CS 360: Machine Learning

Prof. Sara Mathieson

Fall 2020

HAVERFORD COLLEGE

Admin

- **Lab 1 graded**

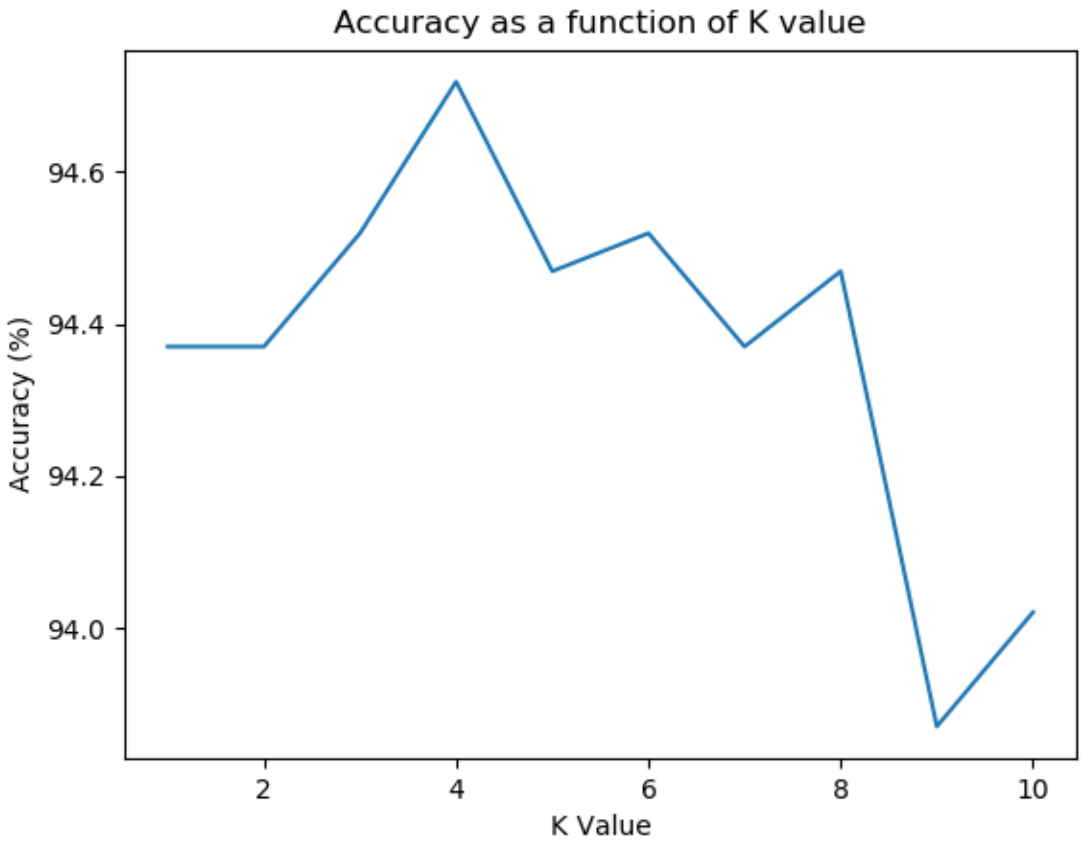
- **Lab 2 collected 5pm today**
  - Last minute questions -> Piazza
  - I am unable to meet this afternoon

- **Lab 3** (shortened, step-by-step) due **Tuesday night** (grace period til Wed night)
  - Starter code: one place coeff -> coef
  - For part (a), easier to use train_data.plot(..)

Lab 1 feedback

- Lab 1 grades posted (10pts for knn, 3 for math)
- Keep large data out of repos (unless it is given as part of the starter)
- Make sure code is in functions and function decomposition makes sense
- Keep lines less than 80 characters
- Official python style for methods/functions
  - my_function
  - ~~MyFunction~~ (uppercase is reserved for classes!)

Lab 1 Extensions: multi-class



Lab 1: how to make KNN faster?

- Runtime: exercise!
- Don't need to sort all distances – for small *K*, we can find the top *K* neighbors in linear time
- Save matrix of pair-wise distances across *K*
- Put each training example in a "zone" or "cluster". For each test example, identify cluster and only consider neighbors within that cluster

Outline for September 25

- Recap bias/variance tradeoff
- Simple linear regression
- SGD (Stochastic Gradient Descent)
- Normal equations solution

Outline for September 25

- Recap bias/variance tradeoff
- Simple linear regression
- SGD (Stochastic Gradient Descent)
- Normal equations solution

Regression setup and Expected Value

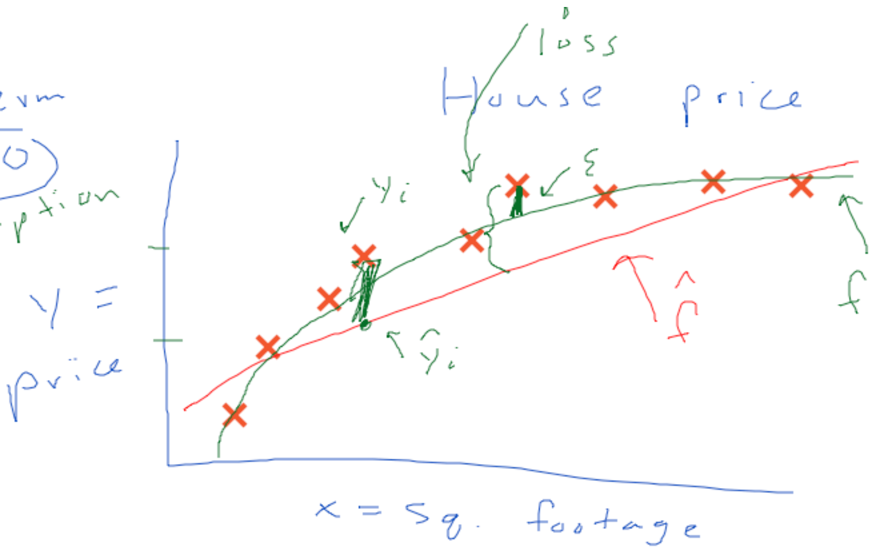
Regression setup

(noise) error term

model : true $y = f(x) + \varepsilon$ mean 0 assumption

$\hat{f}$ = estimate of f

$\hat{y} = \hat{f}(x)$ prediction



GOAL $\ell(y, \hat{y}) = (y - \hat{y})^2$

$E[(y-\hat{y})^2]$ expected loss

expected value = weighted avg

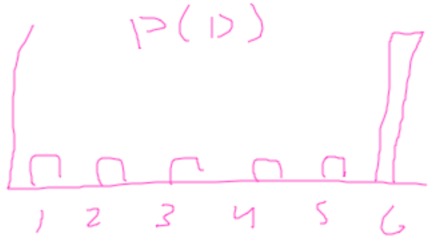
$$E[X] = \sum_{v} p(X=v) \cdot v$$

$$E(D) = \frac{1}{10}(1+2 \cdots 5) + \frac{1}{2} \cdot 6$$

$$= 4.5$$

$P(D=6) = \frac{1}{2}$

$P(D \neq 6) = \frac{1}{10}$



Compute Expected Loss

$$E[(y-\hat{y})^2] = E[(\underbrace{\overbrace{y-f}^{a}}_{\varepsilon} \underbrace{+}_{0} \underbrace{\overbrace{f-\hat{f}}^{b}})^2]$$

$\hat{f} = \hat{y}$, $\hat{f} \leftarrow (x)$

$$= \underbrace{\overbrace{Var(\varepsilon)}^{a^2}}_{\text{noise}} + \underbrace{\overbrace{E[(f-\hat{f})^2]}^{b^2}}_{\text{reducible error}}$$

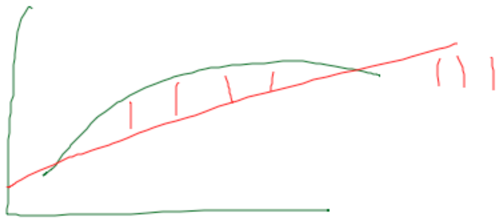
noise

irreducible error

$(a+b)^2$
$= a^2 + b^2 + \cancel{2ab}$ 0

$Var(x) = E[(x-\mu)^2]$

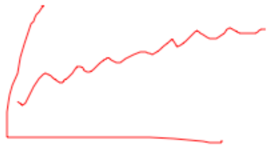
$$E[(f-\hat{f})^2] = E[((f - E[\hat{f}]) \underbrace{+}_{0} \underbrace{E[\hat{f}] - \hat{f}})^2]$$



bias

too simple

variance



too complex

Bias/variance big picture

$$E[(y-\hat{y})^2] = \underbrace{Var(\varepsilon)}_{\text{noise}} + \underbrace{E(f - E(\hat{f}))^2}_{bias(\hat{f})^2} + \underbrace{E(\overbrace{E[\hat{f}]}^{\text{mean}} - \hat{f})^2}_{Var(\hat{f})}$$

