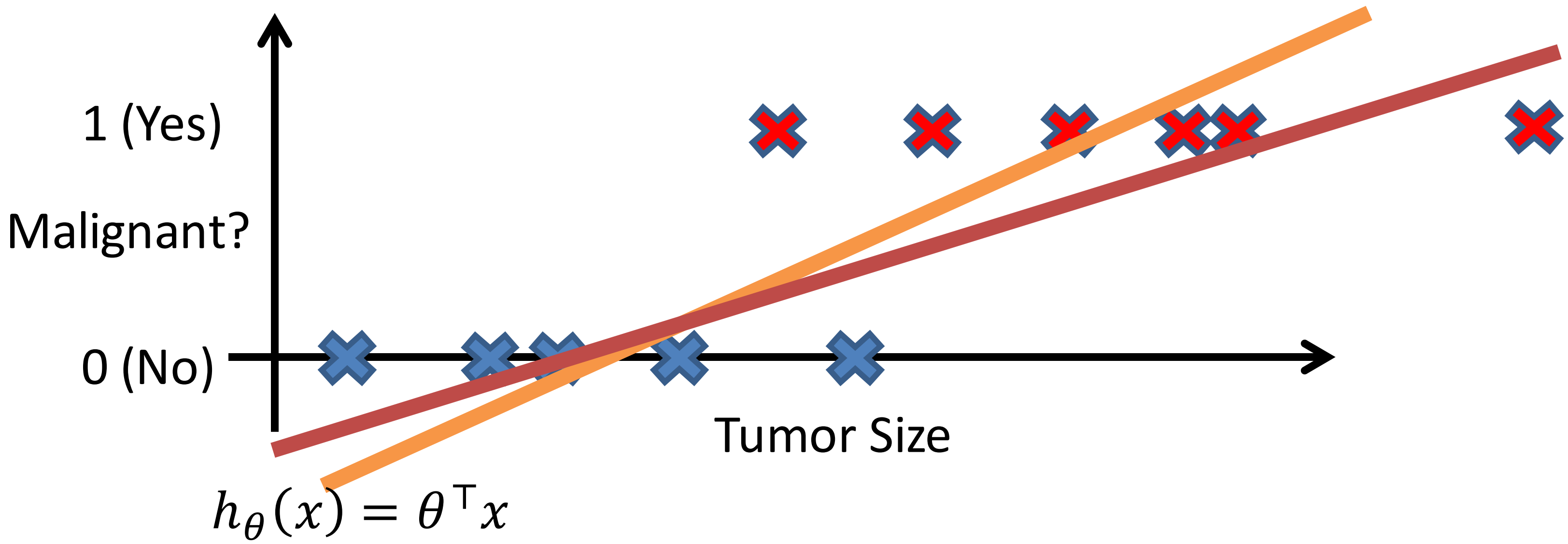
Logistic Regression

Logistic Regression

- Hypothesis representation
- Cost function
- Logistic regression with gradient descent
- Regularization
- Multi-class classification

Logistic Regression

- **Hypothesis representation**
- Cost function
- Logistic regression with gradient descent
- Regularization
- Multi-class classification



- 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 "$y = 0$"

