CS 60050
Machine Learning

Classification: Logistic Regression

Some slides taken from course materials of Andrew Ng

Classification

Email: Spam / Not Spam?

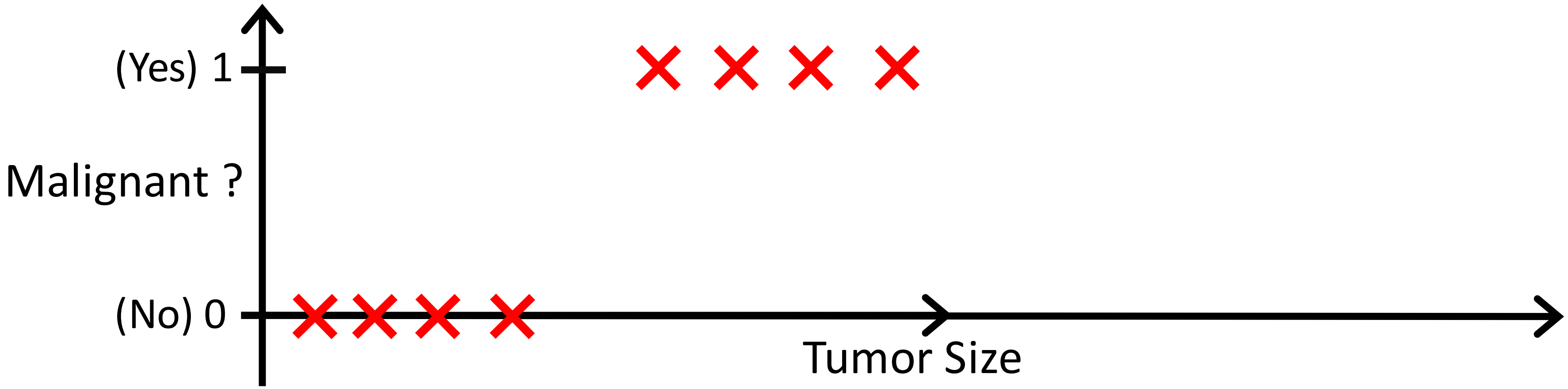
Online Transactions: Fraudulent / Genuine?

Tumor: Malignant / Benign ?

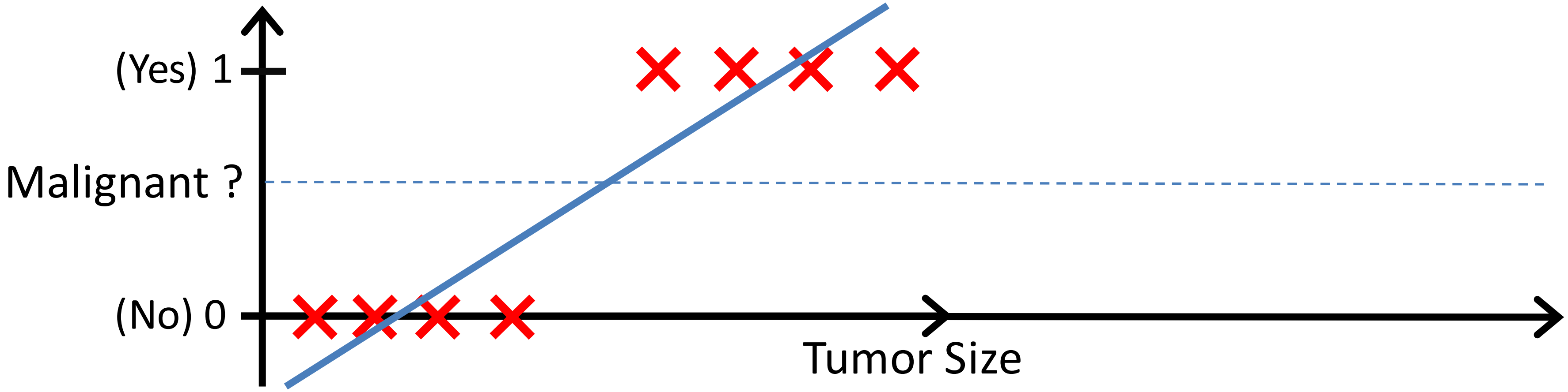
$$y \in \{0, 1\}$$

0: “Negative Class” (e.g., benign tumor)

1: “Positive Class” (e.g., malignant tumor)



Can we solve the problem using linear regression?