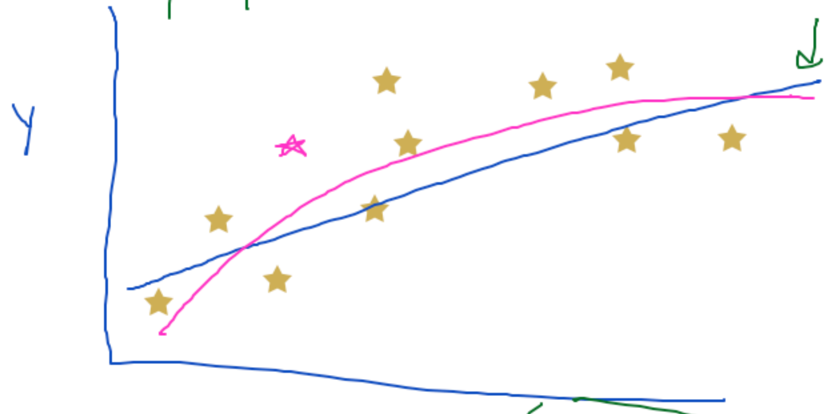
polynomial

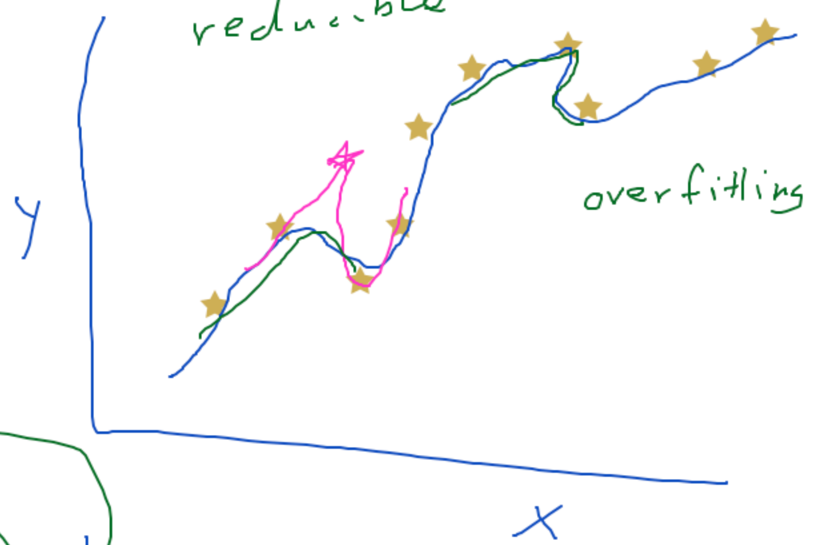
$y = f + \varepsilon$

(true)

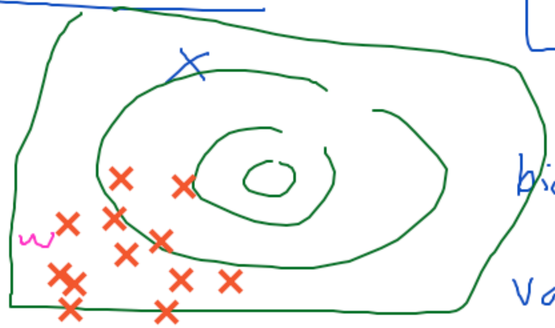
irreducible

reducible





bias : high
variance : low



bias : low
variance : high

Assessing Model Accuracy

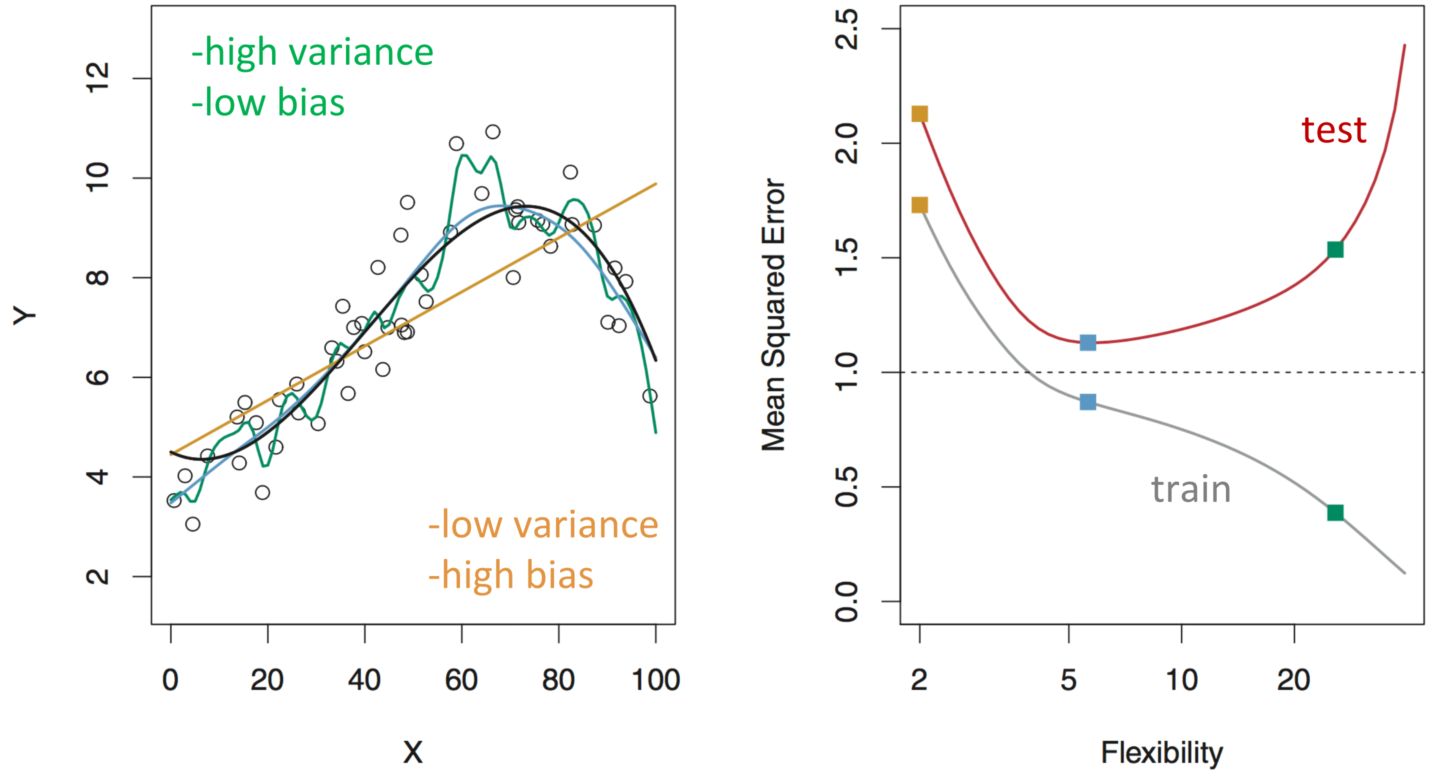


FIGURE 2.9. Left: *Data simulated from f, shown in black. Three estimates of f are shown: the linear regression line (orange curve), and two smoothing spline fits (blue and green curves).* Right: *Training MSE (grey curve), test MSE (red curve), and minimum possible test MSE over all methods (dashed line). Squares represent the training and test MSEs for the three fits shown in the left-hand panel.*

ISL, Fig 2.9

Outline for September 25

- Recap bias/variance tradeoff
- Simple linear regression
- SGD (Stochastic Gradient Descent)
- Normal equations solution