Slide credit: Andrew Ng

Classification: $y = 1$ or $y = 0$

$h_\theta(x) = \theta^\top x$ (from linear regression)
can be > 1 or < 0

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

Logistic regression is actually for **classification**

Slide credit: Andrew Ng

Hypothesis representation

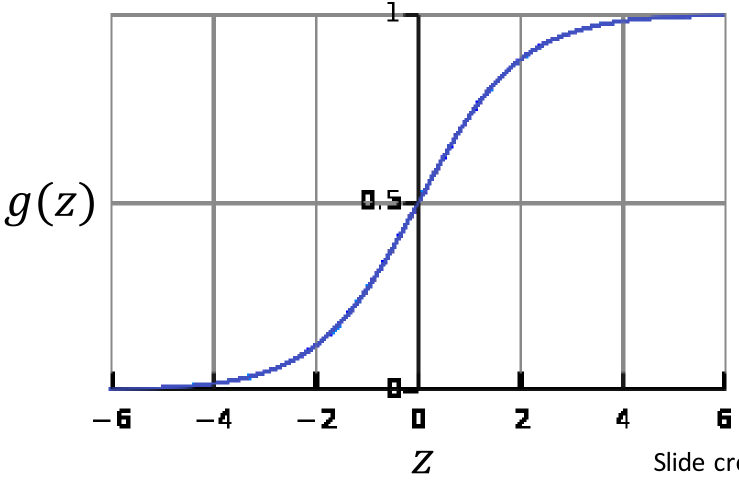
- Want $0 \leq h_\theta(x) \leq 1$

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

- $h_\theta(x) = g(\theta^\top x)$,

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

- Sigmoid function
- Logistic function



Slide credit: Andrew Ng

Interpretation of hypothesis output

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

- Example: If $x = \begin{bmatrix} x_0 \\ \mathrm{x}_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tumorSize} \end{bmatrix}$
- $h_\theta(x) = 0.7$

