Machine Learning

Lecture 3

Participation and Assignment

Pilot News Updates

Zach Introduction (office hours)

Assignment 1 will be on linear regression (yesterday + today's class)

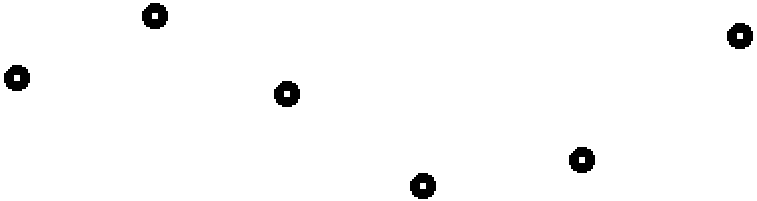
Categories Within Supervised ML

Classification machine learning systems: Systems where we seek a yes-or-no prediction, such as “Is this tumer cancerous?”, “Does this cookie meet our quality standards?”

a. Binary or Multiclass classifier : Output y in {-1, 1} or y in {1,..k}

Regression machine learning systems: Systems where the value being predicted falls somewhere on a continuous spectrum. These systems help us with questions of “How much?” or “How many?”.

Regression



Regression

How do we choose the “right function”?

How do we measure the “rightness”?

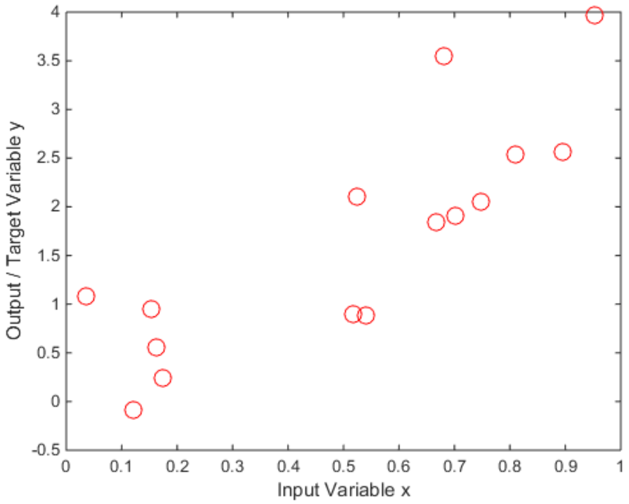
How do we trade off between the degree of fit and the complexity of solution?

Linear Regression: single variable

Consider the single independent variable case:

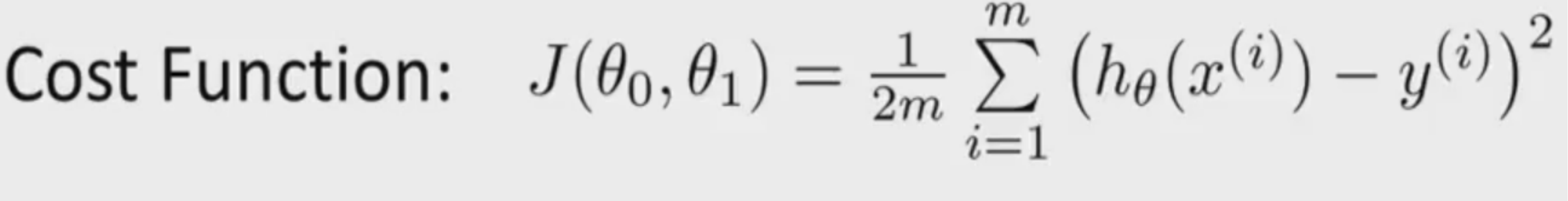
y = hθ(x) s. t.

$$h_\theta(x) = \theta_0 + \theta_1 x$$



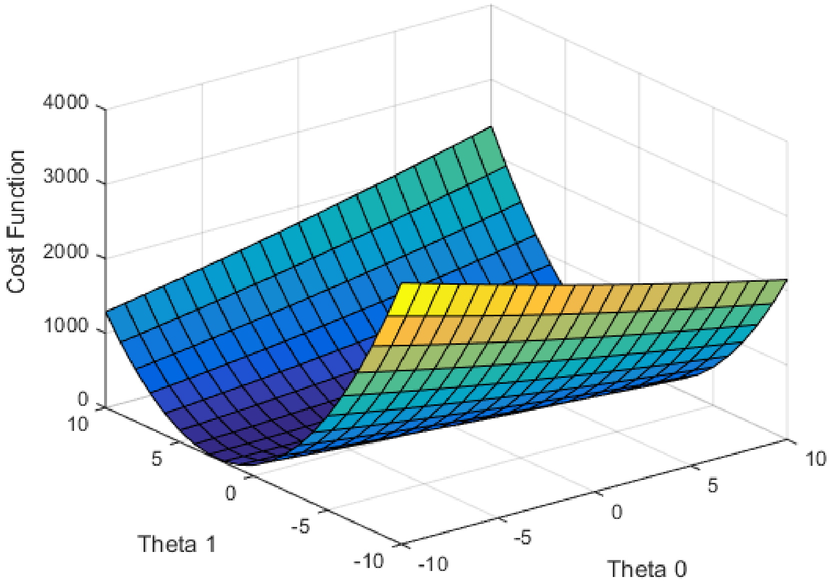
Cost Function

Find the values of Θ_0 & Θ_1 to minimize this expression:



Cost Function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$

Cost Function



Optimization

We have created a cost function that we want to minimize over the training data samples

We want to experiment with different values of Θ_0 & Θ_1 so that the cost function $J(\Theta_0,\Theta_1)$ keeps reducing so we can end up in the minimum (hopefully)

Gradient Descent

α is called the Learning Rate (some books also use ***η***), ∈ [0,1]

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

Gradient Descent

α is called the Learning Rate, ∈ [0,1]

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

What is the global minimum?

What happens to the 2nd term once we reach there?