Goals of Inference

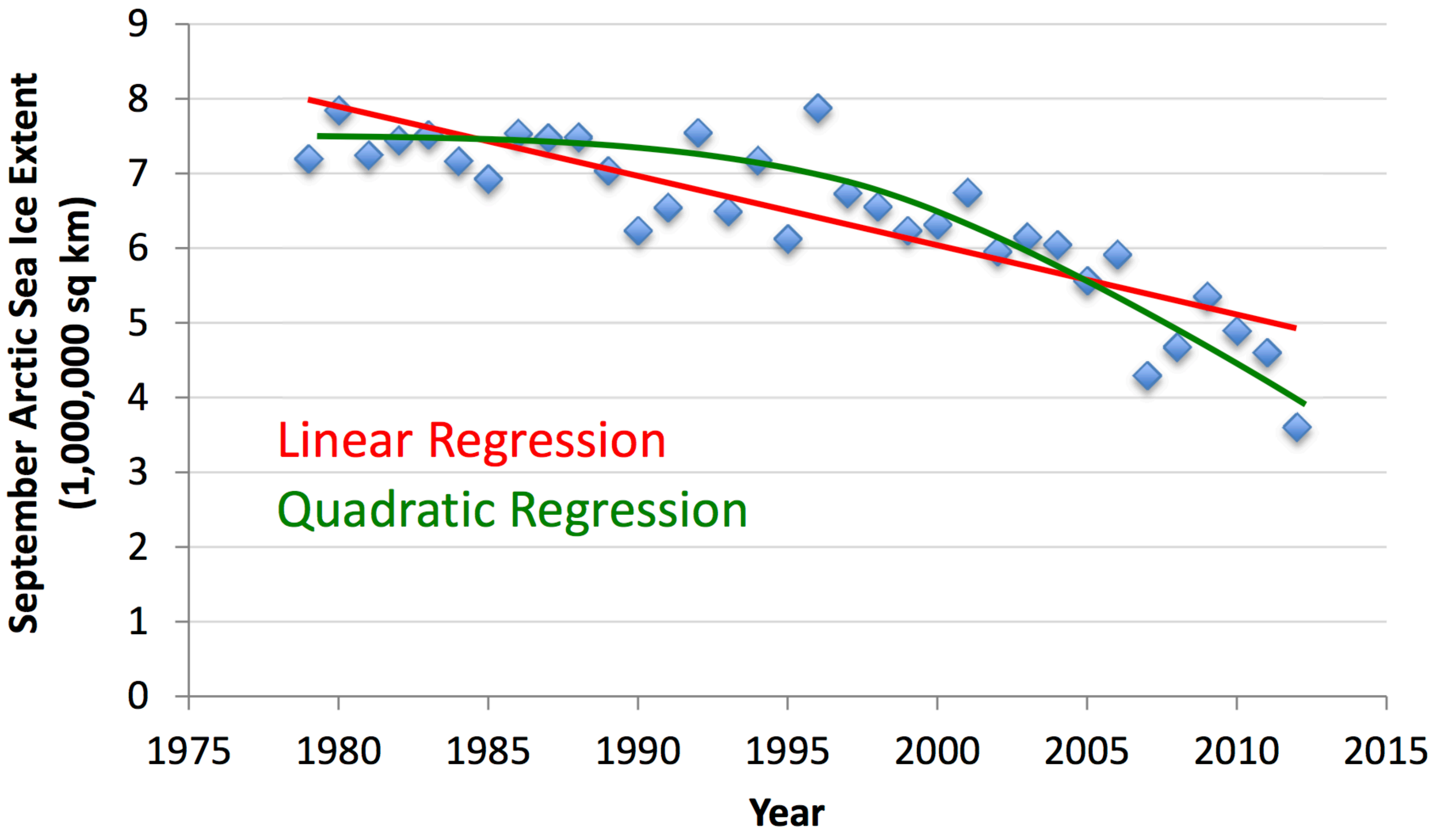
1) Which of the features/explanatory variables/ predictors (x) are associated with the response variable (y)?

2) What is the relationship between x and y?

3) Is a linear model enough?

4) Can we predict y given a new x?

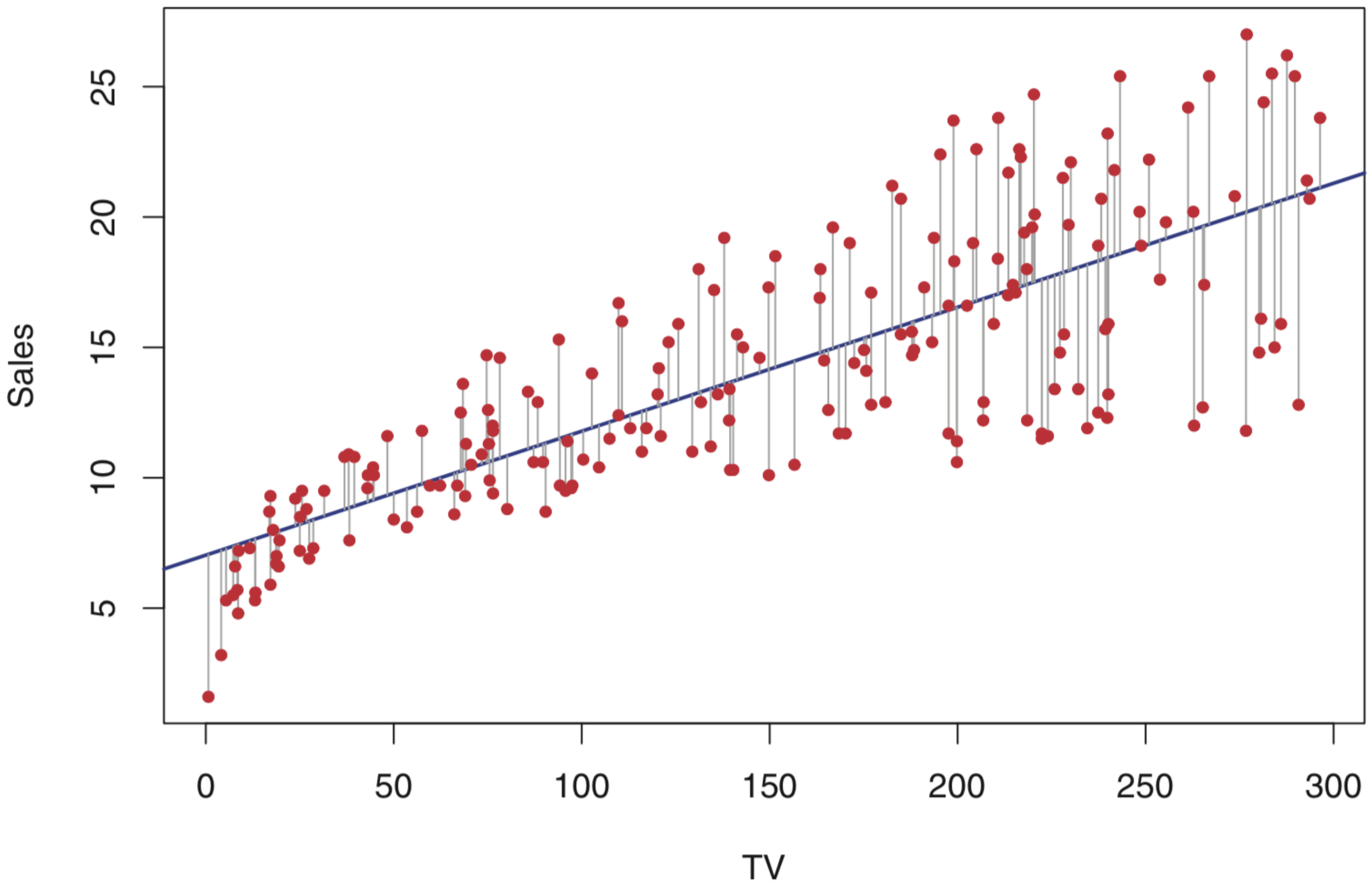
Regression Example



Slide: modified from Jessica Wu

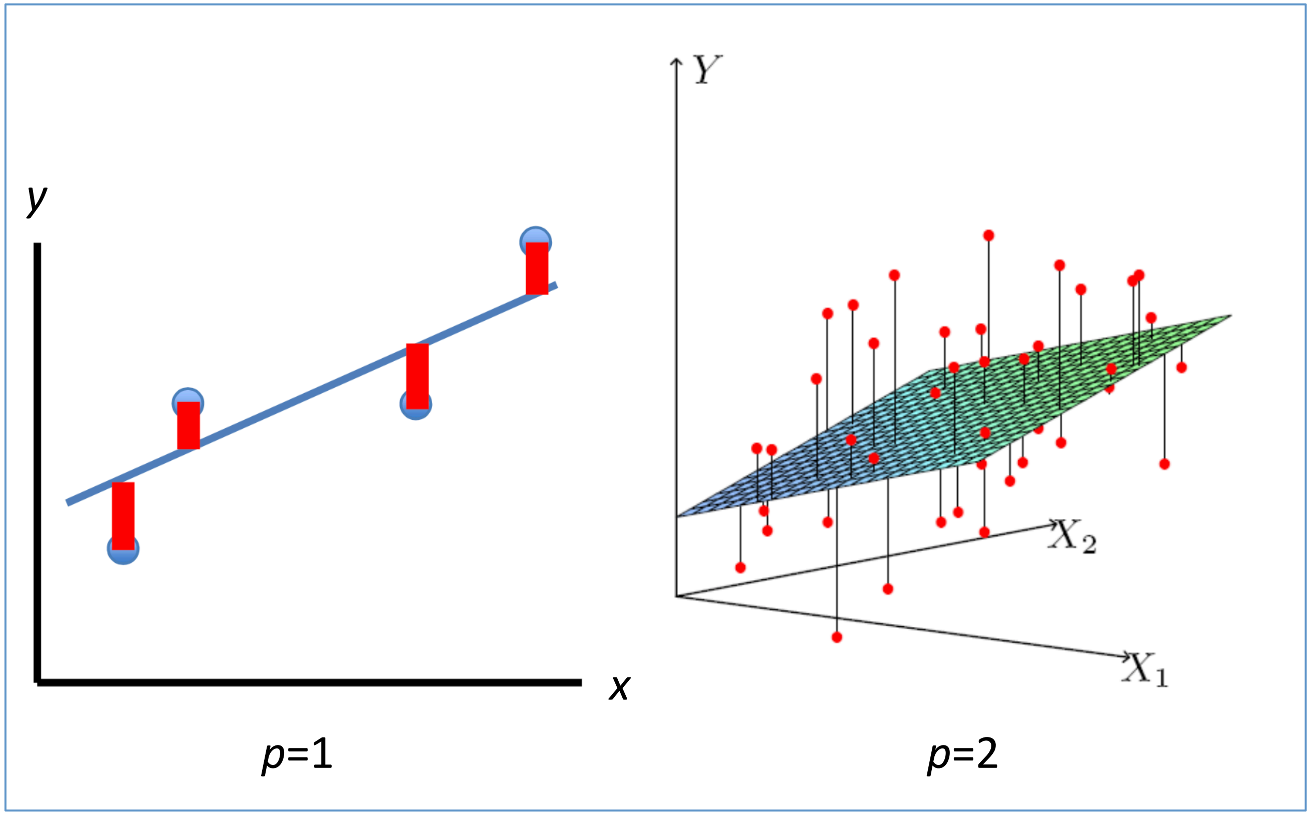
[Data from G. Witt. Journal of Statistics Education, Volume 21, Number 1 (2013)]

