Lecture 7 - Overfitting and Regularization

Dan Sheldon

September 26, 2012

Plan

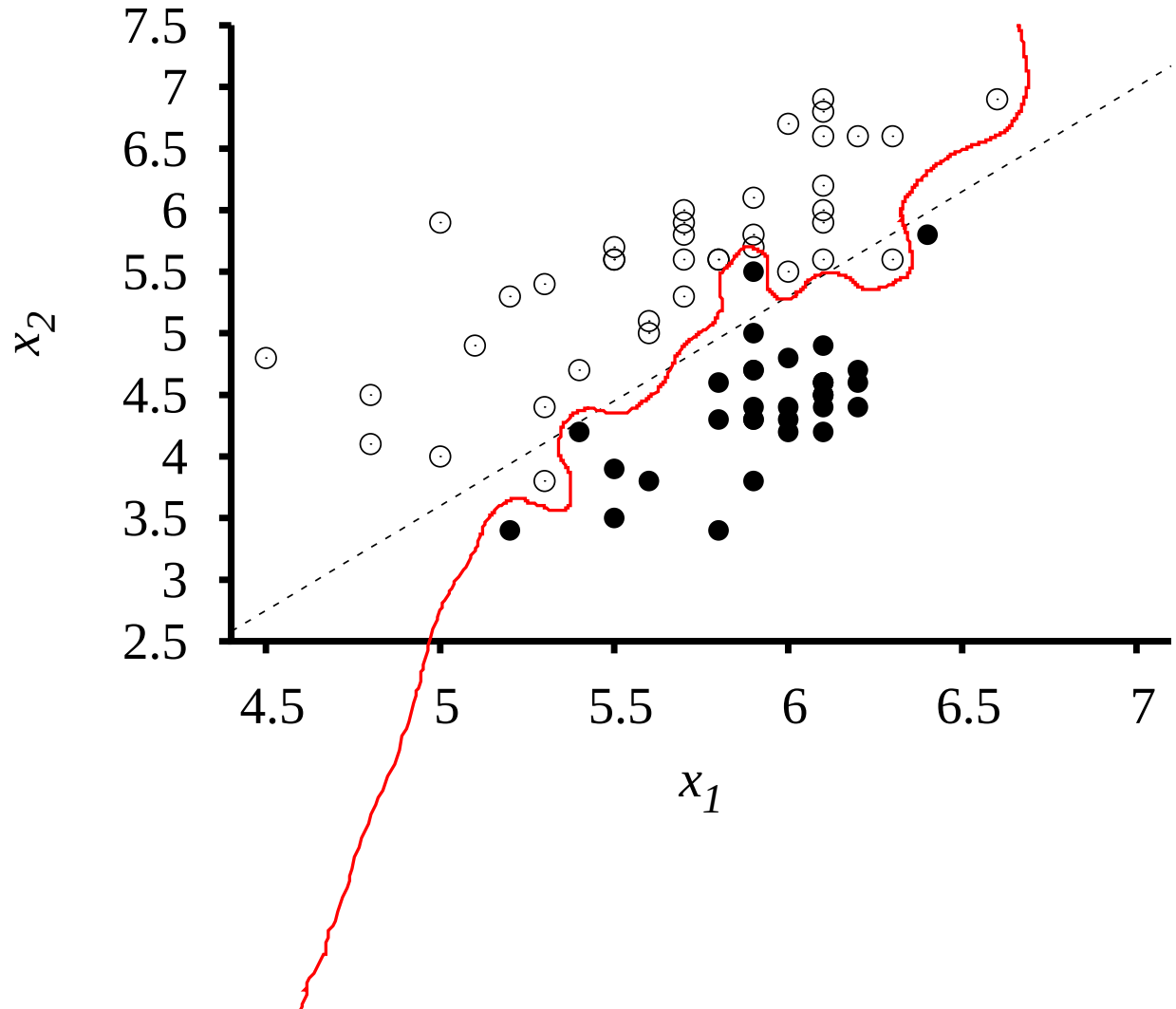
- What is Overfitting?
- How to Diagnose Overfitting
- Regularization

What is Overfitting?

Demo: polynomials

What is Overfitting?

Complex decision boundaries



What is Overfitting?

Overfitting is learning a model that fits the training data very well, but does not *generalize* well.

(Generalize well = predict accurately for new examples.)

How to Diagnose Overfitting?

