CS 360: Machine Learning

Prof. Sara Mathieson

Fall 2019

HAVERFORD COLLEGE

Admin

- **Office hour** modifications (all in Hilles 110)
  - Mon 2-3:45pm
  - Tues 12:30-1:30pm
  - Friday 4-5pm

- Lab 3 due **TONIGHT**
- Lab 4 due **Sunday night** (short Naïve Bayes exercise)

- Reading quiz Thursday (just **Duame 9.3** => 2 pages!)
- Midterm next Thursday (short in-lab part + take home)

Technology and Justice Keynote: Thursday!

Keynote: Vulnerabilities: How Social Media and Data Infrastructure are Exploited for Fun, Profit, and Politics

Thursday, September 26, 2019

4:30 p.m.

Sharpless Auditorium

danah boyd is a Principal Researcher at Microsoft Research, the founder and president of Data & Society, and a Visiting Professor at New York University. Her research is focused on addressing social and cultural inequities by understanding the relationship between technology and society.

Feedback forms

Understand well

- KNN 20
- Entropy 7
- Decision Trees 16
- Over/under fitting 1

Needs most work

• Linear regression	13
• SGD	4
• Decision trees	1
• Intuition behind entropy	4
• Bias/variance/loss	2
• Continuous features	2
• Tying topics together	2

Additional questions

Difficulty/workload	1	2	3	4	5
Count			10	15	3

	Less	More	As Is
Slides	2	3	21
Board		7	19
Group	2	5	19
Handouts	1	10	15

- Random partners yes 19
- Random partners no 3
- Neutral 3

Computer policy

Students mentioned:

- noise of typing is distracting to other students
- temptation to look at non-class material is distracting for student and those around them
- laptops take up space on the small tables
- laptops can prevent students from seeing the board

Guidelines:

- Laptops strongly discouraged
- If a laptop will help your interaction with the class material, talk to me ahead of time!

Other thoughts

- Most take notes, posted notes are helpful
- Classroom setup is a bit awkward
- Labs have been long, after getting behind it can be hard to catch up
- More examples, more pseudocode during class
- Several mentioned implementation is a struggle

Outline for September 24

- Recap two ways of solving linear regression
  - SGD
  - Normal equations (closed-form solution)

- Regularization

- Introduction to conditional probability
  - Clinical trials example

- Begin: Naïve Bayes

Outline for September 24

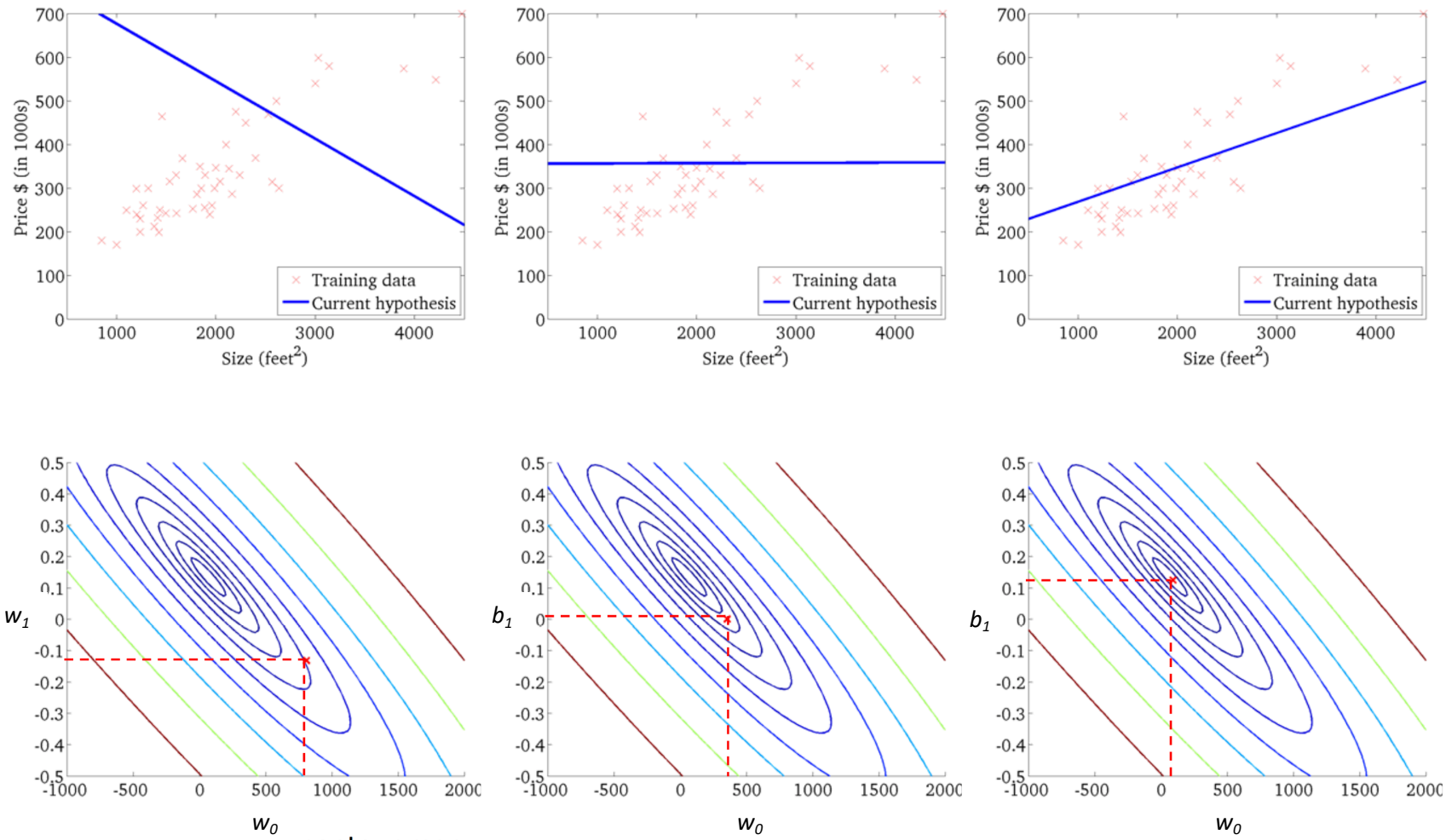
- Recap two ways of solving linear regression
  - SGD
  - Normal equations (closed-form solution)