- Tell patient that 70% chance of tumor being malignant

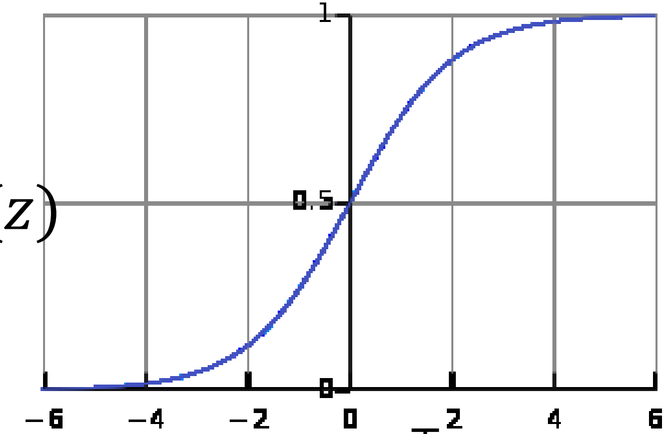
Slide credit: Andrew Ng

Logistic regre

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

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

$g(z)$



$z = \theta^\top x$

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

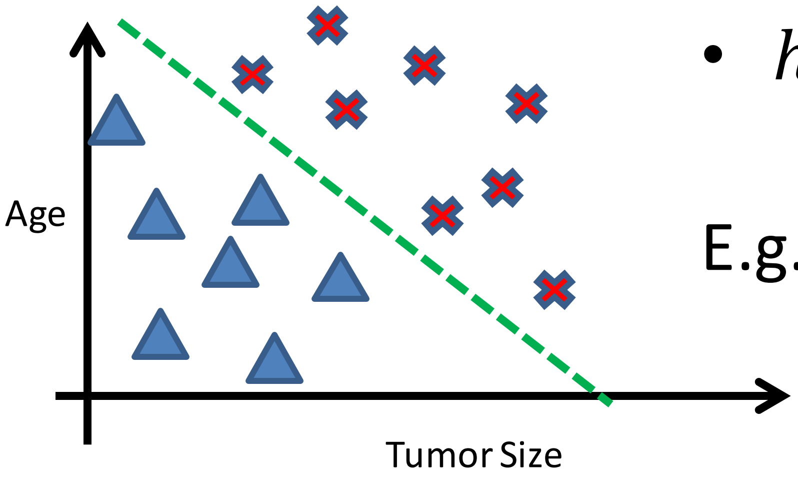
$$z = \ \theta^\top x \geq 0$$

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

$$z = \ \theta^\top x < 0$$

Slide credit: Andrew Ng

Decision boundary

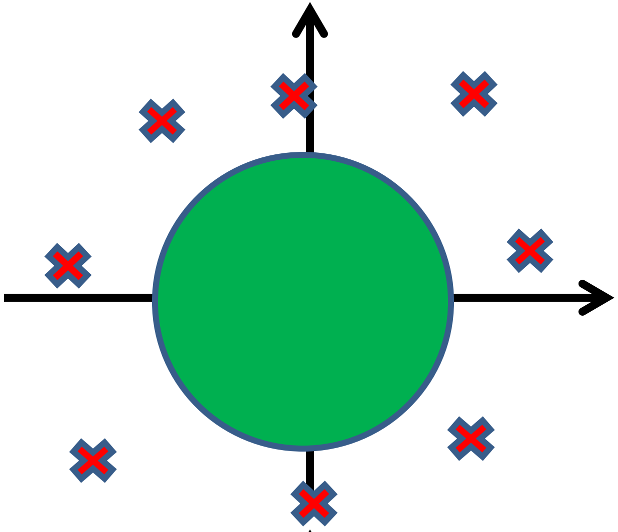


- $h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

E.g., $\theta_0 = -3, \theta_1 = 1, \ \theta_2 = 1$

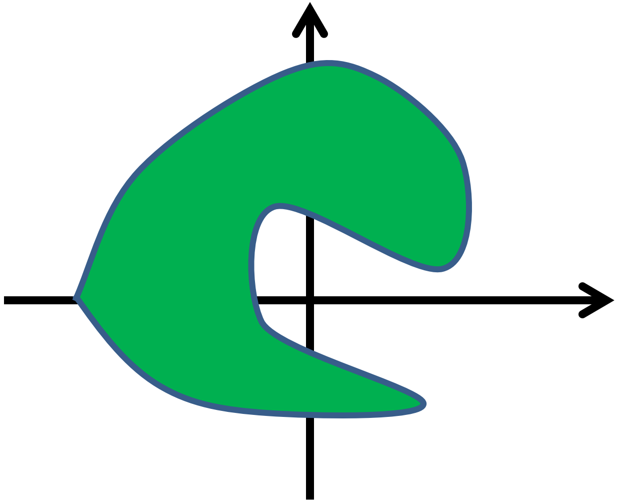
- Predict "$y = 1$" if $-3 + x_1 + x_2 \geq 0$

Slide credit: Andrew Ng



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

E.g., $\theta_0 = -1, \theta_1 = 0, \theta_2 = 0, \theta_3 = 1, \theta_4 = 1$



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

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

Slide credit: Andrew Ng

Logistic Regression

- Hypothesis representation
- **Cost function**
- Logistic regression with gradient descent
- Regularization
- Multi-class classification

Training set with m examples

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

$$x \in \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} \qquad x_0 = 1, y \in \{0, 1\}$$

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

How to choose parameters θ?