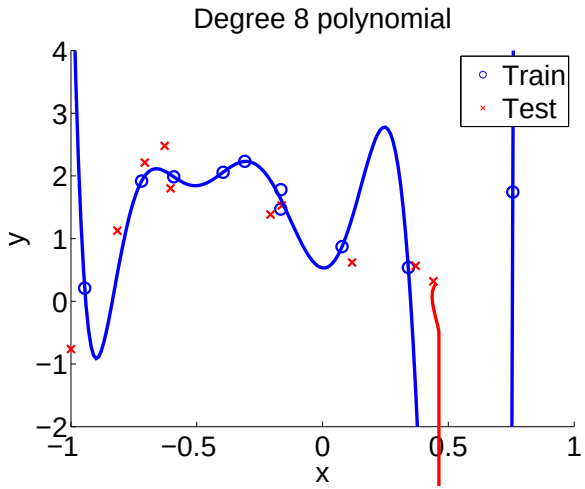
Exercise

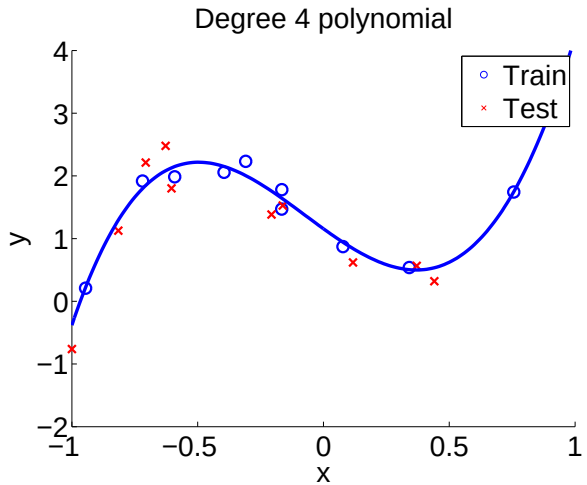
How to Diagnose Overfitting?

Exercise

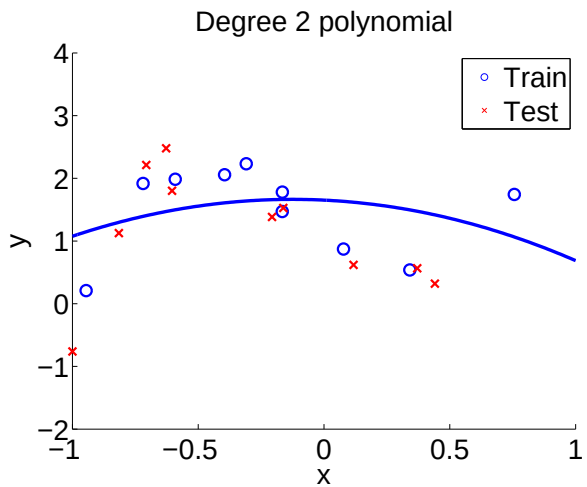
Reserve some data to test whether hypothesis generalizes well

overfit





underfit



Train Data vs. Test Data

Very important (and simple) methodology

- Start with N training examples

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)$$

- Split randomly into *train* and *test* sets
- To fit the model, minimize cost on *train* data only

