Logistic regression

Classification

Fundamentals of Machine Learning

Classification

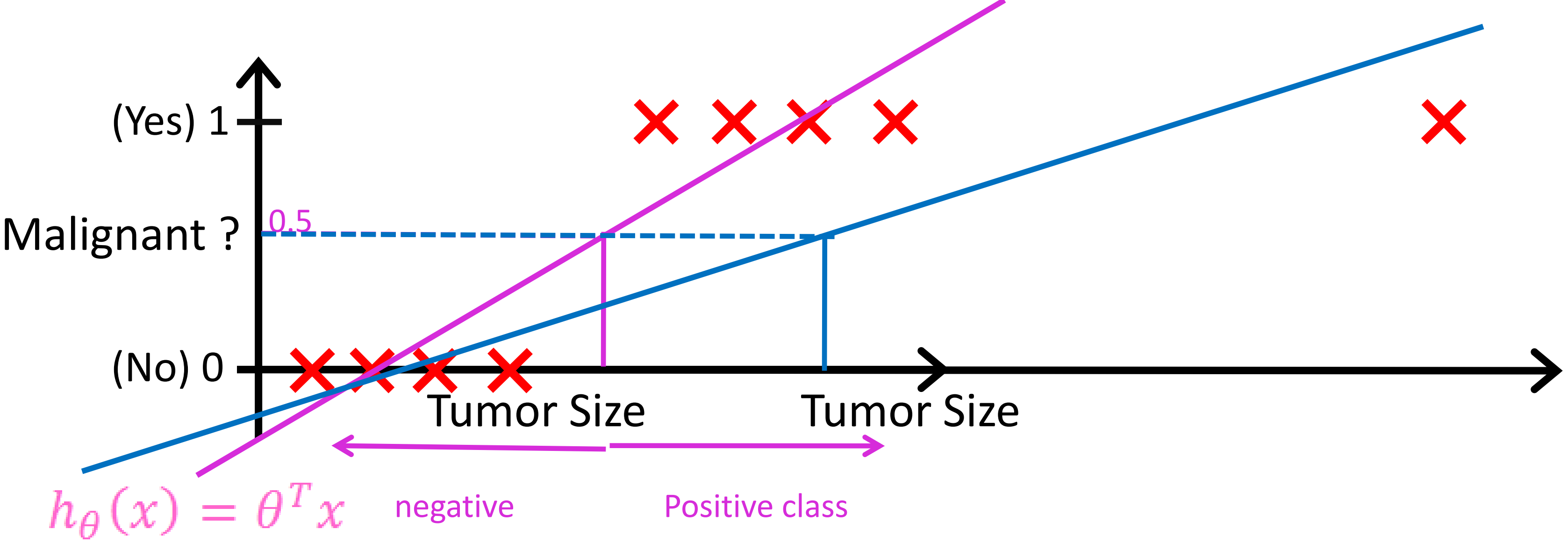
Binary classification problem

Email: Spam / Not Spam?
Online Transactions: Fraudulent (Yes / No)?
Tumor: Malignant / Benign ?

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

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

$y \in \{0,1,2,3\}$ Multi-class classification problem



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"

Classification: y = 0 or 1

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

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

Which of the following statements is true?

- If linear regression doesn't work on a classification task as in the previous example shown in the video, applying feature scaling may help.
- If the training set satisfies $0 \leq y^{(i)} \leq 1$ for every training example $(x^{(i)}, y^{(i)})$, then linear regression's prediction will also satisfy $0 \leq h_\theta(x) \leq 1$ for all values of x.
- If there is a feature x that perfectly predicts y, i.e. if $y = 1$ when $x \geq c$ and $y = 0$ whenever $x < c$ (for some constant c), then linear regression will obtain zero classification error.
- None of the above statements are true.