Example: predict sales from TV advertising budget



ISL: Figure 3.1

Cost Function: sum of squared errors



Linear Regression

- Output (*y*) is continuous, not a discrete label

- Learned model: *linear function* mapping input to output (a *weight* for each feature + *bias*)

- Goal: minimize the *RSS* (residual sum of squared errors) or *SSE* (sum of squared errors)

"Simple" linear regression

$$\hat{f}(x) = \underbrace{\hat{w}_0}_{\text{intercept}} + \hat{w}_1 x = \hat{y} = h_{\vec{w}}(x) \leftarrow \text{hypothesis}$$

minimize: $\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 = MSE \Big\} \approx$ RSS, SSE

cost: $J(\overset{[w_0, w_1]}{\vec{w}}) = \frac{1}{2}\sum_{i=1}^{n}(w_0 + w_1 x_i - y_i)^2 \Big\}$ $\frac{1}{2}$ is for nice derivative

(a)

$$\frac{\partial J}{\partial w_0} = \sum_{i=1}^{n} w_0 + w_1 x_i - y_i = 0$$

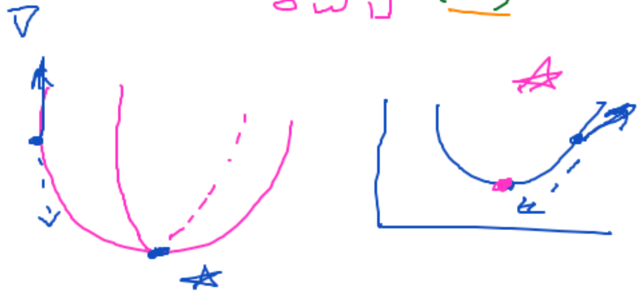
$$\Rightarrow \frac{\cancel{n} w_0}{\cancel{n}} = \underbrace{\frac{1}{n}\sum y_i}_{\bar{y}} - w_1 \underbrace{\frac{1}{n}\sum x_i}_{\bar{x}}$$

$$\boxed{\hat{w}_0 = \bar{y} - \hat{w}_1 \bar{x}_1}$$

gradient — partial derivatives

$$\nabla J_{\vec{w}} = \begin{bmatrix} \frac{\partial J}{\partial w_0} \\ \frac{\partial J}{\partial w_1} \end{bmatrix} = \vec{0}$$

(a) (b)



"Simple" linear regression

$$\boxed{\hat{w}_0 = \bar{y} - \hat{w}_1 \bar{x}}$$

$$\frac{\partial J}{\partial w_1} = \sum (w_0 + w_1 x_i - y_i) x_i = 0$$

solve for w_1

$$\boxed{\hat{w}_1 = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{\mathrm{Cov}(X,Y)}{\mathrm{Var}(X)}}$$

feature

true output

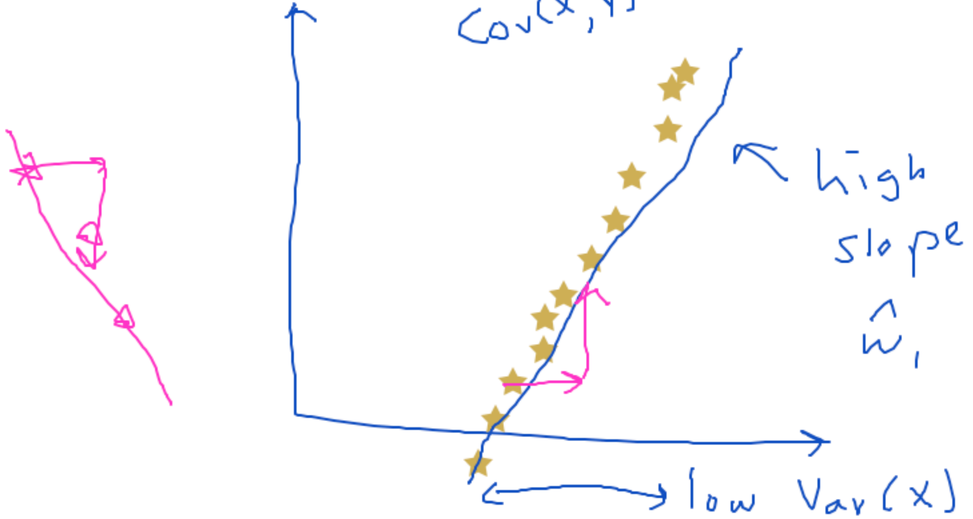
mean

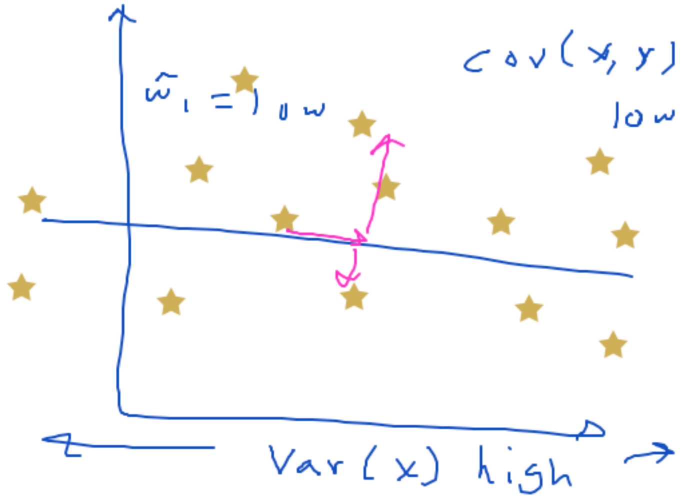
~~input~~

orig data

pred

$$\hat{y}_i = \hat{w}_0 + \hat{w}_1 x_i$$





Handout 4
(on piazza, try on your own!)

Outline for September 25

- Recap bias/variance tradeoff
- Simple linear regression
- SGD (Stochastic Gradient Descent)
- Normal equations solution

Multiple linear regression

$\vec{x}_i$ example

x_j feature

x_{ij}

$$\hat{f}(\vec{x}) = \underline{w_0} \cdot 1 + w_1 x_1 + w_2 x_2 + \cdots w_p x_p$$

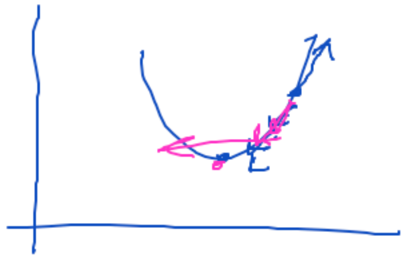
$$= \vec{w} \cdot \vec{x}$$

$$[w_0 \; w_1 \; \cdots \; w_p] \begin{bmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_p \end{bmatrix}$$

fake 1

$$J(\vec{w}) = \frac{1}{2} \sum (\underline{\vec{w}} \cdot \vec{x}_i - y_i)^2$$

$\boxed{\text{SGD}}$ stocartic gradient descent



$$\frac{\partial J}{\partial w_j} = \sum (\vec{w} \cdot \vec{x}_i - y_i) \cdot x_{ij} \leftarrow$$

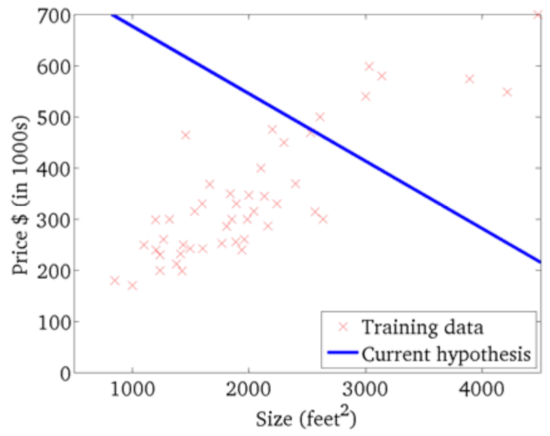
opp. direction of gradient

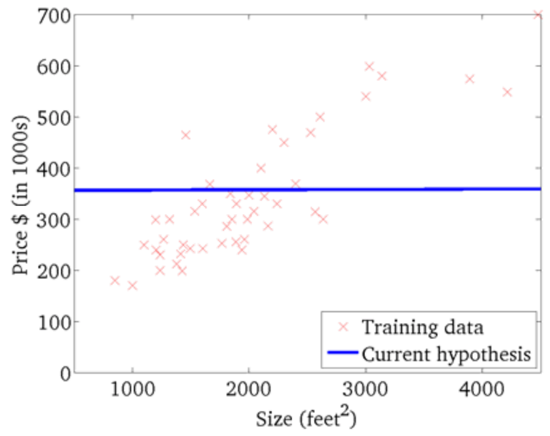
for all i

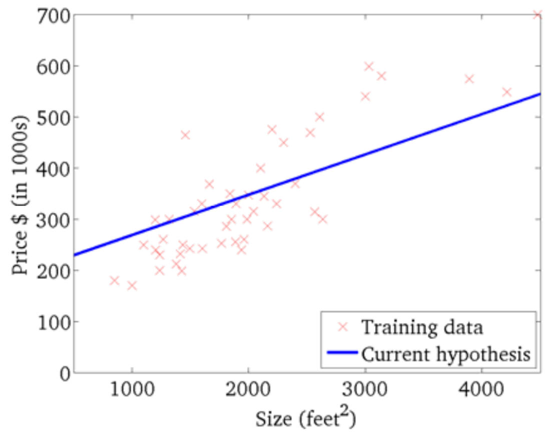
$$\boxed{\vec{w} \leftarrow \vec{w} - \underline{\alpha} \left(\underline{\vec{w} \cdot \vec{x}_i - y_i} \right) \underline{\vec{x}_i}}$$

