DS4023 Machine Learning

Lecture 3:
Logistic Regression

Mathematical Sciences
United International College

Reference: Andrew Ng’s Machine learning course

Outline

- Classification
- Cost function and gradient
- Multi-class
- Regularization

Classification

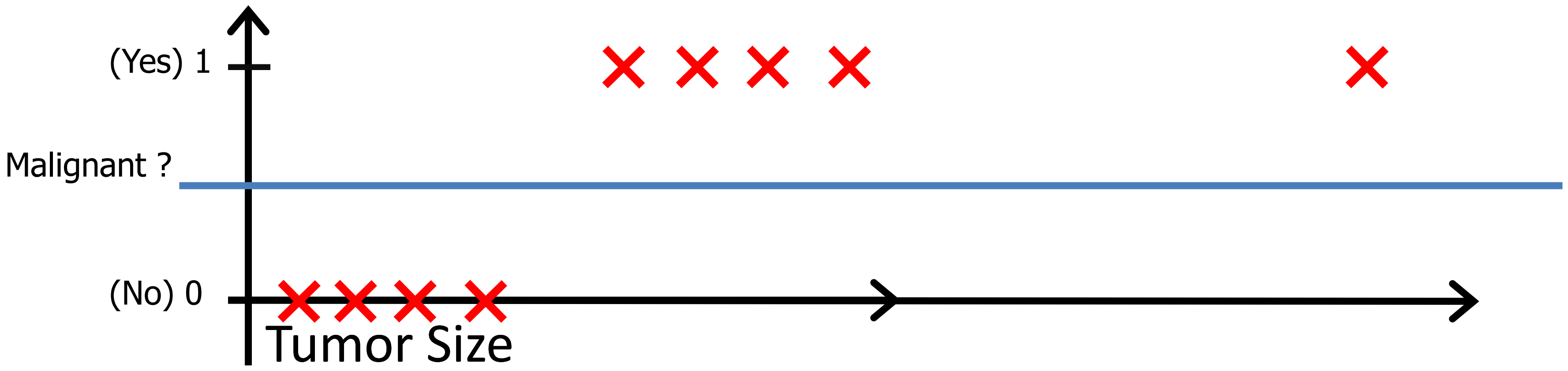
Email: Spam / Not Spam?
Online Transactions: Fraudulent (Yes / No)?
Tumor: Malignant / Benign ?

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

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

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

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 "$y = 0$"

Classification

Classification: $y = 0$ or 1

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

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

Hypothesis Representation

Logistic Regression Model

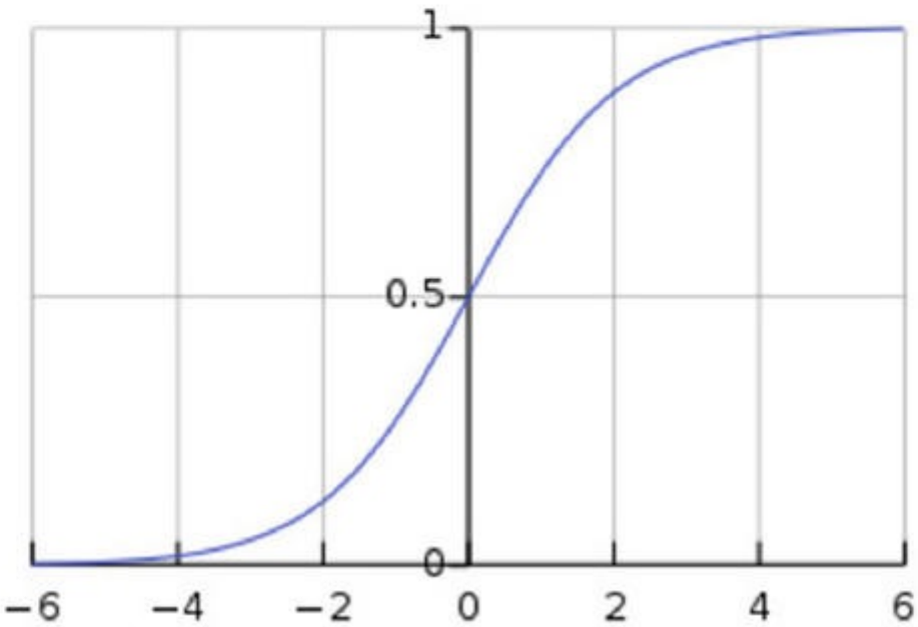
Want $0 \leq h_\theta(x) \leq 1$

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

$$\Downarrow$$

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

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



Sigmoid function

Logistic function

Interpretation of Hypothesis Output

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

Example: If $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ tumorSize \end{bmatrix}$

$$h_\theta(x) = 0.7$$

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 "