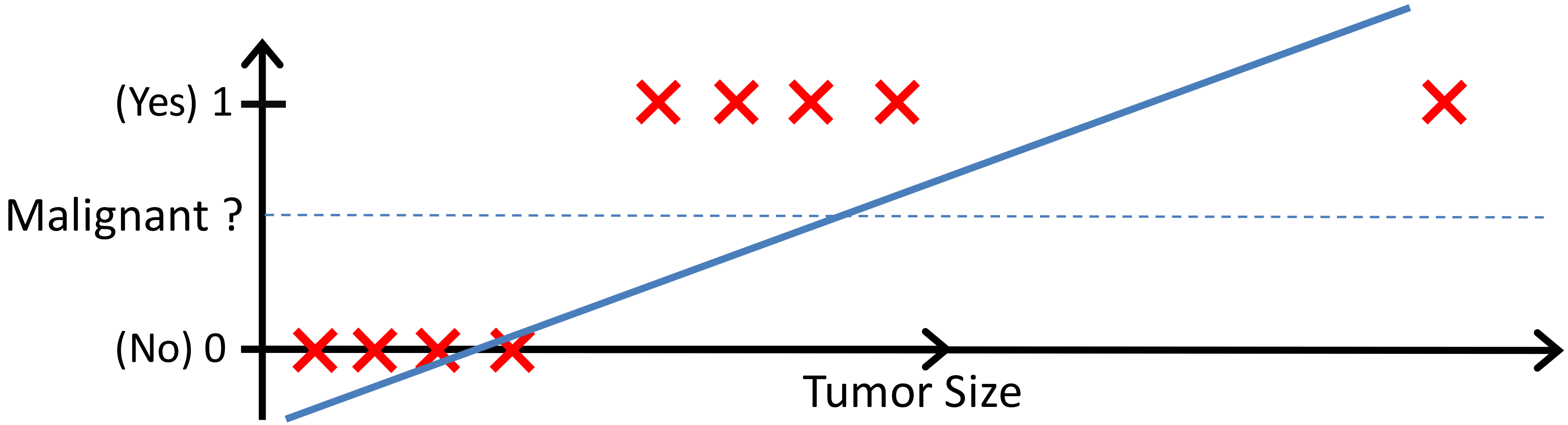


Can we solve the problem using linear regression? E.g., fit a straight line and define a threshold at 0.5

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"



Can we solve the problem using linear regression? E.g., fit a straight line and define a threshold at 0.5

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"

Failure due to adding a new point

Classification: y = 0 or 1

Another drawback of using linear regression for this problem

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

What we need:

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

Logistic Regression Model

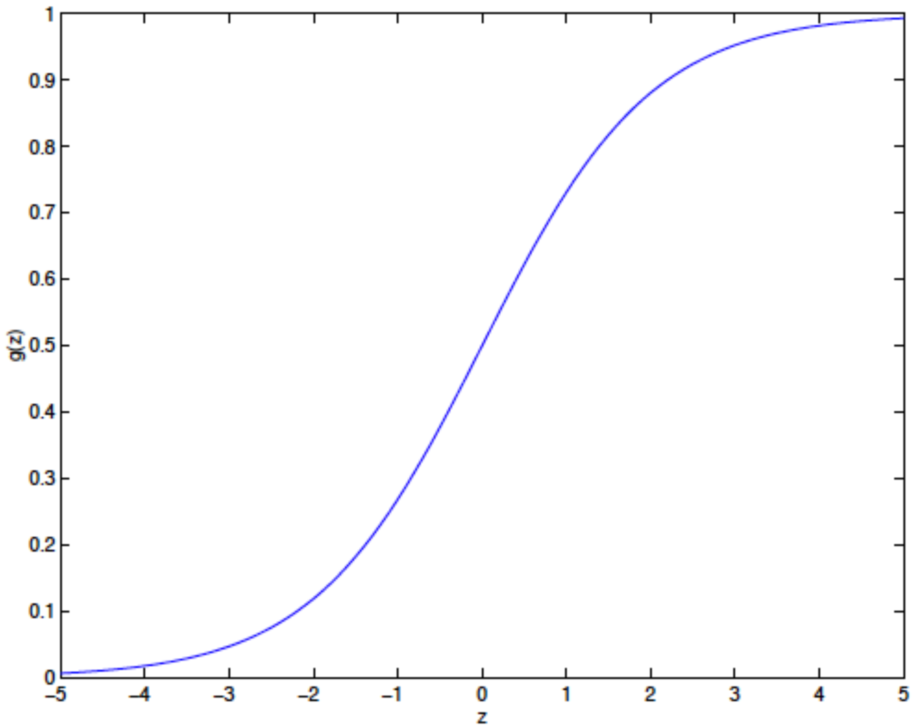
Want $0 \leq h_\theta(x) \leq 1$

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

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

Sigmoid function

Logistic function



A useful property: easy to compute differential at any point

Interpretation of Hypothesis Output

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

"probability that y = 1, given x,
parameterized by θ"

