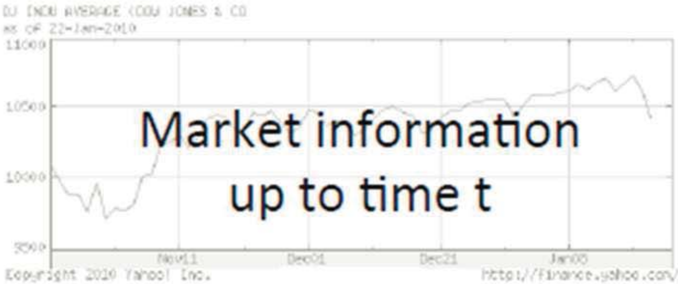
Classification and logistic regression

Supervised Learning

- **Regression**



Share Price
"$ 24.50"

Continuous Labels
Regression

- **Classification**

Feature Space $\mathcal{X}$

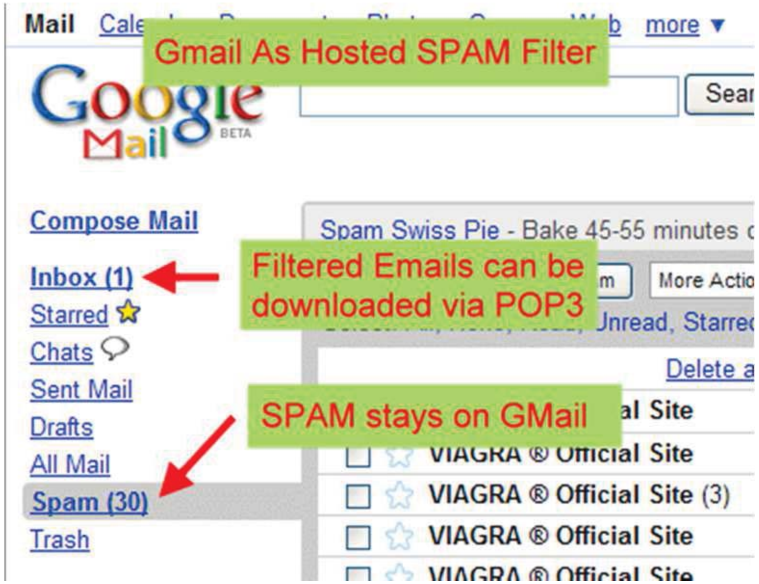
Label Space $\mathcal{Y}$:

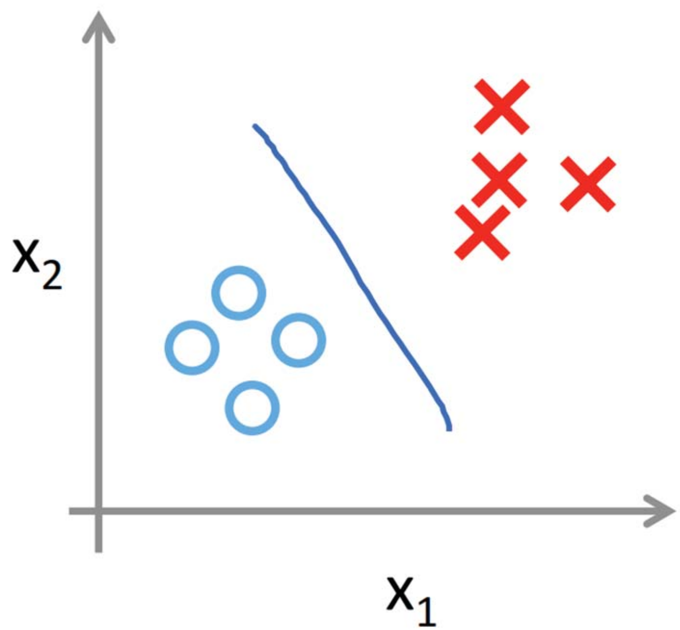
"Sports"
"News"
"Science"
...