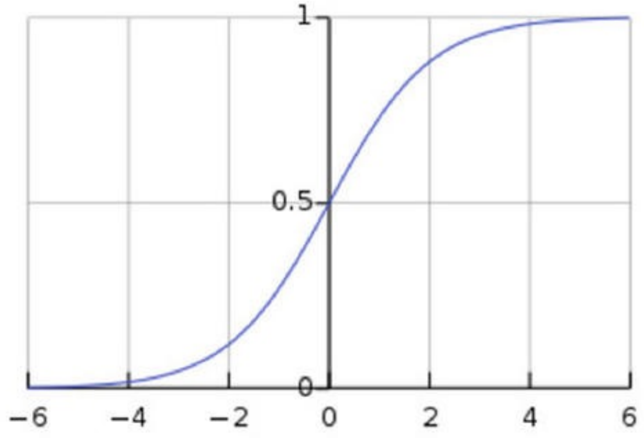
$$P(y = 0|x;\theta) + P(y = 1|x;\theta) = 1$$
$$P(y = 0|x;\theta) = 1 - P(y = 1|x;\theta)$$

Decision boundary

Logistic regression

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

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

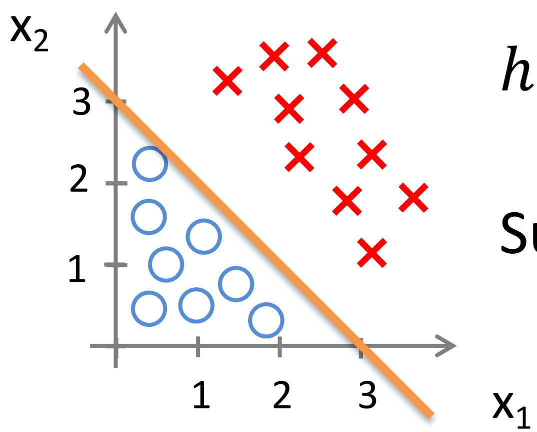


Suppose predict "$y = 1$" if $h_\theta(x) \geq 0.5$ $(\theta^T x \geq 0)$

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

$\theta^T x = 0$: decision boundary

Decision boundary

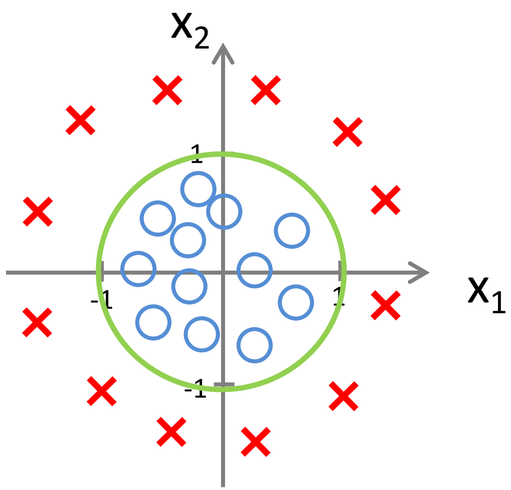


$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$

Suppose we have $\theta_0 = -3, \theta_1 = 1,\ \theta_2 = 1$

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

Non-linear decision boundaries

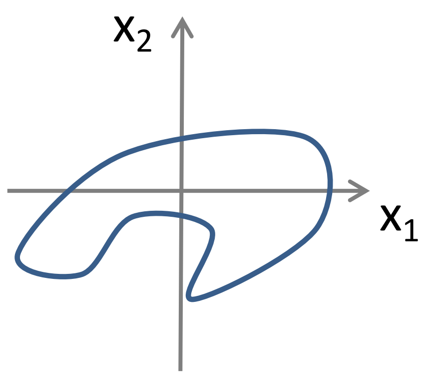


$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 {x_1}^2 + \theta_4 {x_2}^2)$$