Slide credit: Andrew Ng

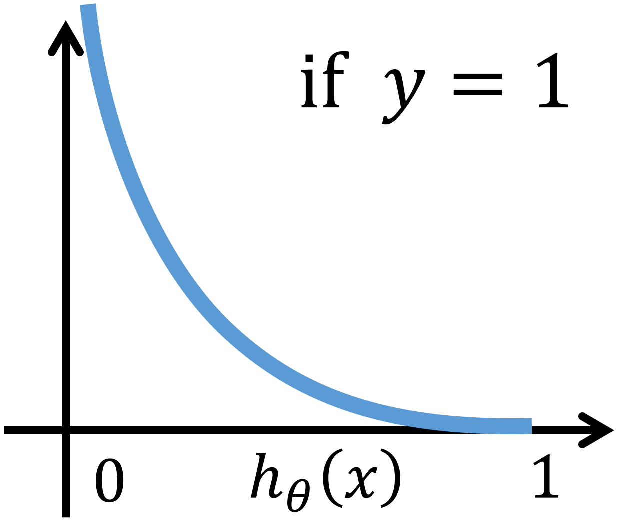
Cost function for **Linear Regression**

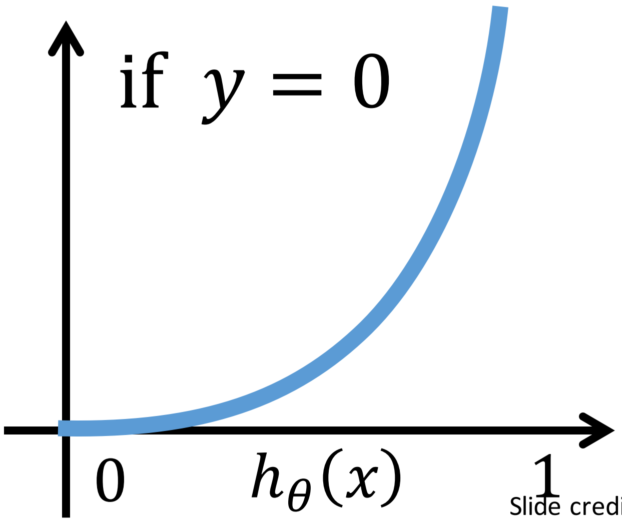
$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 = \frac{1}{m}\sum_{i=1}^{m}\mathrm{Cost}(h_\theta(x^{(i)}), y))$$

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

Cost function for **Logistic Regression**

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





Slide credit: Andrew Ng

Logistic regression cost function

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

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

- If $y = 1$: $\mathrm{Cost}(h_\theta(x), y) = -\log(h_\theta(x))$
- If $y = 0$: $\mathrm{Cost}(h_\theta(x), y) = -\log(1 - h_\theta(x))$

Slide credit: Andrew Ng

Logistic regression

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

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

Learning: fit parameter θ

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

Prediction: given new x

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

Slide credit: Andrew Ng

Logistic Regression

- Hypothesis representation
- Cost function
- **Logistic regression with gradient descent**
- Regularization
- Multi-class classification

Gradient descent

$$J(\theta)$$

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

Goal: $\min_{\theta} J(\theta)$

Good news: Convex function!

Bad news: No analytical solution

Repeat {

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

}

(Simultaneously update all θ_j)

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

Gradient descent

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

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

Repeat {

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

}

(Simultaneously update all θ_j)

Slide credit: Andrew Ng

Gradient descent for Linear Regression

Repeat {

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

}

Gradient descent for Logistic Regression

Repeat {

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

}

Slide credit: Andrew Ng

Logistic Regression

- Hypothesis representation
- Cost function
- Logistic regression with gradient descent
- **Regularization**
- Multi-class classification

Logistic Regression

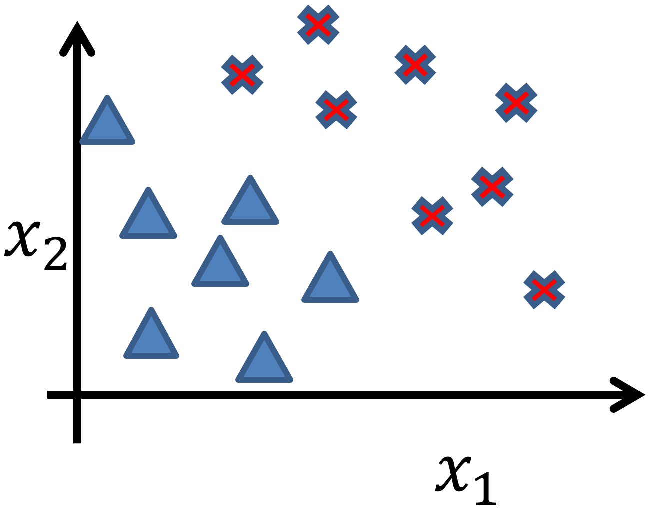
- Hypothesis representation
- Cost function
- Logistic regression with gradient descent
- Regularization
- **Multi-class classification**