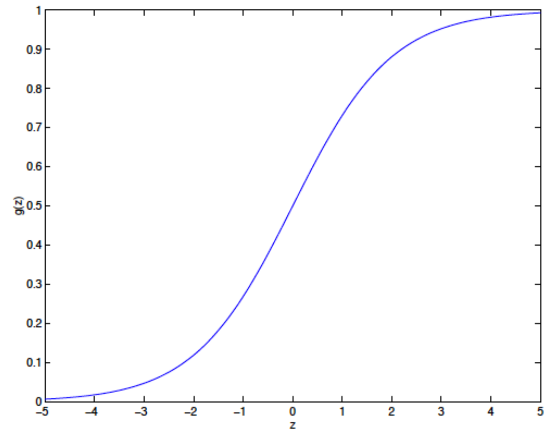
Example: If $h_\theta(x) = 0.7$

Tell patient that 70% chance of tumor being malignant

Logistic regression

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

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



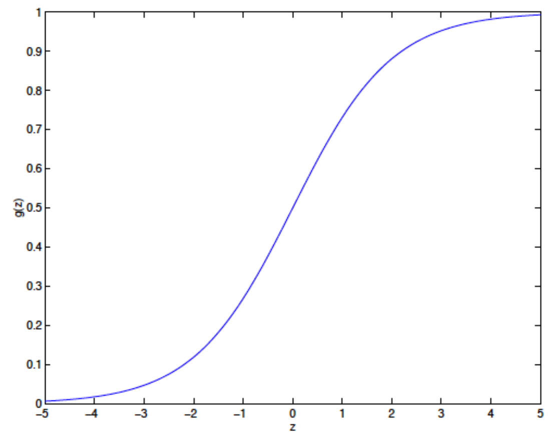
Suppose predict "$y = 1$" if $h_\theta(x) \geq 0.5$

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

Logistic regression

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

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



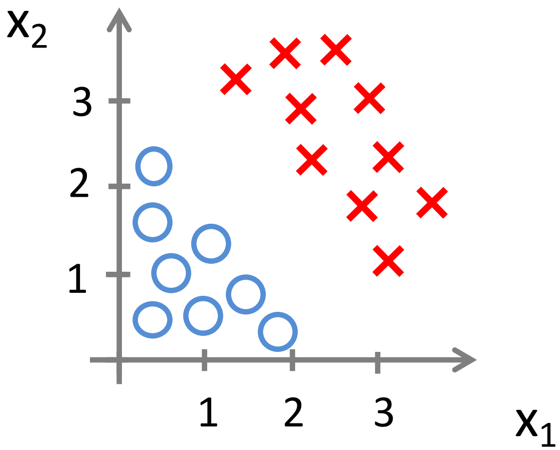
Suppose predict "$y = 1$" if $h_\theta(x) \geq 0.5$ When $\Theta^T x \geq 0$

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

Separating two classes of points

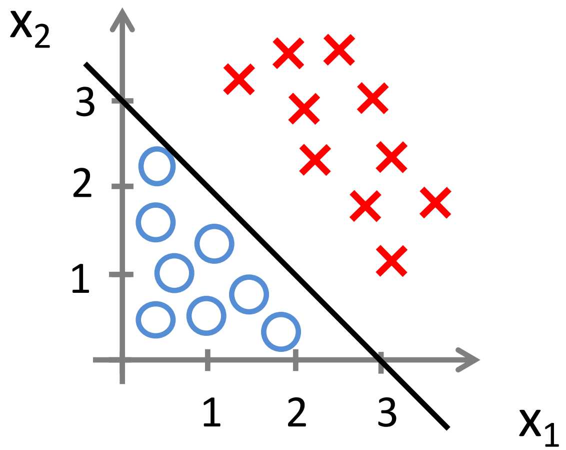
- We are attempting to separate two given sets / classes of points
- Separate two regions of the feature space
- Concept of Decision Boundary
- Finding a good decision boundary => learn appropriate values for the parameters Θ