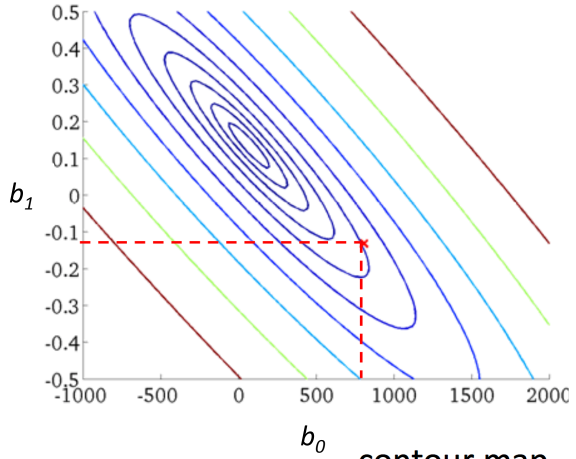
step size

Cost Function

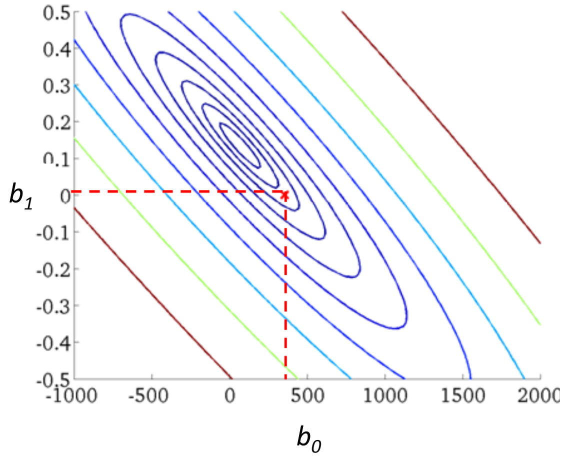


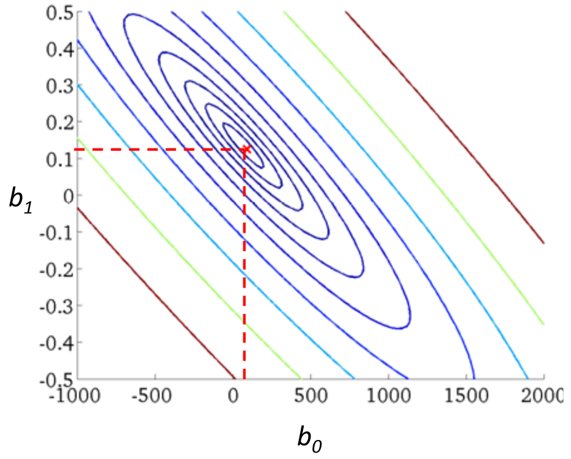






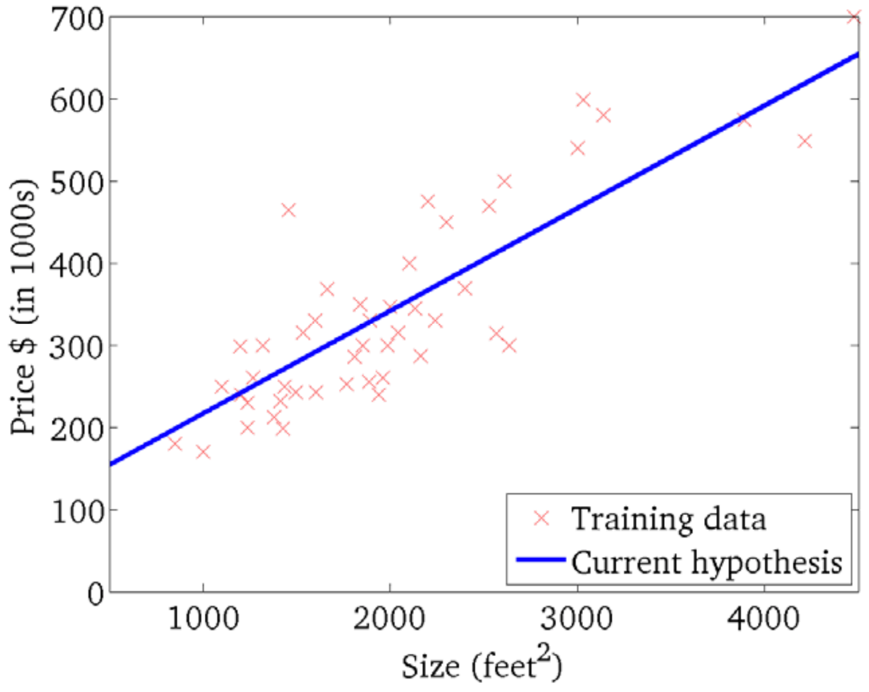
contour map

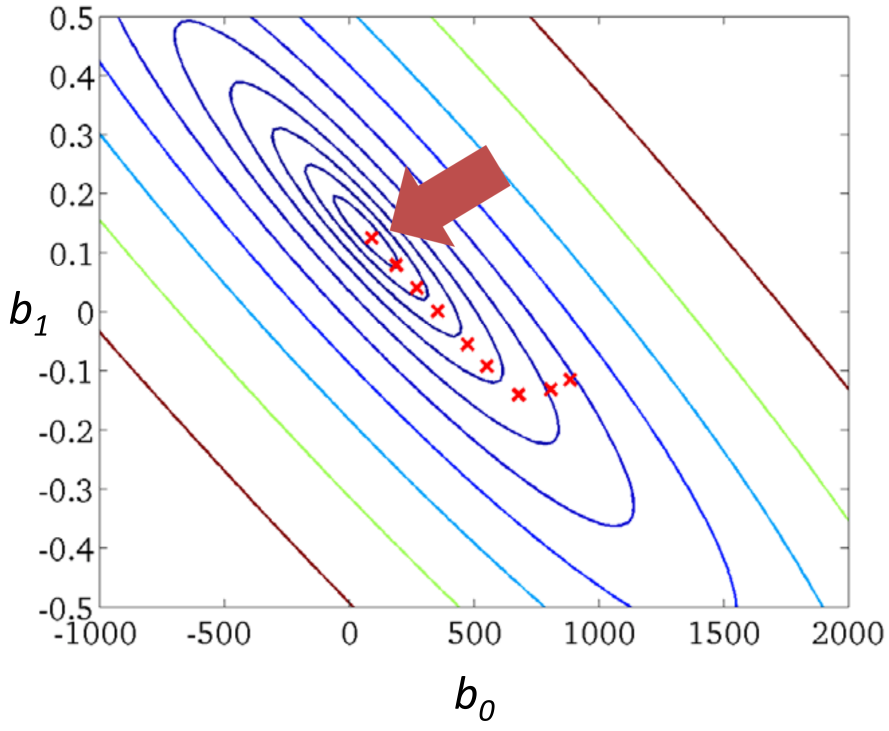




Slide: modified from Jessica Wu
Based on slide by Andrew Ng

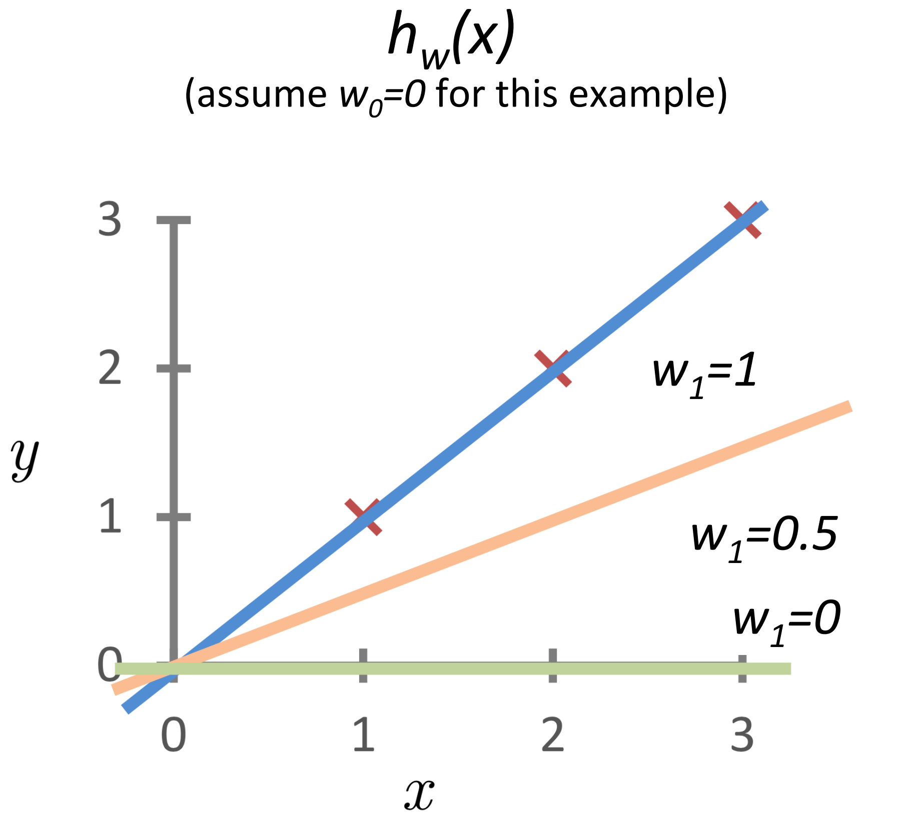
Gradient Descent: walking toward the minimum