Multi-class classification

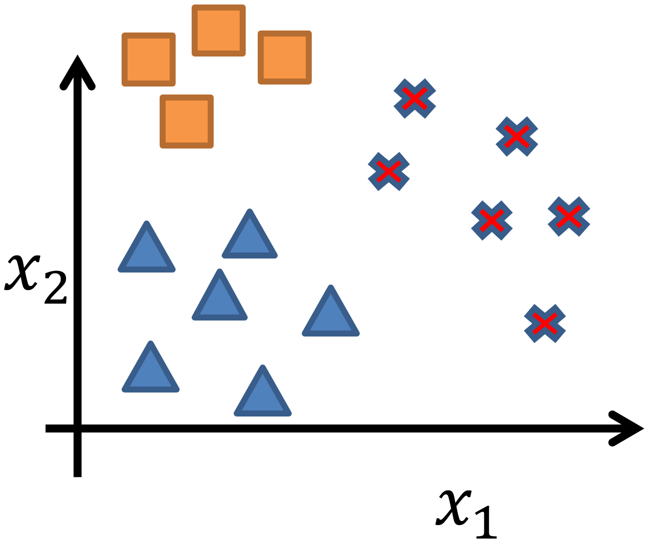
- Email foldering/taggning: Work, Friends, Family, Hobby
- Medical diagrams: Not ill, Cold, Flu
- Weather: Sunny, Cloudy, Rain, Snow

