REGRESSION

INTRODUCTION TO DATA SCIENCE

TIM KRASKA

teaching datascience .org

Recap

Data Management

- Relational Model
- SQL
- Star Schema
- ..

Data Clearning

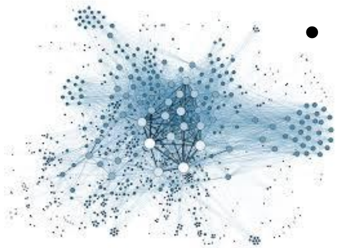
- Schema Matching
- Entity Resolution
- Data Fusion
- Jaccard Similarity
- IDF Weight
- …

Information Retrieval

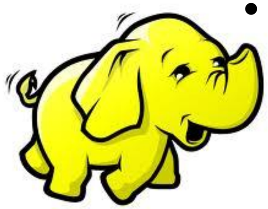
- Text Search
- Tokenization/Stemming
- Inverted Index
- Ranking
- Precision/Recall
- …

Probabilities and Statistics

- Law of Large Numbers
- Central Limit Theorem
- Testing
- …



- **Visualization**
  - Types
  - Senses
  - Design goals
  - …



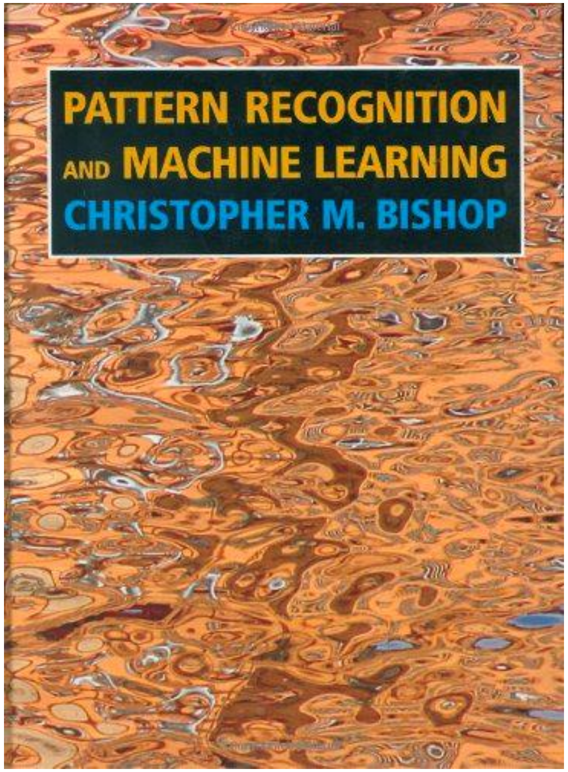
- **Map/Reduce**
  - Abstraction
  - Implementation
  - Spark
  - …

- **Machine Learning**
  - Naïve Bayes
  - Kmeans
  - Page Rank
  - Testing/Cross-Validation
  - ..

MATRIX

	Supervised Learning	*Unsupervised Learning*
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

RECOMMENDED READING



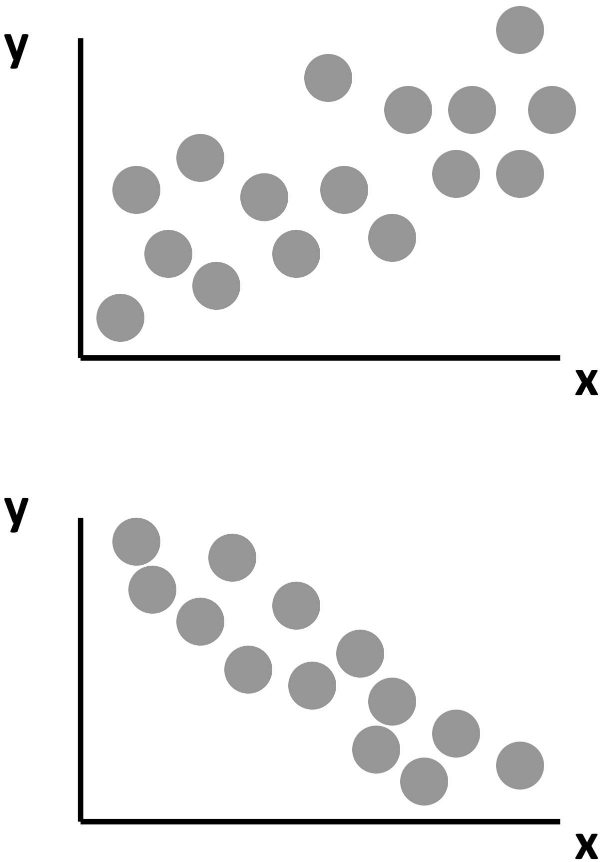
Chapter 3
(math heavy, but complete)

A bit easier to digest:
http://cs229.stanford.edu/notes/cs229-notes1.pdf

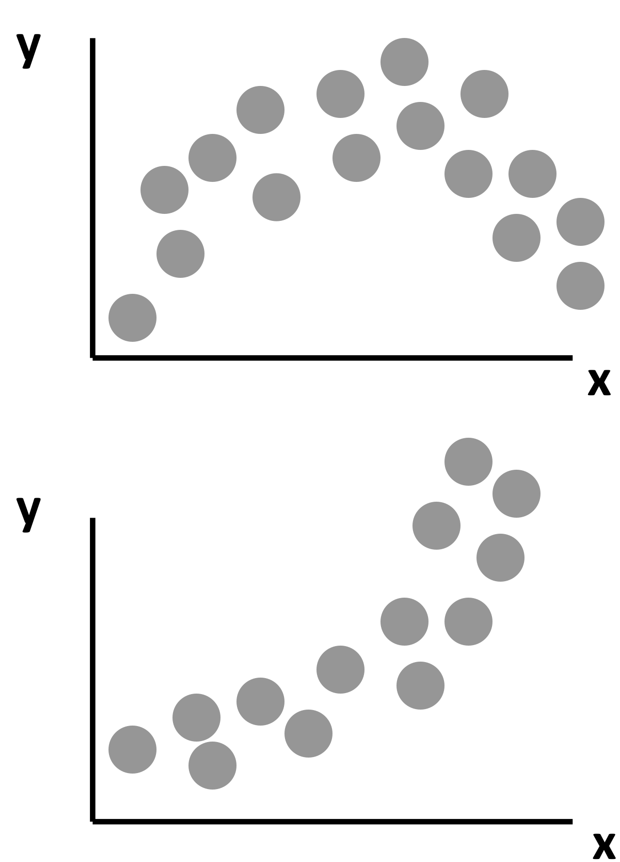
LINEAR REGRESSION

SCATTER PLOT EXAMPLES

Linear relationships



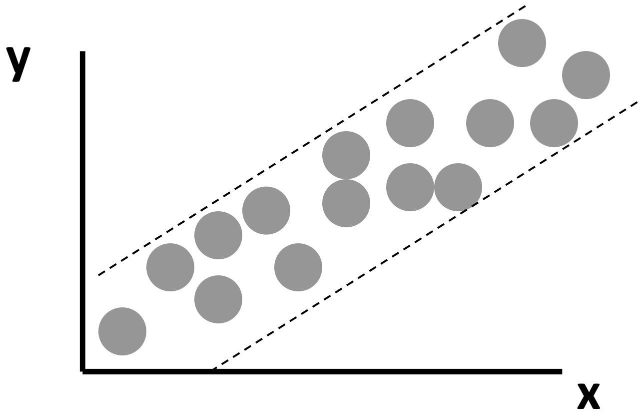
Curvilinear relationships

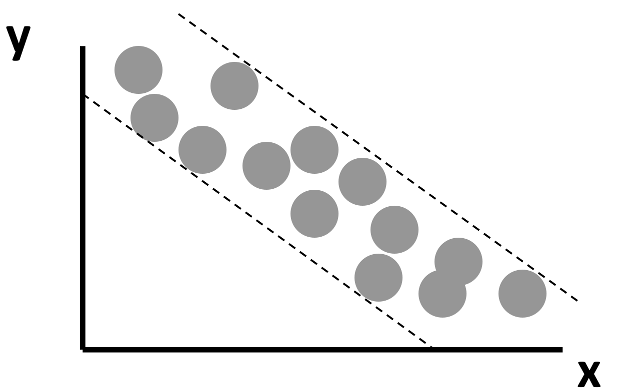


SCATTER PLOT EXAMPLES

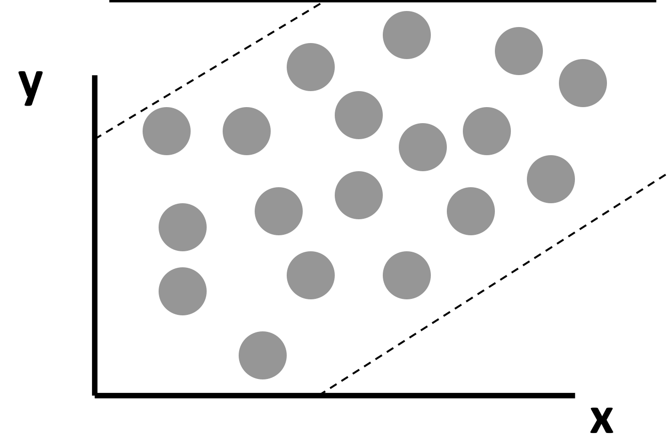
(continued)

