(y^{(i)} \mid x^{(i)}; \theta) \\
&= \prod_{i=1}^{m} \left(h_\theta(x^{(i)})\right)^{y^{(i)}} \left(1 - h_\theta(x^{(i)})\right)^{1-y^{(i)}}
\end{aligned}
$$

log likelihood

$$
\begin{aligned}
\ell(\theta) &= \log L(\theta) \\
&= \sum_{i=1}^{m} y^{(i)} \log h(x^{(i)}) + (1 - y^{(i)}) \log(1 - h(x^{(i)}))
\end{aligned}
$$

2. optimize log-likelihood function

- No closed form for the maximum
- Best values of theta by using an optimization algorithm
- Compute partial derivative of log likelihood with respect to each parameter

Derivative of Sigmoid Function

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

$$\frac{d\big(\sigma(\mathrm{x})\big)}{dx} = \frac{0 * (1 + e^{-\mathrm{x}}) - (1) * \big(e^{-\mathrm{x}} * (-1)\big)}{(1 + e^{-\mathrm{x}})^2}$$

$$\frac{d\big(\sigma(\mathrm{x})\big)}{dx} = \frac{(e^{-\mathrm{x}})}{(1 + e^{-\mathrm{x}})^2} = \frac{1\text{-}1 + (e^{-\mathrm{x}})}{(1 + e^{-\mathrm{x}})^2} = \frac{1 + e^{-\mathrm{x}}}{(1 + e^{-\mathrm{x}})^2} - \frac{1}{(1 + e^{-\mathrm{x}})^2}$$

$$\frac{d\big(\sigma(\mathrm{x})\big)}{dx} = \frac{1}{1 + e^{-\mathrm{x}}} * \left(1 - \frac{1}{1 + e^{-\mathrm{x}}}\right) = \sigma(\mathrm{x})\big(1 - \sigma(\mathrm{x})\big)$$

Partial derivative of log-likelihood

$$\ell(\theta) = y \log h(x) + (1-y)\log(1-h(x))$$

$$\begin{aligned}
\frac{\partial}{\partial\theta_j}\ell(\theta) &= \left(y\frac{1}{g(\theta^T x)} - (1-y)\frac{1}{1-g(\theta^T x)}\right)\frac{\partial}{\partial\theta_j}g(\theta^T x) \\
&= \left(y\frac{1}{g(\theta^T x)} - (1-y)\frac{1}{1-g(\theta^T x)}\right) g(\theta^T x)(1-g(\theta^T x))\frac{\partial}{\partial\theta_j}\theta^T x \\
&= \left(y(1-g(\theta^T x)) - (1-y)g(\theta^T x)\right)x_j \\
&= \left(y - h_\theta(x)\right)x_j
\end{aligned}$$

Gradient Descent Optimization

- partial derivative of log likelihood with respect to each parameter

$$= \quad (y - h_\theta(x))\, x_j$$

Take small steps in the direction of gradient, to reach local maximum

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

Gradient Descent for Logistic Regression

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

Want $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$

- Initialize $\boldsymbol{\theta}$
- Repeat until convergence

$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\boldsymbol{\theta})$$

simultaneous update
for j = 0 ... d

Example

	NPG	PGL	DIA	TSF	INS	BMI	DPF	AGE	Diabet
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

```
#Predictor variables:
#NPG= number of times pregnant
#PGL= Plasma glucose concentration a 2 hours in an oral
#DIA= Diastolic blood pressure (mm Hg)
#TSF=Triceps skin fold thickness (mm)
#INS= 2-Hour serum insulin (mu U/ml)
#BMI=Body mass index (weight in kg/(height in m)^2)
#DPF= Diabetes pedigree function
#AGE= Age (years)

#Output variable:
#Diabet= 0/1
```

Learned Parameters

```
NPG     0.123182
PGL     0.035164
DIA    -0.013296
TSF     0.000619
INS    -0.001192
BMI     0.089701
DPF     0.945180
AGE     0.014869
```

Model

y= sigmoid(t0 + t1*NPG + t2*PGL+ t3*DIA + t4*TSF + t5*INS + t6*BMI + t7*DPF + t8*AGE)