- Choose $\theta_0 = -1, \theta_1 = 0, \theta_2 = 0, \theta_3 = 1, \theta_4 = 1$

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

Predict "y=0" otherwise



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

Outline

- Classification
- Cost function and gradient
- Multi-class
- Regularization

Cost function

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

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

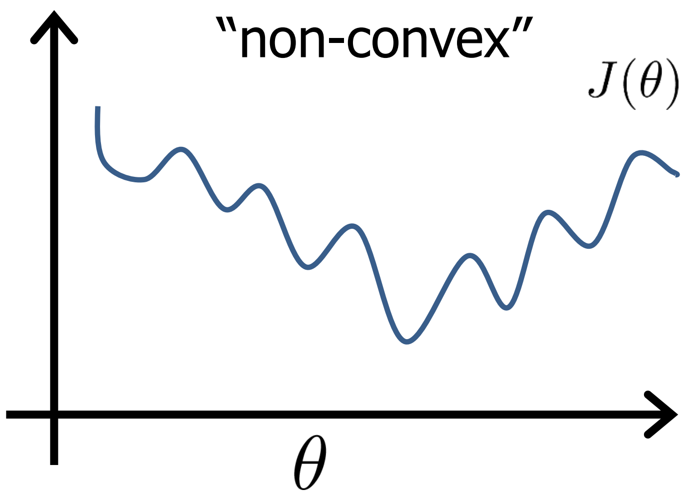
$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

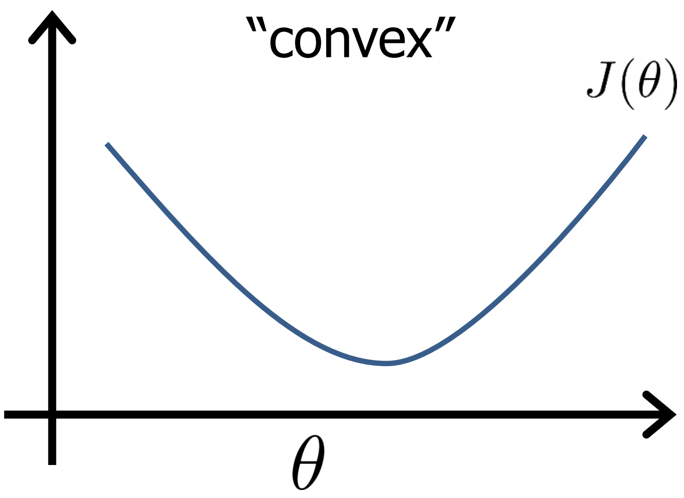
How to choose parameters θ ?

Cost function

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

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

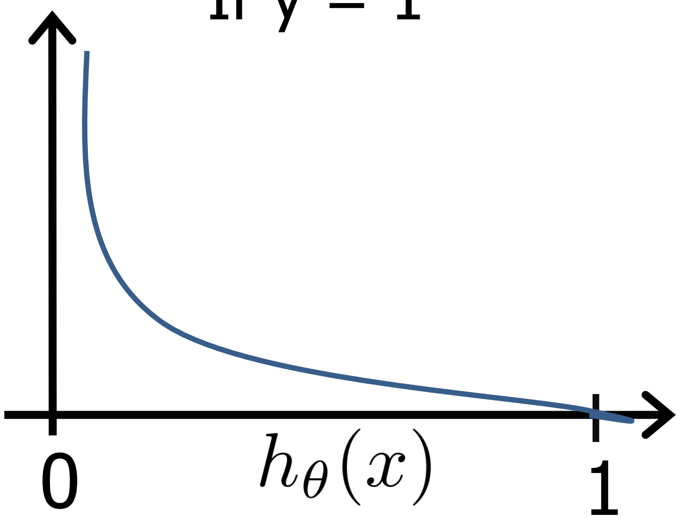




Logistic regression cost function

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

If y = 1



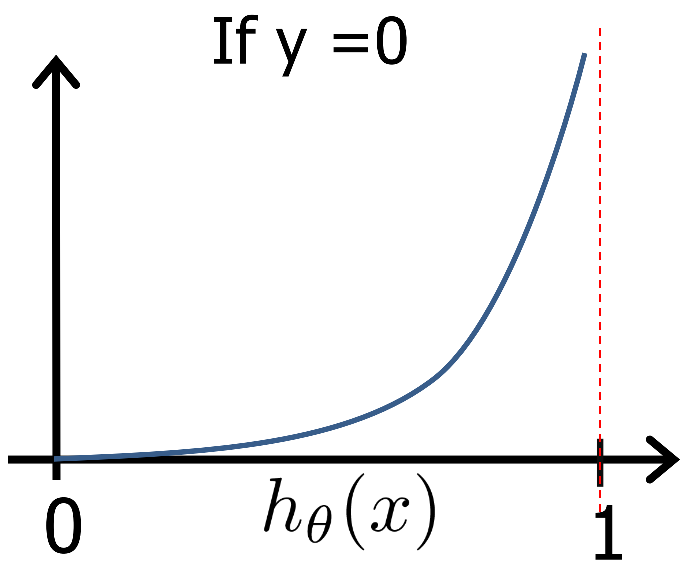
$Cost = 0$ if $y = 1, h_\theta(x) = 1$

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 will penalize learning algorithm by a very large cost.