Decision Boundary



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

Decision Boundary

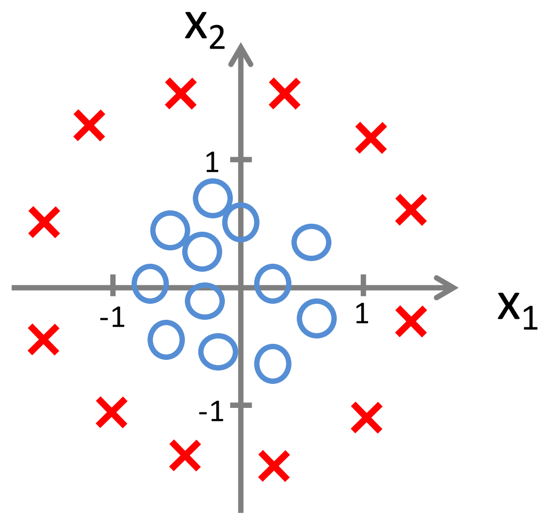


$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

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

How to get the parameter values – will be discussed soon

Non-linear decision boundaries

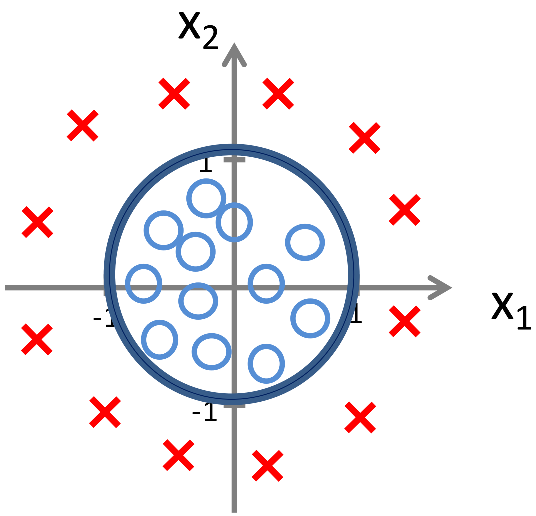


$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

We can learn more complex decision boundaries where the hypothesis function contains higher order terms.

(remember polynomial regression)

Non-linear decision boundaries



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

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

How to get the parameter values – will be discussed soon

Cost function for Logistic Regression

How to get the parameter values?

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

m examples $x \in \begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_n \end{bmatrix}$ $x_0 = 1, y \in \{0, 1\}$

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

How to choose parameters θ ?

Cost function

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

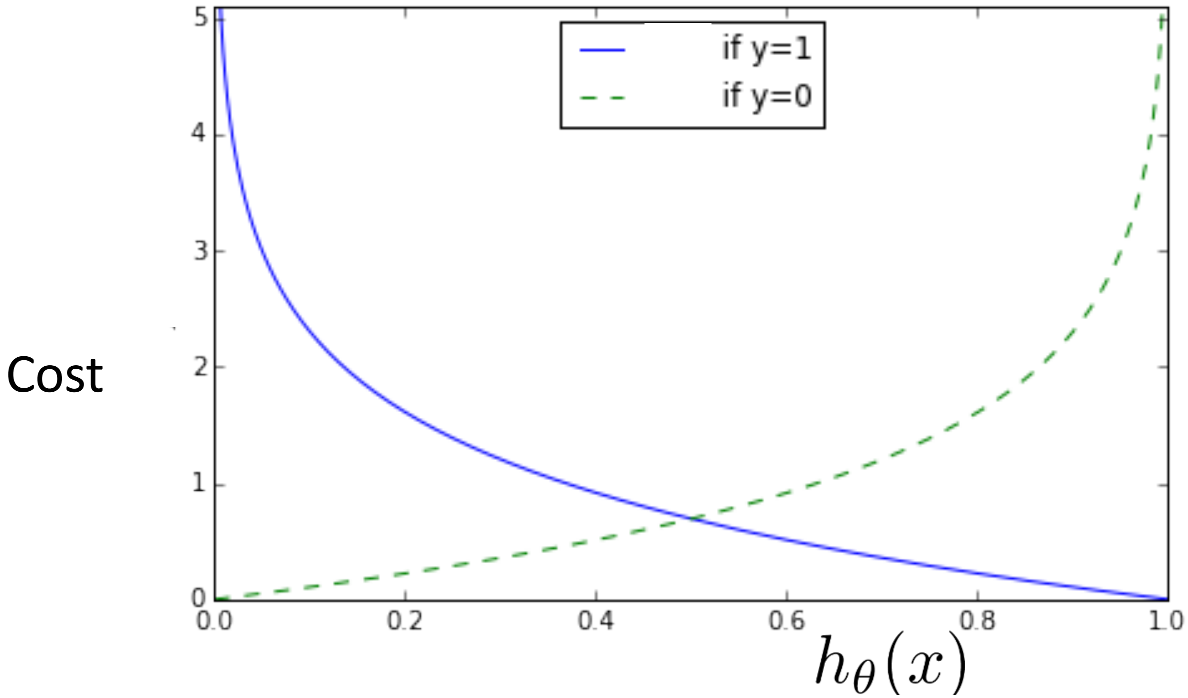
Squared error cost function:

