Machine Learning: Regression and Gradient Descent

NYU Shanghai

Spring 2017

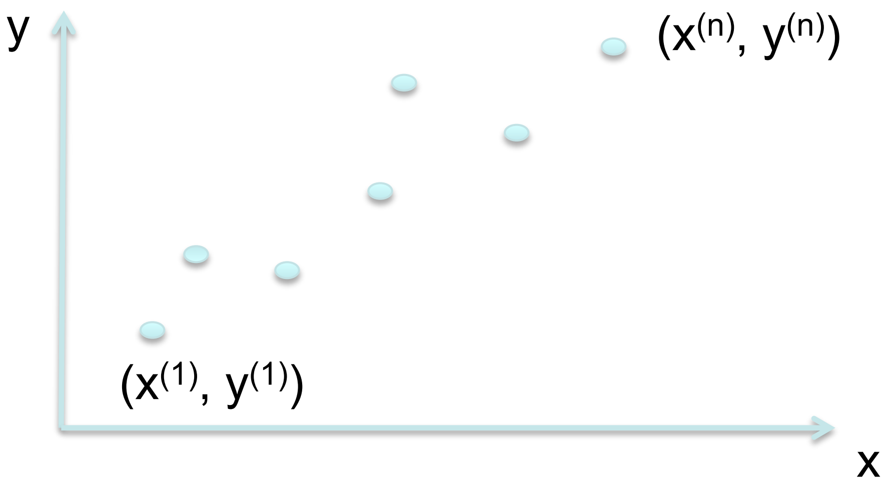
Some of the slides adapted from Andrew Ng

Regression and Gradient Descent

- Regression
  - Single variable and multi-variable
  - Linear and non-linear models
- Overfitting
- Gradient Descent
- Stochastic Gradient Descent

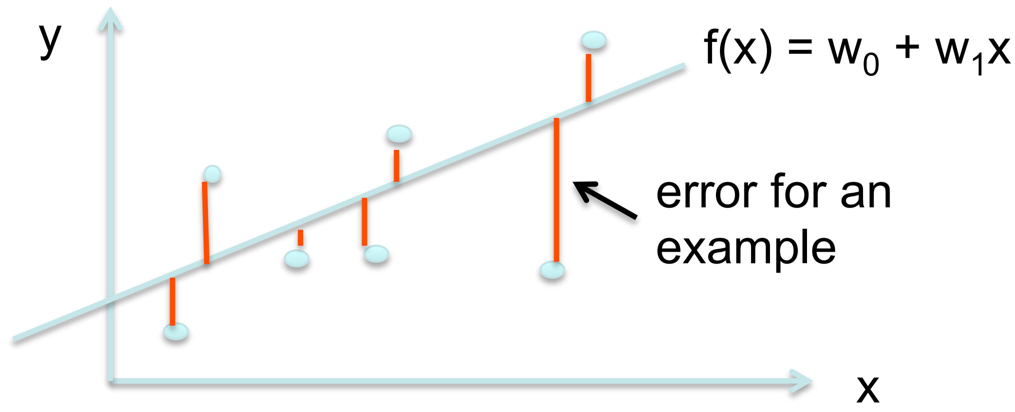
Linear Regression w/ One Variable (1)

- examples: $(x^{(i)}, y^{(i)})$, $i=1,\ldots,m$, where each $x^{(i)}$ and $y^{(i)}$ are real numbers.



- What if given new of value of x? How should we predict y?
- Use linear hypothesis $y = f(x) = w_0 + w_1x$
- How do we choose w_0 and w_1 ?

Linear Regression w/ One Variable (2)