- Regularization

- Introduction to conditional probability
  - Clinical trials example

- Begin: Naïve Bayes

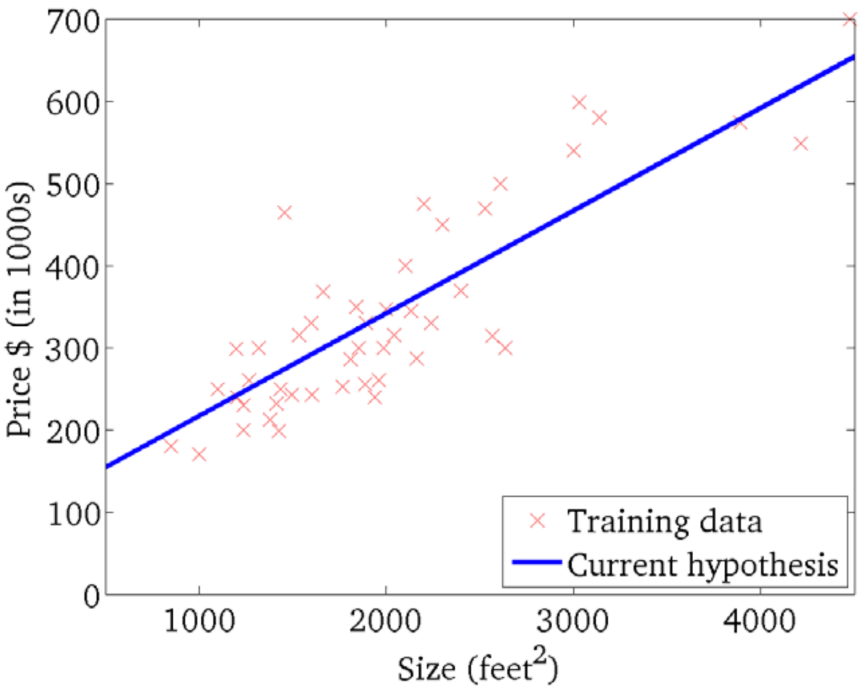
Cost Function

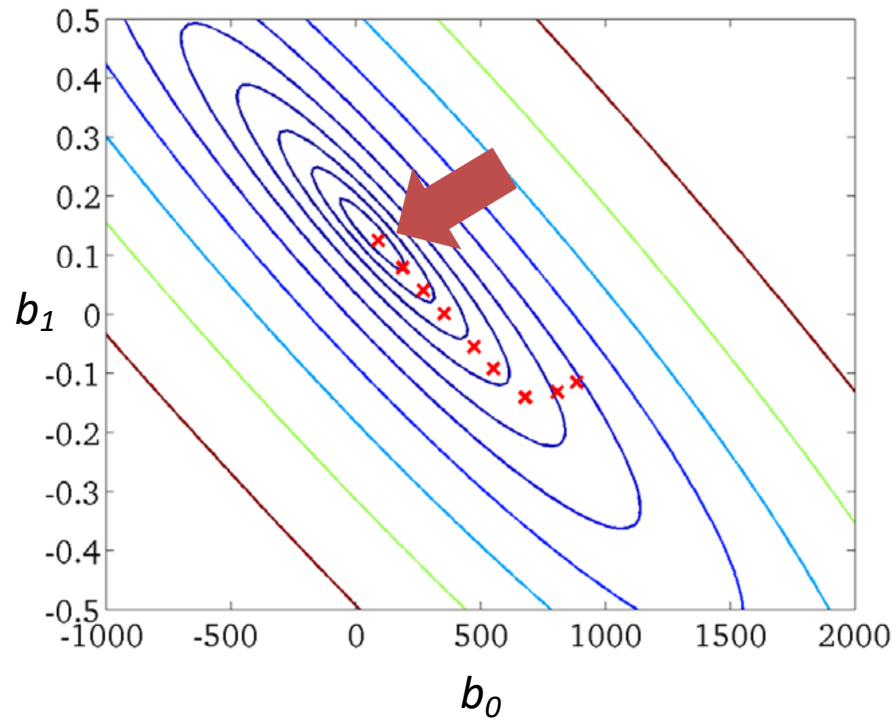


contour map

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