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

However this cost function is non-convex for the hypothesis of logistic regression.

Logistic regression cost function

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



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 $h_\theta(x) \rightarrow 0$
$Cost \rightarrow \infty$

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

Logistic regression cost function

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

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

Note: $y = 0$ or 1 always

Logistic regression cost function

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

This cost function is convex

To fit parameters θ:

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

Gradient Descent

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

Want $\min_\theta J(\theta)$:

Repeat $\{$

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

(simultaneously update all θ_j)

$\}$

Gradient Descent

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

Want $\min_\theta J(\theta)$:

Repeat $\{$

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

(simultaneously update all θ_j)

$\}$

Algorithm looks identical to linear regression, but the hypothesis function is different for logistic regression.

Thus we can gradient descent to learn parameter values, and hence compute for a new input:

To make a prediction given new x :

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

= estimated probability that y = 1 on input x

How to use the estimated probability?

- Refraining from classifying unless confident
- Ranking items
- Multi-class classification

Multi-class classification:
one vs. all

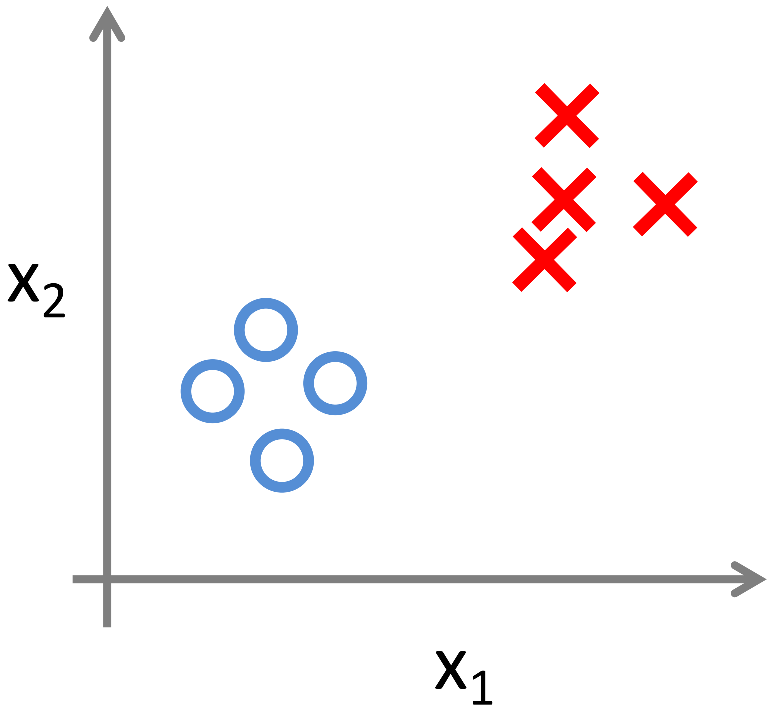
Multiclass classification

News article tagging: Politics, Sports, Movies, Religion, …

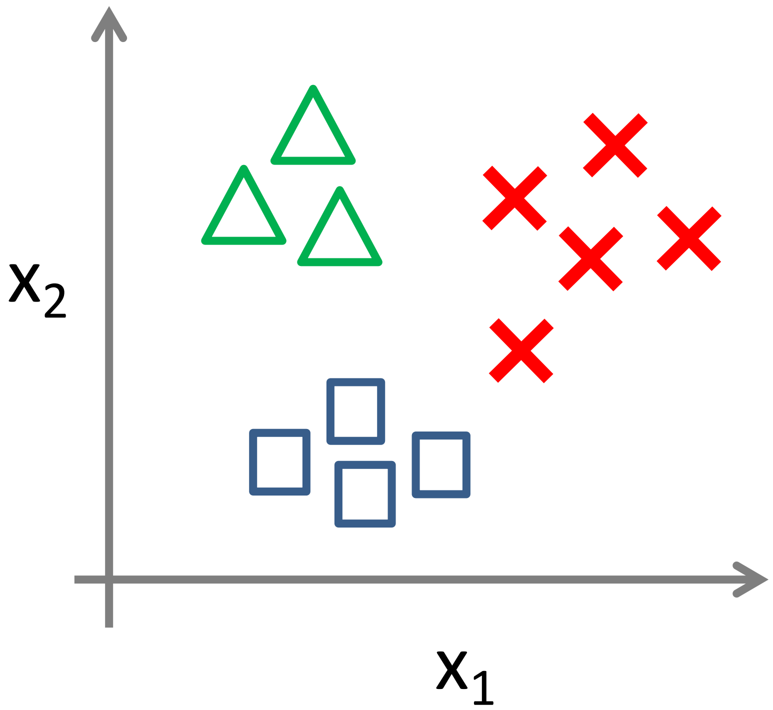
Medical diagnosis: Not ill, Cold, Flu, Fever

Weather: Sunny, Cloudy, Rain, Snow

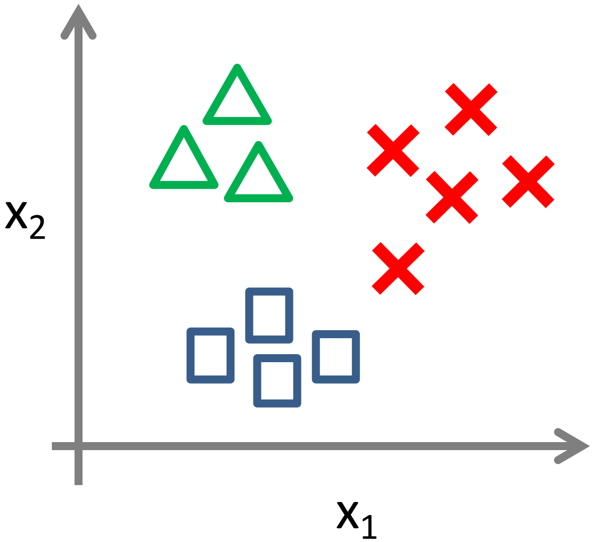
Binary classification:

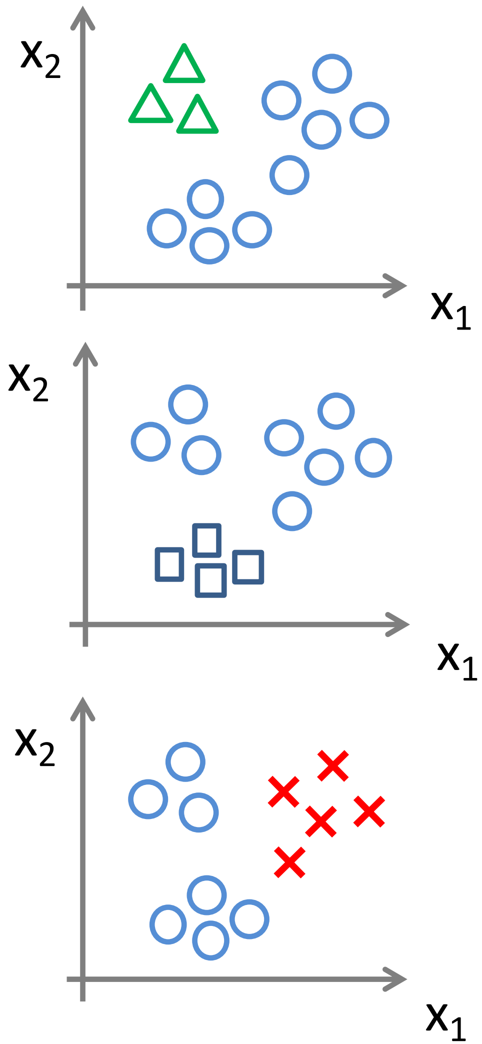


Multi-class classification:



One-vs-all (one-vs-rest):





Class 1: △
Class 2: □
Class 3: ✕

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

One-vs-all

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 a prediction, pick the class i that maximizes

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

Advanced Optimization algorithms (not part of this course)

Optimization algorithms:

- Gradient descent
- Conjugate gradient
- BFGS
- L-BFGS

Advantages of the other algorithms:

- No need to manually pick learning rate
- Often converges faster than gradient descent

Disadvantages:

- More complex