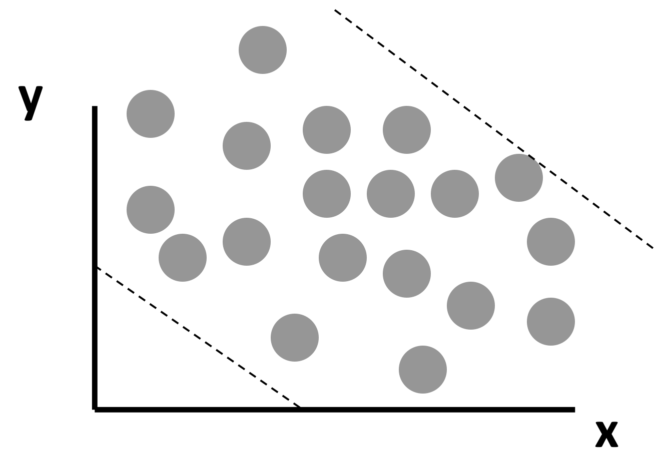
Strong relationships





Weak relationships

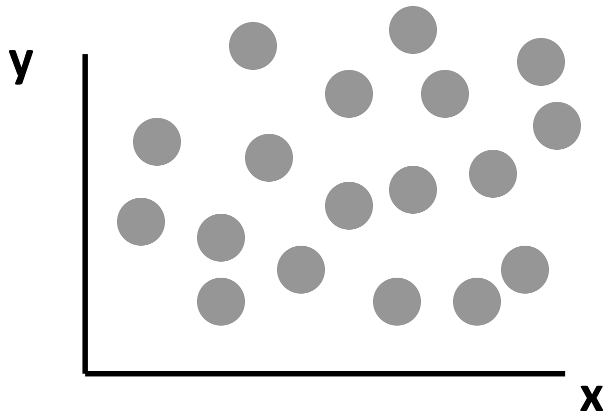


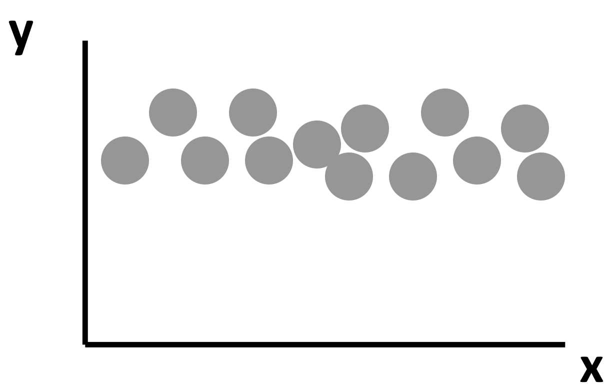


SCATTER PLOT EXAMPLES

(continued)

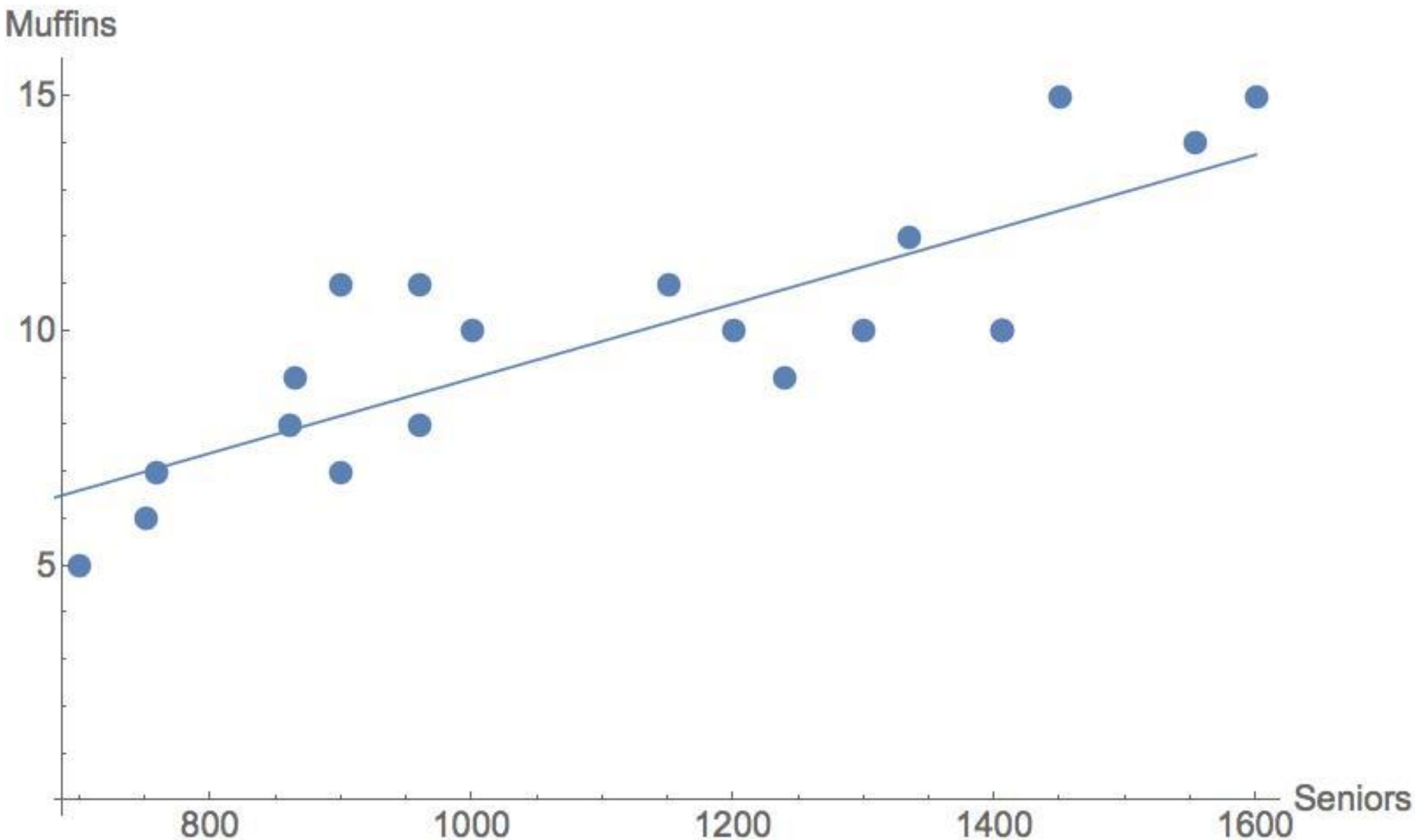
No relationship





Naked Cup-Cake Run

Naked Cup-Cake Run



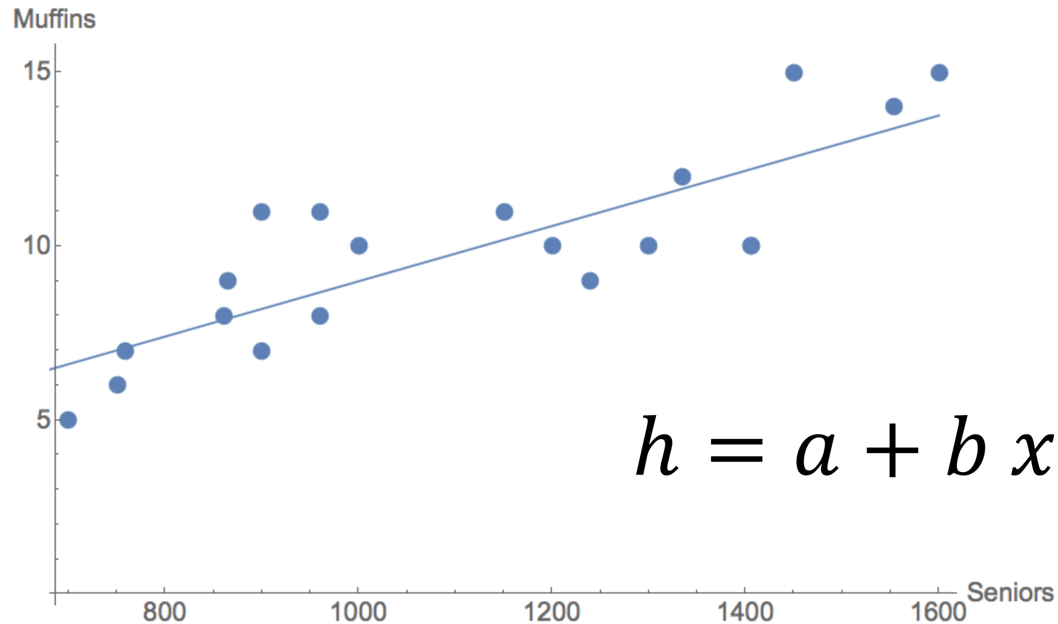
Naked Cup-Cake Run

How do we represent the problem:

$$q = \begin{pmatrix} q_0 \\ q_1 \end{pmatrix}$$

$$x = \begin{pmatrix} 1 \\ x_1 \end{pmatrix} = \begin{pmatrix} 1 \\ Seniors \end{pmatrix}$$

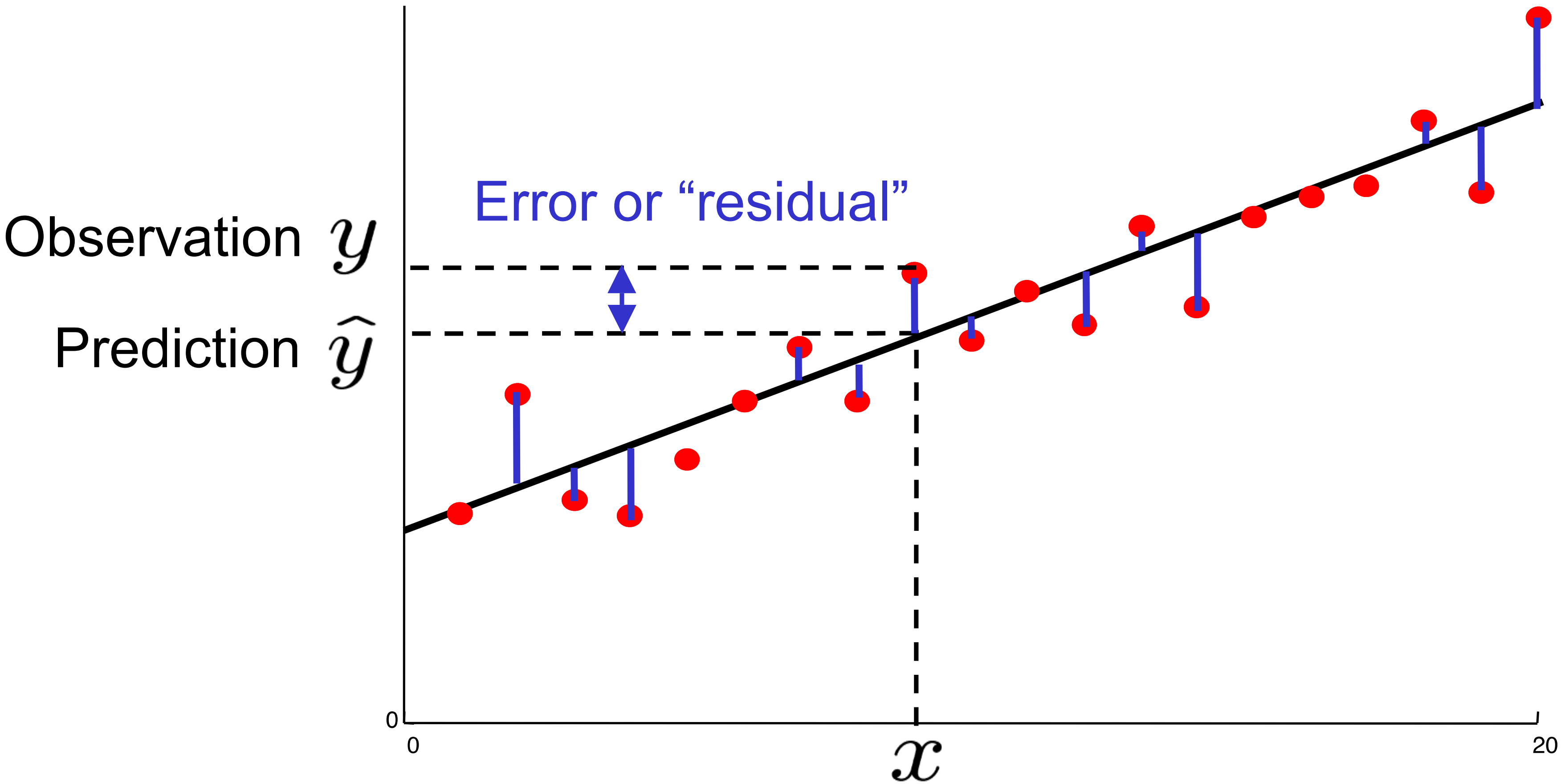
intercept term (x_0=1)