Gradient Descent

Repeat until convergence {

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

}

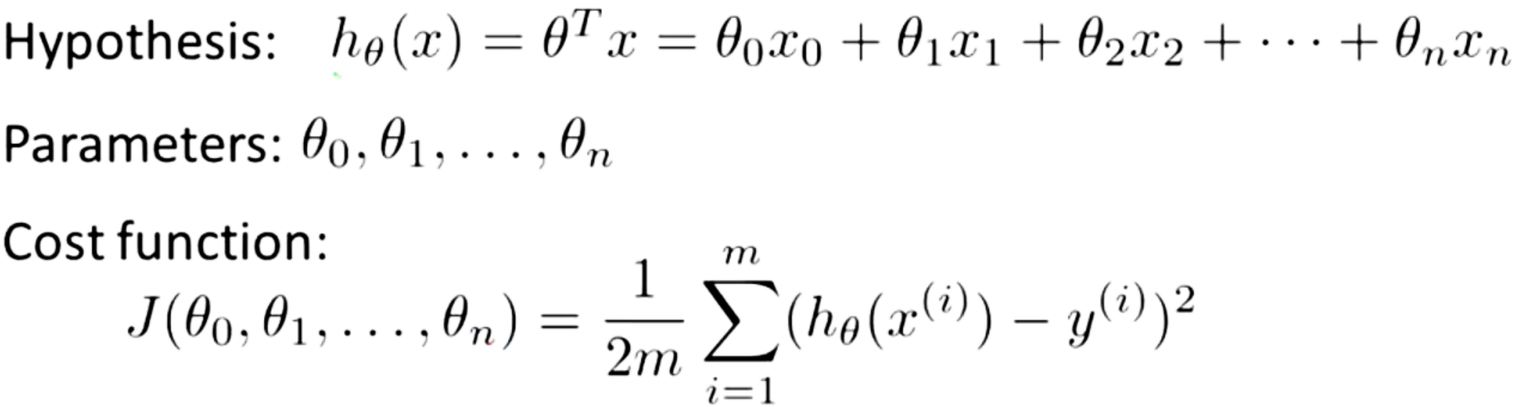
Some Pointers On Implementation:

Using temp variables (2nd parameter is a function of Θ_0 and Θ_1)

Updating the parameters at the end simultaneously, after computing the partial derivatives for each parameter

Values of α : varying from 0 to 1

Multivariate Gradient Descent



Hypothesis: $h_\theta(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

Parameters: $\theta_0, \theta_1, \ldots, \theta_n$

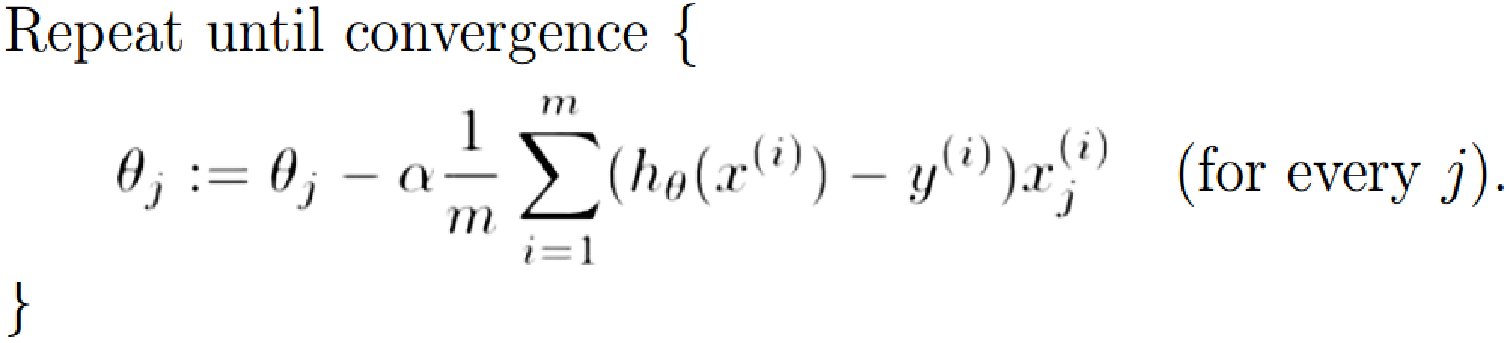
Cost function:

$$J(\theta_0, \theta_1, \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2$$

Note: Both x and theta are now (n+1) dimensional

Gradient Descent

For m training samples, then we get:



Repeat until convergence {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) x_j^{(i)} \quad \text{(for every } j\text{).}$$

}

Gradient Descent

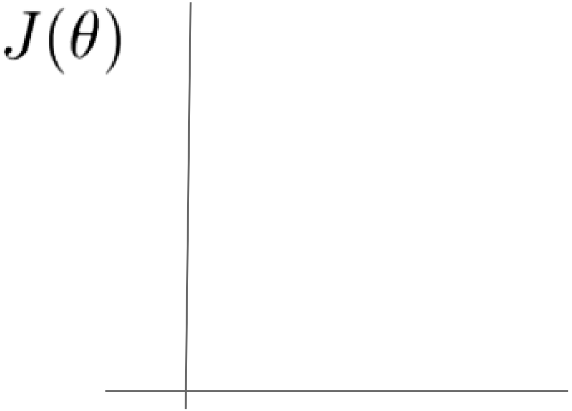
Batch vs Online

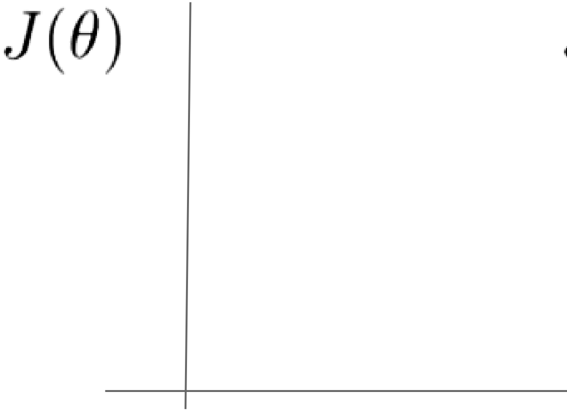
Batch: considers all the training samples in the data (or batches at a time)

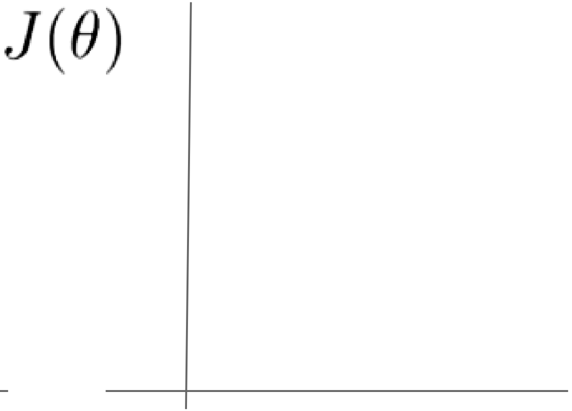
Online: considers each training sample one at a time

Q: Is our method online or batch?