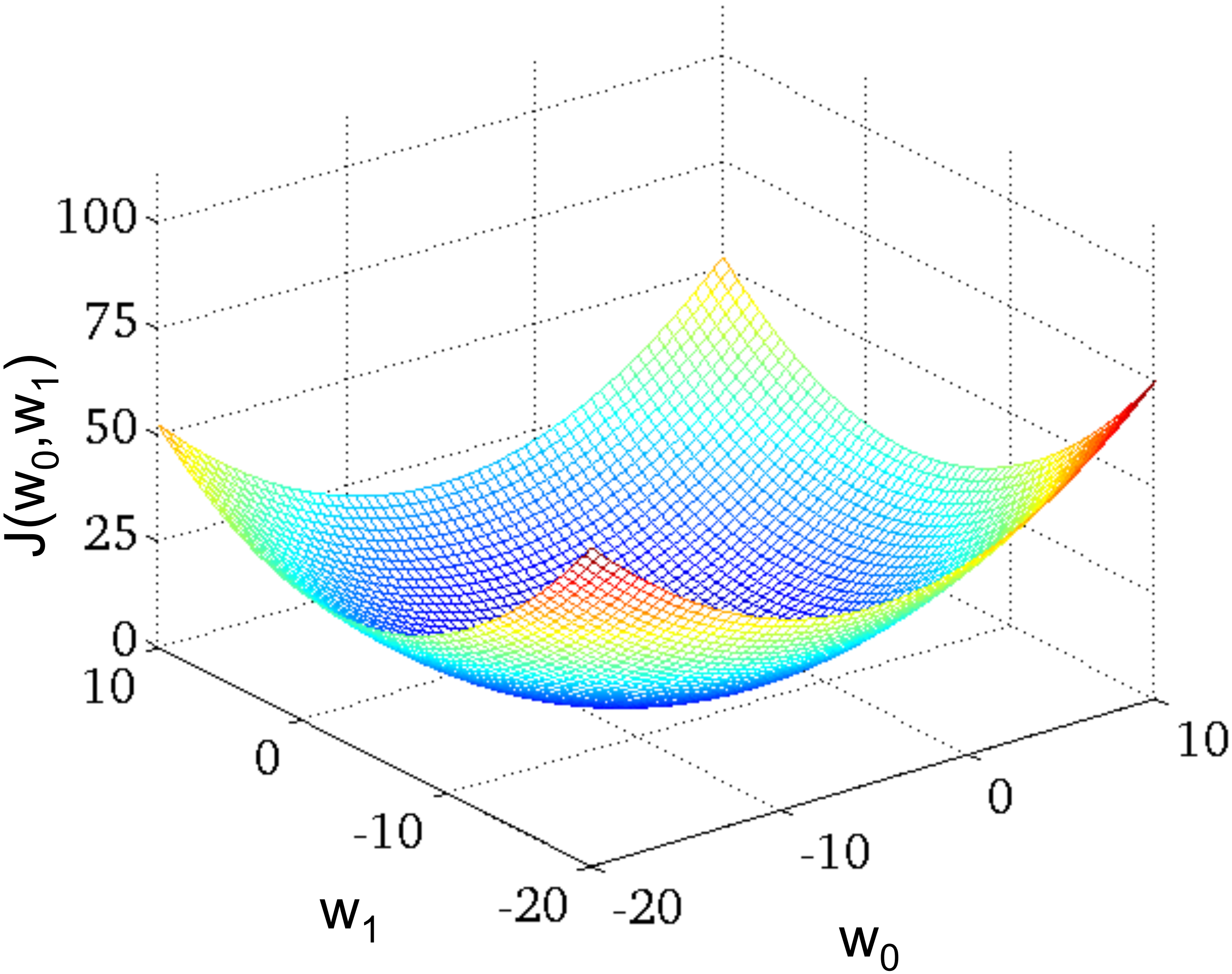
- What is the error for a given choice of w_0 and w_1?
- Error for i^{th} example = $y^{(i)} - [w_0 + w_1 x^{(i)}]$
- Squared error for i^{th} example = $(y^{(i)} - w_0 - w_1 x^{(i)})^2$
- mean squared error across all of the examples =

$$\frac{1}{m}\sum_{i=1}^{m}\left(y^{(i)} - w_0 - w_1 x^{(i)}\right)^2$$

Linear Regression w/ One Variable (3)

- To pick parameters w_0, w_1, natural thing to do is minimize Mean Squared Error (MSE).
- That is, choose w_1 and w_2 to minimize
  - J (w_0, w_1) = $$\frac{1}{2m}\sum_{i=1}^{m}\left(y^{(i)} - w_0 - w_1 x^{(i)}\right)^2$$

(The 2 in the denominator is just there for mathematical convenience. When minimizing a function, you can ignore multiplicative constants.)