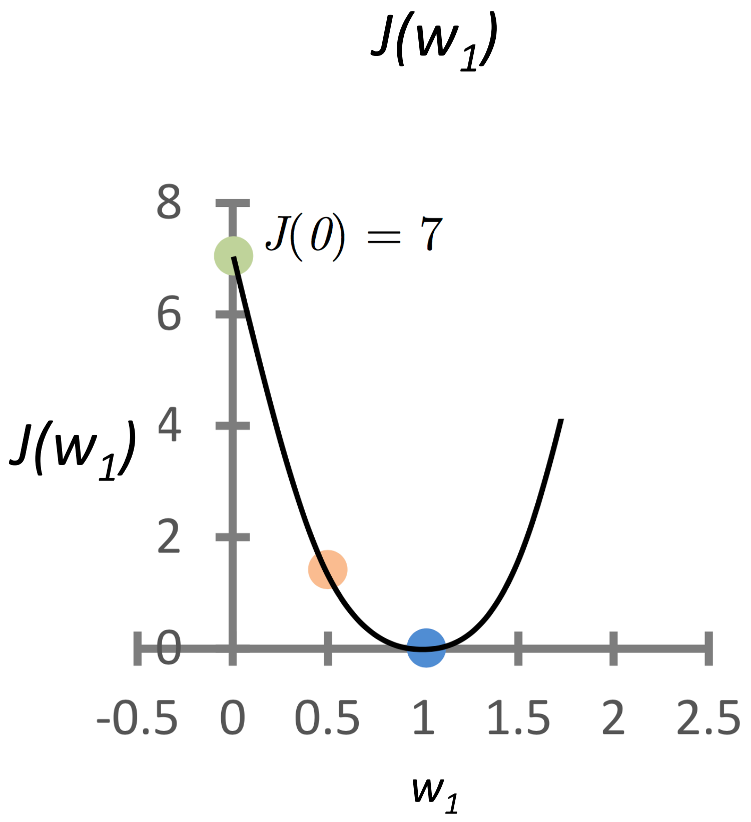




Slide: modified from Jessica Wu
Based on slide by Andrew Ng

Cost Function (extra practice)



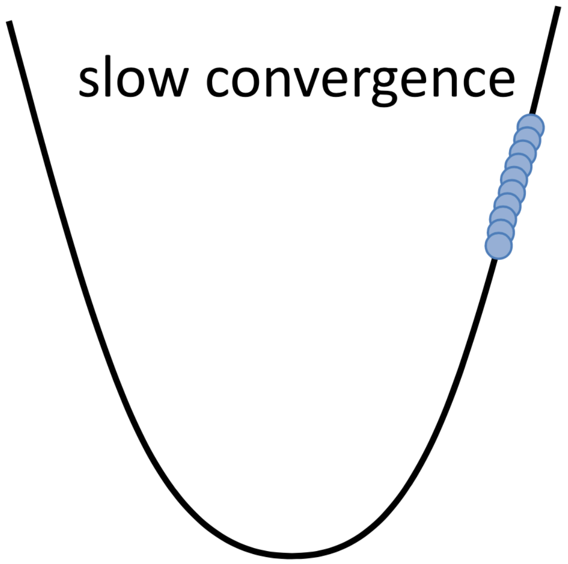


$$J(0.5) = \frac{1}{2}\left[(0.5-1)^2 + (1-2)^2 + (1.5-3)^2\right] = 1.75$$

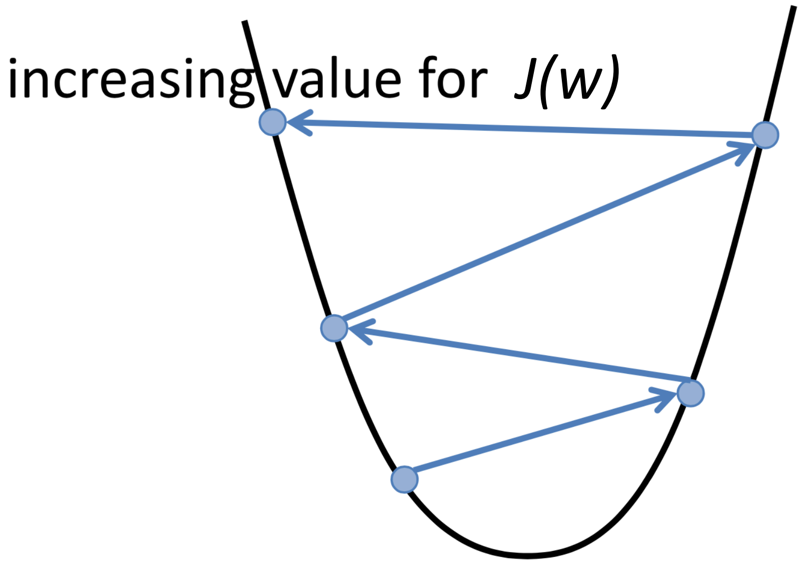
Slide: modified from Jessica Wu
Based on slide by Eric Eaton
[Example by Andrew Ng]

Choosing step size

α too small



α too large

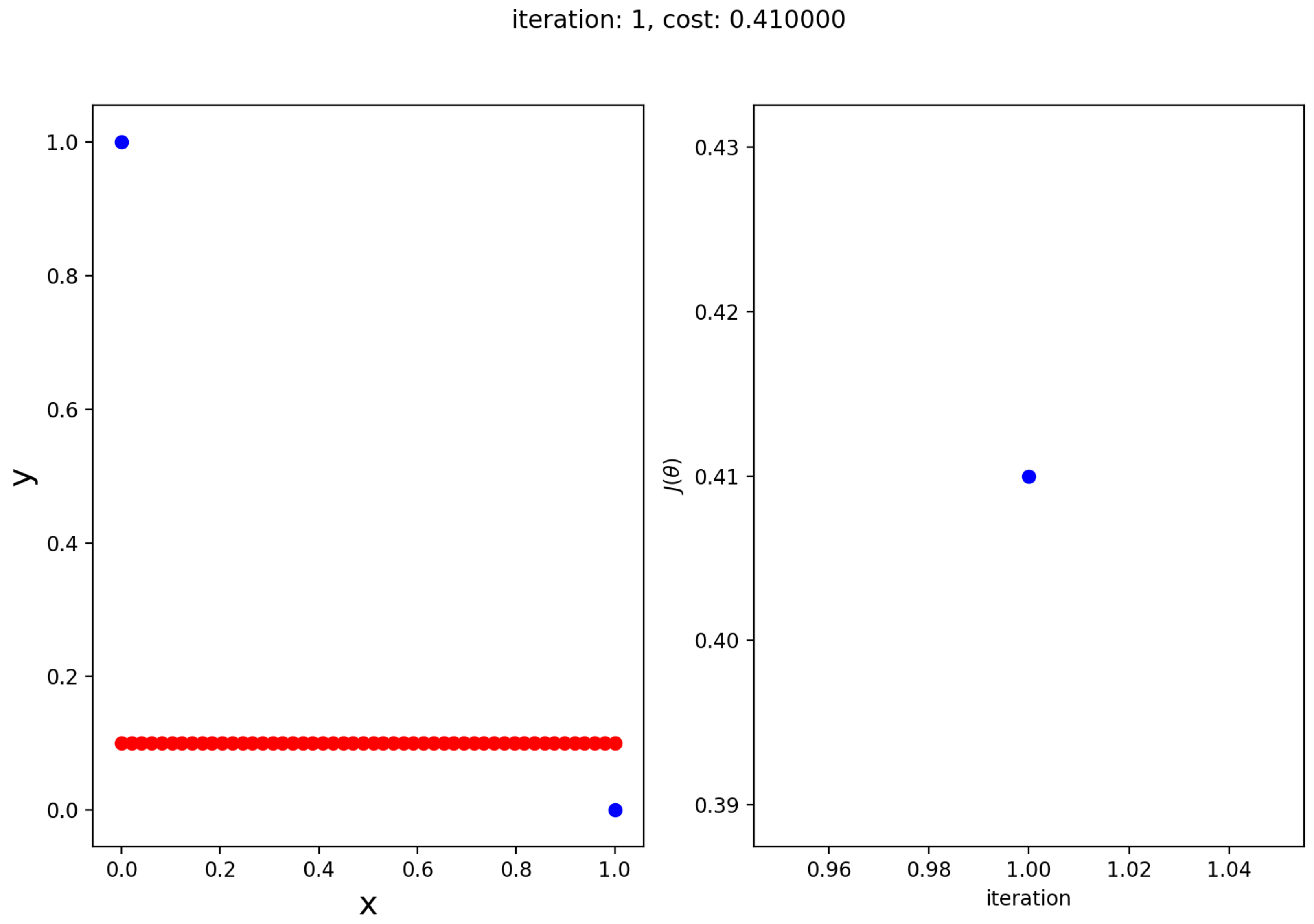


- may overshoot minimum
- may fail to converge (may even diverge)