How to check for the learning rate?







Feature Scaling

Are there any challenges here that could affect our optimization function (cost function)?

Feature 1	Feature 2	Target
0	4100	255
1	6544	422
2	7711	122
1	100	661

Feature Scaling

Replace feature x with:
(x – mean(x)) / (max(x) – min(x))

Feature 1	Feature 2	Target
0	4100	255
1	6544	422
2	7711	122
1	100	661

How to verify that the Algorithm is working as it should?

What should the cost function look like over time? (What is time here??)

Stopping Criteria

This is an iterative algorithm

How do we know when to stop?

- choose a small threshold ε: if the change in cost function is below ε, stop the iterations
- Hard code the number of iterations
- A combination of both (look at the graph first)

Polynomial Features

Suppose I have this function:

$$h_\theta(x) = \Theta_0 + \Theta_1 x + \Theta_2 X^2$$

Does this change the algorithm we have learned so far?

Polynomial Features

Suppose I have this function:

$$h_\theta(x) = \Theta_0 + \Theta_1 x + \Theta_2 X^2$$

Does this change the algorithm we have learned so far?

Is feature scaling important?

Assignment 1

Will be posted online

Technical report addressing all the questions posed

Analysis is the main difference in graduate assignments - Why? What? How?

Question: How many planning to do a thesis?

Assignment 1

Training Vs Test Data

- Can they overlap?
- Which would give the highest performance?
- What could be a concern?
- What are alternatives?

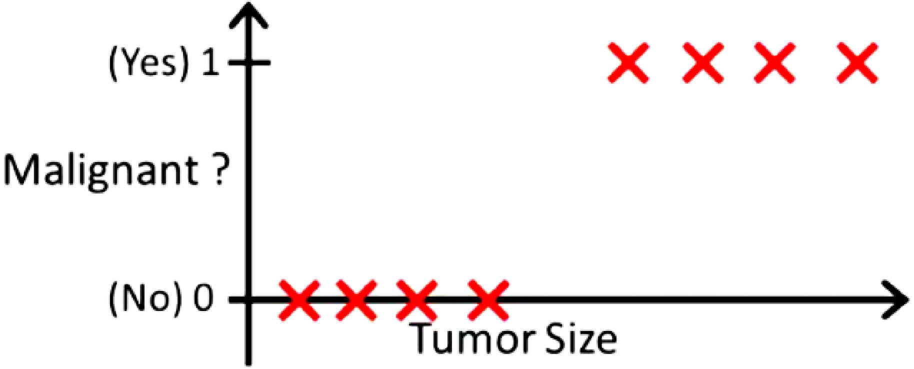
Logistic Regression

Classification

Logistic Regression

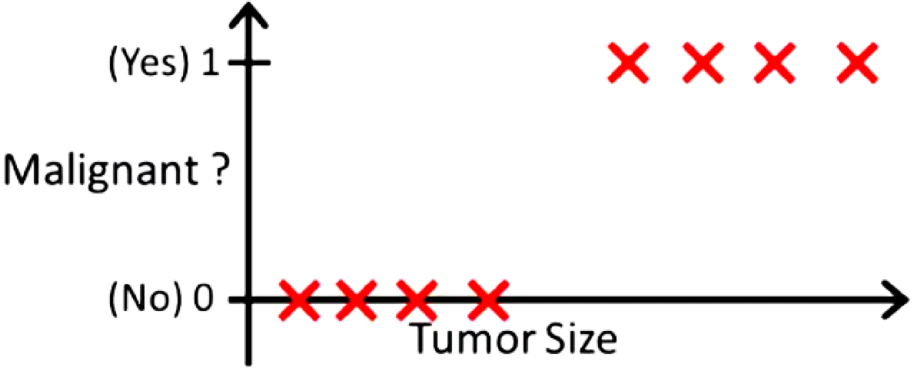
We need a transformation that ‘forces’ the output to {0, 1}

$h_\theta(x)$



Logistic Regression

Fitting a straight line: how will that look? Is it a good idea?



Logistic Regression

Fitting a straight line: how will that look? Is it a good idea?

Concerns:

1. Linear representation of hypothesis function
2. Range

Logistic Regression

Specifically, we need a transformation s.t.

$$0 \leq h\ominus(x) \leq 1$$

linear regression:

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

Clearly, we are looking at classification

Logistic Regression

Specifically, we need a transformation s.t.

$$0 \leq h\ominus(x) \leq 1$$

linear regression:

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

Let’s consider sigmoid function

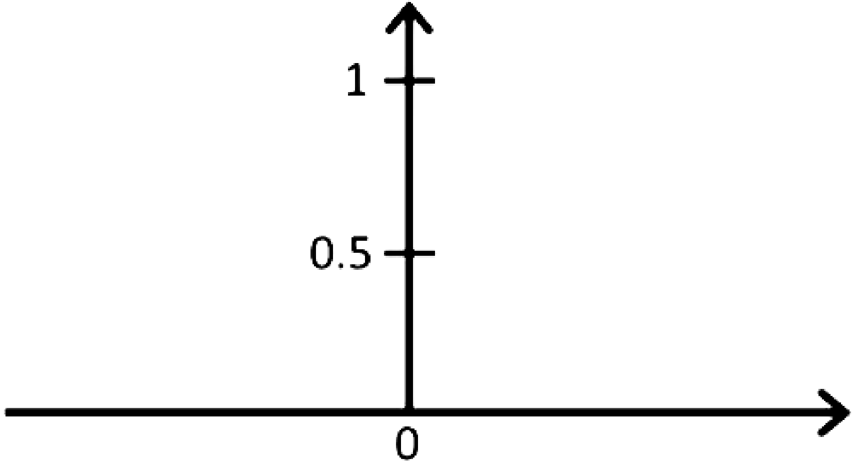
$$f(x) = \frac{1}{1+e^{-x}}$$

Logistic Regression

So does this equation satisfy our criteria? Plot the curve to find out

$$f(x) = \frac{1}{1+e^{-x}}$$

What if x=0?
What if x<0?
What if x>0?



Logistic Regression

So what will be the sigmoid function for the logistic regression

Interpretation

So how do we interpret h⊖(x)? What if h⊖(x)=0.8? Let’s use the malignant tumor use case.