Discrete Labels
Classification

Introduction

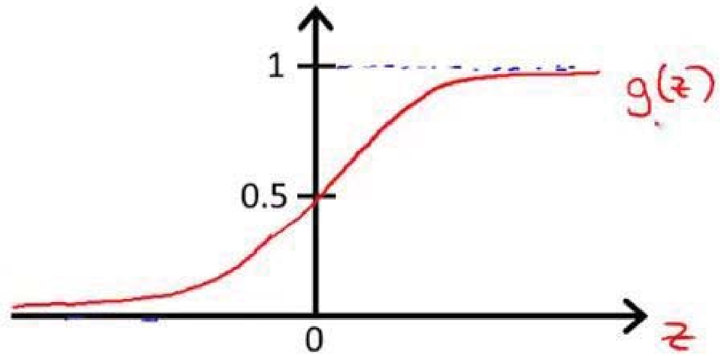
- Logistic Regression is a classification model, although it is called “regression”
- It is a binary classification model
- It is a linear classification model





The logistic function

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



Model Description

- **Hypothesis**

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

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

- **Compact Form**

$$p(y \mid x;\theta) = (h_\theta(x))^y \left(1 - h_\theta(x)\right)^{1-y}$$

- **Parameters** θ

Maximum Likelihood Estimation

- **(Conditional) 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**

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

Also known as the Cross-Entropy cost function

Unconstraint Optimization Methods

- **Problem**

$$\arg\max_{\theta} \sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))$$

where $h(x) = \frac{1}{1+\exp -\theta^T x}$.

- **Optimization Methods**
  - **Gradient Descent/Ascent**
  - **Stochastic Gradient Descent/Ascent**
  - **Newton Method**
  - **Quasi-Newton Method**
  - **Conjugate Gradient**
  - **…**

Gradient Ascent

- Property of sigmoid function:

$$
\begin{aligned}
g'(z) &= \frac{d}{dz}\frac{1}{1+e^{-z}} \\
&= \frac{1}{(1+e^{-z})^2}\left(e^{-z}\right) \\
&= \frac{1}{(1+e^{-z})}\cdot\left(1-\frac{1}{(1+e^{-z})}\right) \\
&= g(z)(1-g(z)).
\end{aligned}
$$