Logistic regression cost function

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



When $y = 0$,
Cost =0, if $h_\theta(x) = 0$
Cost goes to infinite if $h_\theta(x) = 1$

Logistic regression cost function

Simplification:

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

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

Note: $y = 0$ or 1 always

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

Logistic regression cost function

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

To fit parameters θ :

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

To make a prediction given new x :

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

Gradient Descent

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

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

Repeat{

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

} (simultaneously update all θ_j)

Notice the fact that for

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

$$g'(z) = g(z)(1-g(z))$$

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)

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

Algorithm looks identical to linear regression!

Outline

- Classification
- Cost function and gradient
- Multi-class
- Regularization

Multiclass classification

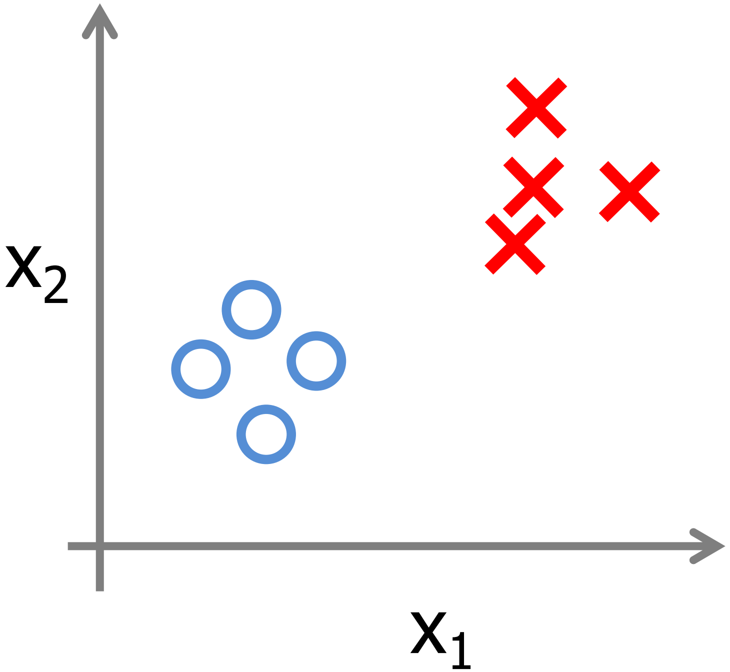
Email foldering/tagging: Work, Friends, Family, Hobby

Medical diagrams: Not ill, Cold, Flu, Coro-V

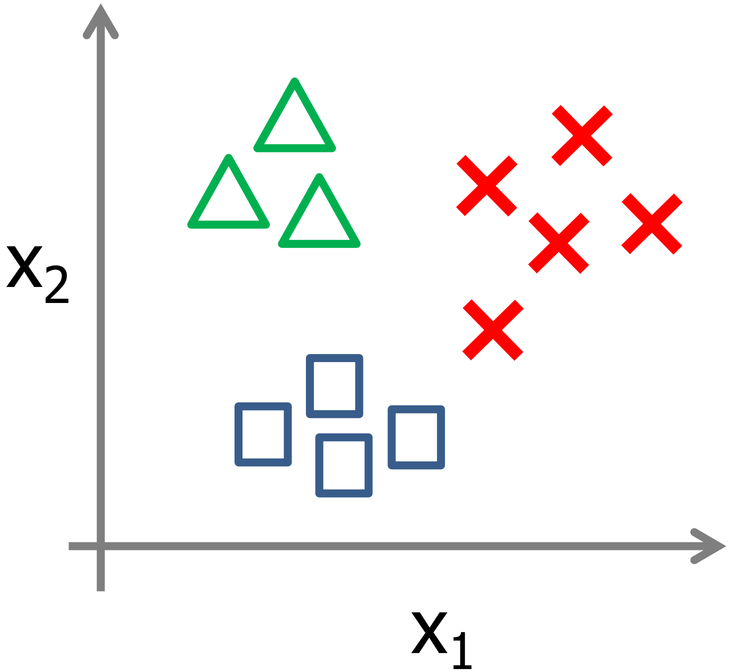
Weather: Sunny, Cloudy, Rain, Snow

Multiclass classification

Binary classification:

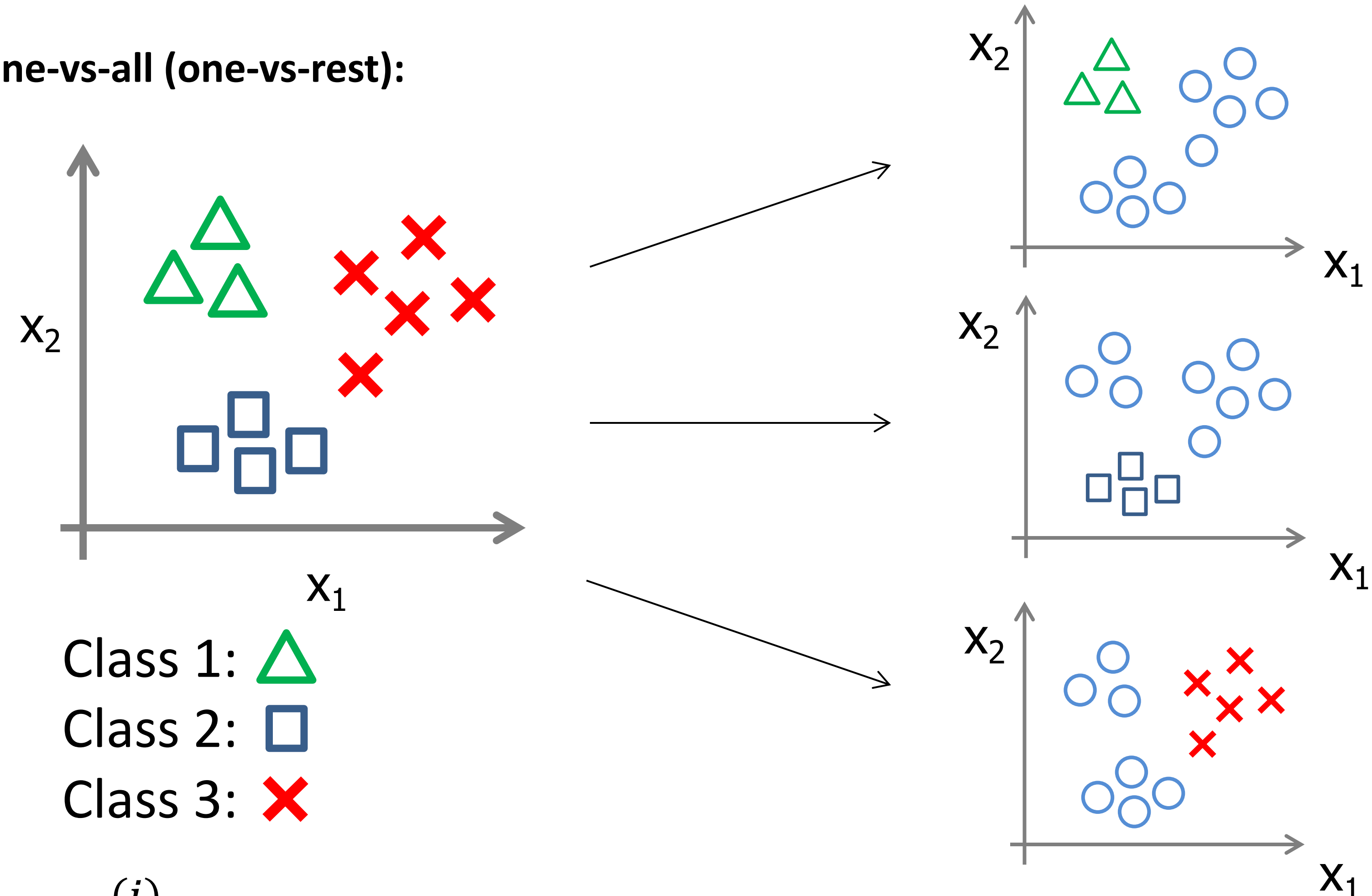


Multi-class classification:



Multiclass classification

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



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

Multiclass classification

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

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

Lab Exercise 4

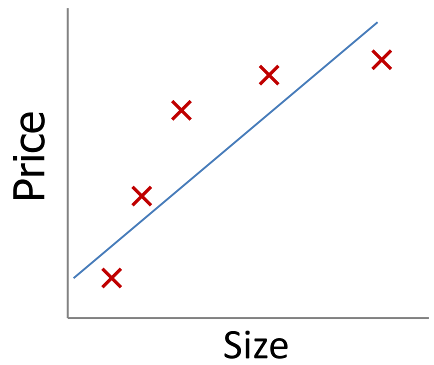
- In this exercise, you will implement logistic regression and get to see it work on data.
- Download *LogisticRegression.ipynb* and Ex2D*ata1.txt* from iSpace, and finish the code implementation in section 1. Logistic Regression (section 2. Regularization is for next lab)
- Submit the completed notebook on iSpace.