Interpretation

What if $h\ominus(x)=0.8$?
Classification?

Decision Boundary

A **decision boundary (DB)** is the region of a problem space in which the output label of a classifier is ambiguous. So it separates the data space into “y=1” and “y=0” in the malignant tumor example.

If the **decision** surface is a hyperplane, then the classification problem is linear, and the classes are linearly separable.

Decision Boundary

A **decision boundary (DB)** is the region of a problem space in which the output label of a classifier is ambiguous. So it separates the data space into “y=1” and “y=0” in the malignant tumor example.

What should the value of y be at the DB?

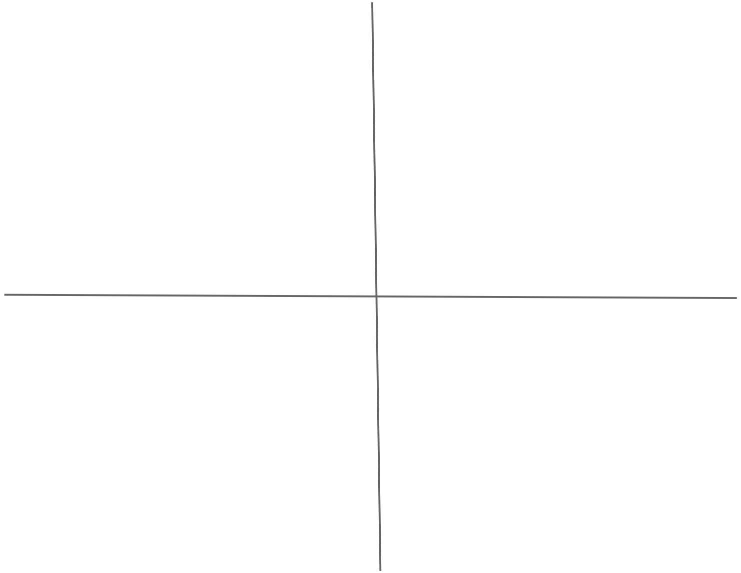
DB

If we assign y=1 when h⊖(x)>=0.5, what is the value of $\theta^T x$?

What is the DB for polynomial data?

y=1 if $\theta^T x$ >=0

What if theta2 =1,
theta1=0, and theta0=2?



Logistic Regression Model

Training Set {($x^{(1)}$,$y^{(1)}$), ($x^{(2)}$,$y^{(2)}$) …..($x^{()}$,$y^{()}$)}   1 through m ; y ∈ {0,1}
Consider the earlier cost function for Linear Regression

Cost Function: $$J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

But now we have J(Ɵ) is a non-convex function!! (Remember, what is the logistic hypothesis function?)

Logistic Regression Model

Consider:

Cost (h⊖(x) ,y) = -log(h⊖(x)) if y = 1
-log(1 - h⊖(x)) if y = 0

For each of the training samples .

Can we write this cost function in a single equation?

Logistic Regression Model

So how do we merge the two cases for $y = 0$ or $y = 1$?

Logistic Regression Model

Consider:

Cost (h⊖(x) ,y) = (y)* (-log(h⊖(x))) + (1-y)*(-log(1 - h⊖(x)))

= -y*log(h⊖(x)) - (1-y)*log(1 - h⊖(x))

And now we need to consider this for each of the training samples .

Logistic Regression Model

Final cost function:

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

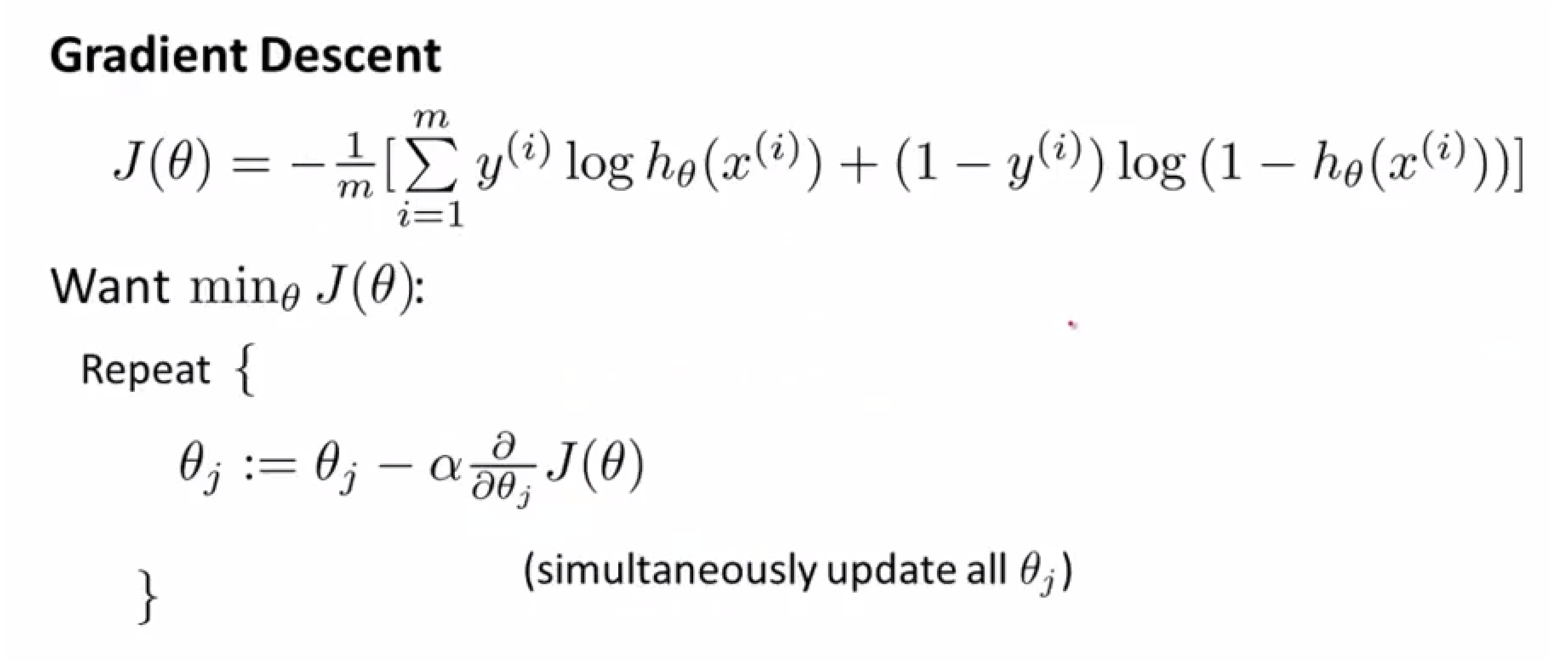
How does this model work?