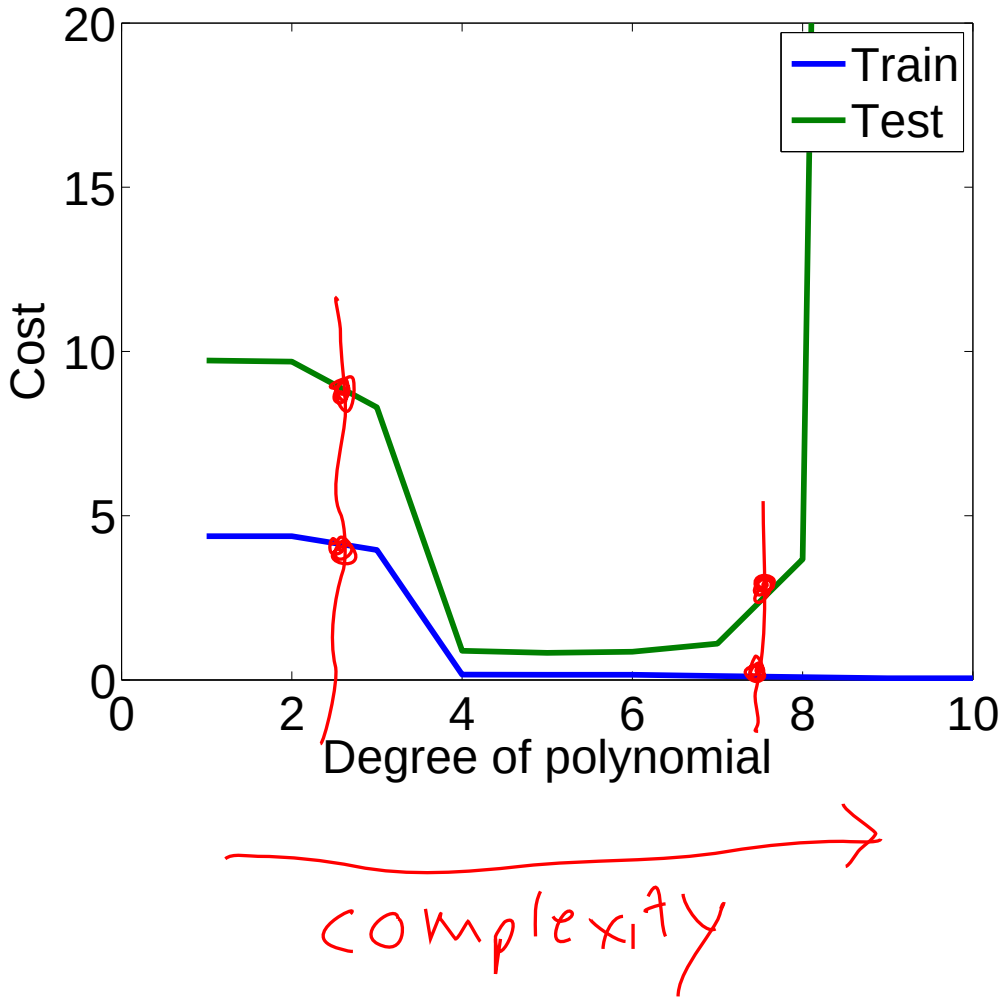
$$J_{\text{train}}(\mathbf{w}) = \sum_{i \in \text{train}} \text{cost}(h(\mathbf{x}_i), y_i)$$

- To evaluate the fit, measure cost on test set

$$J_{\text{test}}(\mathbf{w}) = \sum_{i \in \text{test}} \text{cost}(h(\mathbf{x}_i), y_i)$$

How to Diagnose Overfitting?

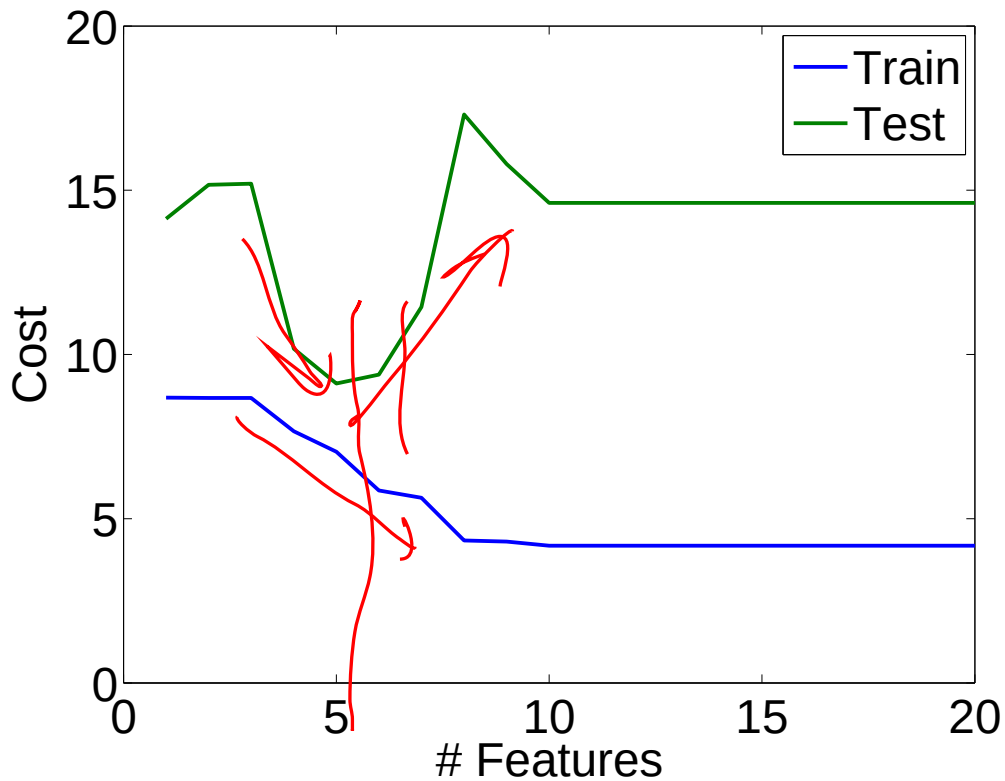
Example: cost function vs. degree of polynomial