100
75
50
25
0
$J(w_0,w_1)$
10
0
-10
-20
w_1
-20
-10
0
10
w_0

Linear Regression w/ One Variabe (4)

- **Theorem:** Given n examples $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}),\ldots,(x^{(m)},y^{(m)})$, the straight line $y = w_0+w_1x$ that minimizes the MSE error is:
  - $w_1 = \dfrac{\dfrac{\sum x^{(i)}y^{(i)}}{m} - \overline{x}\cdot\overline{y}}{\dfrac{\sum (x^{(i)})^2}{m} - \overline{x}^2}$
  - $w_0 = \overline{y} - w_1\overline{x}$

where $\overline{x} = \dfrac{1}{m}\sum_{i=1}^{m} x^{(i)}$ $\quad \overline{y} = \dfrac{1}{m}\sum_{i=1}^{m} y^{(i)}$

Linear Regression w/ One Variable (5)

- Average weight of a football player at U Texas:

year	weight (lb)
1905	164
1932	181
1945	192
1965	199

- Find w_0 and w_1 and that minimizes the MSE.
- Find predicted weight for 1970.

Linear Regression w/ One Variable (6)

- How do we prove the theorem?
- Need to choose w_0 and w_1 to minimize:

$$\frac{1}{2m}\sum_{i=1}^{m}\left(y^{(i)} - w_0 - w_1 x^{(i)}\right)^2$$

- Minimizing a function $J(w_0,w_1)$ of two variables.

Minimizing a Function of Multiple Variables

- Function of multiple variables: $J(w_0, w_1)$
- Partial Derivatives:

$$\frac{\partial}{\partial w_0} J(w_0, w_1) \qquad \frac{\partial}{\partial w_1} J(w_0, w_1)$$

- Under some convexity conditions, minimum occurs at (w_0^*, w_1^*) where

$$\frac{\partial}{\partial w_0} J(w_0^*, w_1^*) = 0 \qquad \frac{\partial}{\partial w_1} J(w_0^*, w_1^*) = 0$$

Let's go through it for MSE linear regression on whitebaord.

Degree-N Polynomial Regression w/ One Variable

- Consider now fitting a polynomial to the data: $f_{\mathbf{w}}(x) = w_o + w_1 x + w_2\, x^2 + \ldots + w_n x^n$
- $\mathbf{w} = (w_0, w_1, \ldots, w_n)$
- $J(\mathbf{w}) = (1/2m)\, \Sigma_i \,[\, y^{(i)} - f_{\mathbf{w}}(x^{(i)})\,]^2$

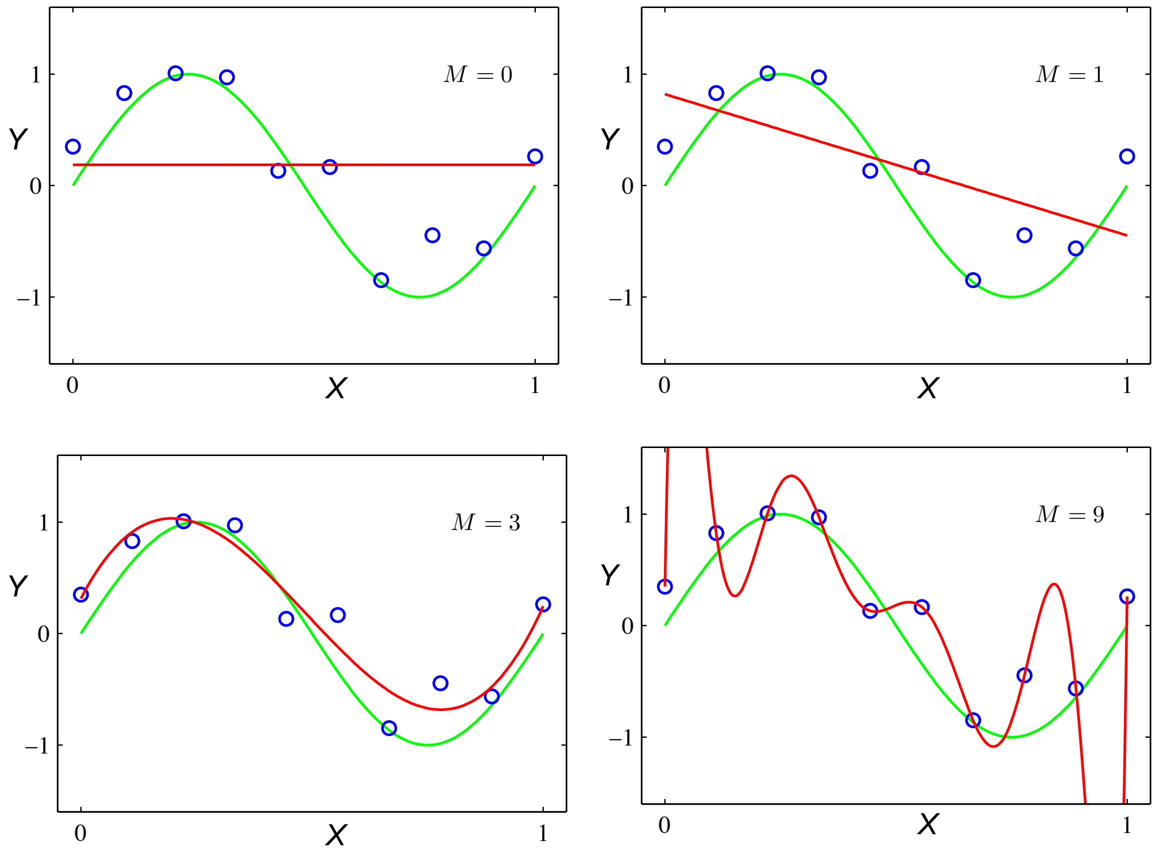
$= (1/2m)\Sigma_i\, [y^{(i)} - w_o - w_1 x^{(i)} - w_2\, x^{(i)2} - \ldots - w_n x^{(i)n}\,]^2$