Logistic regression

Hypothesis Representation

Fundamentals of Machine Learning

Logistic Regression Model

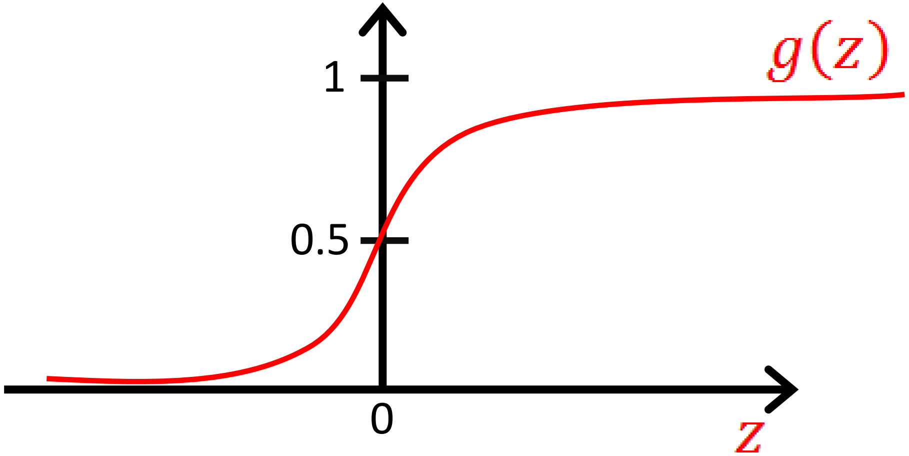
Want $0 \leq h_\theta(x) \leq 1$

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

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

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

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



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

$$h_\theta(x) = p(y = 1|x\,;\ \theta)$$

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

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

y = 0 or 1

Andrew Ng

Suppose we want to predict, from data x about a tumor, whether it is malignant ($y = 1$) or benign ($y = 0$). Our logistic regression classifier outputs, for a specific tumor, $h_\theta(x) = P(y = 1|x;\theta) = 0.7$, so we estimate that there is a 70% chance of this tumor being malignant. What should be our estimate for $P(y = 0|x;\theta)$, the probability the tumor is benign?

- ○ $P(y = 0|x;\theta) = 0.3$
- ○ $P(y = 0|x;\theta) = 0.7$
- ○ $P(y = 0|x;\theta) = 0.7^2$
- ○ $P(y = 0|x;\theta) = 0.3 \times 0.7$