Slide credit: Andrew Ng

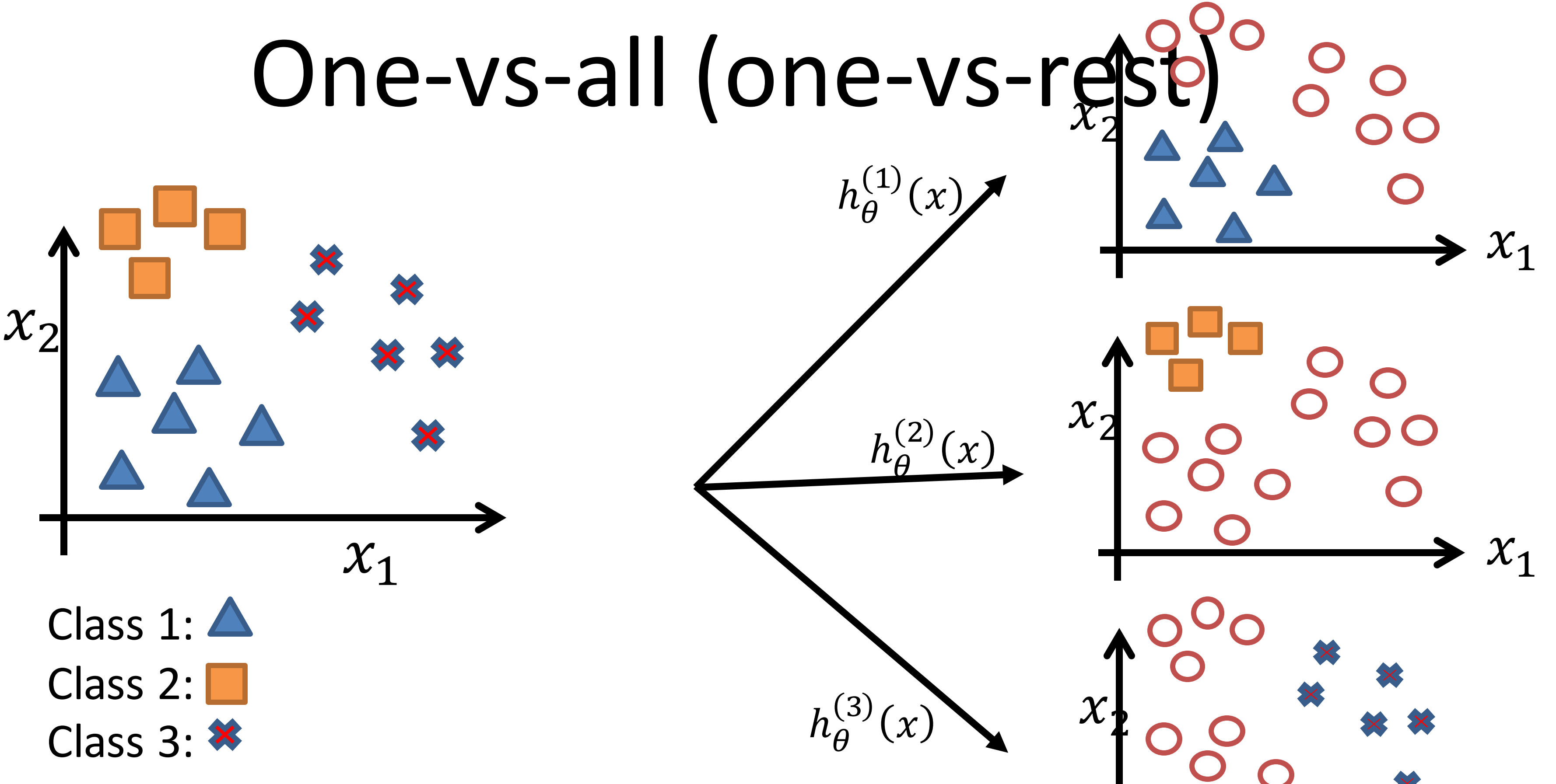
Binary classification



Multiclass classification



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



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

Slide credit: Andrew Ng

One-vs-all

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

- Given a new input x, pick the class i that maximizes

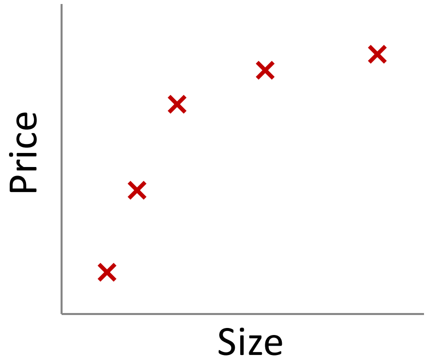
$$\max_{\mathrm{i}} h_\theta^{(i)}(x)$$

Slide credit: Andrew Ng

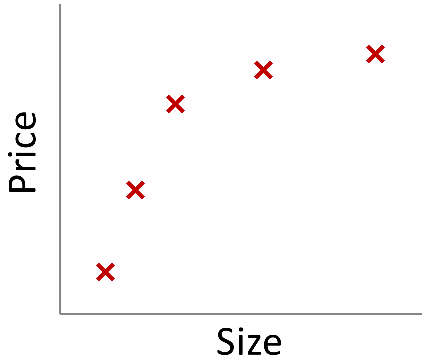
Regularization

The problem of overfitting

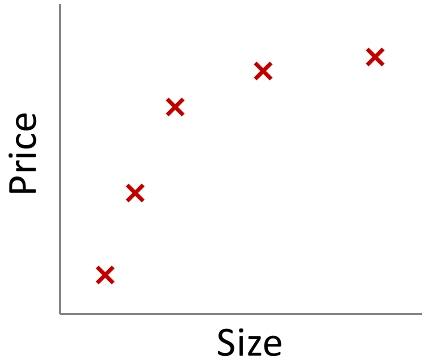
Example: Linear regression (housing prices)



[redacted]



[redacted]



[redacted]

Overfitting: If we have too many features, the learned hypothesis may fit the training set very well ($J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2 \approx 0$), but fail to generalize to new examples (predict prices on new examples).

Addressing overfitting:

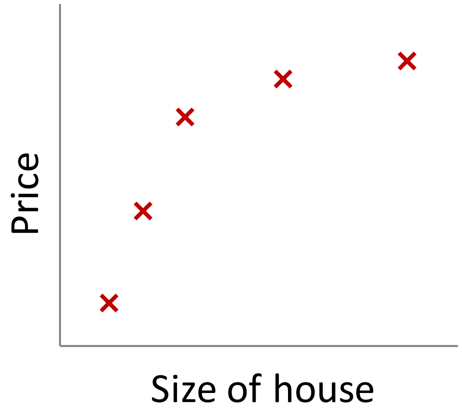
Options:

1. Reduce number of features.
   - Manually select which features to keep.
   - Model selection algorithm (later in course).
2. Regularization.
   - Keep all the features, but reduce magnitude/values of parameters [redacted].
   - Works well when we have a lot of features, each of which contributes a bit to predicting [redacted].