Test data

<8,196,30,38,230,45,0.180,34>

P= 0.971684882874

97% chance of the above patient getting diabetes

Is Logistic Regression a Linear Classifier ?

$$P(Y=0|X) = \frac{1}{1+\exp(\boldsymbol{\theta}_0 + \sum_i \boldsymbol{\theta}_i X_i)}$$

$$P(Y=1|X) = 1 - P(Y=0|X) = \frac{exp(\boldsymbol{\theta}_0 + \sum_i \boldsymbol{\theta}_i X_i)}{1+exp(\boldsymbol{\theta}_0 + \sum_i \boldsymbol{\theta}_i X_i)}$$

predict positive if $P(Y=1|X) > P(Y=0|X)$, or equivalently:

$$\frac{P(Y=1|X)}{P(Y=0|X)} > 1$$

Taking logs on both sides

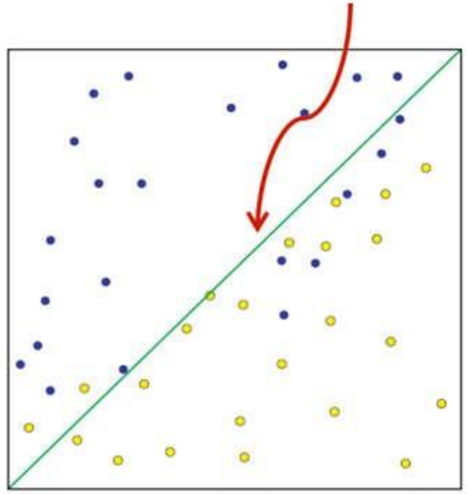
$$log\left(\frac{P(Y=1|X)}{P(Y=0|X)}\right) > 0$$

$$log(exp(\boldsymbol{\theta}_0 + \sum_i \boldsymbol{\theta}_i X_i) \;-\; log(1+exp(\boldsymbol{\theta}_0 + \sum_i \boldsymbol{\theta}_i X_i)$$

$$-\; \cancel{log(1)} \;+\; \cancel{log(1+exp}(\boldsymbol{\theta}_0 + \sum_i \boldsymbol{\theta}_i X_i) \;> 0$$

$$(\boldsymbol{\theta}_0 + \sum_i \boldsymbol{\theta}_i X_i) > 0$$

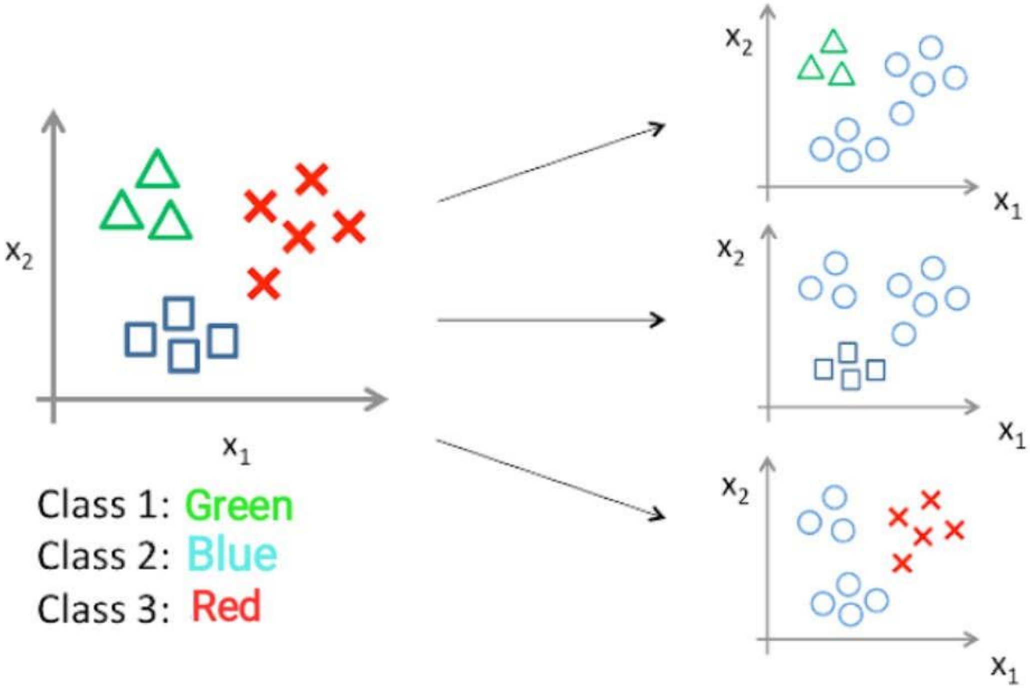
$(\boldsymbol{\theta}_0 + \sum_i \boldsymbol{\theta}_i X_i) = 0$