Logistic regression

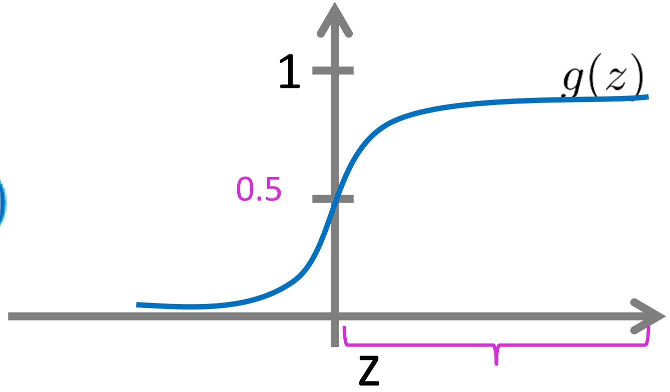
Decision boundary

Fundamentals of Machine Learning

Logistic regression

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

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



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

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

$\theta^T x \geq 0$

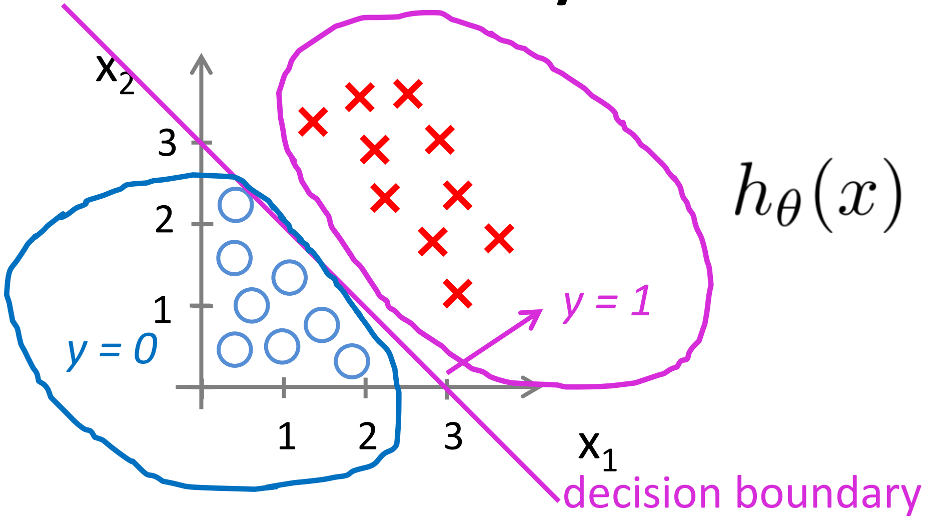
$h_\theta(x) = g(\theta^T x) \geq 0.5$

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

whenever $\theta^T x \geq 0$

$\theta^T x < 0$

Decision Boundary



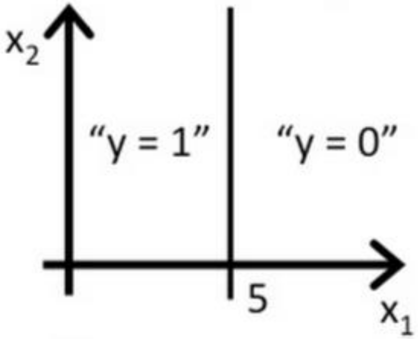
$$h_\theta(x) = g(\underset{=-3}{\theta_0} + \underset{=1}{\theta_1} x_1 + \underset{=1}{\theta_2} x_2)$$

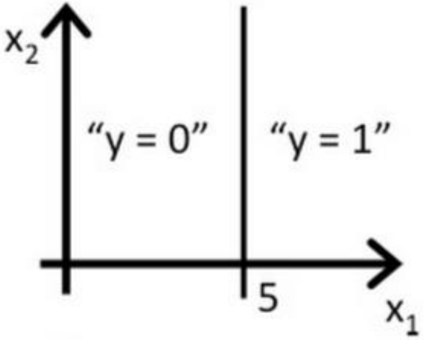
Predict "$y = 1$" if $\underbrace{-3 + x_1 + x_2}_{\theta^T x \geq 0} \geq 0$

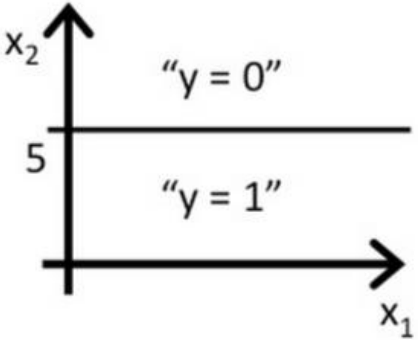
$x_1 + x_2 = 3$

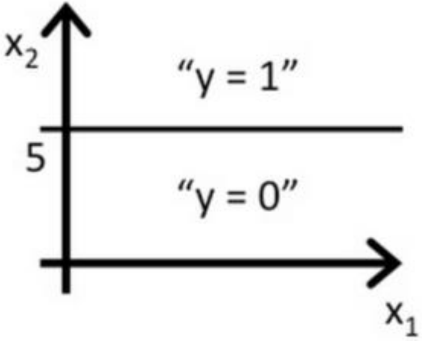
Andrew Ng

Consider logistic regression with two features x_1 and x_2. Suppose $\theta_0 = 5$, $\theta_1 = -1$, $\theta_2 = 0$, so t hat $h_\theta(x) = g(5 - x_1)$. Which of these shows the decision boundary of $h_\theta(x)$?

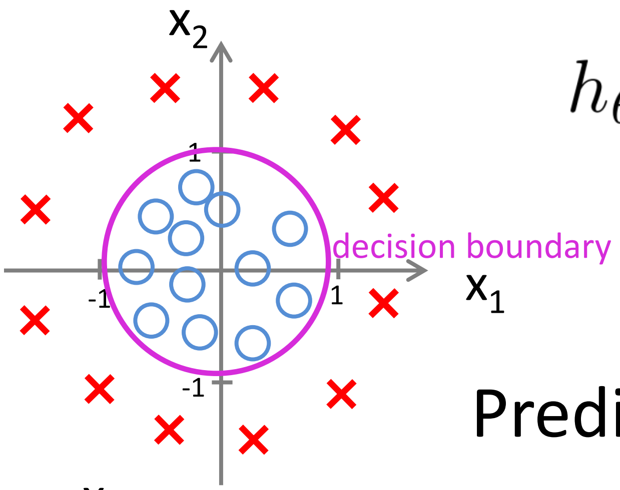








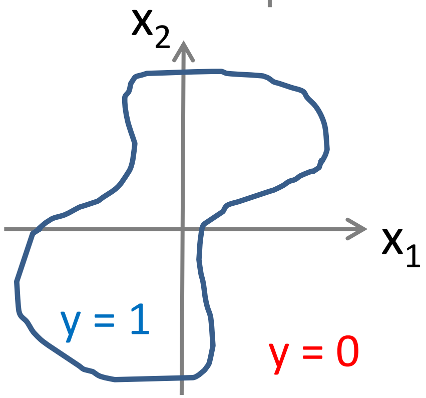
Non-linear decision boundaries



$$\theta = \begin{bmatrix} -1 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