Degree-N Polynomials

1 parameter

2 parameters

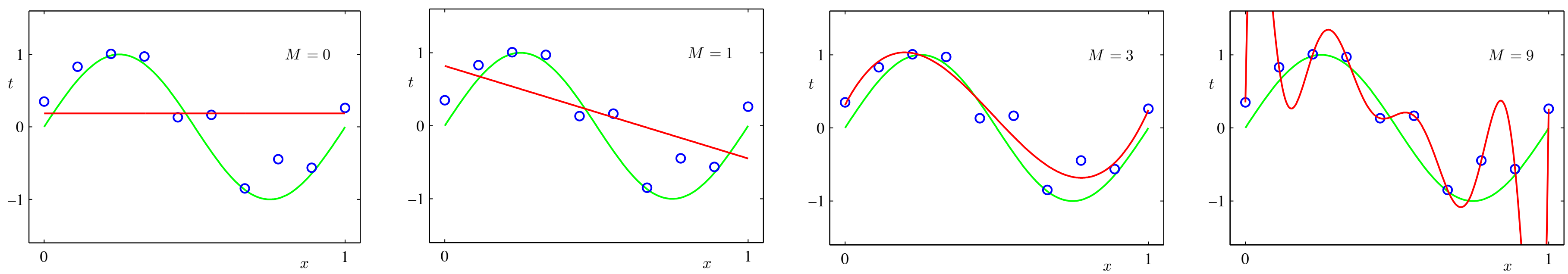
- Which one is **best**?