Gradient Descent: walking toward the minimum



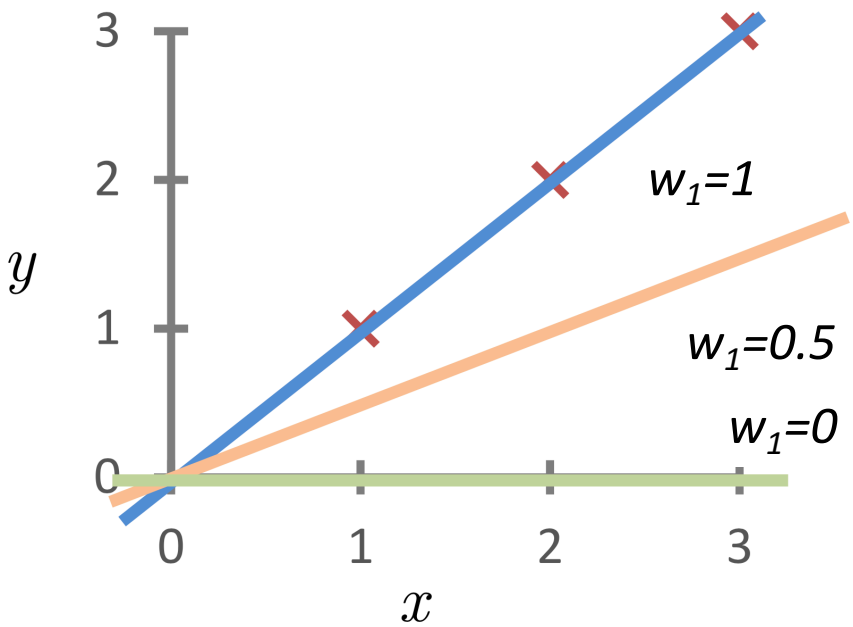


Slide: modified from Jessica Wu
Based on slide by Andrew Ng

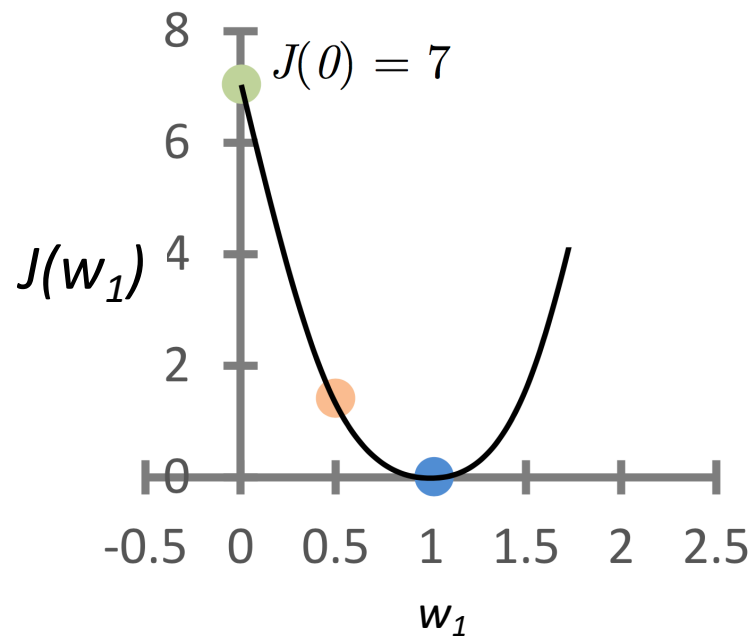
Cost Function (extra practice)

$h_w(x)$

(assume $w_0=0$ for this example)