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

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



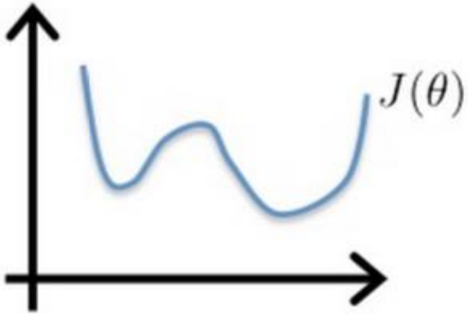
$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^2 x_2^2 + \theta_6 x_1^3 x_2 + \dots)$$

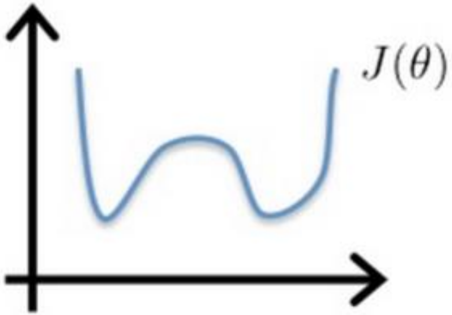
Logistic regression

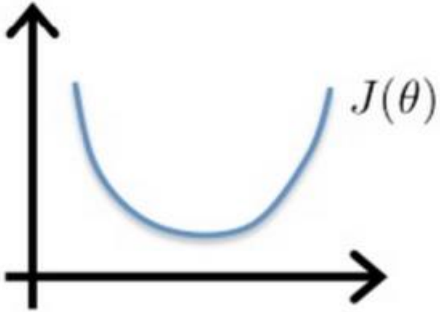
Cost function

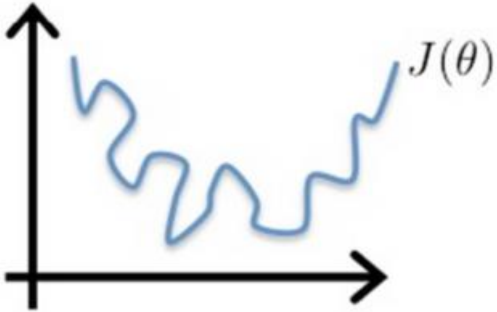
Fundamentals of Machine Learning

Consider minimizing a cost function $J(\theta)$. Which one of these functions is convex?









Andrew Ng

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

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

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

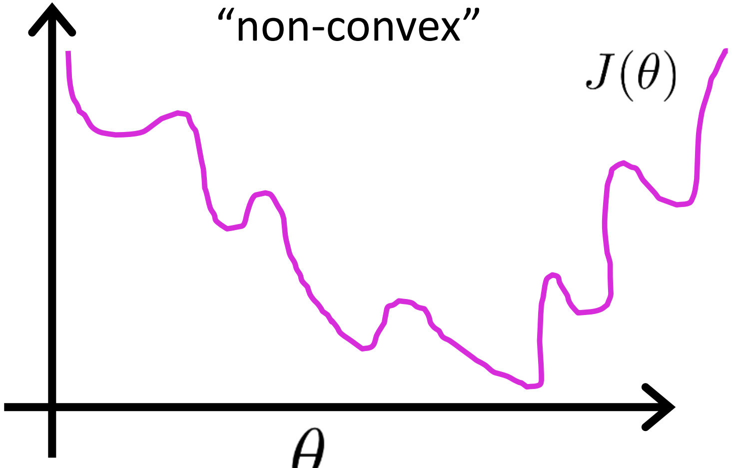
How to choose parameters θ ?