Outline

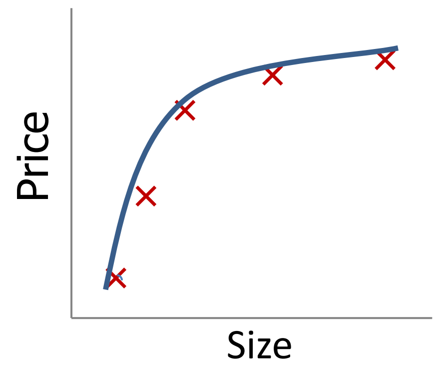
- Classification
- Cost function and gradient
- Multi-class
- Regularization

Regularization-The problem of overfitting

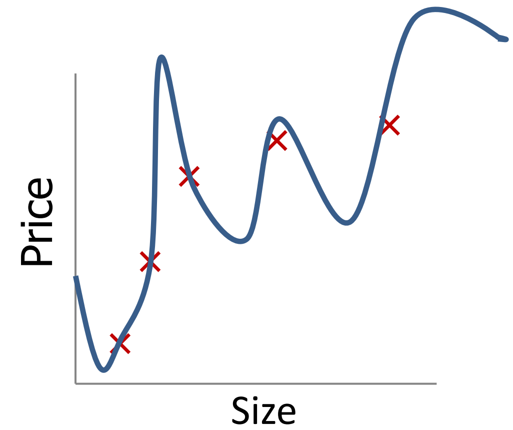
Example: Linear regression (housing prices)



$$\theta_0 + \theta_1 x$$



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

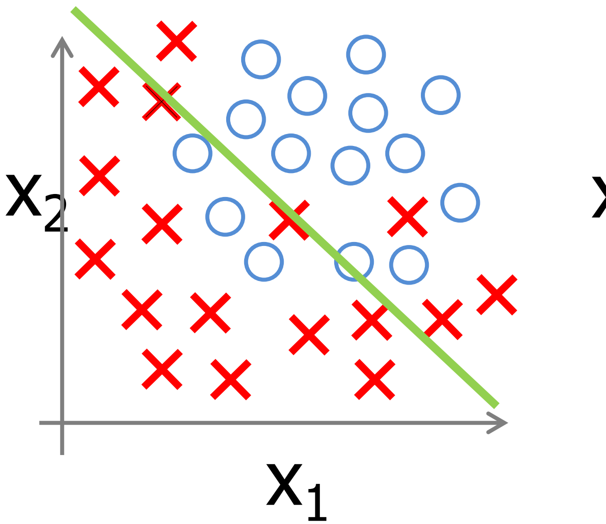


$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

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).

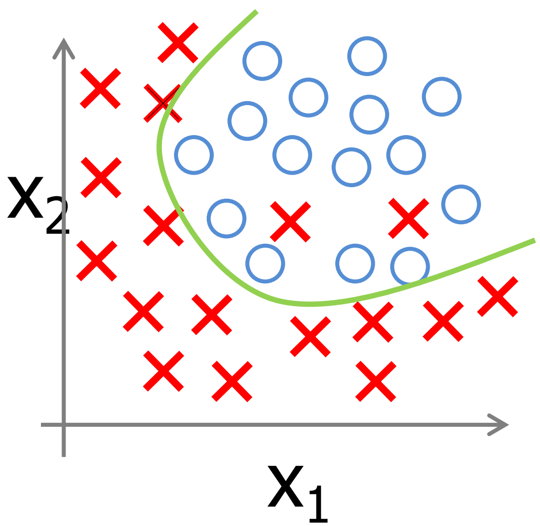
Regularization

Example: Logistic regression

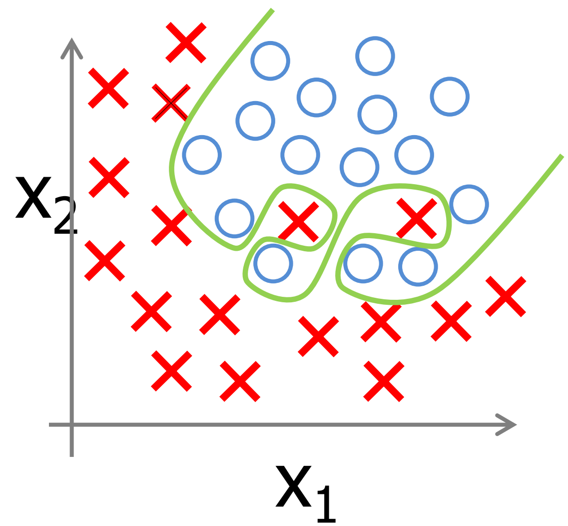


$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

(g = sigmoid function)