Hypothesis:

$$h = a + b\,x$$
$$h_\Theta = \Theta_0 x_0 + \Theta_1 x_1$$
$$h_\Theta = \Theta^T x$$

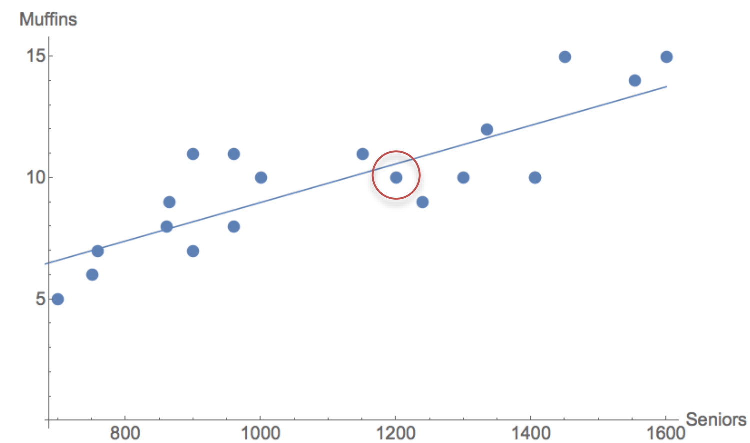
Idea: Chose θ so that $h_\theta(x)$ is close to h (muffins) for training examples (Seniors, Muffins)


Error or “residual”
Observation y
Prediction $\widehat{y}$
0
0
x
20

Naked Cup-Cake Run

Hypothesis:

$$h_\Theta(x) = \Theta^T x$$
$$h_\Theta(x) = \Theta_0 x_0 + \Theta_1 x_1$$



Notation:

$$x^i = \begin{pmatrix} 1 \\ x_1^i \\ \vdots \\ x_n^i \end{pmatrix}$$

Example

$$x^{2o} = \begin{pmatrix} 1 \\ 1200 \end{pmatrix}$$
$$y^{20} = 9$$

Minimize: Squared Error (there are others)

#Data Points

Prediction

Observed

$$\min_q J(q) = \frac{1}{m} \sum_{i=1}^{m} \left(h_q(x^{(i)}) - y^{(i)}\right)^2$$

Cost-Function

Minimize: Squared Error (there are others)

#Data Points

Prediction

Observed

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

Cost-Function

Alternative Versions of Squared Error Cost-Functions

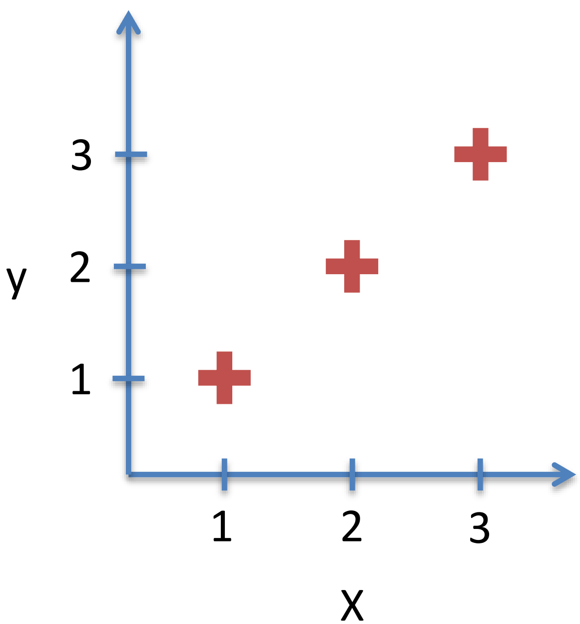
$$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_{\theta}(x^{(i)}) - y^{(i)}\right)^2$$

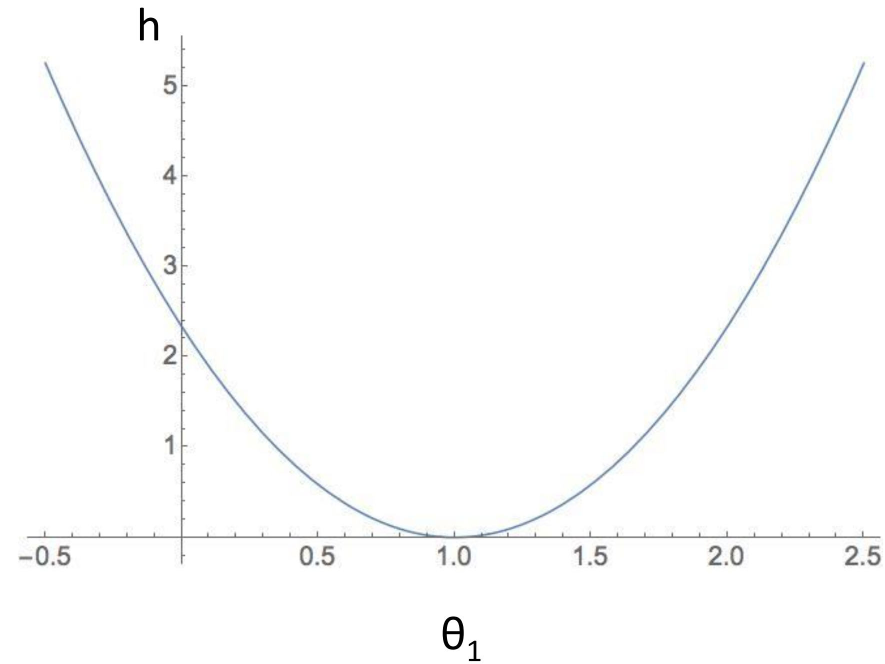
$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}\left(h_{\theta}(x^{(i)}) - y^{(i)}\right)^2$$

1/2 is just for mathematical convenience

The Shape of the Cost Function

Lets assume θ_0=0: $h_{\theta_1} = \theta_1 x_1$

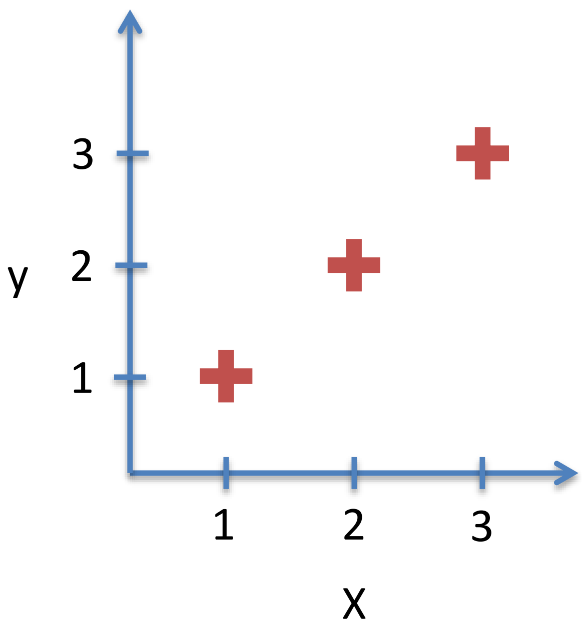




$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

Clicker Question

Suppose our hypothesis is $h_{\theta_1} = \theta_1 x_1$ and we have the following training set (m=3):



Our cost function is:

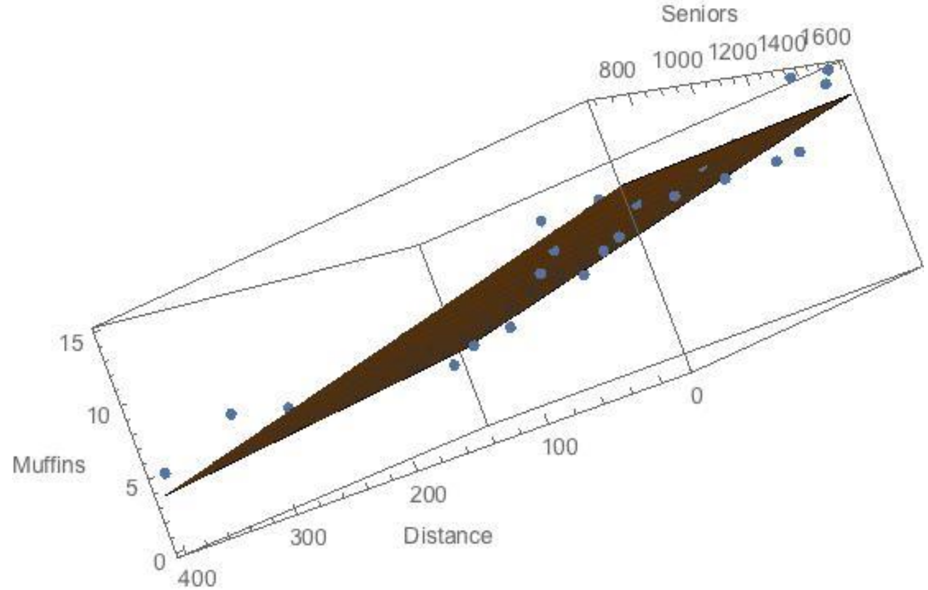
$$J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