Gradient Ascent

■ Gradient

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

Error × Feature

■ Batch Gradient Ascent Method

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

Stochastic Gradient Ascent

- **Randomly choose a training sample** (x, y)
- **Compute gradient**

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

- **Updating weights**

$$\theta_j := \theta_j + \alpha\,(y - h_\theta(x))\, x_j$$

- **Repeat…**

Gradient Ascent -- batch updating

Stochastic Gradient Ascent -- online updating

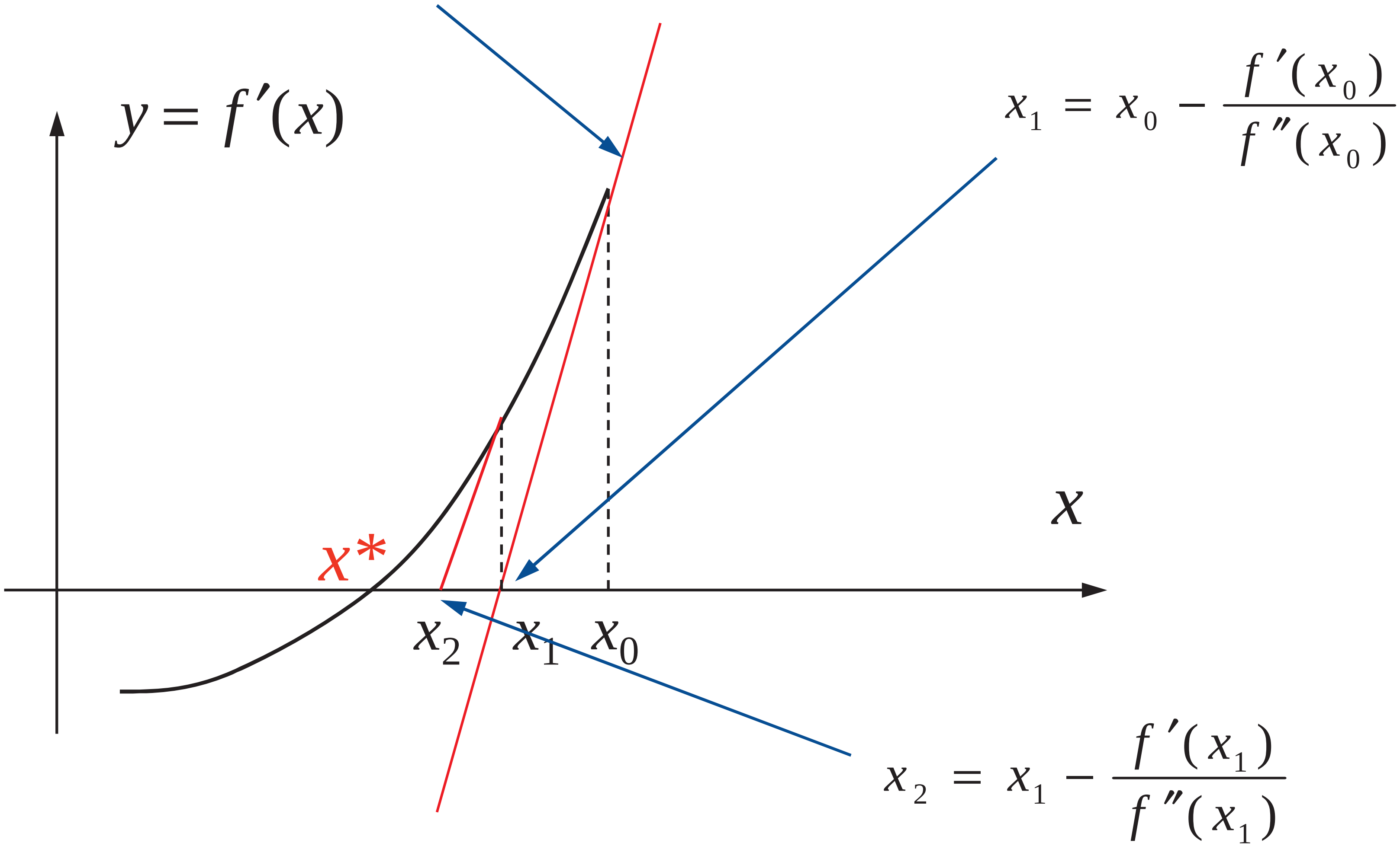
The Newton's method

- Finding a zero of a function

$$\theta^{t+1} := \theta^t - \frac{f(\theta^t)}{f'(\theta^t)}$$

Illustration of Newton's Method

tangent line: $y = f'(x_0) + f''(x_0)(x - x_0)$



Newton's Method

- **Problem**

$$\arg\min f(x) \iff \text{solve}: \nabla f(x) = 0$$

- **Second-order Taylor expansion**

$$\phi(x) = f(x^{(k)}) + \nabla f(x^{(k)})(x - x^{(k)}) + \frac{1}{2}\nabla^2 f(x^{(k)})(x - x^{(k)})^2 \approx f(x)$$

$$\Downarrow$$

$$\nabla\phi(x) = 0 \implies x = x^{(k)} - \nabla^2 f(x^{(k)})^{-1} \nabla f(x^{(k)})$$

- **Newton's method (also called Newton-Raphson method)**

$$x^{(k+1)} = x^{(k)} - \underbrace{\boxed{\nabla^2 f(x^{(k)})}}_{\text{Hessian Matrix}}{}^{-1} \nabla f(x^{(k)})$$

The Newton-Raphson method

- In LR the θ is vector-valued, thus we need the following generalization:

$$\theta := \theta - H^{-1}\nabla_\theta \ell(\theta)$$

Here, $\nabla_\theta \ell(\theta)$ is, as usual, the vector of partial derivatives of $\ell(\theta)$ with respect to the θ_i's; and H is an n-by-n matrix (actually, $n+1$-by-$n+1$, assuming that we include the intercept term) called the **Hessian**, whose entries are given by

$$H_{ij} = \frac{\partial^2 \ell(\theta)}{\partial \theta_i \partial \theta_j}.$$