(Linear Decision Boundary)

Multiclass Classification

- One vs. All:- **N-class instances** then **N binary classifier models**
- One vs. One:- **N-class instances** then **N* (N-1)/2 binary classifier models**

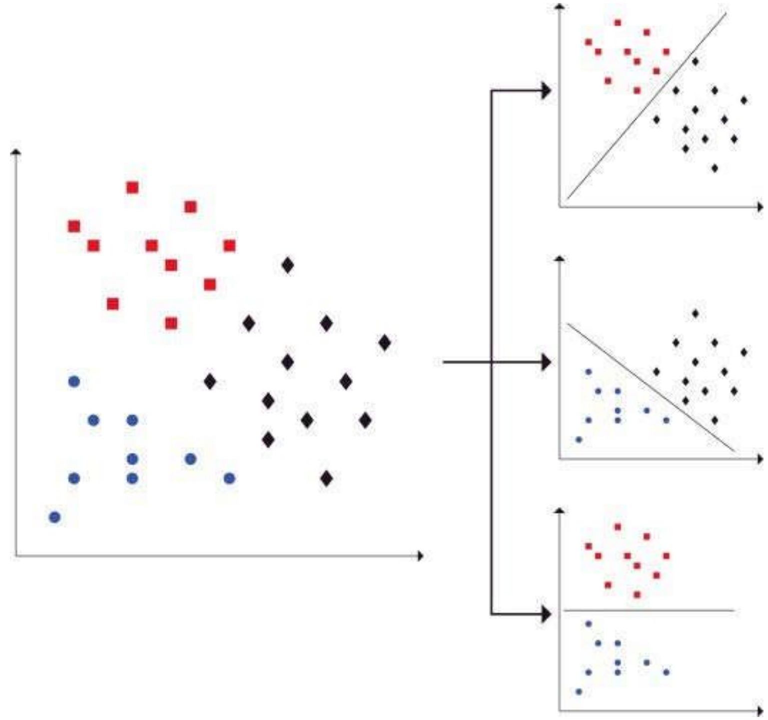


- **Classifier 1:- [Green] vs [Red, Blue]**
- **Classifier 2:- [Blue] vs [Green, Red]**
- **Classifier 3:- [Red] vs [Blue, Green]**

Multiclass Classification

- One vs. One:- **N-class instances** then **N* (N-1)/2 binary classifier models**

  Split the dataset into one binary classification dataset for each pair of classes



N* (N-1)/2 = 3

- Classifier 1: Red vs. Black
- Classifier 2: Blue vs. Black
- Classifier 3: Blue vs. Red

model with majority

https://towardsdatascience.com/multi-class-classification-one-vs-all-one-vs-one-94daed32a87b

Probabilistic classifier

1. A **feature representation** of the input.
2. A classification function that computes the estimated class, via $p(y|x)$. Ex. **sigmoid**
3. An objective function for learning, **loss function**
4. An algorithm for optimizing the objective function. **gradient descent** algorithm

Discriminative Classifier

- Problem: To distinguish dog images from cat images

- Discriminative model
  - learn to distinguish the classes without learning much about them
  - Estimate parameters of P(Y|X) directly from training data

- Generative model
  - Estimate parameters of P(X|Y), P(Y) directly from training dat
  - Indirect computation of P(Y|X) through Bayes rule

Logistic Regression is a Discriminative model

Summary

- Logistic Regression is a probabilistic classifier $p(y|x)$
- Logistic Regression is a linear classifier
  - decision rule is a hyperplane
- Logistic Regression is a discriminative classifier
- Uses sigmoid function to map output to 0-1 range
- Uses a log loss error function
- Logistic Regression is optimized by conditional likelihood
  - no closed-form solution
  - global optimum with gradient descent