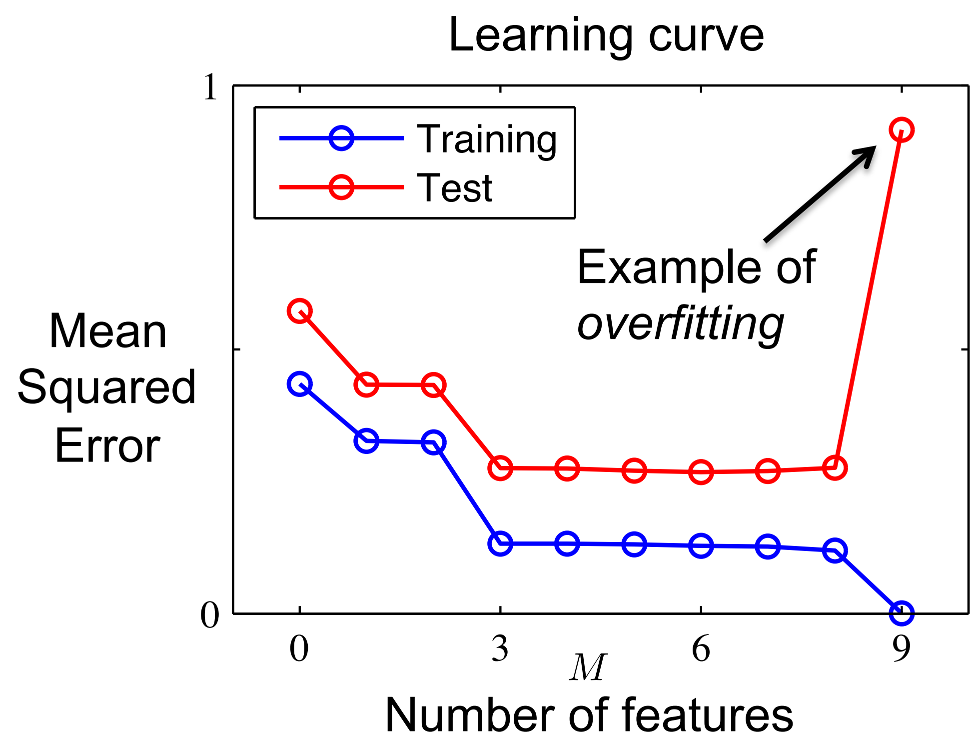
4 parameters

10 parameters

Degree-N Polynomials



Very rough rule of thumb:
If the ratio of the number of parameters (weights) to the number of training examples is large, can result in over-fitting.



Linear Regression w/ Multiple Variables

x_1	x_2	x_3	y
Living Area (ft^2)	No. of Bedrooms	Age of home	Prices (in $1000s)
2104	3	14	400
1600	3	32	330
2400	3	35	369
1416	2	41	232
….	….		…..

- Suppose have data for 60 houses.
- # of features = ?
- # parameters = ?
- m = training examples = ?

Linear Regression w/ Multiple Variables