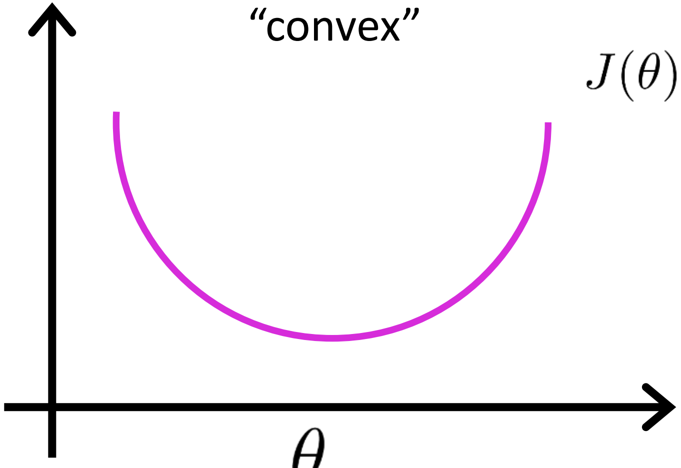
Cost function

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

$\mathrm{cost}(h_\theta(x^{(i)}), y)$

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

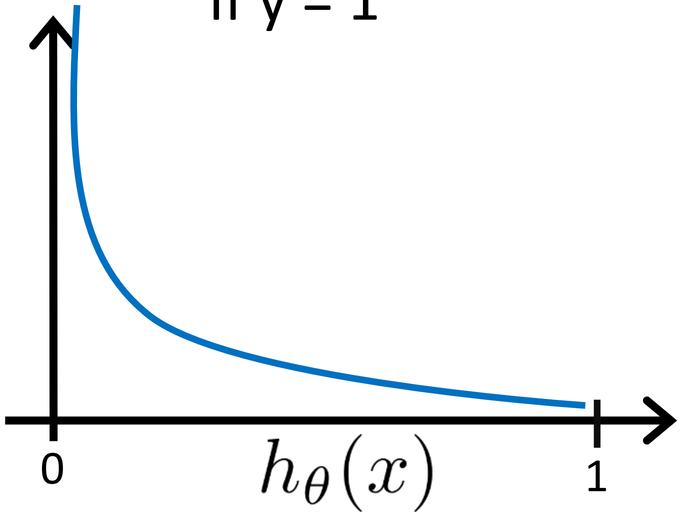




Logistic regression cost function

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

If y = 1



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

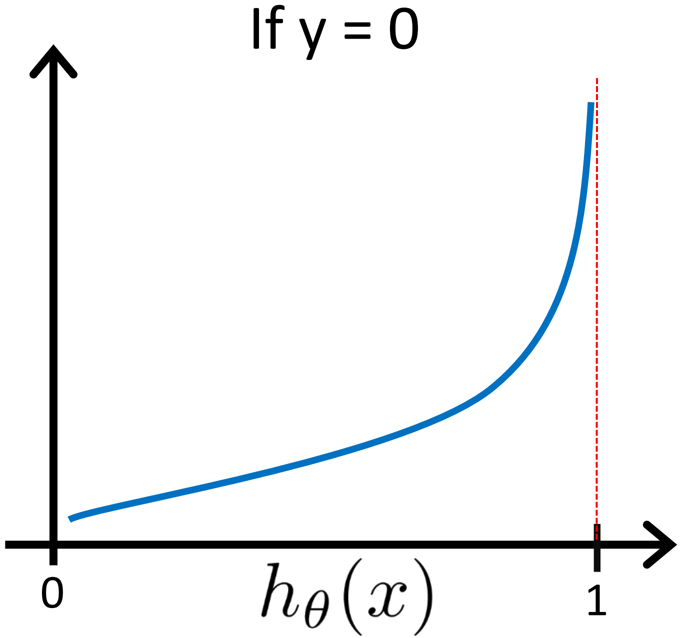
But as $h_\theta(x) \to 0$

$Cost \to \infty$

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

Logistic regression cost function

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



In logistic regression, the cost function for our hypothesis outputting (predicting) $h_\theta(x)$ on a training example that has label $y \in \{0, 1\}$ is:

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

Which of the following are true? Check all that apply.

- ☐ If $h_\theta(x) = y$, then $\mathrm{cost}(h_\theta(x), y) = 0$ (for $y = 0$ and $y = 1$).
- ☐ If $y = 0$, then $\mathrm{cost}(h_\theta(x), y) \rightarrow \infty$ as $h_\theta(x) \rightarrow 1$.
- ☐ If $y = 0$, then $\mathrm{cost}(h_\theta(x), y) \rightarrow \infty$ as $h_\theta(x) \rightarrow 0$.
- ☐ Regardless of whether $y = 0$ or $y = 1$, if $h_\theta(x) = 0.5$, then $\mathrm{cost}(h_\theta(x), y) > 0$.