How to Diagnose Overfitting?

$(x_1, x_2) \longmapsto (x_1, x_2, x_1^2, x_2^2, x_1 x_2)$

Example: cost function vs. number of features in book data

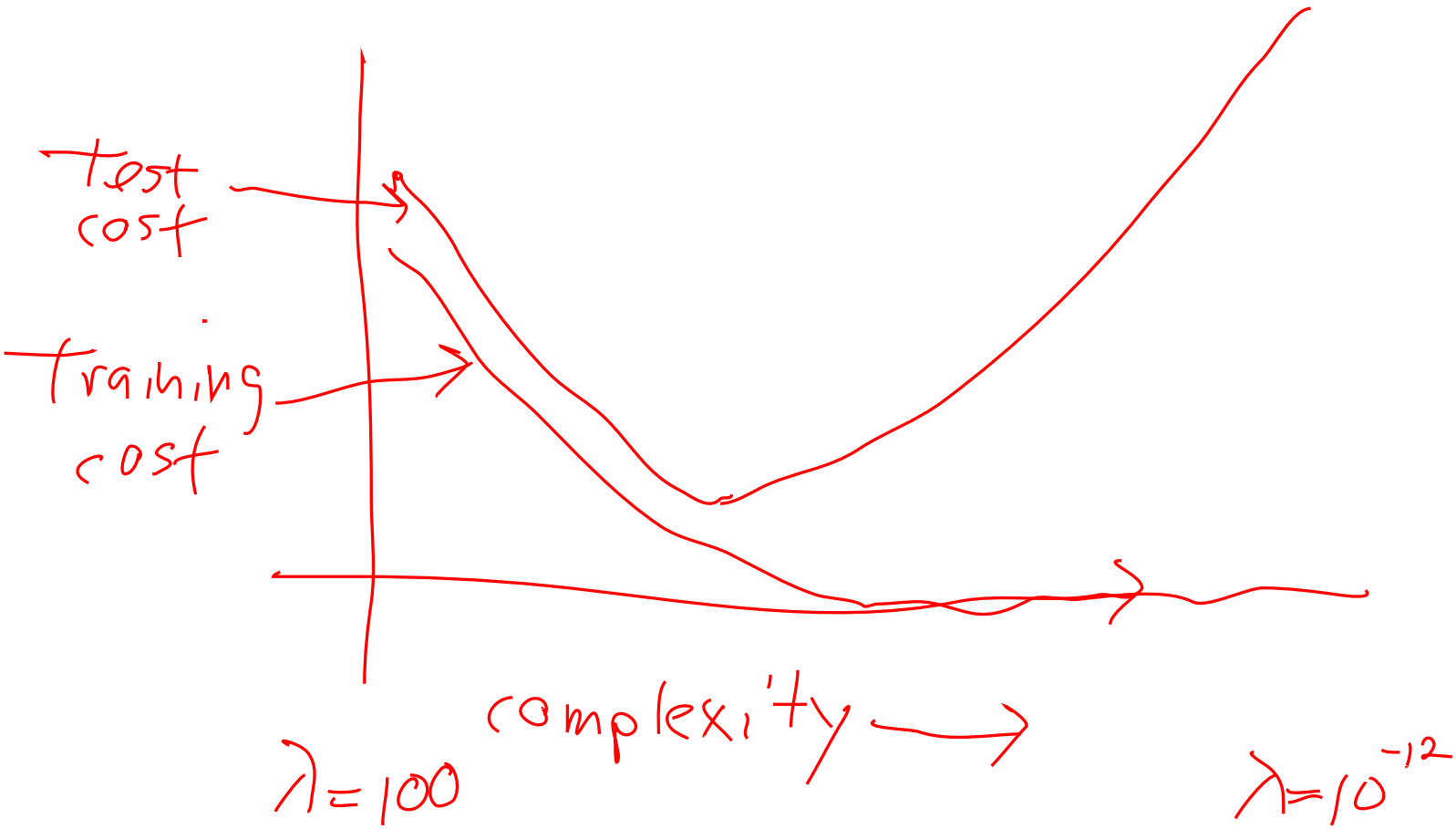


Cost vs. Complexity

~~$\lambda = 1, 2, 3, \ldots, 10$~~ use powers of 10

$\lambda = 10^{-12}, 10^{-11}, 10^{-10}, \ldots, 1, 10, 100, 1000$

General phenomenon: training/test cost vs. model "complexity"



What Makes a Model Complex?

- Polynomial: higher degree
- Book data: more features
- Linear functions ($h_{\mathbf{w}}(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$): large weights

$$h(x) = w_0 + w_1 x_1 + w_2 x_2 + \dots + w_d x_d$$

$$h(x) = w_0 + w_1 x + w_2 x^2 + \dots + w_d x^d$$

$w_{d-1} = 0$ $w_d = 0$

Large Weights

Example

Width	Thickness	Height	Weight
8	1.8	10	4.4
8	0.9	9	2.7
	...		

Which is more complex?

$$y = -3.94 + 0.18x_1 + .34x_2$$

vs.

$$y = 2842 - 957x_1 + 300x_2$$