- i^{th} example: $(x_1^{(i)}, x_2^{(i)},\ldots,x_n^{(i)} , y^{(i)})$
- $\mathbf{x}^{(i)} = (x_1^{(i)}, x_2^{(i)},\ldots,x_n^{(i)}),\ \ 1=1,\ldots,m$
- Linear Hypothesis (model):
  $y = f_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1x_1 + w_2x_2 +\ldots+ w_nw_n$

  $\mathbf{w} = (w_0,w_1,\ldots,w_n)$ (n+1 weights)
- *Note: same model as in spam email problem, but now we're doing regression.*
- Loss function:

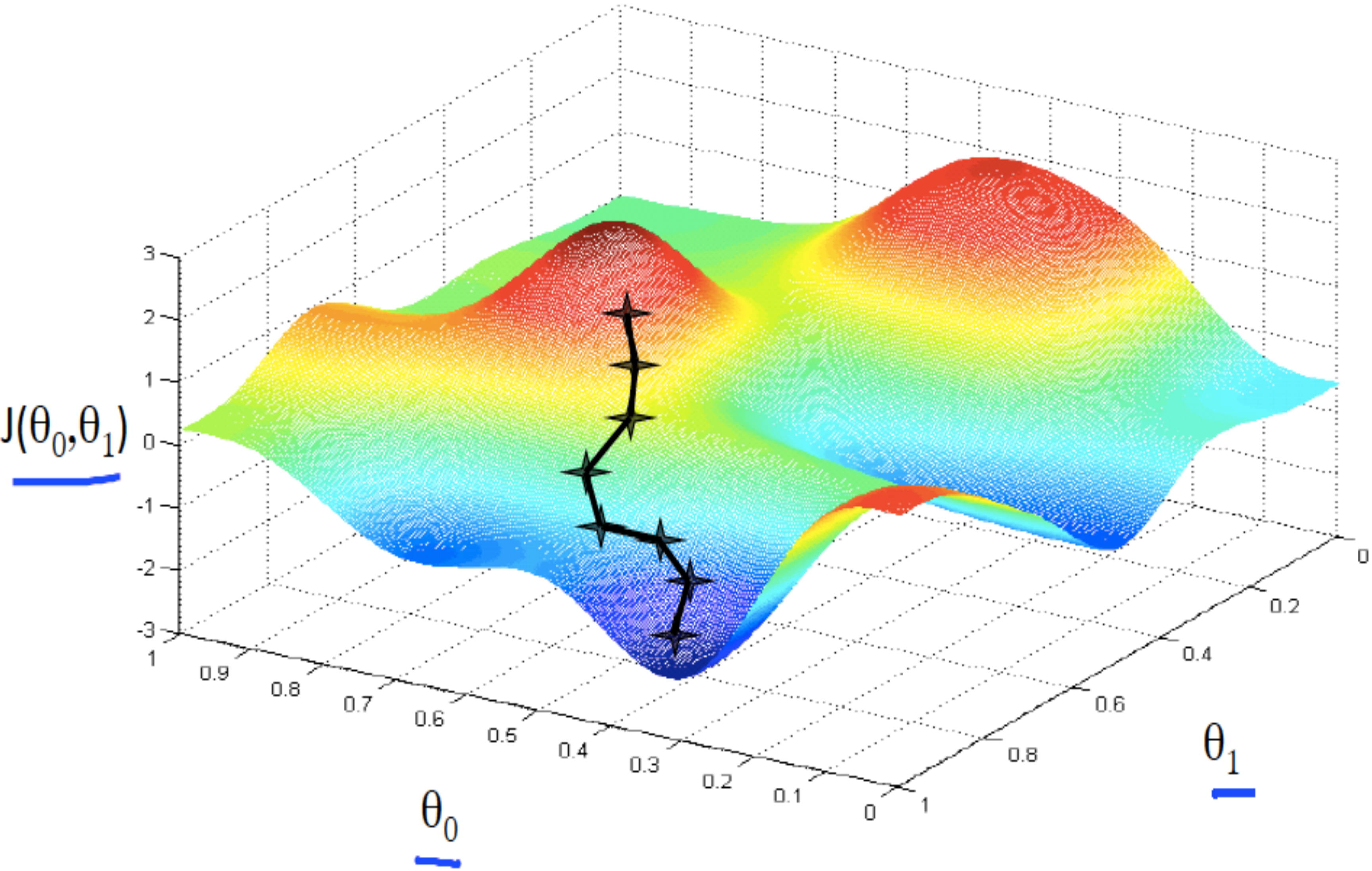
  $J(\mathbf{w}) = (1/2m)\ \Sigma_i\ [y^{(i)} - f_{\mathbf{w}}(\mathbf{x}^{(i)})\ ]^2$