Logistic regression

Simplified cost function and gradient descent

Fundamentals of Machine Learning

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

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

Logistic regression cost function

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

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

To fit parameters θ:

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

To make a prediction given new x:

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

$p(y = 1|x;\theta)$

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)

$\}$

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

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 $\{$

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

(simultaneously update all θ_j)

$\}$

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

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

Algorithm looks identical to linear regression!

Suppose you are running gradient descent to fit a logistic regression model with parameter $\theta \in \mathbb{R}^{n+1}$. Which of the following is a reasonable way to make sure the learning rate α is set properly and that gradient descent is running correctly?

- Plot $J(\theta) = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$ as a function of the number of iterations (i.e. the horizontal axis is the iteration number) and make sure $J(\theta)$ is decreasing on every iteration.
- Plot $J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^{(i)}\log h_\theta(x^{(i)}) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))]$ as a function of the number of iterations and make sure $J(\theta)$ is decreasing on every iteration.
- Plot $J(\theta)$ as a function of θ and make sure it is decreasing on every iteration.
- Plot $J(\theta)$ as a function of θ and make sure it is convex.

Andrew Ng

One iteration of gradient descent simultaneously performs these updates:

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_0^{(i)}$$
$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_1^{(i)}$$
$$\vdots$$
$$\theta_n := \theta_n - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_n^{(i)}$$

We would like a vectorized implementation of the form $\theta := \theta - \alpha\delta$ (for some vector $\delta \in \mathbb{R}^{n+1}$). What should the vectorized implementation be?

- $\theta := \theta - \alpha \frac{1}{m} \sum_{i=1}^{m} [(h_\theta(x^{(i)}) - y^{(i)}) \cdot x^{(i)}]$
- $\theta := \theta - \alpha \frac{1}{m} [\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})] \cdot x^{(i)}$
- $\theta := \theta - \alpha \frac{1}{m} x^{(i)} [\sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})]$
- All of the above are correct implementations.

Logistic regression

Advanced optimization

Fundamentals of Machine Learning

Optimization algorithm

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

Given θ, we have code that can compute

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

Gradient descent:

Repeat $\{$

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

$\}$

Optimization algorithm

Given θ, we have code that can compute

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

Optimization algorithms:

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

Line search algorithm –
Automatically picks a good learning

Advantages:

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

Disadvantages:

- More complex

Example:

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

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

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

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

```
function [jVal, gradient]
            = costFunction(theta)
  jVal = (theta(1)-5)^2 + ...
            (theta(2)-5)^2;
  gradient = zeros(2,1);
  gradient(1) = 2*(theta(1)-5);
  gradient(2) = 2*(theta(2)-5);
```

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

theta = $\begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix}$

function [jVal, gradient] = costFunction(theta)

jVal = [code to compute $J(\theta)$**];**

gradient(1) = [code to compute $\frac{\partial}{\partial \theta_0} J(\theta)$**];**

gradient(2) = [code to compute $\frac{\partial}{\partial \theta_1} J(\theta)$**];**

$\vdots$

gradient(n+1) = [code to compute $\frac{\partial}{\partial \theta_n} J(\theta)$ **];**

Suppose you want to use an advanced optimization algorithm to minimize the cost function for logistic regression with parameters θ_0 and θ_1. You write the following code:

```
function [jVal, gradient] = costFunction(theta)
  jVal = % code to compute J(theta)
  gradient(1) = CODE#1 % derivative for theta_0
  gradient(2) = CODE#2 % derivative for theta_1
```

What should CODE#1 and CODE#2 above compute?

- CODE#1 and CODE#2 should compute $J(\theta)$.
- CODE#1 should be `theta(1)` and CODE#2 should be `theta(2)`.
- CODE#1 should compute $\frac{1}{m}\sum_{i=1}^{m}[(h_\theta(x^{(i)}) - y^{(i)}) \cdot x_0^{(i)}] (= \frac{\partial}{\partial\theta_0}J(\theta))$, and CODE#2 should compute $\frac{1}{m}\sum_{i=1}^{m}[(h_\theta(x^{(i)}) - y^{(i)}) \cdot x_1^{(i)}] (= \frac{\partial}{\partial\theta_1}J(\theta))$.
- None of the above.