Given a new input x, compute the hypothesis

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

Which we then interpret as the p(y=1|x;⊖)

Algorithm



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

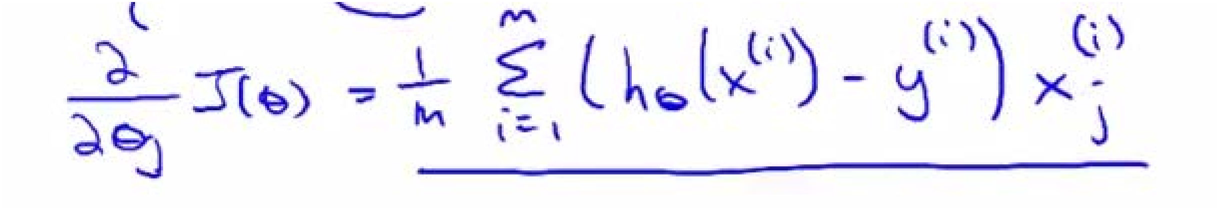
Repeat $\{$

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

$\}$

(simultaneously update all θ_j)

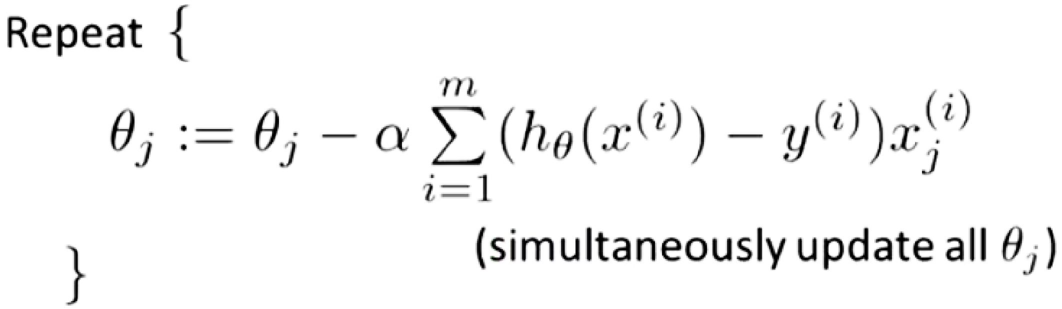
hide slide



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

Algorithm

Substitute the partial derivative term



Repeat {

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

(simultaneously update all θ_j)

}

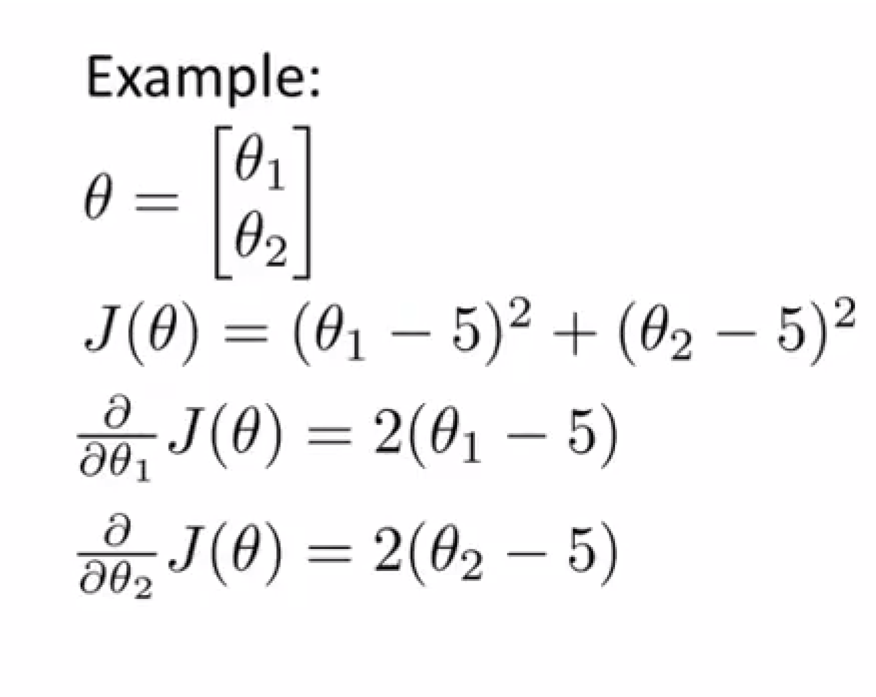
So how can we check the performance in this case?

Optimization Concepts

Other sophisticated algorithms

BFGS, conjugate gradient (may want to consider implementing for project)

Example



Example:

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

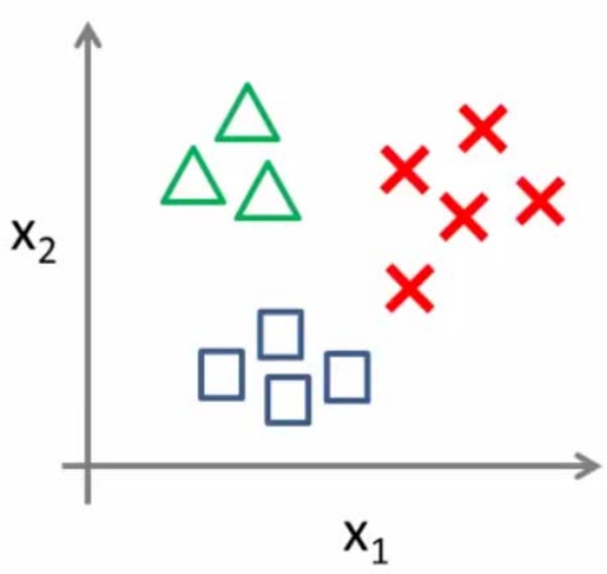
$$J(\theta) = (\theta_1 - 5)^2 + (\theta_2 - 5)^2$$