$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$$

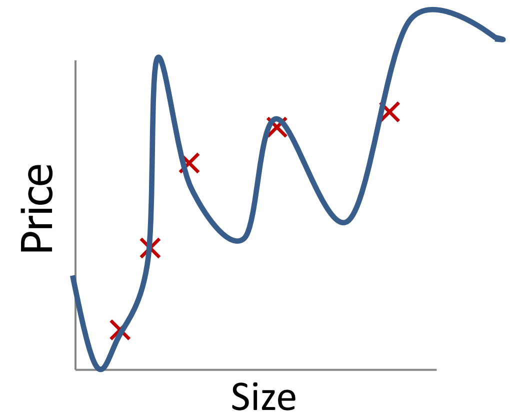


$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots)$$

Regularization

Addressing overfitting:

$$
\begin{aligned}
x_1 &= \text{size of house} \\
x_2 &= \text{no. of bedrooms} \\
x_3 &= \text{no. of floors} \\
x_4 &= \text{age of house} \\
x_5 &= \text{average income in neighborhood} \\
x_6 &= \text{kitchen size} \\
&\vdots \\
x_{100}
\end{aligned}
$$



Regularization

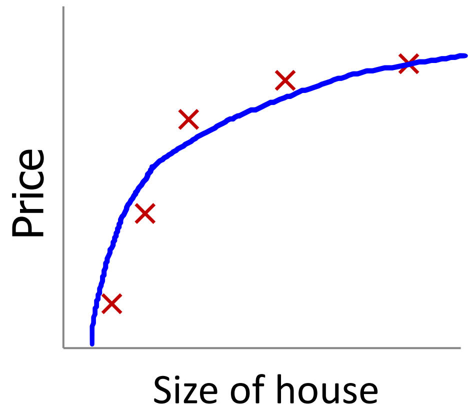
Addressing overfitting:

Options:

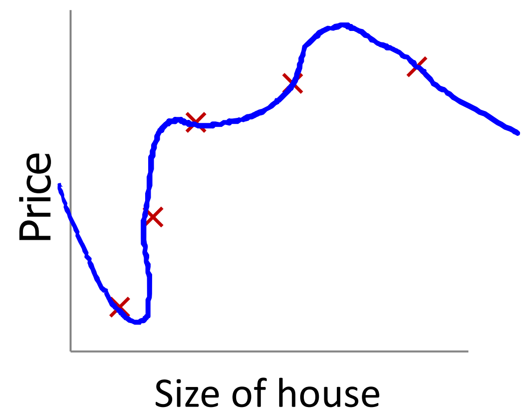
1. Reduce number of features.
   - Manually select which features to keep.
   - Model selection algorithm.
2. Regularization.
   - Keep all the features, but reduce magnitude/values of parameters.
   - Works well when we have a lot of features, each of which contributes a bit to predicting.

Regularization

Intuition



$\theta_0 + \theta_1 x + \theta_2 x^2$



$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$

Suppose we penalize and make θ_3, θ_4 really small.

$$\min_\theta \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + 1000\theta_3^2 + 1000\theta_4^2$$

Regularization

Small values for parameters $\theta_0, \theta_1, \dots, \theta_n$

- "Simpler" hypothesis
- Less prone to overfitting

Housing:

- Features: $x_0, x_1, \dots, x_{100}$
- Parameters: $\theta_0, \theta_1, \dots, \theta_{100}$

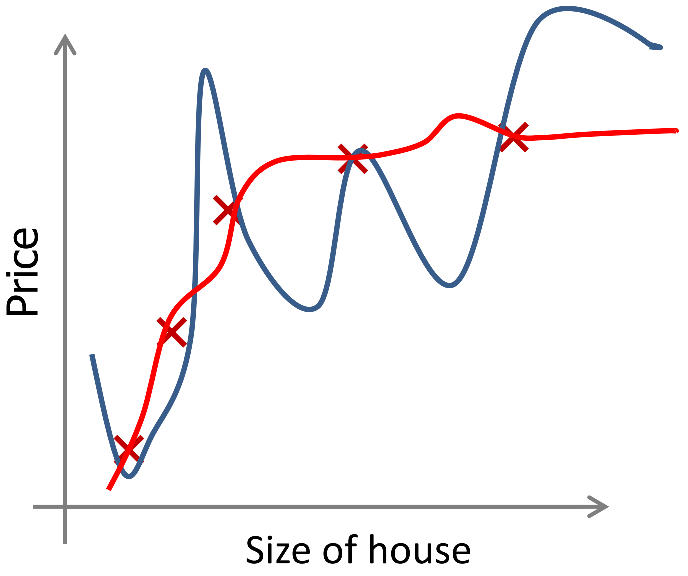
$$\min_\theta \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \boxed{\lambda \sum_{j=1}^{n} \theta_j^2}\ \right]$$