What is the cost of θ_1=0

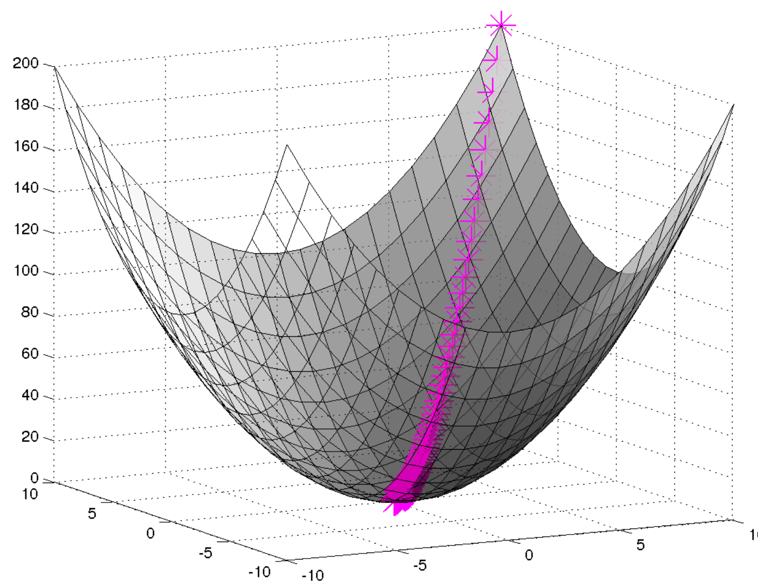
a) 0
b) 1/6
c) 1
d) 14/6

Works also in Higher Dimensions

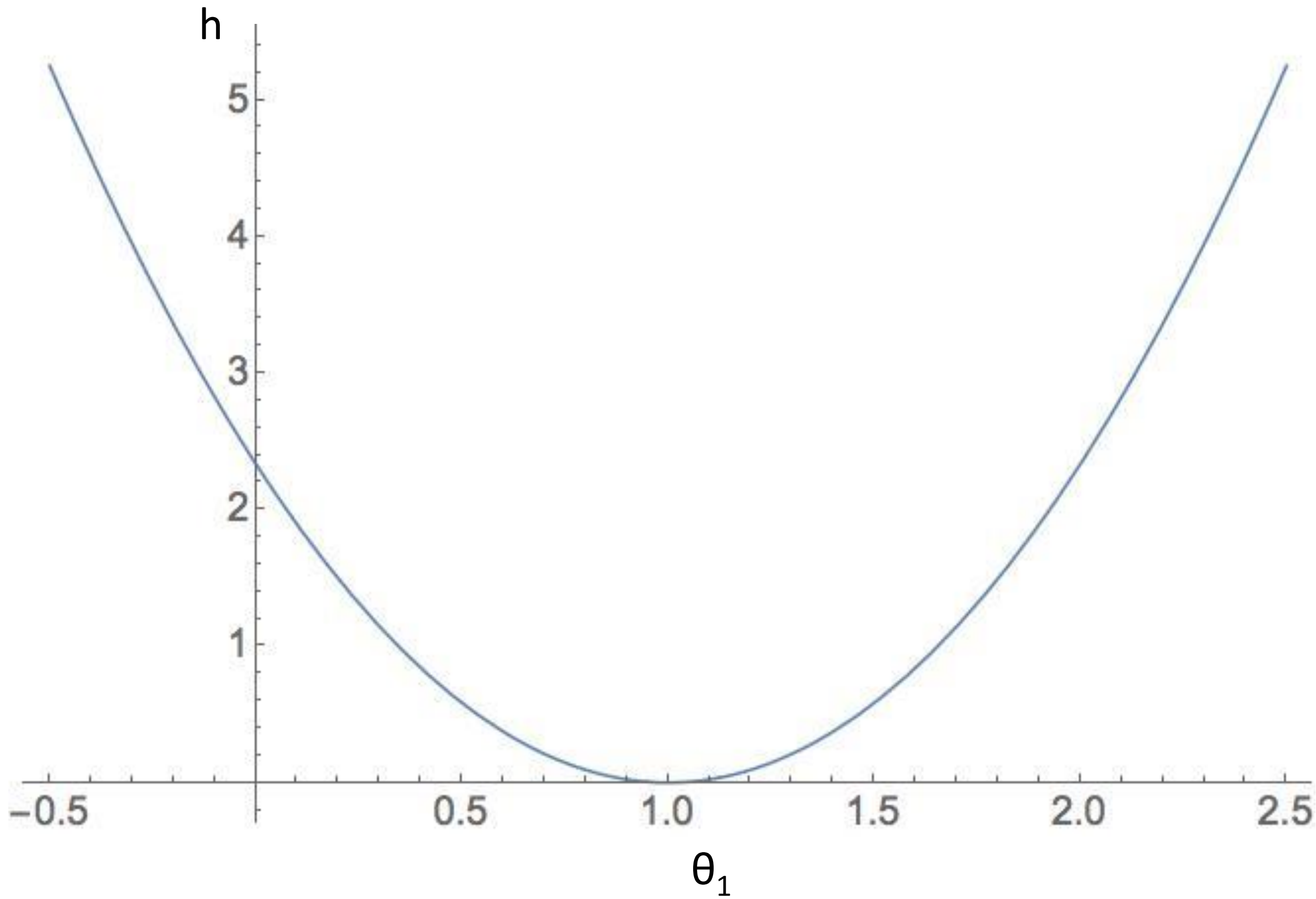
$$h_q(x) = q^T x$$



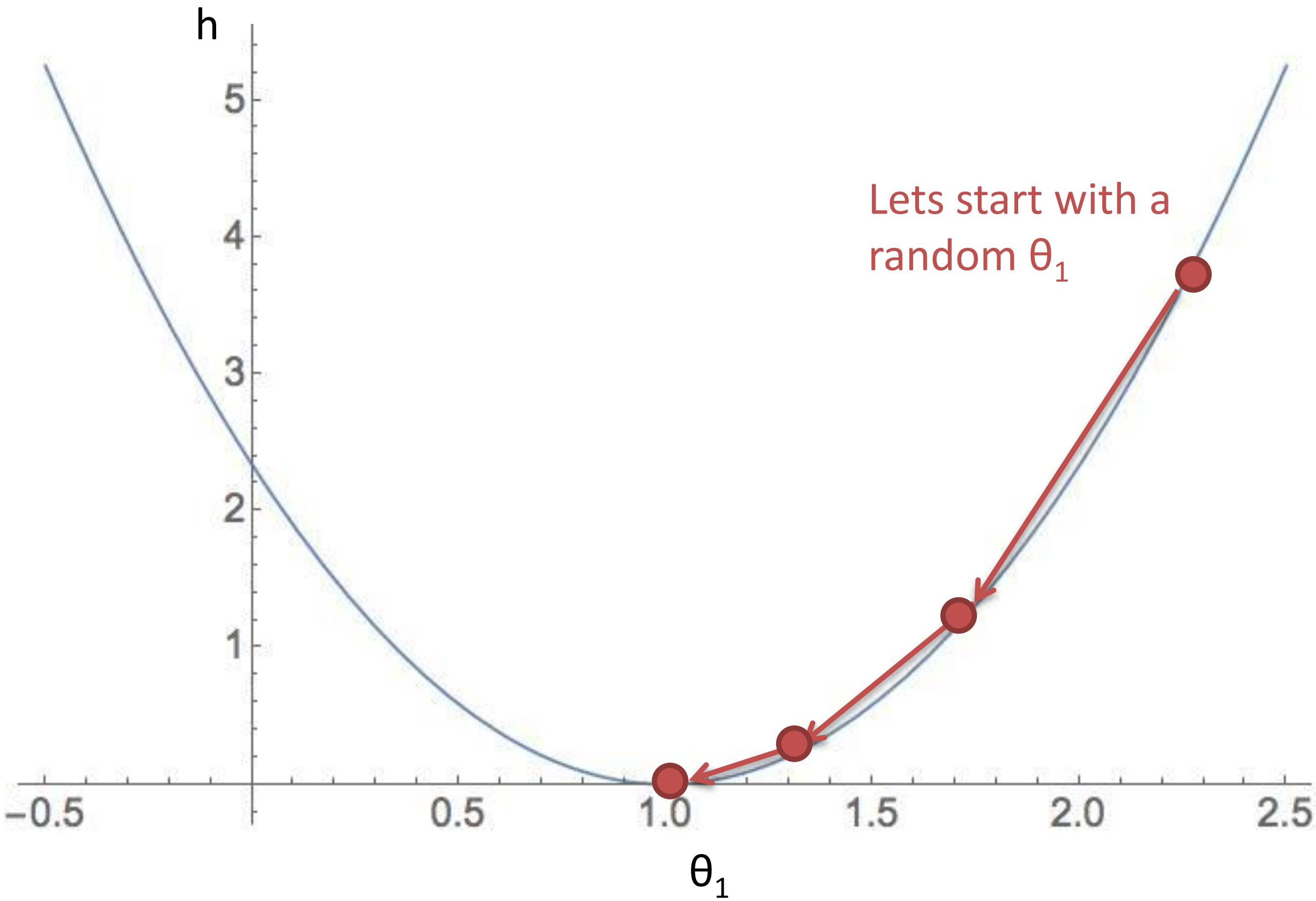
$$J(q_1) = \frac{1}{2m} \mathring{a}_{i=1}^{m} \left(h_q(x^{(i)}) - y^{(i)}\right)^2$$



How Can We Find The Minimum?