Regularization (Linear Regression)

intercept term often not regularized

$w_0 + w_1 x_1 + \ldots + w_d x_d$

Intuition: large weights $\longrightarrow$ high complexity

So, modify the cost function to penalize large weights. For linear regression, the new cost function is:

$$J(\mathbf{w}) = \lambda \sum_{j=0}^{d} w_j^2 + \sum_{i=1}^{N} (h_{\mathbf{w}}(\mathbf{x}_i) - y_i)^2$$

new term $\rightarrow$ penalty on high weights

λ controls trade-off between model complexity and fit

$\lambda = .01 \rightarrow$ more emphasis on fit

$\lambda = 1000 \rightarrow$ more emphasis on weights

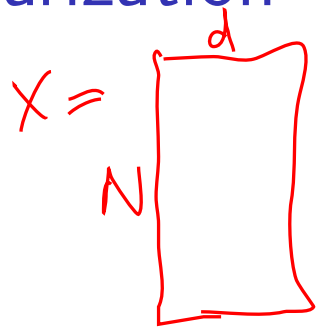
Discussion

Regularization is really important!!!

Why?

- A knob to turn to control complexity
- Don't need to know anything about problem, e.g., which features are not important

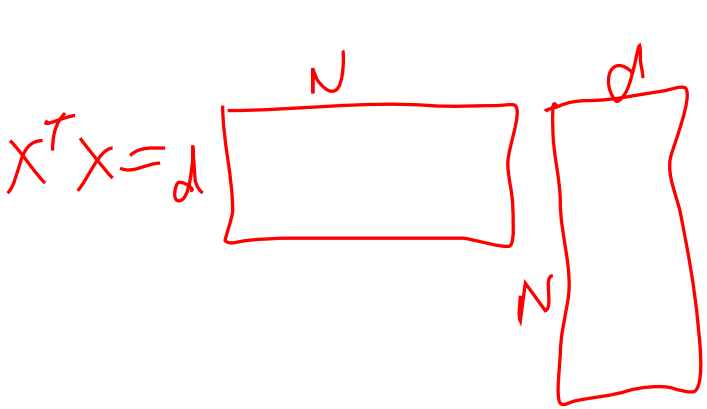
Normal Equations with Regularization



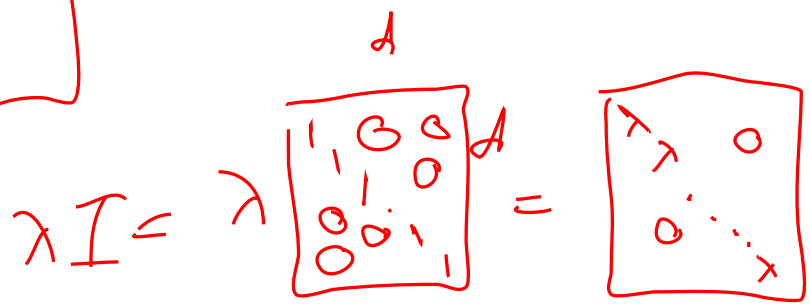
Old:

$$w = (X^T X)^{-1} X^T y$$

$$\mathbf{w} = (X^T X + \lambda I)^{-1} X^T y$$



I = d x d identity



Derivation (review on your own)

$$J(\mathbf{w}) = \sum_{i=1}^{N} (h_{\mathbf{w}}(\mathbf{x}_i) - y_i)^2 + \lambda \sum_{j=0}^{d} w_j^2$$
$$= (X\mathbf{w} - \mathbf{y})^T (X\mathbf{w} - \mathbf{y}) + \lambda \mathbf{w}^T \mathbf{w}.$$

Set derivative to zero

$$\begin{aligned}
0 &= \frac{d}{d\mathbf{w}} J(\mathbf{w}) \\
0 &= 2(X\mathbf{w} - \mathbf{y})^T X + 2\lambda \mathbf{w}^T \\
0 &= X^T (X\mathbf{w} - \mathbf{y}) + \lambda \mathbf{w} \\
X^T X\mathbf{w} + \lambda \mathbf{w} &= X^T \mathbf{y} \\
(X^T X + \lambda I)\mathbf{w} &= X^T \mathbf{y} \\
\mathbf{w} &= (X^T X + \lambda I)^{-1} X^T \mathbf{y}
\end{aligned}$$

Regularized Gradient Descent for Linear Regression

$$J(\mathbf{w}) = \lambda \sum_{j=0}^{d} w_j^2 + \sum_{i=1}^{N} (h_{\mathbf{w}}(\mathbf{x}_i) - y_i)^2$$

Repeat until convergence

$$w_j \leftarrow w_j - \alpha \frac{\partial}{\partial w_j} J(\mathbf{w}), \qquad j = 0, \ldots, d.$$

Regularized Gradient Descent for Linear Regression

$\frac{d}{dw_j} \lambda w_j^2 = 2\lambda w_j$

$$J(\mathbf{w}) = \lambda \sum_{j=0}^{d} w_j^2 + \sum_{i=1}^{N} (h_{\mathbf{w}}(\mathbf{x}_i) - y_i)^2$$

Repeat until convergence

$$w_j \leftarrow w_j - \alpha \frac{\partial}{\partial w_j} J(\mathbf{w}), \qquad j = 0, \ldots, d.$$

$$w_j \leftarrow w_j - \alpha \left(2\lambda w_j + 2 \sum_{i=1}^{N} (h_{\mathbf{w}}(\mathbf{x}_i) - y_i) x_{i,j} \right)$$

Regularized Gradient Descent for Linear Regression

$$J(\mathbf{w}) = \lambda \sum_{j=0}^{d} w_j^2 + \sum_{i=1}^{N} (h_{\mathbf{w}}(\mathbf{x}_i) - y_i)^2$$

Repeat until convergence

$$w_j \leftarrow w_j - \alpha \frac{\partial}{\partial w_j} J(\mathbf{w}), \qquad j = 0, \ldots, d.$$

$$w_j \leftarrow w_j - \alpha \left(2\lambda w_j + 2 \sum_{i=1}^{N} (h_{\mathbf{w}}(\mathbf{x}_i) - y_i) x_{i,j} \right)$$

$$w_j \leftarrow w_j (1 - 2\lambda\alpha) - 2\alpha \sum_{i=1}^{N} (h_{\mathbf{w}}(\mathbf{x}_i) - y_i) x_{i,j}$$

0.99

shrinking

old gradient step

Regularized Gradient Descent for Logistic Regression

$$J(\mathbf{w}) = \lambda \sum_{j=0}^{d} w_j^2 + \sum_{i=1}^{N} (h_{\mathbf{w}}(\mathbf{x}_i) - y_i)^2$$

$(-y_i \log h(x_i) - (1-y_i) \log(1-h(x_i))$

Repeat until convergence. For $j = 0, \ldots, d$:

$$w_j = w_j(1 - 2\lambda\alpha) - 2\alpha \sum_{i=1}^{N} (h_{\mathbf{w}}(\mathbf{x}_i) - y_i) x_{i,j}.$$

shrinking

gradient of old cost function

What You Need To Know

- Concept of overfitting
- Diagnosis: train/test sets
- Regularized cost function (penalize weights)
- Regularized gradient descent
- See it work