Regularization

$$\min_{\theta} \frac{1}{2m} \left[\sum_{i=1}^{m} \boxed{\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2} + \boxed{\lambda \sum_{j=1}^{n} \theta_j^2} \right]$$

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

The two parts in the cost function make a balance between bias and variance, so that the model doesn't go underfitting or overfitting too much



Regularization

In regularized linear regression, we choose θ to minimize

$$\min_{\theta} \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right)-y^{(i)}\right)^2+\lambda\sum_{j=1}^{n}\theta_j^2\ \right]$$

What if λ is set to an extremely large value (perhaps for too large for our problem, say $\lambda = 10^{10}$)?

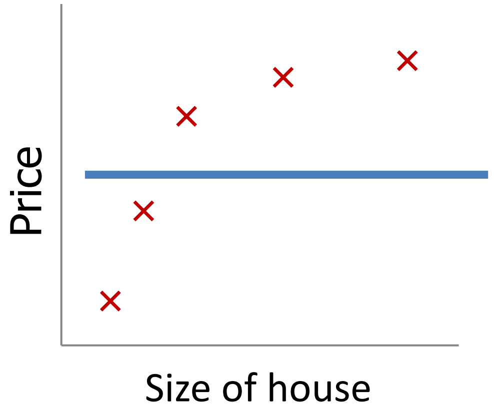
- Algorithm works fine; setting λ to be very large can't hurt it
- Algorithm fails to eliminate overfitting.
- Algorithm results in underfitting. (Fails to fit even training data well).
- Gradient descent will fail to converge.

Regularization

In regularized linear regression, we choose θ to minimize

$$\min_{\theta} \frac{1}{2m} \left[\sum_{i=1}^{m} \left(h_\theta\left(x^{(i)}\right) - y^{(i)} \right)^2 + \lambda \sum_{j=1}^{n} \theta_j^2 \right]$$

What if λ is set to an extremely large value (perhaps for too large for our problem, say $\lambda = 10^{10}$)?



underfit

$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

All theta are penalized to almost 0

Regularization: linear regression

$$\min_\theta \frac{1}{2m}\left[\sum_{i=1}^m \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2 + \lambda \sum_{j=1}^n \theta_j^2 \;\right]$$

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

Gradient descent

Repeat $\{$

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^m (h_\theta\left(x^{(i)}\right) - y^{(i)}) x_0^{(i)}$$

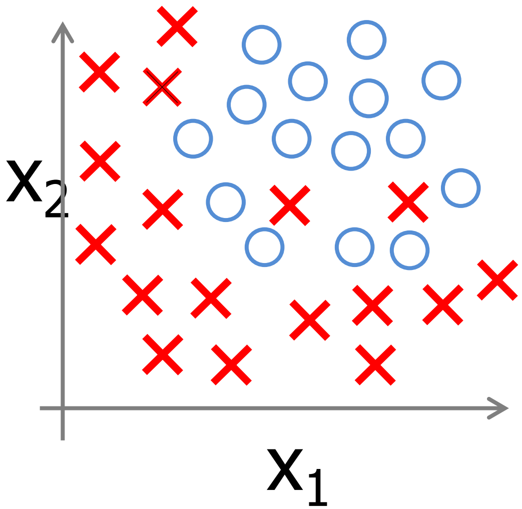
$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^m \left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right) x_j^{(i)} - \alpha \frac{\lambda}{m} \theta_j$$

$$(j = 1,2, \ldots, n)$$

$\}$

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

Regularization: logistic regression



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

Cost function:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log\left(h_\theta\left(x^{(i)}\right)\right) - \left(1-y^{(i)}\right)\log\left(1-h_\theta\left(x^{(i)}\right)\right)\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

Regularization: **logistic regression**

Gradient descent

Repeat $\{$

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_0^{(i)}$$

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

$$(j = 1,2, \dots, n)$$

$\}$

The above gradient formula looks exactly like what we have in linear regression, the only difference is the form of $h_\theta(x^{(i)})$.

Lab Exercise 5

- In this exercise, you will implement logistic regression and get to see it work on data.
- Download *LogisticRegression.ipynb* and Ex2D*ata2.txt* from iSpace, and finish the code implementation in both section 1. Logistic Regression (you should have done it already) and section 2. Regularization
- (Note that some of the import and function definition needed in section 2 are defined in section 1)
- Submit the completed notebook on iSpace.