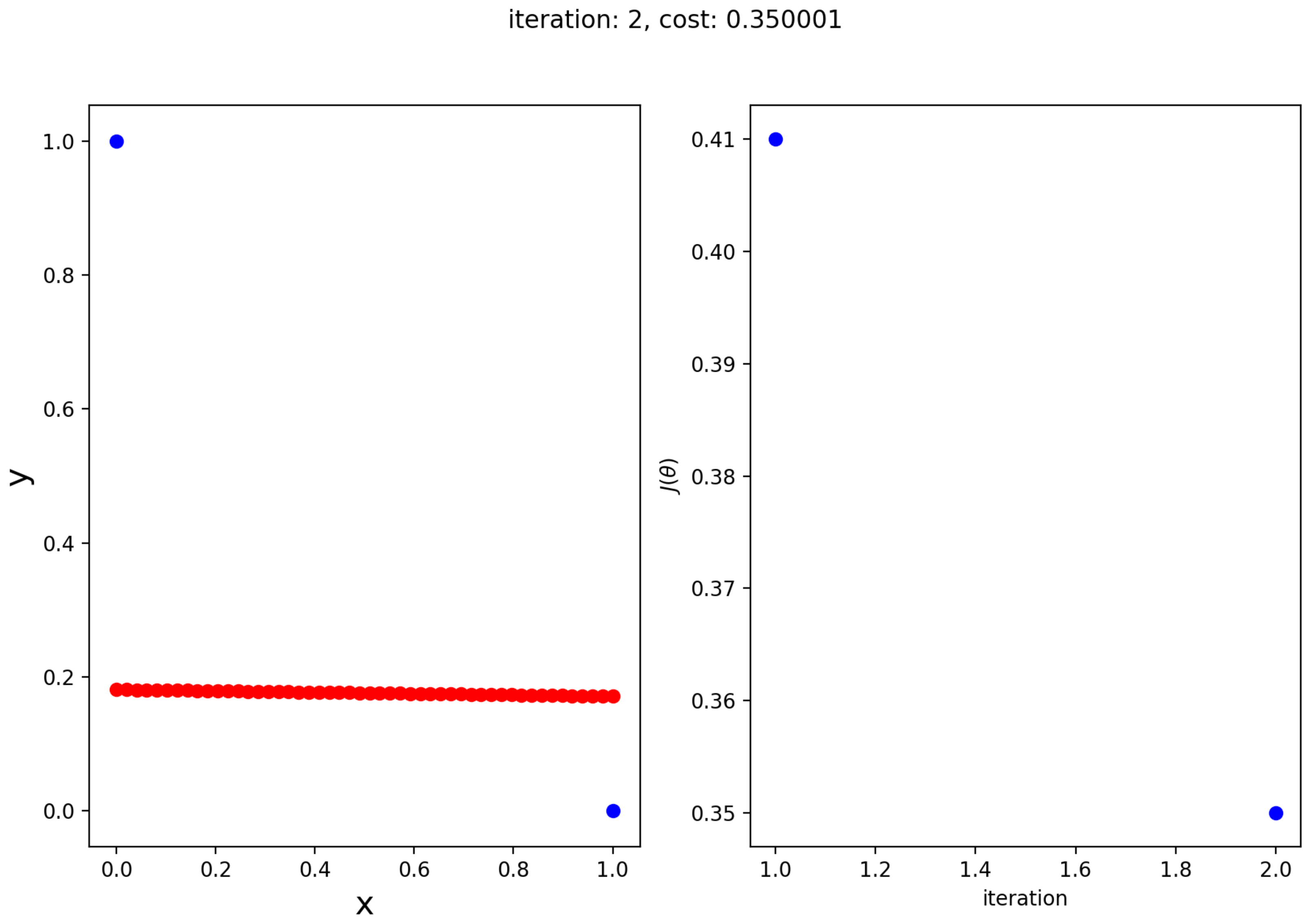
Slide: modified from Jessica Wu
Based on slide by Eric Eaton
[Originally by Andrew Ng]