Newton's Method for Logistic Regression

- **Problem**

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

- **Gradient and Hessian Matrix**

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

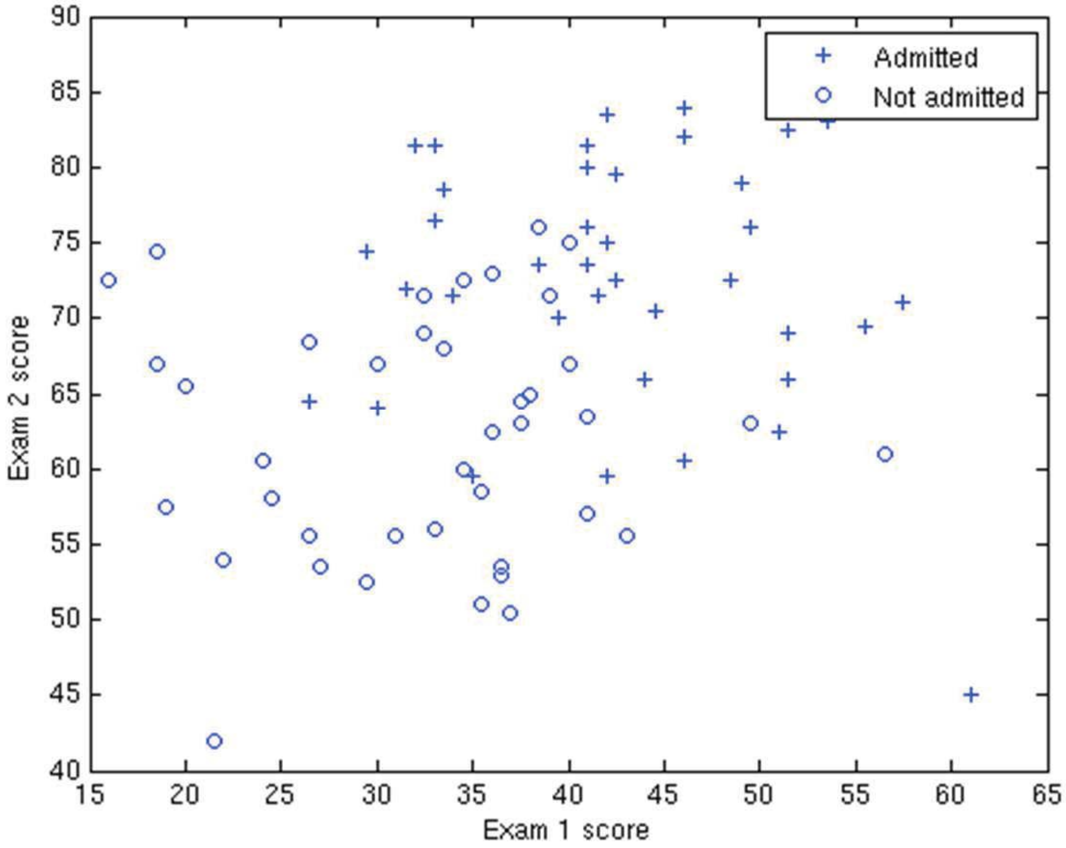
$$H = \frac{1}{m}\sum_{i=1}^{m} h_{\theta}(x^{(i)})(1 - h_{\theta}(x^{(i)}))x^{(i)}\left(x^{(i)}\right)^{T}$$