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

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

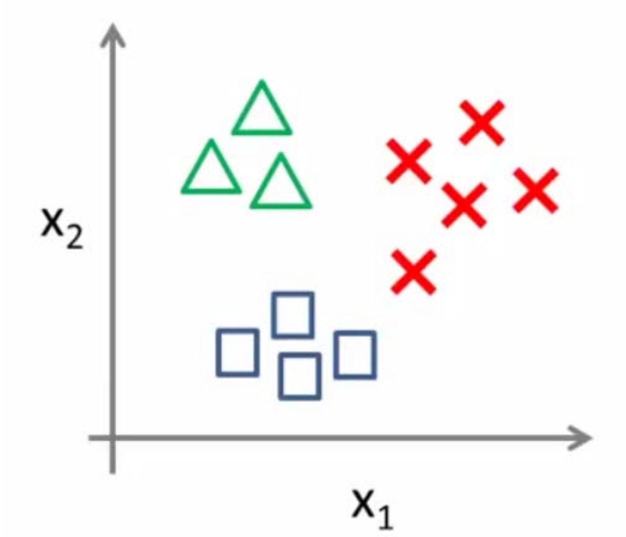
Multiclass classification

Cookies = {chocolate, oatmeal, raisin}



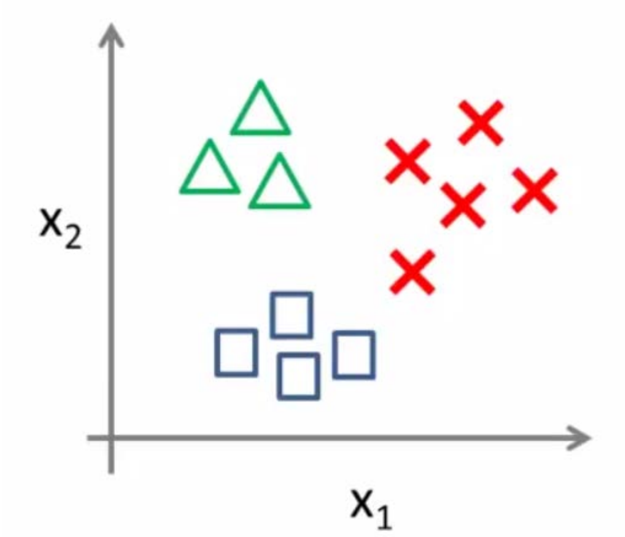
Multiclass classification

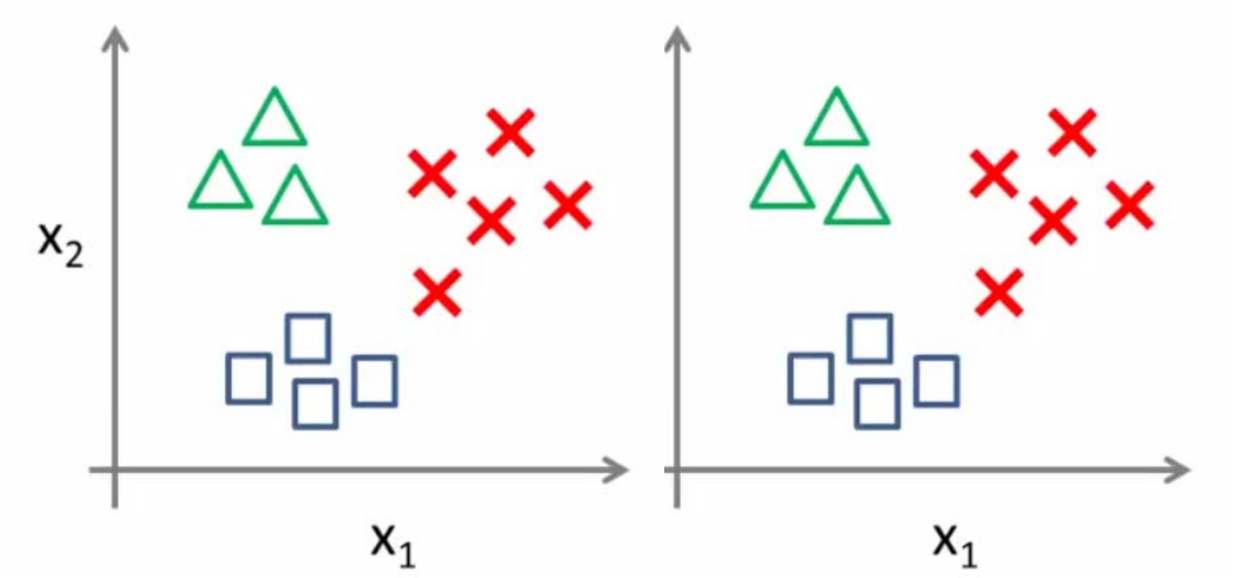
Cookies = {chocolate, oatmeal, raisin}



Multiclass classification

Cookies = {chocolate, oatmeal, raisin}





One - vs - all

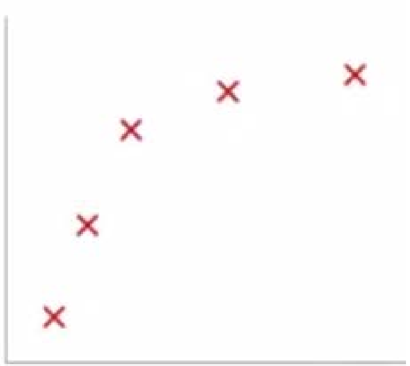
For a new input x, to make a prediction, pick the class label i such that it is the max value of the hypothesis function