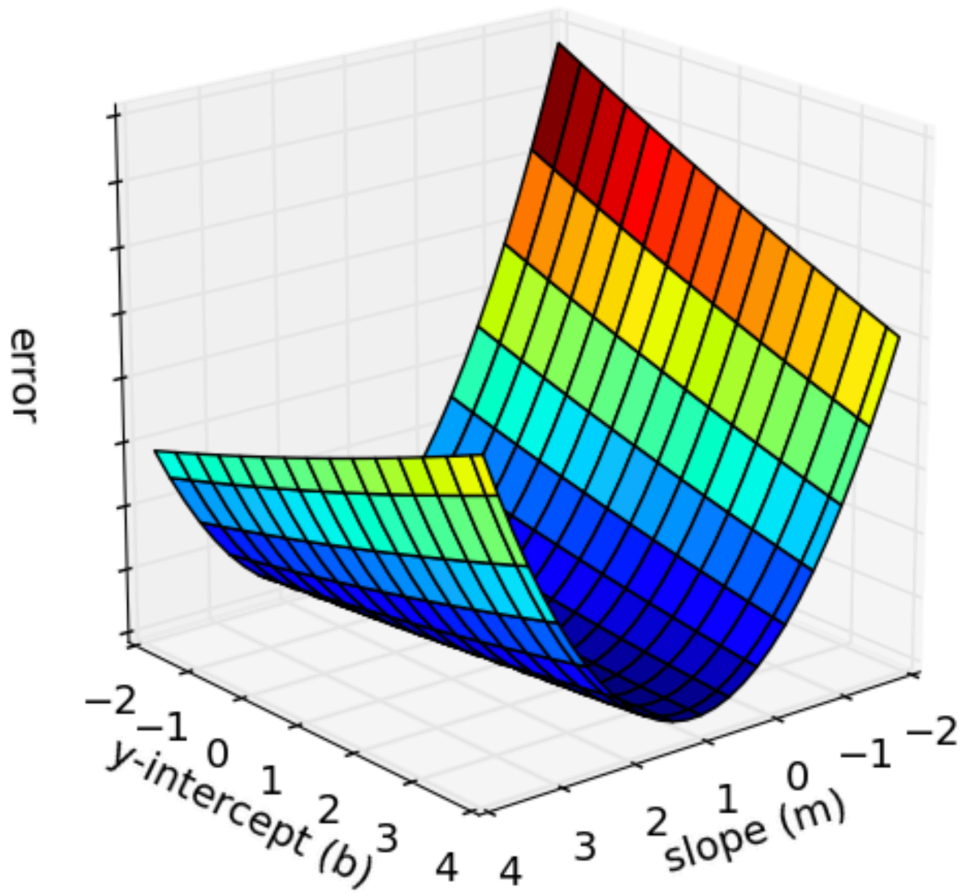
$J(w_1)$

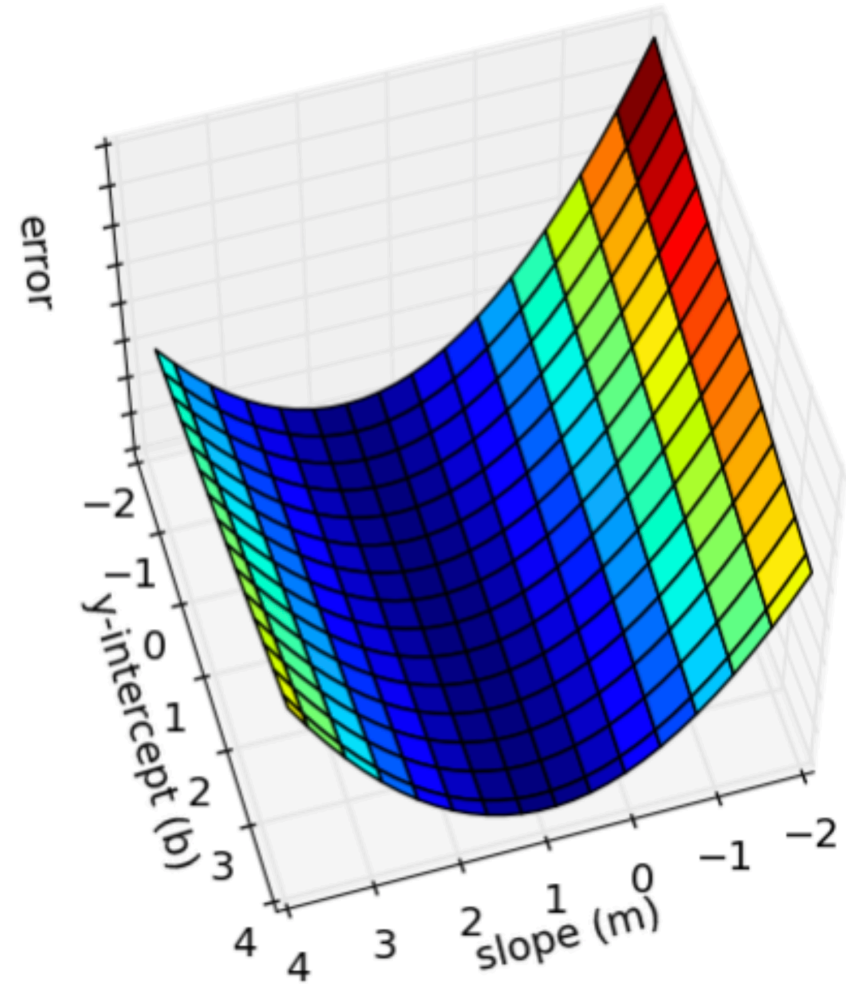


$$J(0.5) = \frac{1}{2}\left[(0.5-1)^2 + (1-2)^2 + (1.5-3)^2\right] = 1.75$$

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

Error as a function of slope & y-intercept



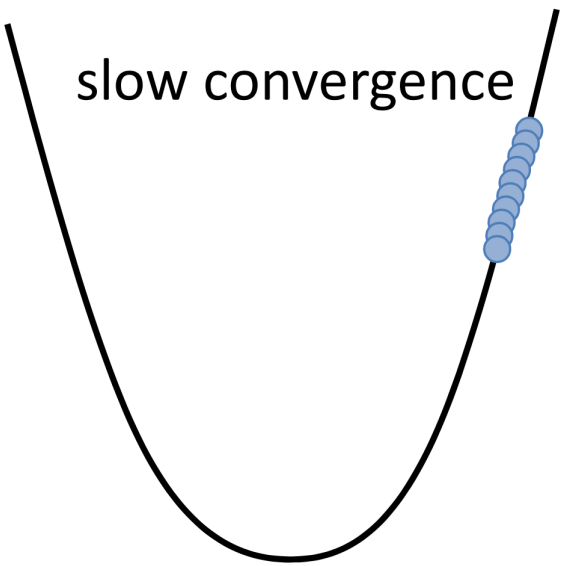


Convex => single global optimum (can prove it is a minimum with second derivative)!