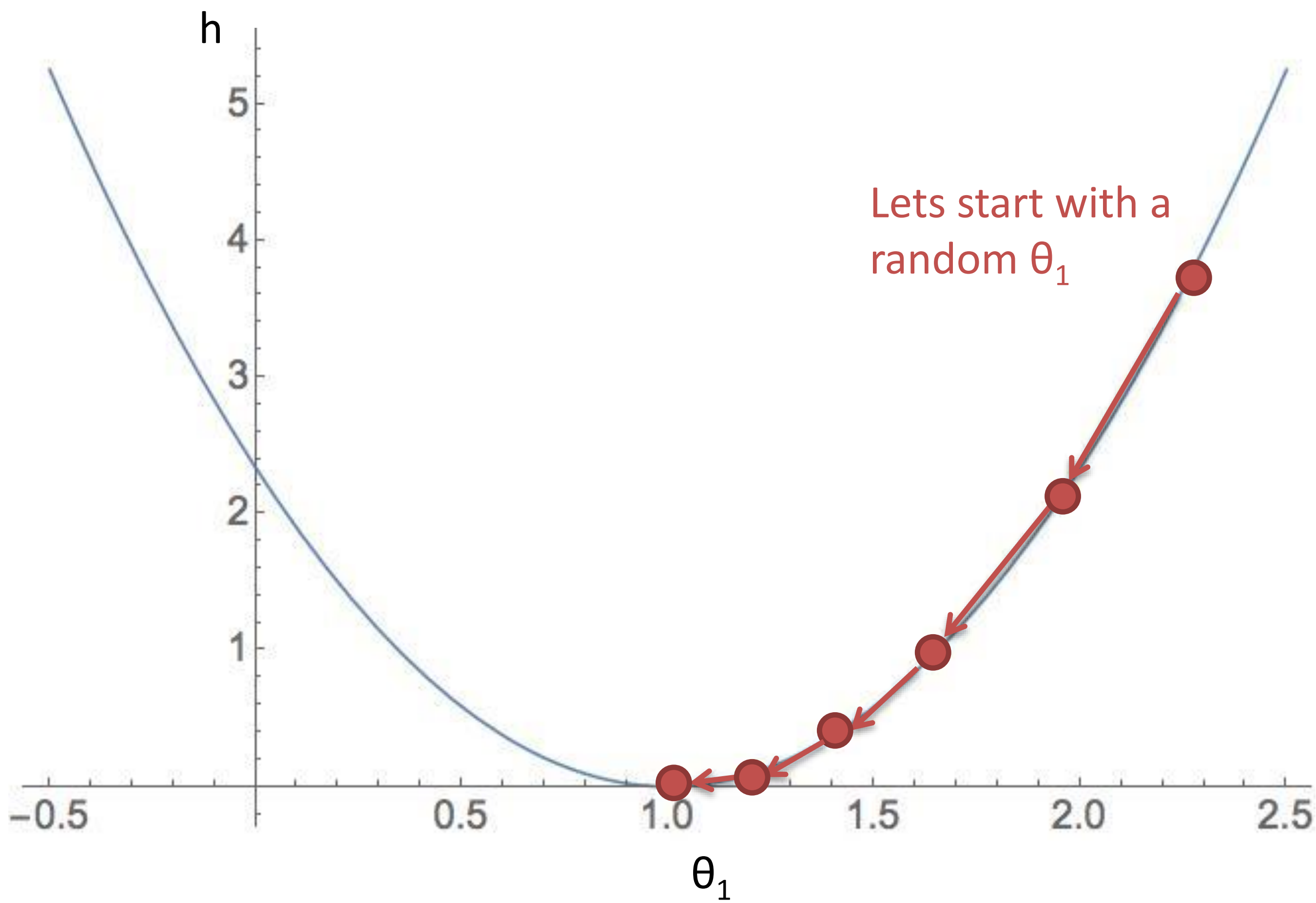
How Can We Find The Minimum?



$$\theta_1 = \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

Step Size / Learning Rate

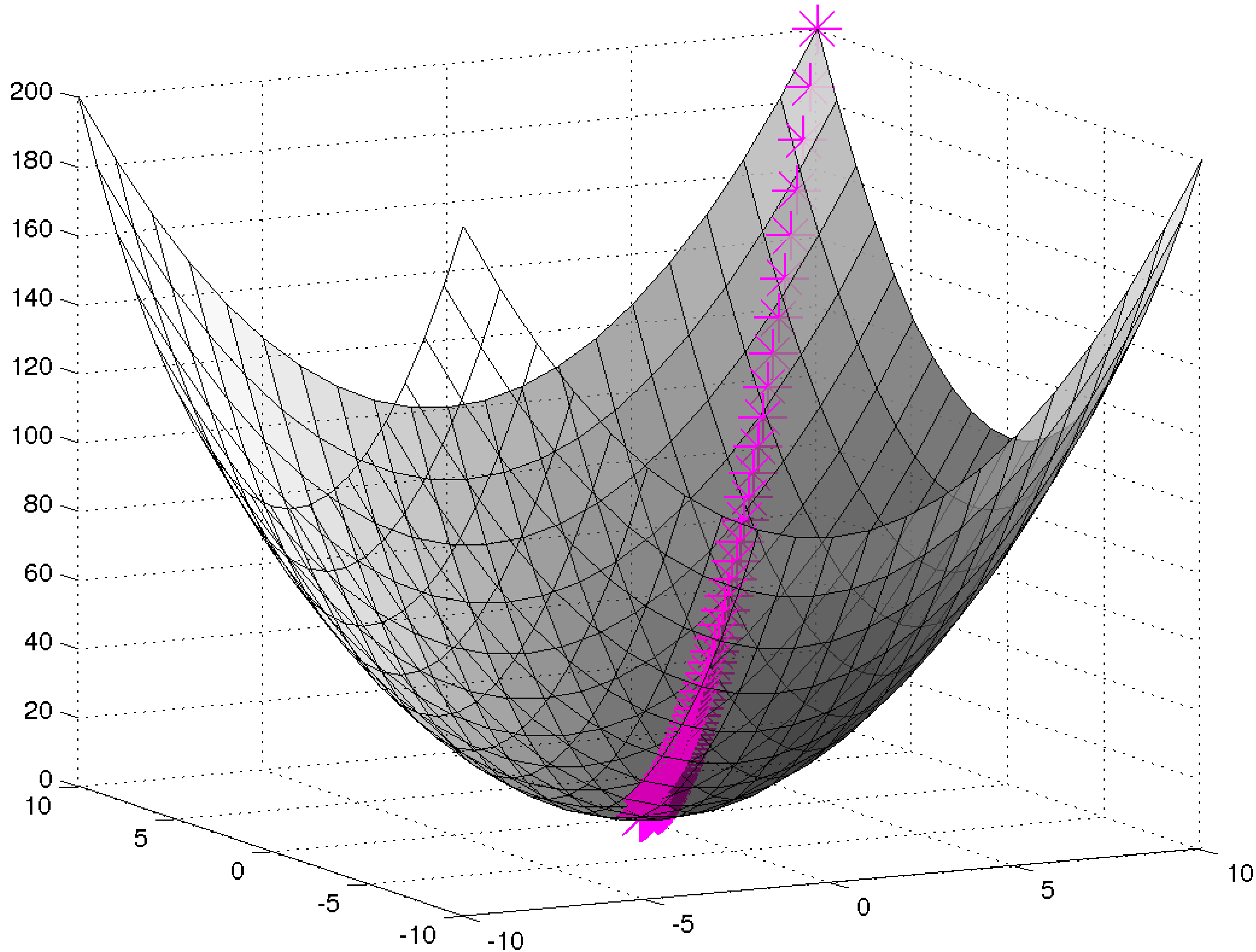
How Can We Find The Minimum?



Steps are automatically smaller the closer they get to the minimum

$$\theta_1 = \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$

Step Size / Learning Rate



200
180
160
140
120
100
80
60
40
20
0
10
5
0
-5
-10
-10
-5
0
5
10

Gradient Descent

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

Step Size / Learning Rate

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

$$h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x$$

Gradient Descent

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

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

$$h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x$$

Lets assume single data point (m=1)

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2} (h_\theta(x) - y)^2$$

Chain Rule

$$\frac{d}{dx}[f(g(x))] = f'(g(x))g'(x)$$

Gradient Descent

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

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

$$h(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x$$

Lets assume single data point (m=1)

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{\partial}{\partial \theta_j} \frac{1}{2} (h_\theta(x) - y)^2$$

$$= 2 \cdot \frac{1}{2} (h_\theta(x) - y) \cdot \frac{\partial}{\partial \theta_j} (h_\theta(x) - y)$$

Chain Rule

$$\frac{d}{dx}[f(g(x))] = f'(g(x))g'(x)$$

Gradient Descent

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

$$h(x) = \sum_{i=0}^{n}\theta_i x_i = \theta^T x$$

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

Lets assume single data point (m=1)

$$\begin{aligned}
\frac{\partial}{\partial\theta_j}J(\theta) &= \frac{\partial}{\partial\theta_j}\frac{1}{2}(h_\theta(x) - y)^2 \\
&= 2\cdot\frac{1}{2}(h_\theta(x) - y)\cdot\frac{\partial}{\partial\theta_j}(h_\theta(x) - y) \\
&= (h_\theta(x) - y)\cdot\frac{\partial}{\partial\theta_j}\left(\sum_{i=0}^{n}\theta_i x_i - y\right)
\end{aligned}$$

Chain Rule

$$\frac{d}{dx}[f(g(x))] = f'(g(x))g'(x)$$

Gradient Descent

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

$$h(x) = \sum_{i=0}^{n}\theta_i x_i = \theta^T x$$

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

Lets assume single data point (m=1)

$$\begin{aligned}
\frac{\partial}{\partial\theta_j}J(\theta) &= \frac{\partial}{\partial\theta_j}\frac{1}{2}(h_\theta(x) - y)^2 \\
&= 2\cdot\frac{1}{2}(h_\theta(x) - y)\cdot\frac{\partial}{\partial\theta_j}(h_\theta(x) - y) \\
&= (h_\theta(x) - y)\cdot\frac{\partial}{\partial\theta_j}\left(\sum_{i=0}^{n}\theta_i x_i - y\right) \\
&= (h_\theta(x) - y)\, x_j
\end{aligned}$$

Chain Rule

$$\frac{d}{dx}[f(g(x))] = f'(g(x))g'(x)$$

LMS Update Rule

(Widrow-Hoff Learning)