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

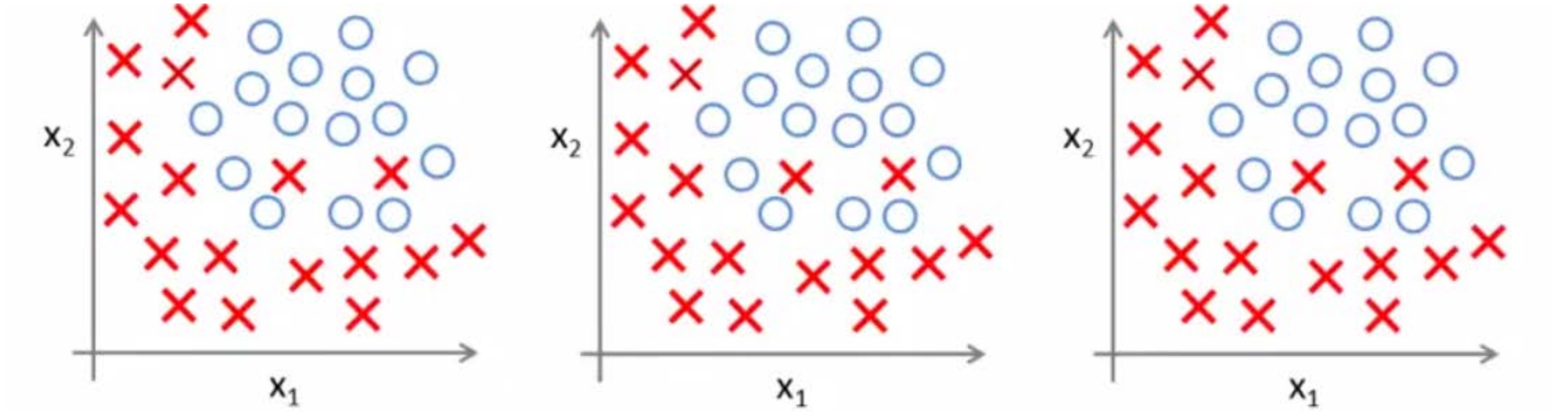
Overfitting

Overfitting

If we have too many features, the learned hypothesis may fit the training data very well but fail to generalize to new examples



Overfitting: Logistic Regression



Addressing Overfitting

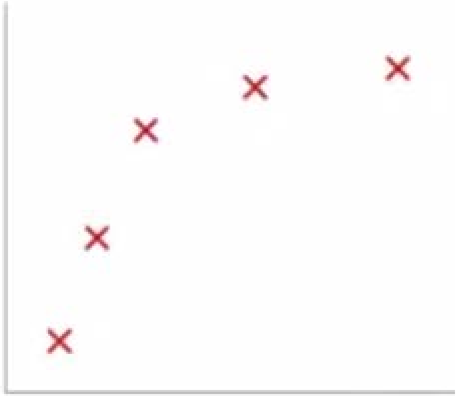
- Reducing # of features
  - Some models help screen out less important features
- Regularization
  - Keep all features but reduce magnitudes or values of the theta parameters
  - Useful for cases with a lot of ‘weak’ features

Regularization

Suppose we try to penalize the parameters in the equation

~g($\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$)

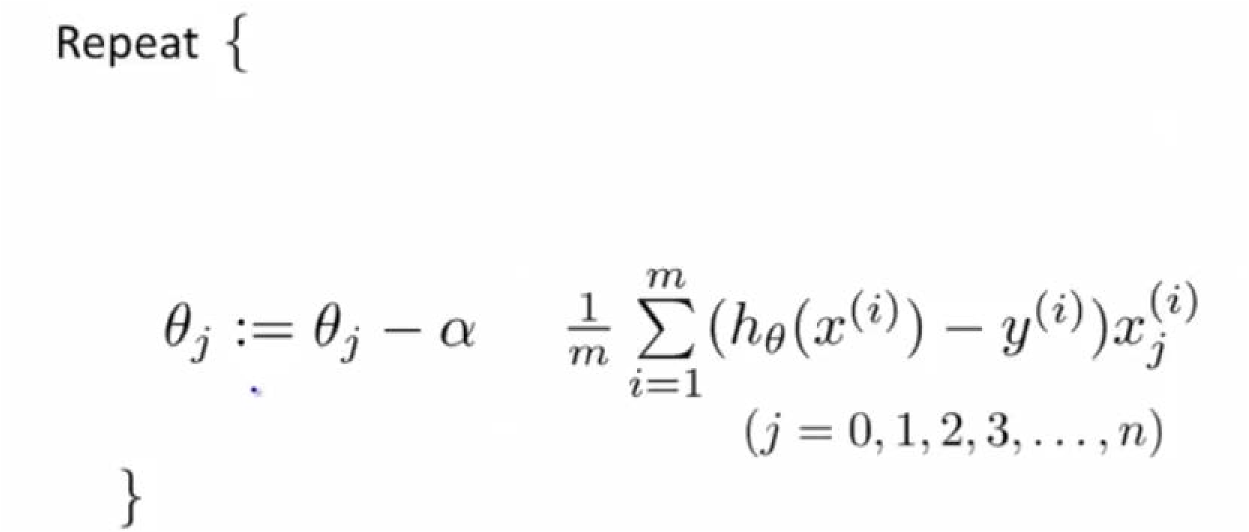
$$\min_\theta \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)})^2 \quad + 1000 * \theta_3^2$$