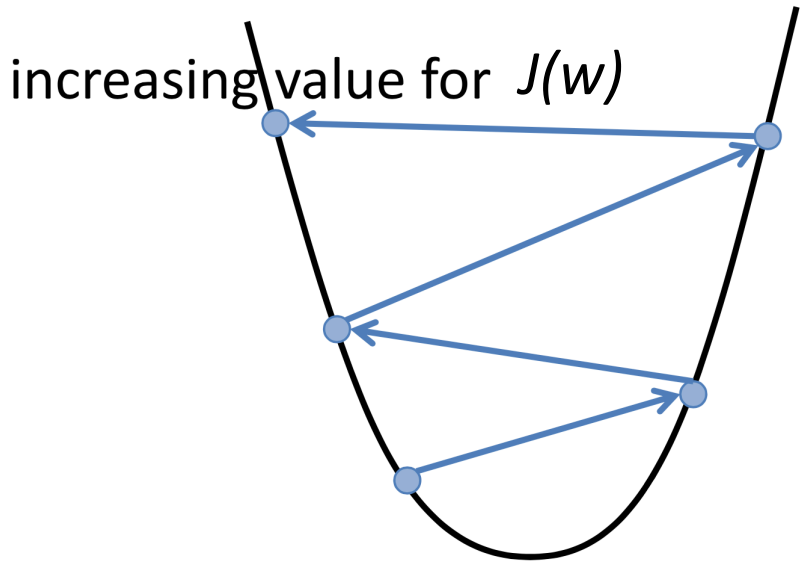
Image: Bryce Wiedenbeck

Choosing step size α

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]

Pros and Cons

Gradient Descent

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

Normal Equations

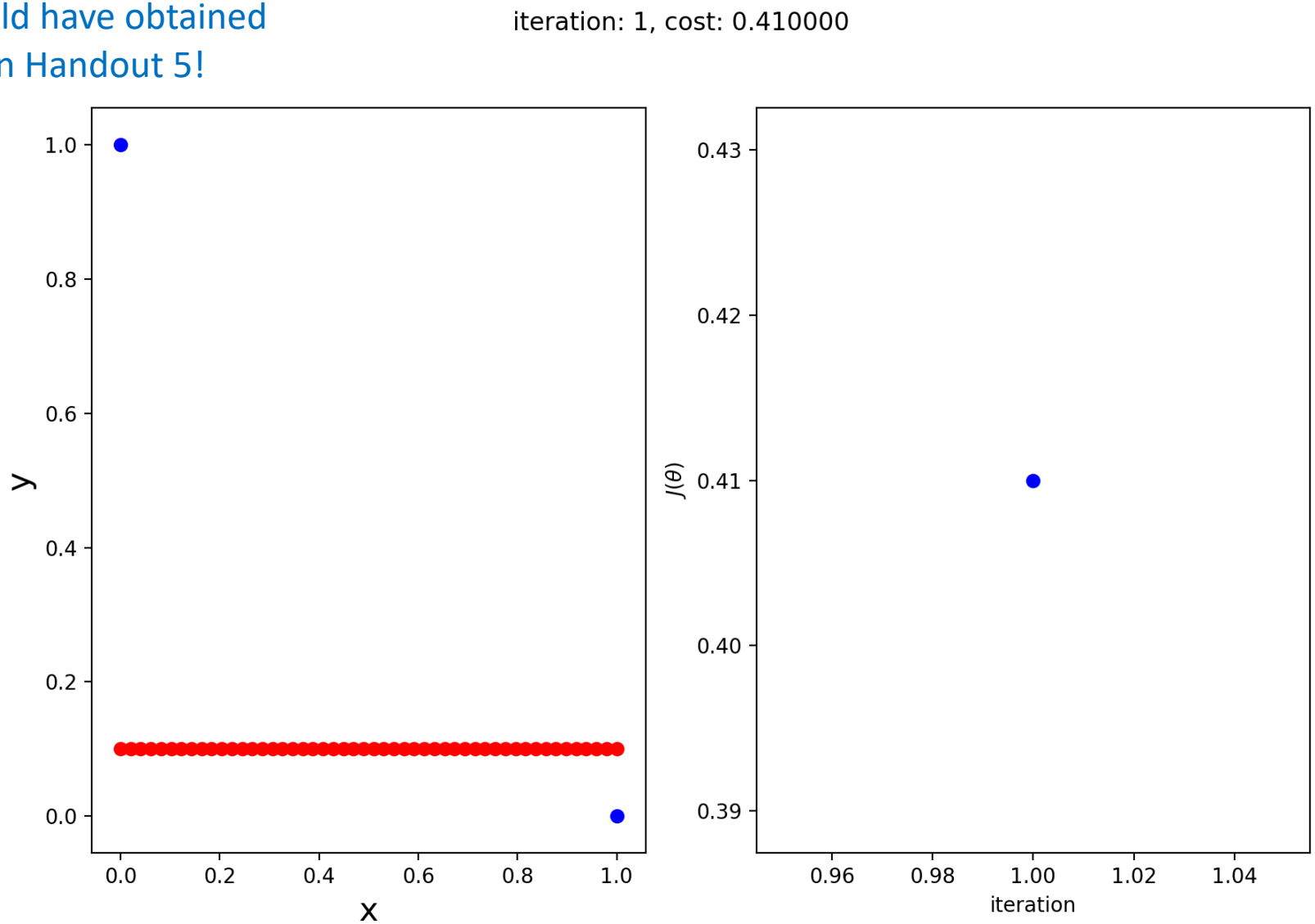
- non-iterative
- no need for α
- slow if p is large
  - matrix inversion is $O(p^3)$