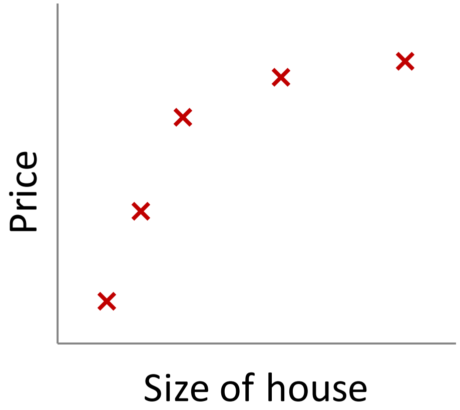
Regularization

Cost function

Intuition



[redacted]



[redacted]

Suppose we penalize and make θ_3, θ_4 really small.

[redacted]

Regularization.

Small values for parameters [redacted]

- "Simpler" hypothesis
- Less prone to overfitting

Housing:

- Features: [redacted]
- Parameters: [redacted]

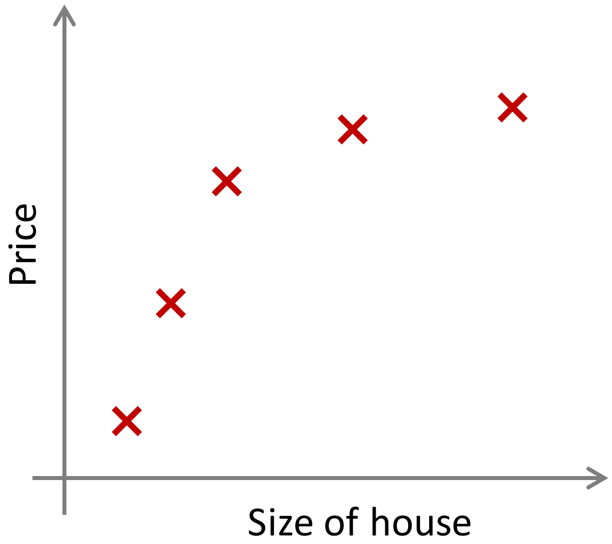
[redacted]

[redacted]

Regularization.

[redacted]

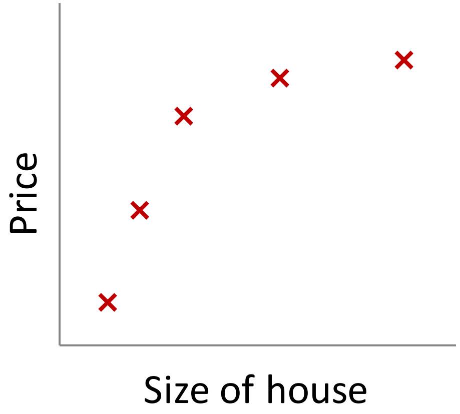
[redacted]



In regularized linear regression, we choose [redacted] to minimize

[redacted]

What if [redacted] is set to an extremely large value (perhaps for too large for our problem, say [redacted])?



[redacted]

Regularization

Regularized linear regression

Regularized linear regression

[redacted]

[redacted]

Gradient descent

Repeat [redacted]

[redacted]

[redacted] [redacted]

$$(j = \cancel{0}, 1, 2, 3, \ldots, n)$$

[redacted]

Normal equation

$$X = \begin{bmatrix} (x^{(1)})^T \\ \vdots \\ (x^{(m)})^T \end{bmatrix}$$

[redacted]

[redacted]

Non-invertibility (optional/advanced).

Suppose [redacted],

(#examples) (#features)

[redacted]

If [redacted],

[redacted]

References

- Andrew Ng's slides on Multiple Linear Regression from his Machine Learning Course on Coursera.

Disclaimer

- Content of this presentation is not original and it has been prepared from various sources for teaching purpose.