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

Batch Gradient Descent

Repeat until convergence {

$$\theta_j := \theta_j + \alpha \sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)}) \right) x_j^{(i)} \qquad \text{(for every } j\text{)}$$

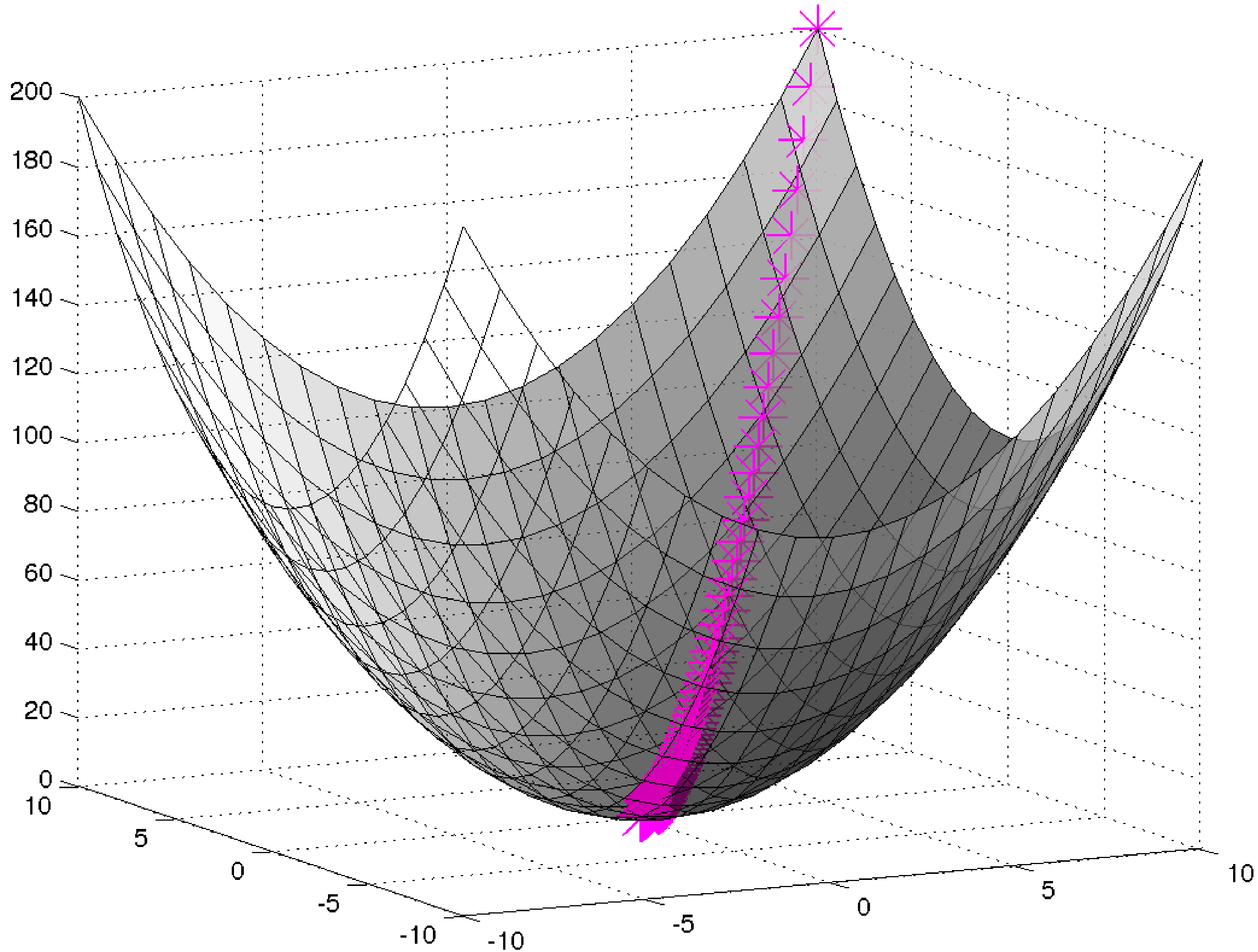
}

Clicker:

Can we get “stuck” in a local minima?

a) Yes

b) No


200
180
160
140
120
100
80
60
40
20
0
10
5
0
-5
-10
-10
-5
0
5
10

Clicker Question

Batch Gradient Descent

Repeat until convergence {

$$\theta_j := \theta_j + \alpha \sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)} \qquad \text{(for every } j\text{)}$$

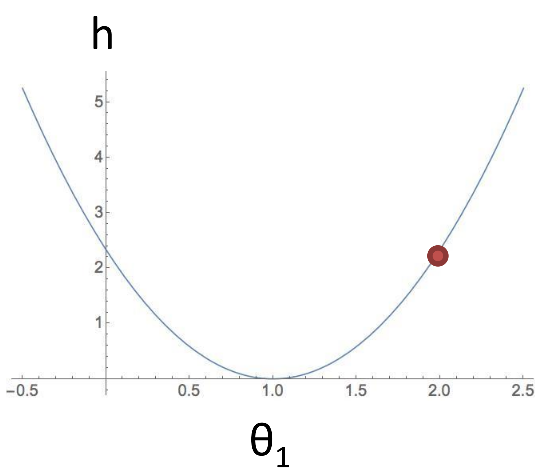
}

Data (x,y)

(1,1)
(2,2)
(3,3)

Our hypothesis $h_q(x) = q_0 + q_1 x_1$

Current values: $a = 0.01 \qquad q_0 = 0 \qquad q_1 = 2$



What is the next θ_1

a) 1
b) 1.54
c) 1.86

Batch Gradient Descent

Repeat until convergence {

$$\theta_j := \theta_j + \alpha \sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)} \qquad \text{(for every j)}$$

}

Data (x,y)

(1,1)
(2,2)
(3,3)

Our hypothesis

$$h_q\left(x\right) = q_0 + q_1 x_1$$

Current values:

$$a = 0.01$$

$$q_0 = 0$$

$$q_1 = 2$$

$$q_1 = q_1 + a \sum_{i=1}^{m} \left(y^{(i)} - h_q(x^{(i)})\right) x_j^{(i)}$$

$$q_1 = q_1 + a \begin{pmatrix} \left(y^{(1)} - \left(q_0 + q_1 x^{(1)}\right)\right) x^{(1)} + \\ \left(y^{(2)} - \left(q_0 + q_1 x^{(2)}\right)\right) x^{(2)} + \\ \left(y^{(3)} - \left(q_0 + q_1 x^{(3)}\right)\right) x^{(3)} \end{pmatrix}$$

$$q_1 = 2 + 0.01 \begin{pmatrix} (1 - 2)1 + \\ (2 - 4)2 + \\ (3 - 6)3 \end{pmatrix}$$

$$q_1 = 2 + 0.01(-1 - 4 - 9) = 2 - 0.14 = 1.86$$

How To Implement It

Repeat Until Convergence {

$$Temp_0 = \theta_0 + \alpha \sum_{i=0}^{m} \left(y^{(i)} - h_\theta(x^{(i)})\right) x_0^{(i)}$$

$$Temp_1 = \theta_1 + \alpha \sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)})\right) x_1^{(i)}$$

$$\vdots \qquad \vdots$$

$$Temp_n = \theta_n + \alpha \sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)})\right) x_n^{(i)}$$

$$\theta_0 = Temp_0$$

$$\theta_1 = Temp_1$$

$$\vdots$$

$$\theta_n = Temp_n$$

}

What is the problem with Batch Gradient Descent for Big Data?

Repeat until convergence {

$$\theta_j := \theta_j + \alpha \sum_{i=1}^{m} \left(y^{(i)} - h_\theta(x^{(i)}) \right) x_j^{(i)} \qquad \text{(for every } j\text{)}$$

}

Stochastic Gradient Descent (Incremental Gradient Descent)

Loop {

for i = 1 to m, {

$$\theta_j := \theta_j + \alpha \left(y^{(i)} - h_\theta(x^{(i)})\right) x_j^{(i)} \qquad \text{(for every } j\text{)}.$$

}

}

Clicker: Does this algorithm always converge?

a) Yes

b) No

Compromise

- **Batch gradient descent**:
  Use all m examples in each step

- **Stochastic gradient descent**:
  Use 1 example in each step

- **Mini-batch gradient descent**:
  Use *b* (e.g., 10) examples in each step
  - Easier to parallelize (e.g., SIMD)
  - Bit more robust against local minima (depending on b)

Important!

For linear regression there exists a closed form

- Good for smaller data sizes

Practical Considerations: Features

X_1 = Age (e.g., 24; 32; 100)
X_2 = Salary (e.g., \$120000; \$200000; \$210 000)