Logistic regression

Multi-class classification: One-vs-all

Fundamentals of Machine Learning

Multiclass classification

Email foldering/tagging: Work, Friends, Family, Hobby

$y=1$, $y=2$, $y=3$, $y=4$

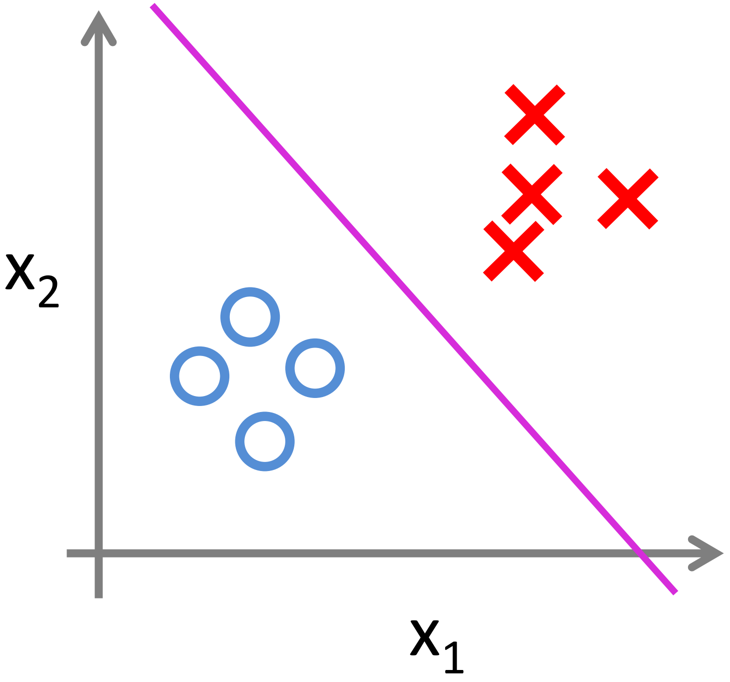
Medical diagrams: Not ill, Cold, Flu

$y=1$, $y=2$, $y=3$

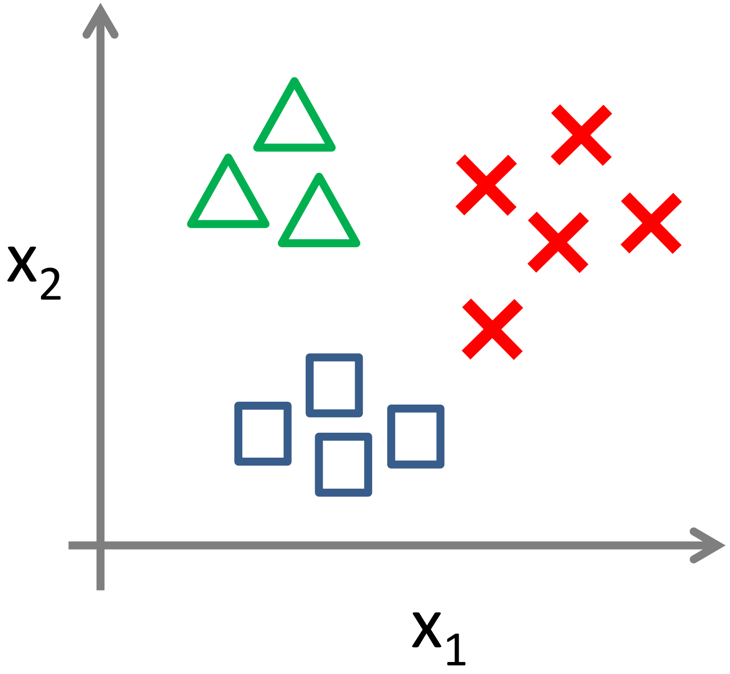
Weather: Sunny, Cloudy, Rain, Snow

$y=1$, $y=2$, $y=3$, $y=4$

Binary classification:

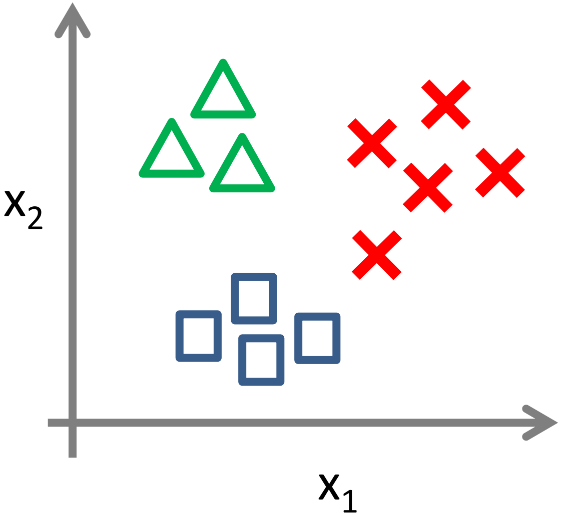


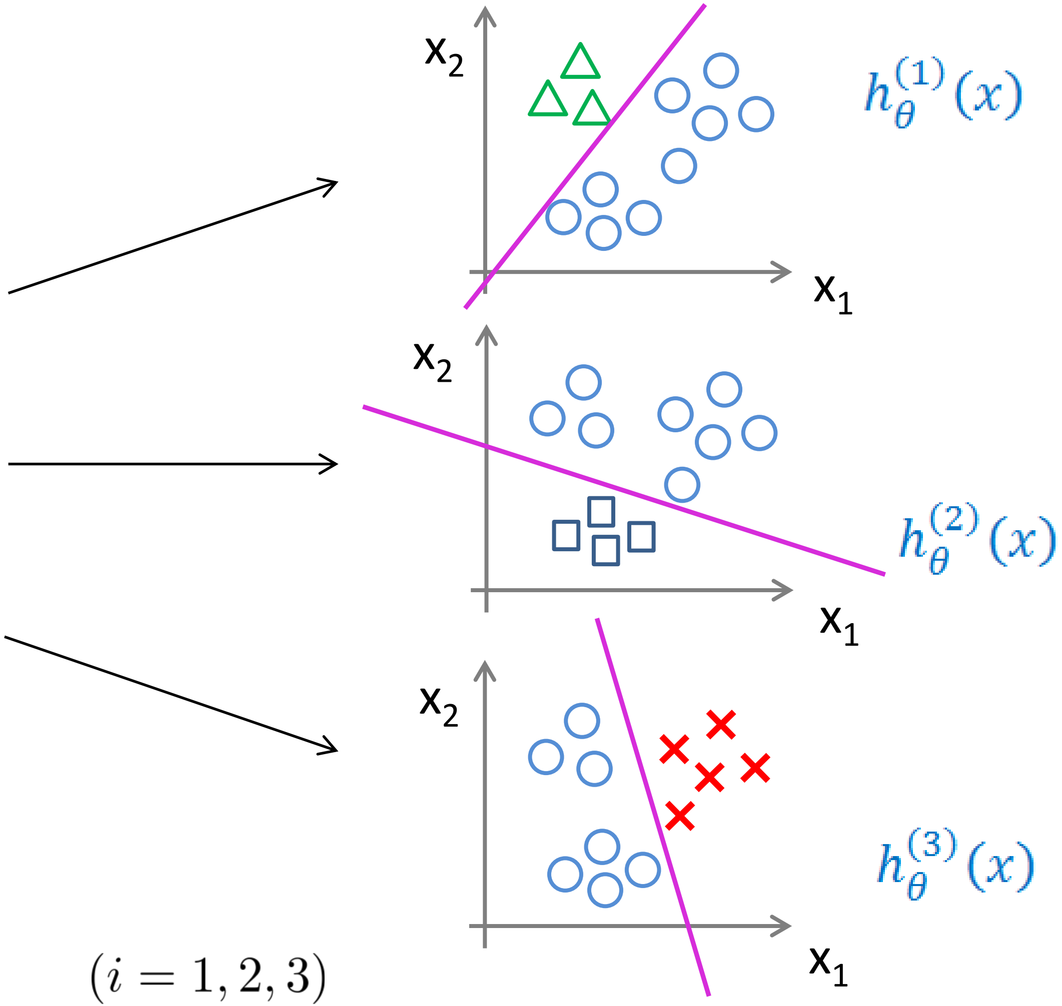
Multi-class classification:



Andrew Ng

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

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

Suppose you have a multi-class classification problem with k classes (so $y \in \{1, 2, \ldots, k\}$). Using the 1-vs.-all method, how many different logistic regression classifiers will you end up training?

- ○ $k - 1$
- ○ k
- ○ $k + 1$
- ○ Approximately $\log_2(k)$