Lab 3 applied to Handout 4

Toy example, iteration 1

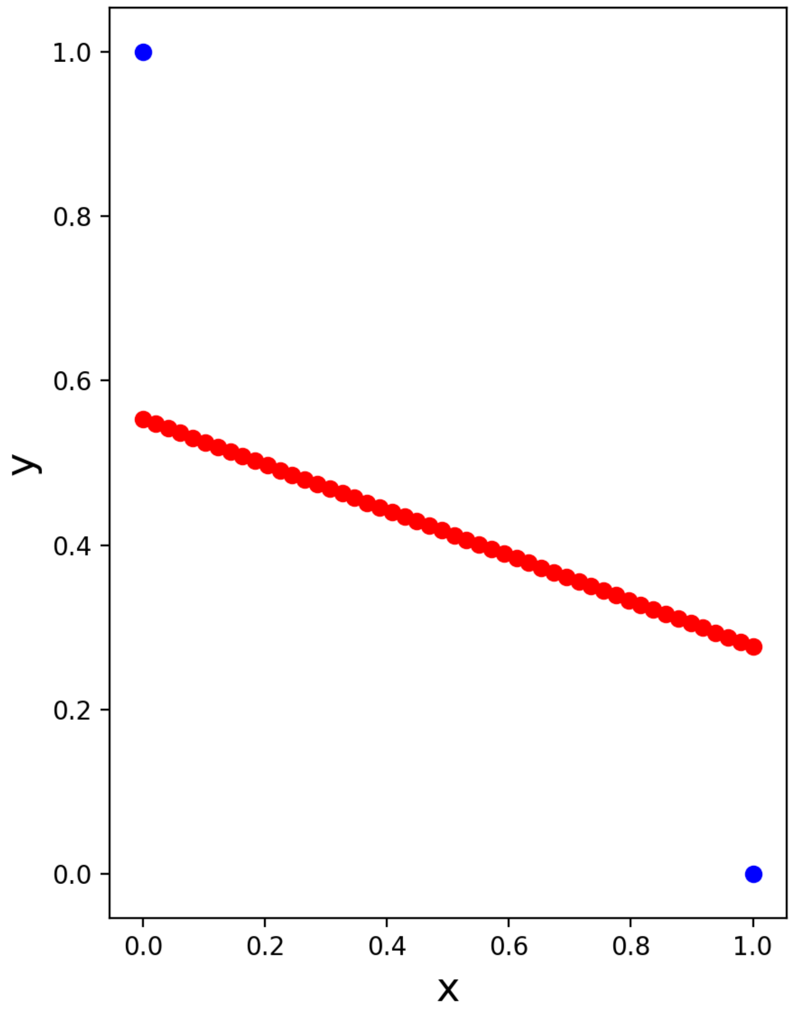


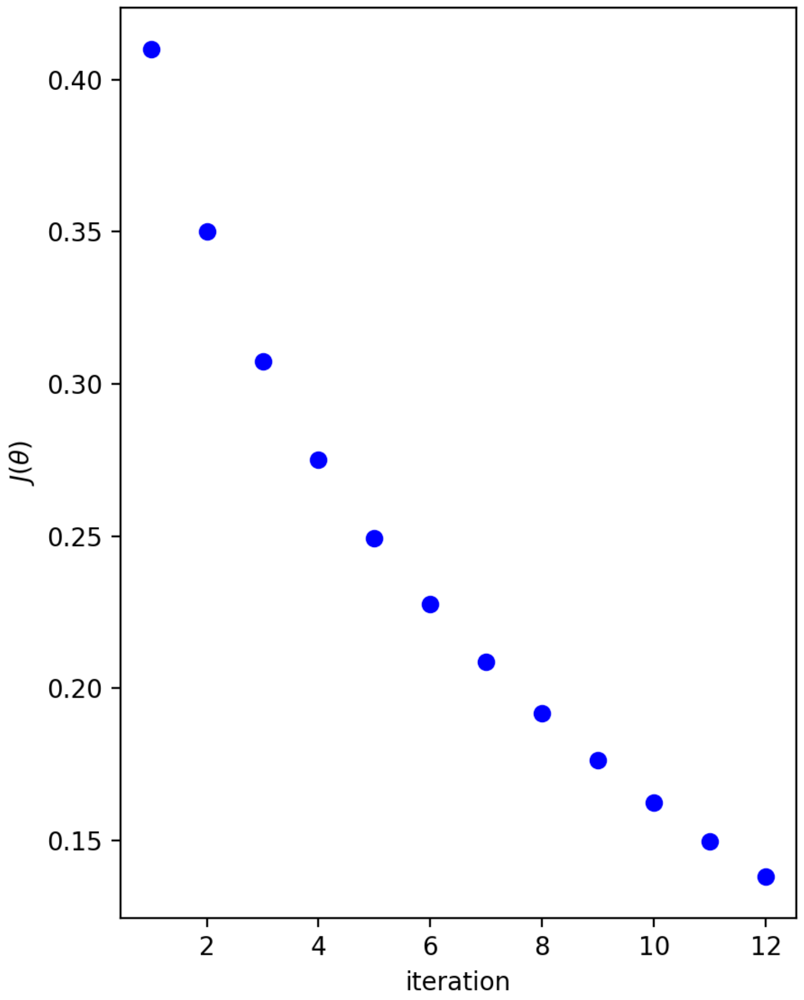
Toy example, iteration 2



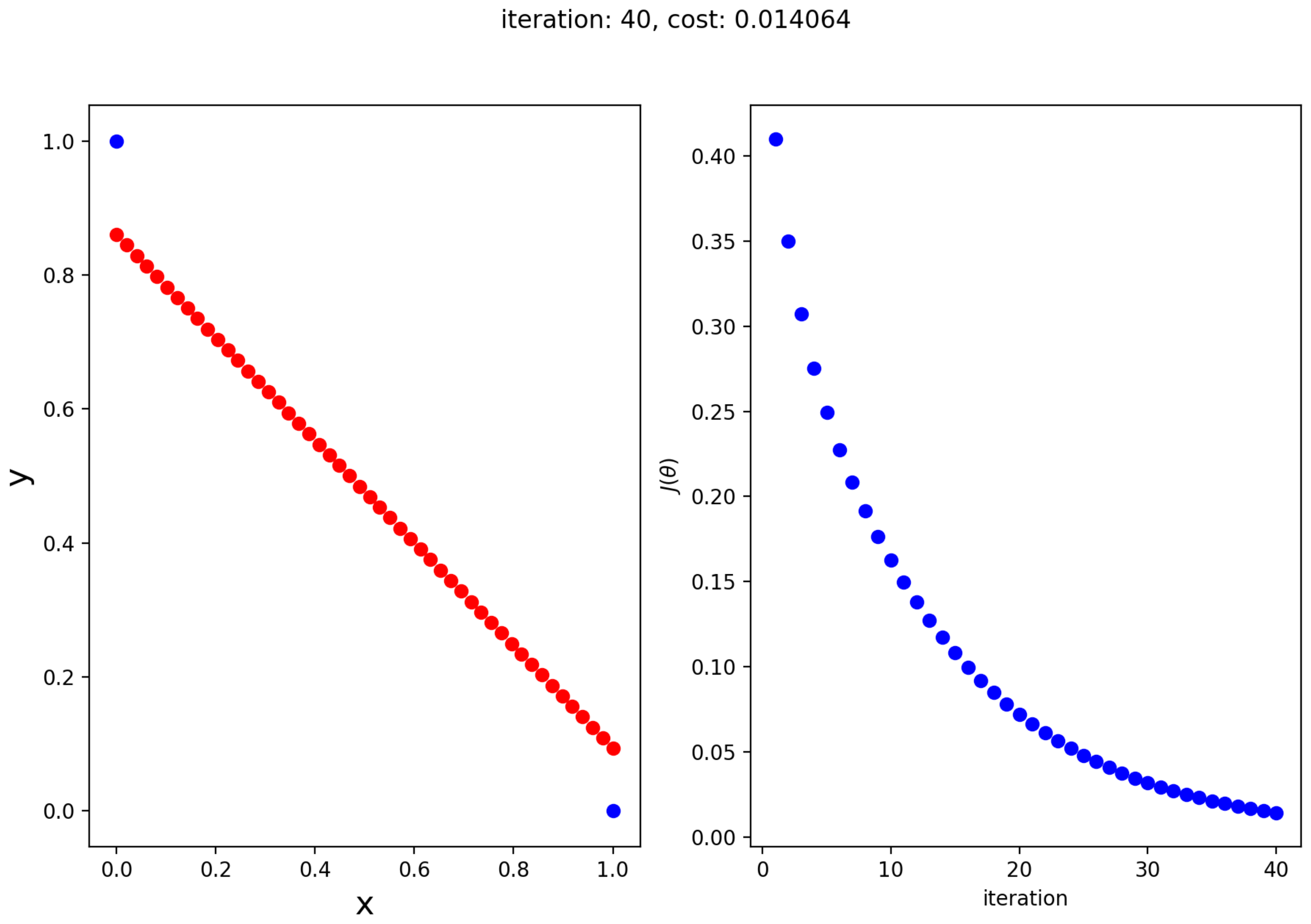
Toy example, iteration 12

iteration: 12, cost: 0.138047

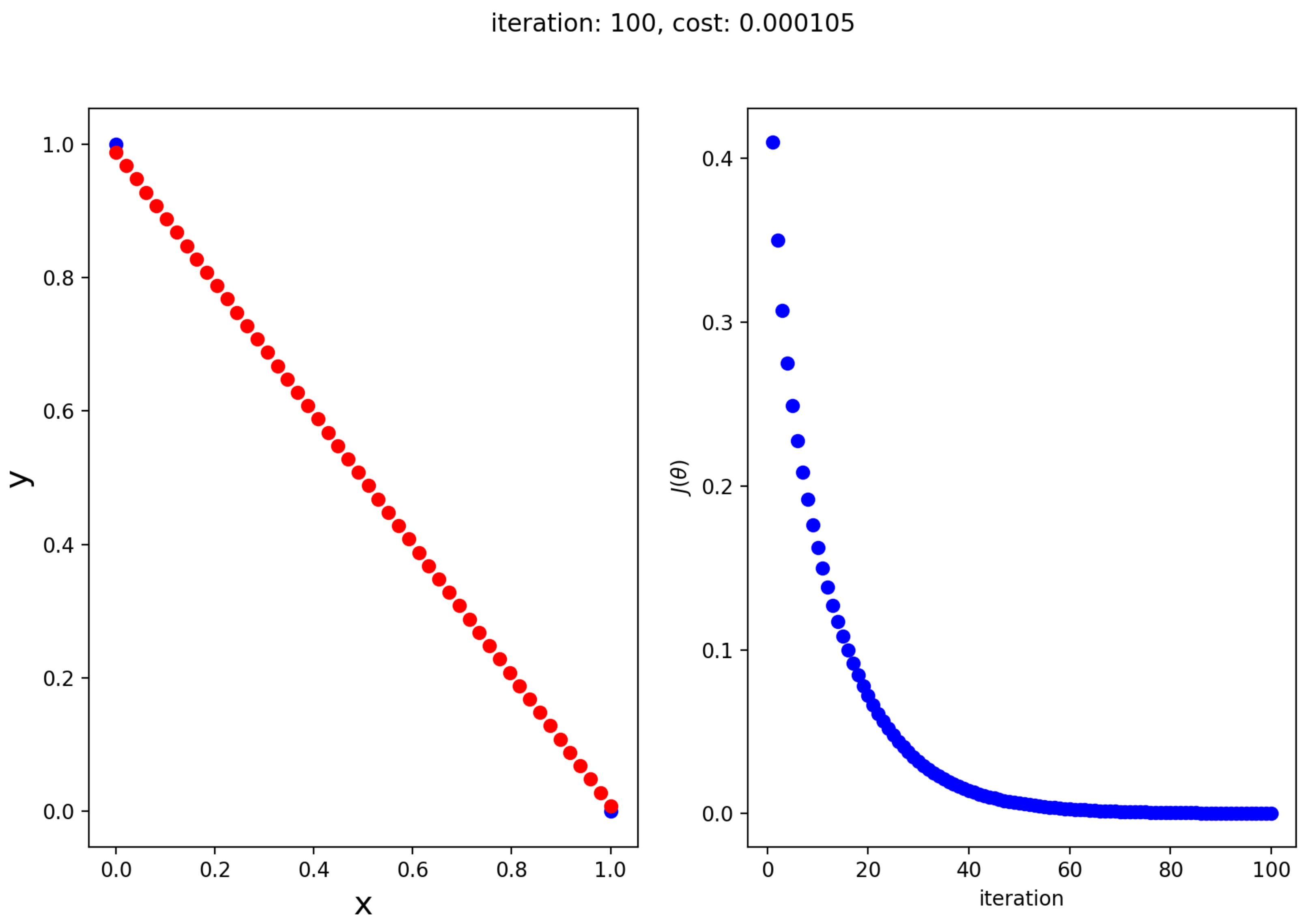




Toy example, iteration 40



Toy example, iteration 100



Outline for September 25

- Recap bias/variance tradeoff
- Simple linear regression
- SGD (Stochastic Gradient Descent)
- Normal equations solution

Pros and Cons

Gradient Descent

- requires multiple iterations
- need to choose α
- works well when p is large
- can support online learning

Normal Equations

- non-iterative
- no need for α
- slow if p is large
  - matrix inversion is $O(p^3)$

Slide: modified from Jessica Wu
Based on slide by Eric Eaton