Regularization

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

Gradient Descent without Regularization



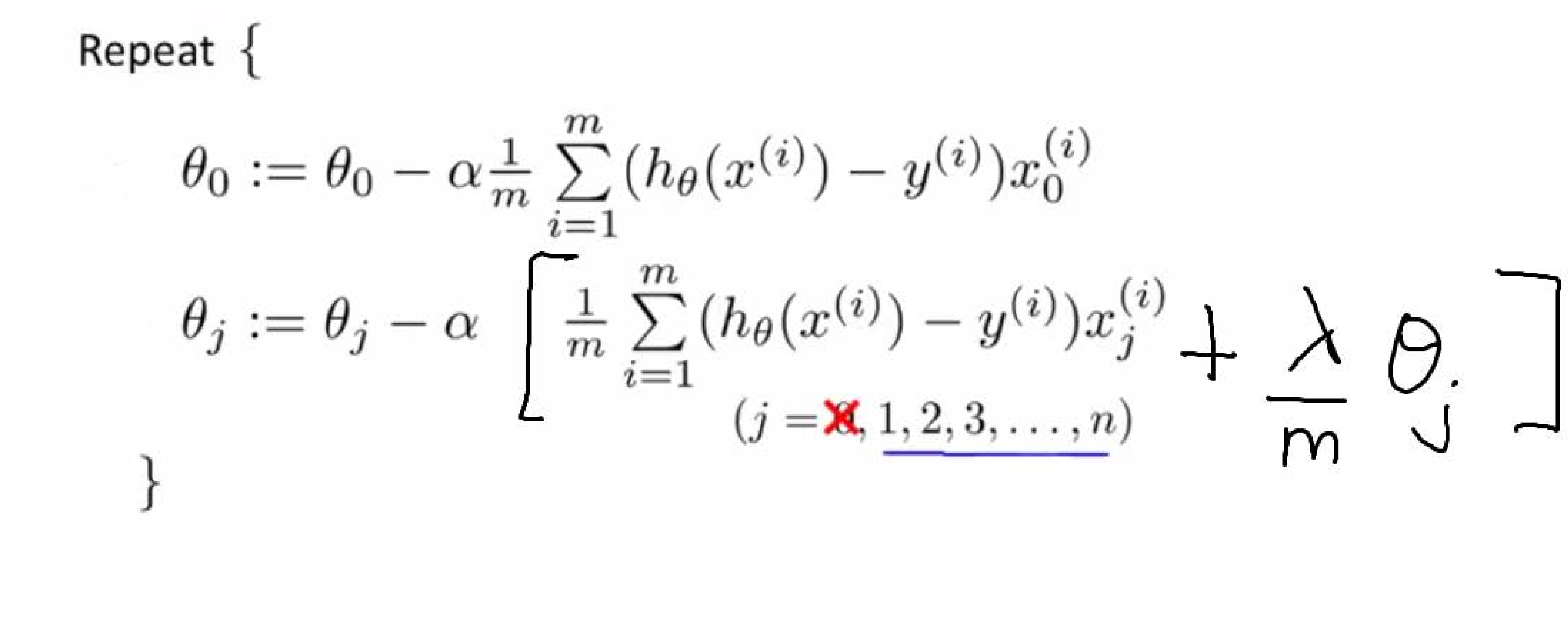
Repeat $\{$

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

$$(j = 0, 1, 2, 3, \dots, n)$$

$\}$

Gradient Descent with Regularization



Repeat {

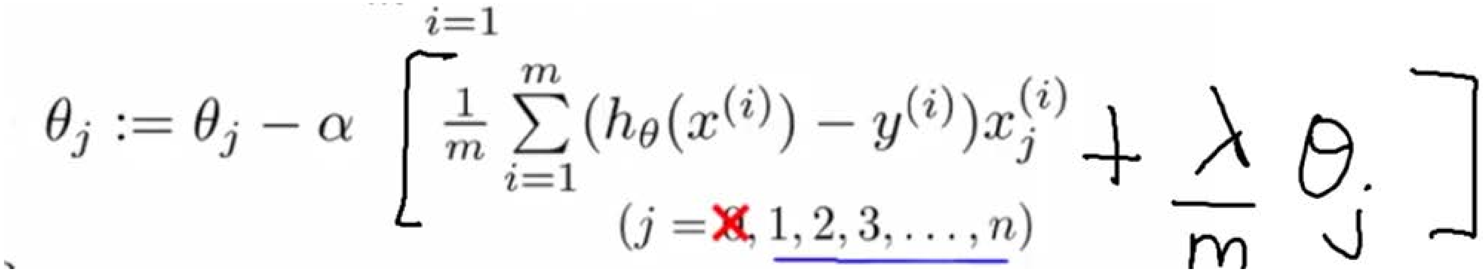
$$\theta_0 := \theta_0 - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_0^{(i)}$$

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

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

}

Gradient Descent with Regularization



$i=1$

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

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

‖

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

Regularized Logistic Regression

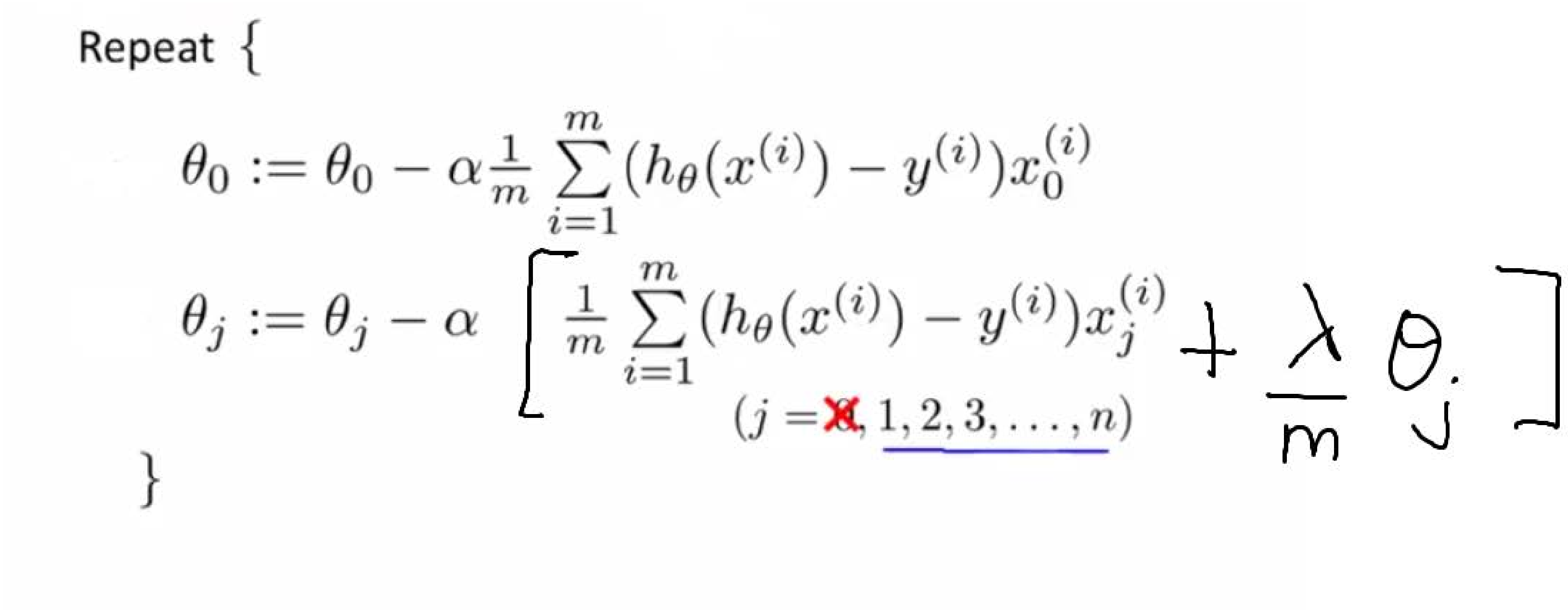
Recap: Previously without regularization

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

So now we add the regularization term

$$+\frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

Regularized Logistic Regression



Repeat {

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

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

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

}

Debugging

Plot $J(\Theta)$ and make sure it is decreasing over the iterations