- **Weight updating using Newton's method**

$$\theta^{(t+1)} = \theta^{(t)} - H^{-1}\nabla J(\theta^{(t)})$$

An Example of Logistic Regression

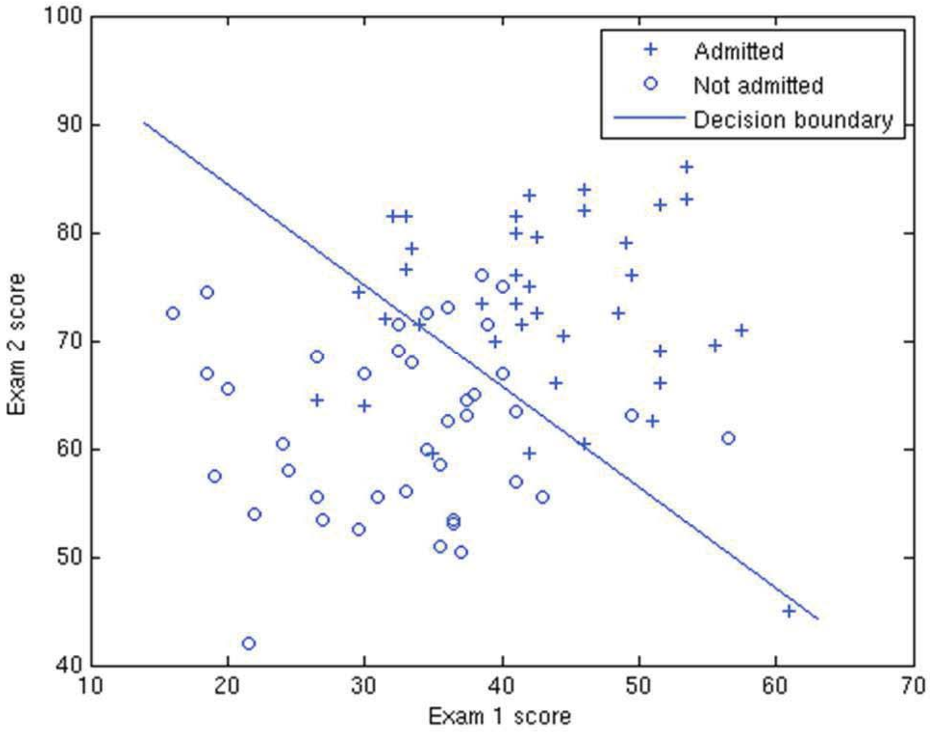
- **Training data set**



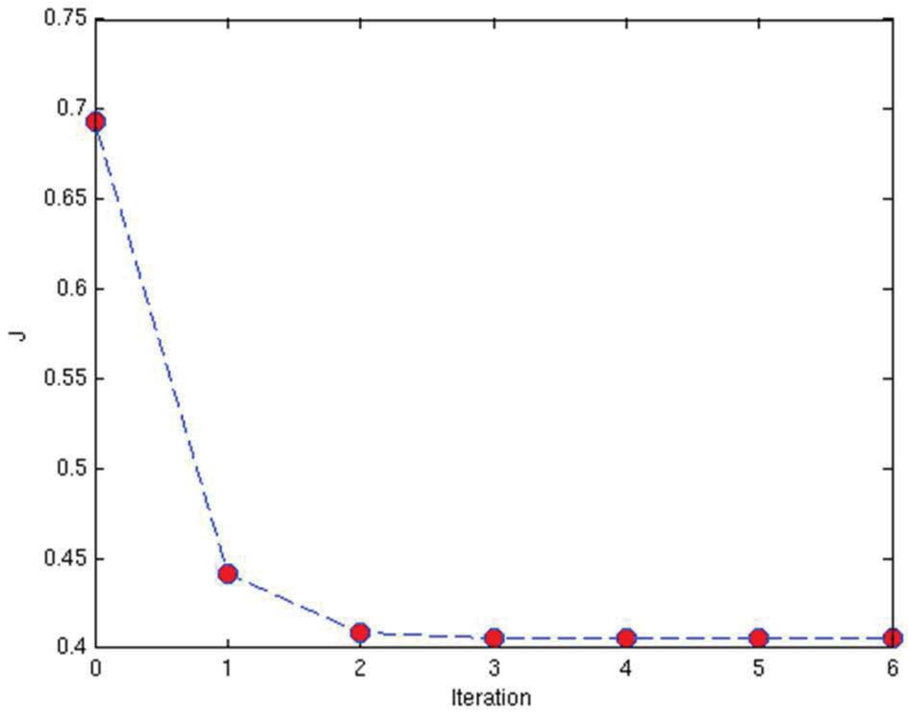
http://openclassroom.stanford.edu/MainFolder/DocumentPage.php?course=DeepLearning&doc=exercises/ex4/ex4.html

An Example (using Newton's method)