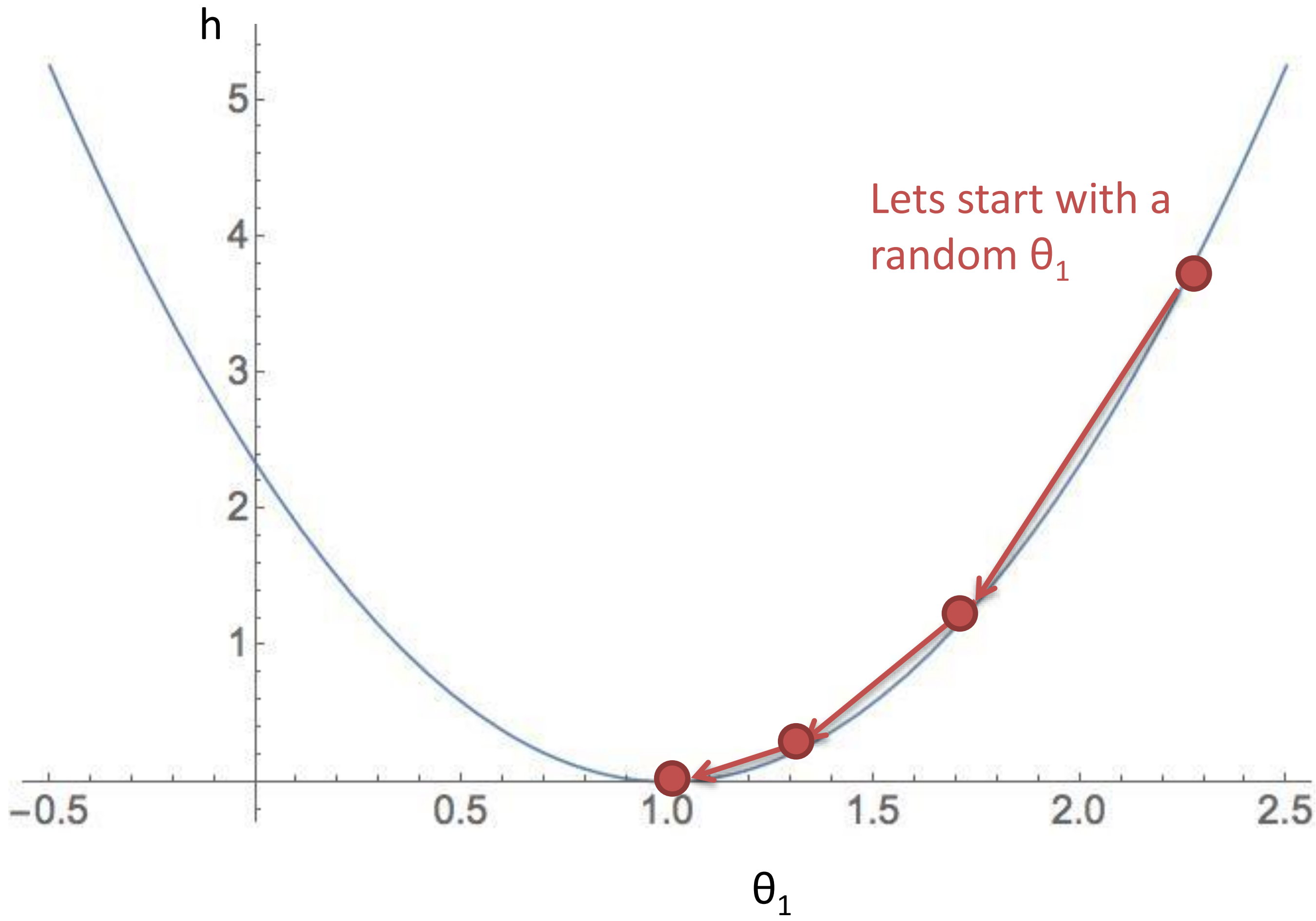
Solution: Feature Scaling $-1 \le x \le 1$

Mean Normalization $-0.5 \le x \le 0.5$

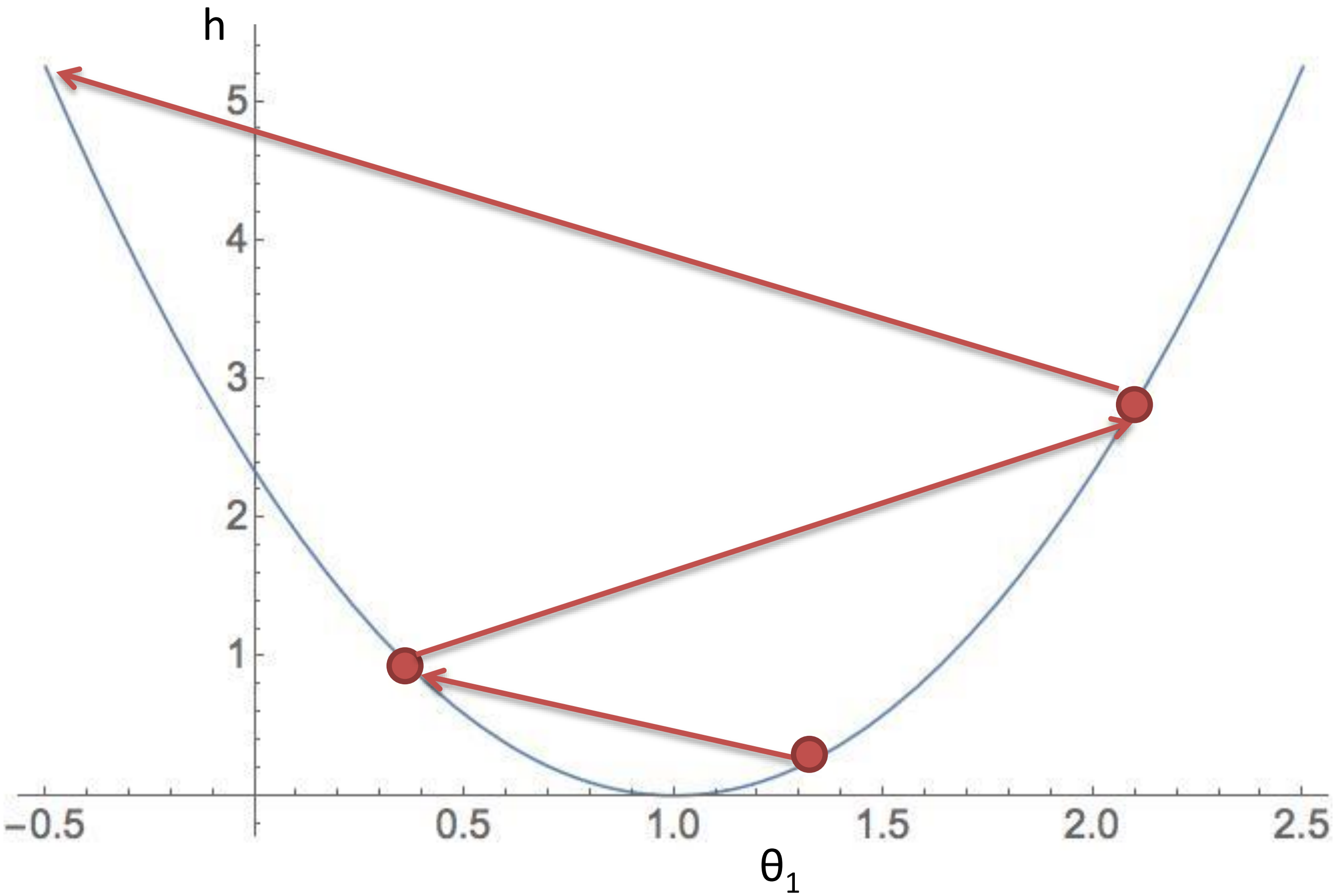
$$x_j = \frac{x_j - m_j}{\max(x_j) - \min(x_j)}$$

$$x_j^i = \frac{x_j^i - m_j}{S_j}$$

Practical Considerations: Learning Rate



Practical Considerations: Learning Rate



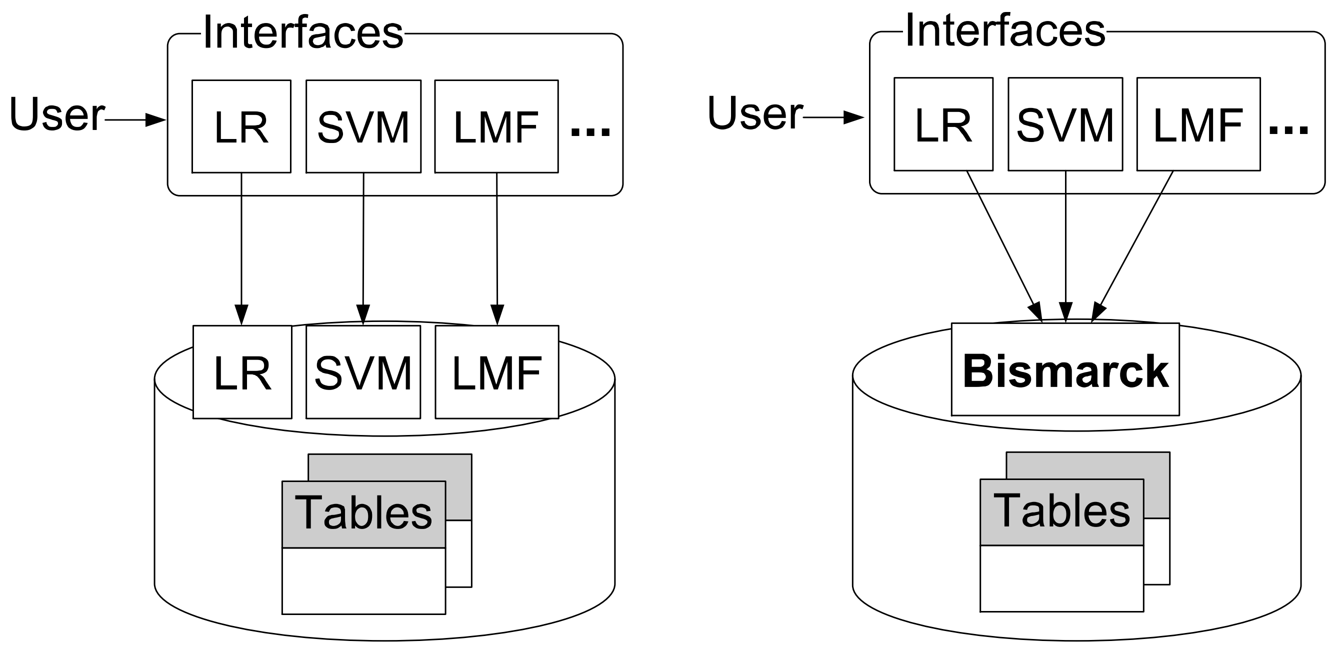
For sufficiently small *α, J*(θ) should decrease on every iteration

Bug if *α* is too small, it will take a long time to converge

Try 2x or 3x steps. For example: 0.001, 0.003, 0.01, 0.03, 0.1, 0.3, 1.0

Observe J and pick the one which has a good convergence rate

Towards a Unified Architecture for in-RDBMS Analytics



Current In-RDBMS Analytics   Bismarck In-RDBMS Analytics

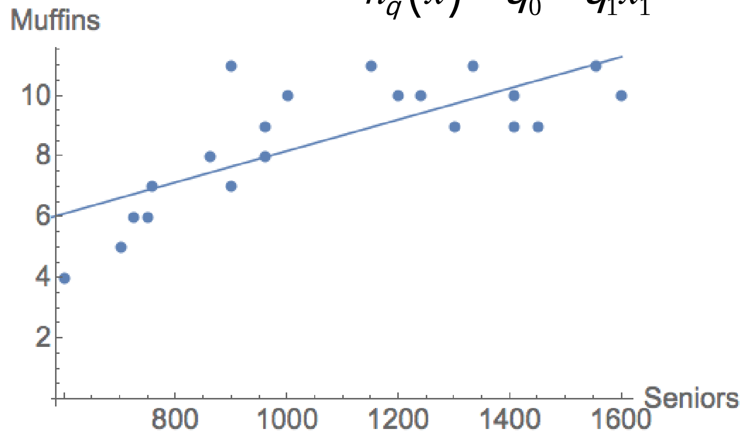
Xixuan Feng, Arun Kumar, Benjamin Recht, and Christopher Ré. 2012. Towards a unified architecture for in-RDBMS analytics. In Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data (SIGMOD '12). ACM, New York, NY, USA, 325-336.