Gradient Descent

- Goal: find **w** to minimize some function J(**w**)
- Iterative Approach: begin with some initial **w**, for example **w** = (0,0,…,0)
- Evaluate partial derivate of J(**w**) at current value of **w**.
- update $$w_j = w_j - \alpha \frac{\partial J(\mathbf{w})}{\partial w_j}$$
- Then update again…
- The gradient $$[\frac{\partial}{\partial w_1} J(\mathbf{w}), \dots, \frac{\partial}{\partial w_n} J(\mathbf{w})]$$

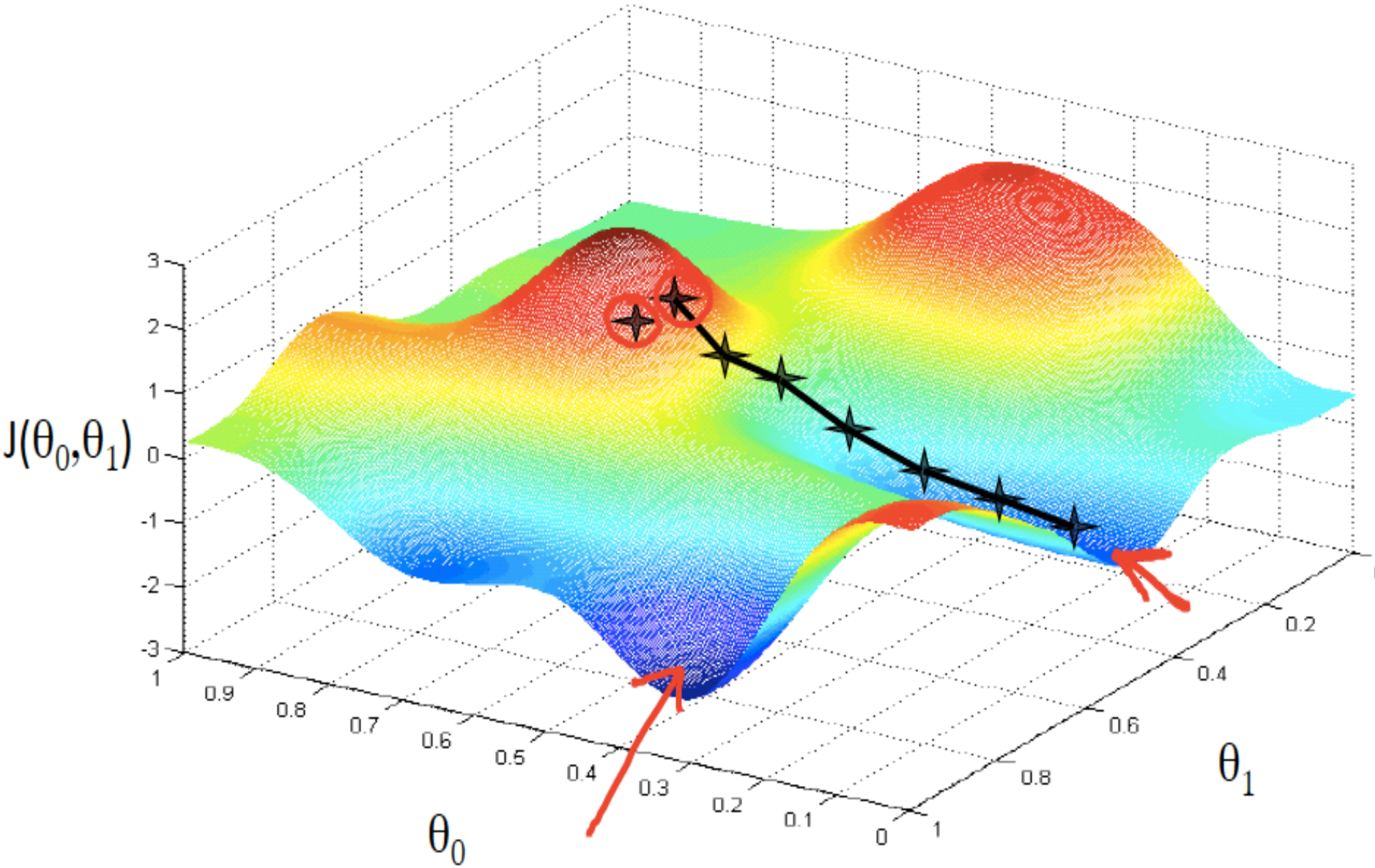
is the direction of steepest ascent

Gradient Descent: multiple local minima



Slide from Andrew Ng's Course

Gradient Descent: multiple local minima



Slide from Andrew Ng's Course

Gradient Descent / Linear Model

$$f_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + ... + w_n x_n$$

$$J(\mathbf{w}) = \frac{1}{2m} \sum_{i=1}^{m} [f_{\mathbf{w}}(\mathbf{x}^{(i)}) - y^{(i)}]^2$$

$$\frac{\partial J(\mathbf{w})}{\partial w_j} = \frac{1}{m} \sum_{i=1}^{m} [f_{\mathbf{w}}(\mathbf{x}^{(i)}) - y^{(i)}] x_j^{(i)}$$

$$w_j = w_j - \alpha \frac{\partial J(\mathbf{w})}{\partial w_j}$$

Gradient Descent Algorithm

$$f_{\mathbf{w}}(\mathbf{x}) = w_0 + w_1 x_1 + w_2 x_2 + ... + w_n x_n$$

Initialize $\mathbf{w}$

Repeat:

For j = 0,1,...,n

$$\widetilde{w}_j = w_j - \alpha \sum_{i=1}^{m} [f_{\mathbf{w}}(\mathbf{x}^{(i)}) - y^{(i)}] x_j^{(i)}$$

$$\mathbf{w} \leftarrow \widetilde{\mathbf{w}}$$

Gradient Descent: Succinct Notation

$$\mathbf{w} = \mathbf{w} - \alpha \nabla_{\mathbf{w}} J(\mathbf{w})$$

Gradient Descent Issues

- Scaling data:
  - Converges faster if the features have roughly the same range
  - For each feature, redefine by subtracting mean and dividing by standard deviation
  - When predicting, also need to scale
- Learning rate:
  - Too small, convergence rate slow.
  - Too big, may not converge at all

Stochastic Gradient Descent (SGD)

Initialize $\mathbf{w}$

Repeat:

For i = 1,2,...,m

For j = 0,1,...,n

$$\widetilde{w}_j = w_j - \alpha[f_{\mathbf{w}}(\mathbf{x}^{(i)}) - y^{(i)}]x_j^{(i)}$$

$$\mathbf{w} \leftarrow \widetilde{\mathbf{w}}$$

Each update of **w** uses a single data point $\mathbf{x}^{(i)}$!

Often better when there are many training examples

Homework Assignment

1. Derive some equations for gradient descent.
2. Use gradient descent & SGD to find optimal regression parameters for housing data with two variables and three parameters.
   - Several hundred data points in file
   - Need to first scale the data
3. Proof convergence of perceptron algo