- **Hyper-plane**

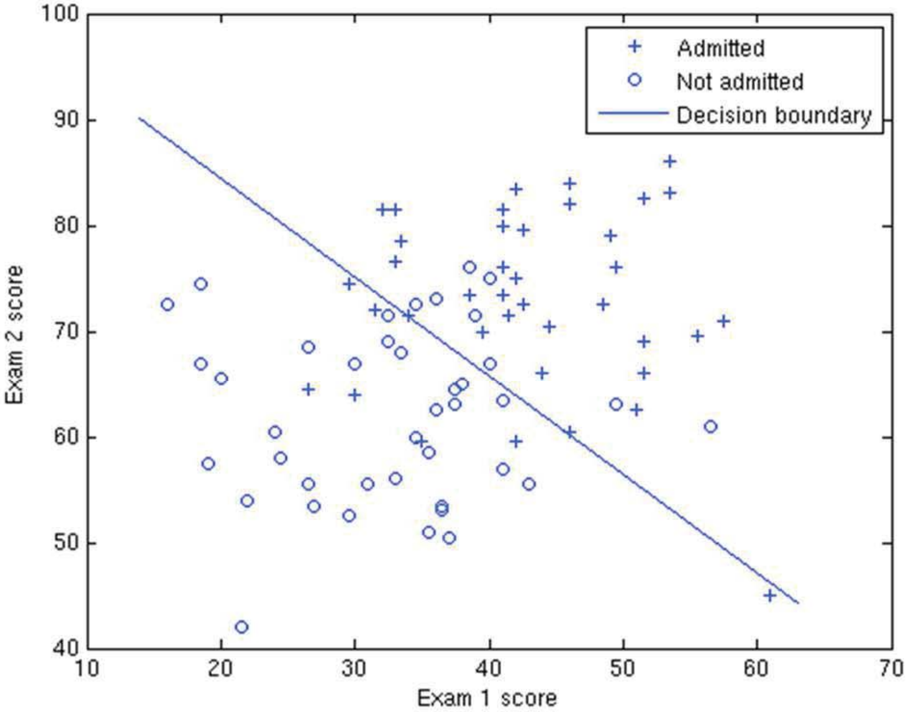


- **Cost functions**

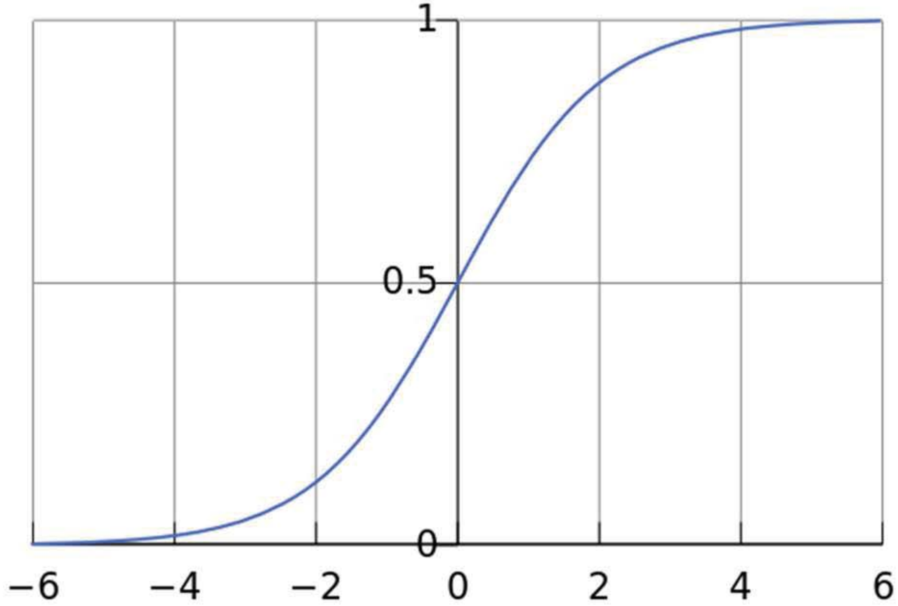


A Linear Classification Model

- **Logistic regression has a linear decision boundary (hyperplane)**



- **But with a nonlinear activation function (Sigmoid function) to model the posterior probability**



$$h(x) = \frac{1}{1+\exp -\theta^T x}$$