Towards a Unified Architecture for in-RDBMS Analytics

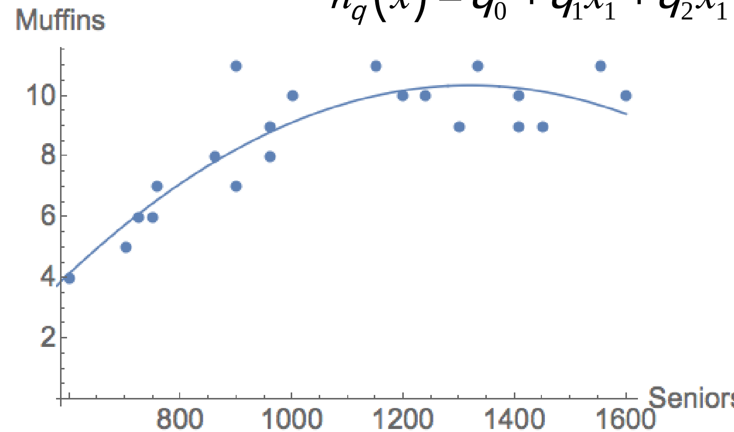
Analytics Task	Objective
Logistic Regression (LR)	$\sum_i \log(1 + exp(-y_i w^T x_i)) + \mu \|\vec{w}\|_1$
Classification (SVM)	$\sum_i (1 - y_i w^T x_i)_+ + \mu \|\vec{w}\|_1$
Recommendation (LMF)	$\sum_{(i,j)\in\Omega} (L_i^T R_j - M_{ij})^2 + \mu \|L, R\|_F^2$
Labeling (CRF) [48]	$\sum_k \left[\sum_j w_j F_j(y_k, x_k) - \log Z(x_k) \right]$
Kalman Filters	$\sum_{t=1}^T \|\|Cw_t - f(y_t)\|\|_2^2 + \|\|w_t - Aw_{t-1}\|\|_2^2$
Portfolio Optimization	$p^T w + w^T \Sigma w \quad \text{s.t.} \quad w \in \Delta$

Polynomial Regression

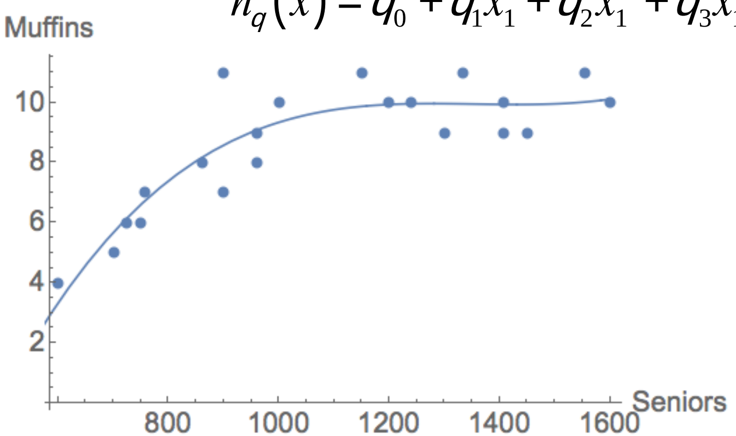
$$h_q(x) = q_0 + q_1 x_1$$



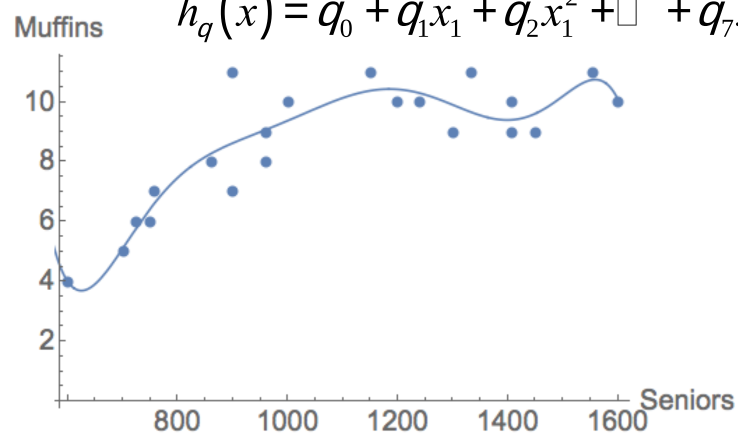
$$h_q(x) = q_0 + q_1 x_1 + q_2 x_1^2$$



$$h_q(x) = q_0 + q_1 x_1 + q_2 x_1^2 + q_3 x_1^3$$



$$h_q(x) = q_0 + q_1 x_1 + q_2 x_1^2 + \cdots + q_7 x_1^7$$

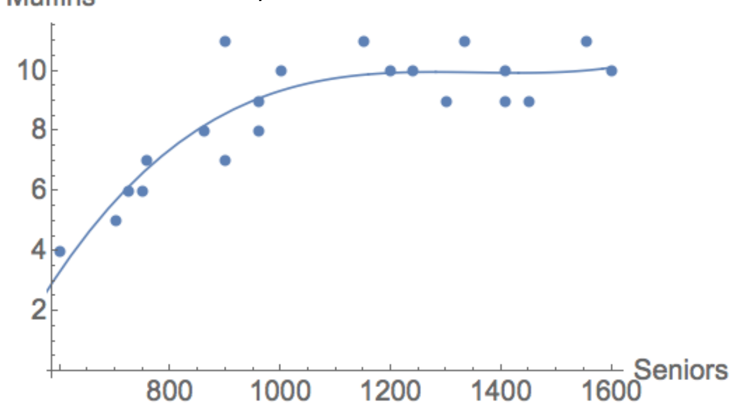


How To Prevent Overfitting

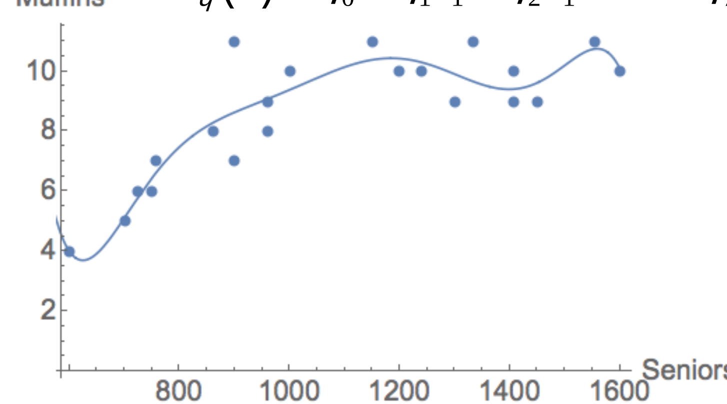
- Adjust features
  - Reduce number of polynomials
  - Reduce number of features
- Regularization
  - Keep all the features, but reduce their impact
  - Works well when we have a lot of features, each of which contributes a little

Regularization: Intuition

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^3$$



$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \cdots + \theta_7 x_1^7$$



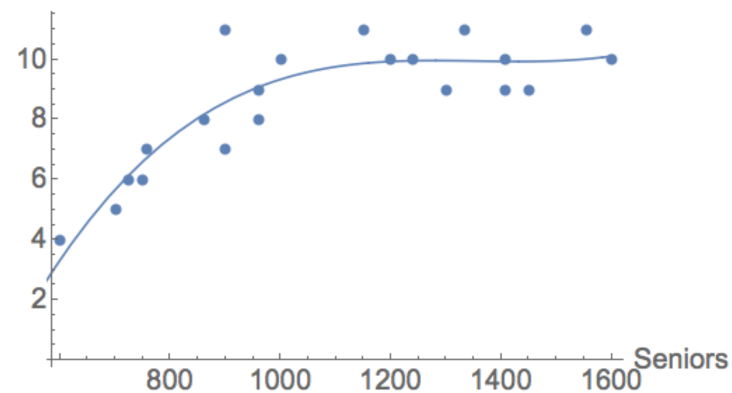
$$\min_\theta J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)} \right)^2$$

$$+ 1000\theta_4 + 1000\theta_5 + 1000\theta_6 + 1000\theta_7$$

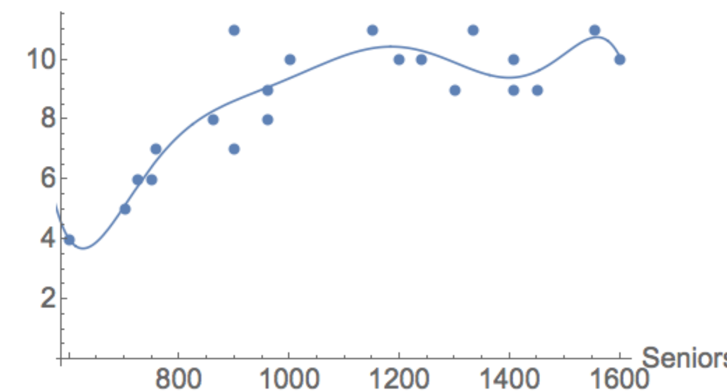
$$\theta_4, \theta_5, \theta_6, \theta_7 \rightarrow 0$$

Regularization: Intuition

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^3$$



$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \cdots + \theta_7 x_1^7$$



$$\min_\theta J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_\theta(x^{(i)}) - y^{(i)}\right)^2 + \underbrace{\lambda \sum_{j=1}^{n} \theta_j^2}_{\text{Regularization term}}$$

Works ok for reducing the impact of polynomials (better to reduce nb of polynomials directly)
Works great for a bag of equally important features.