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

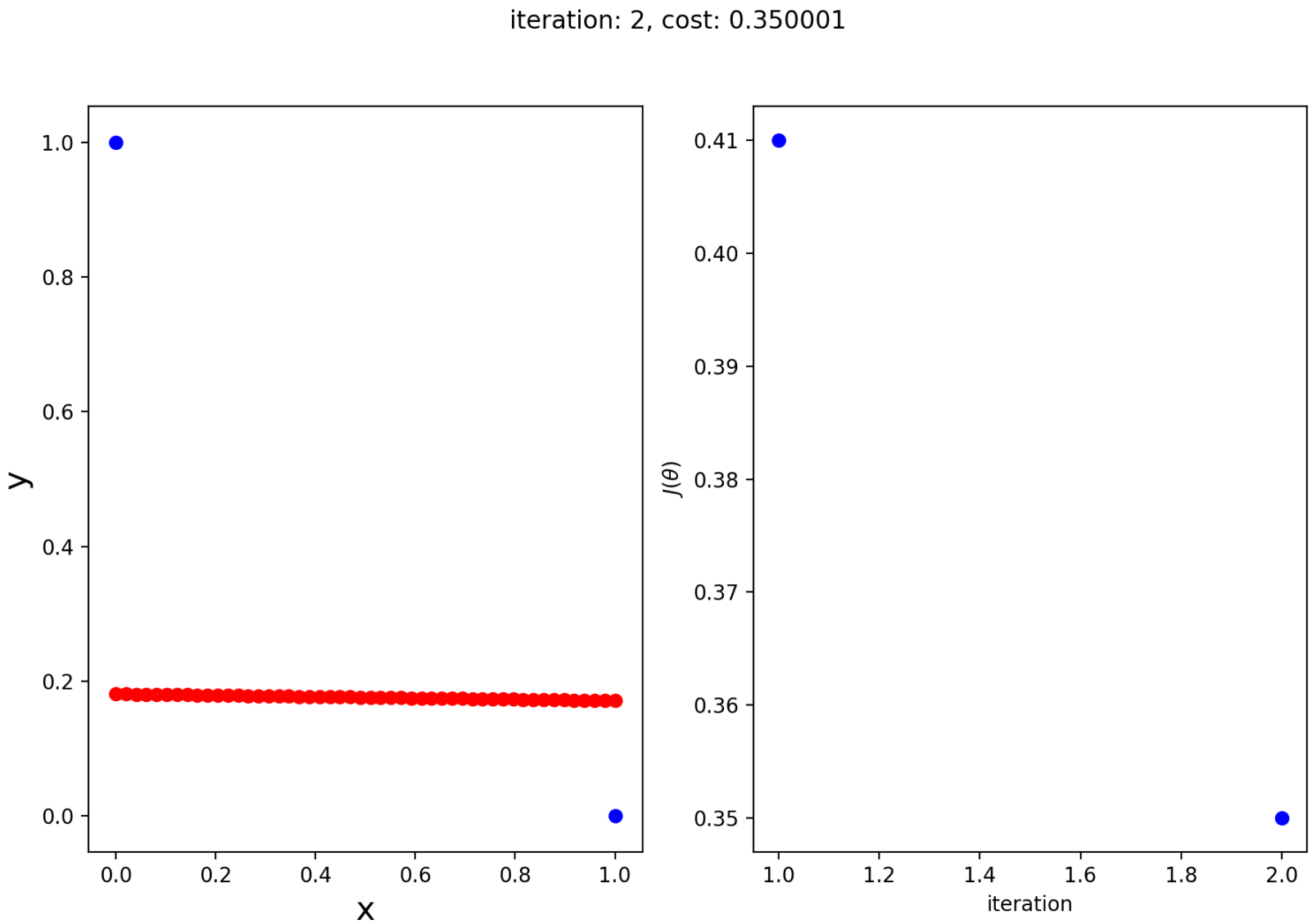
Lab 3 applied to Handout 5

Toy example, iteration 1

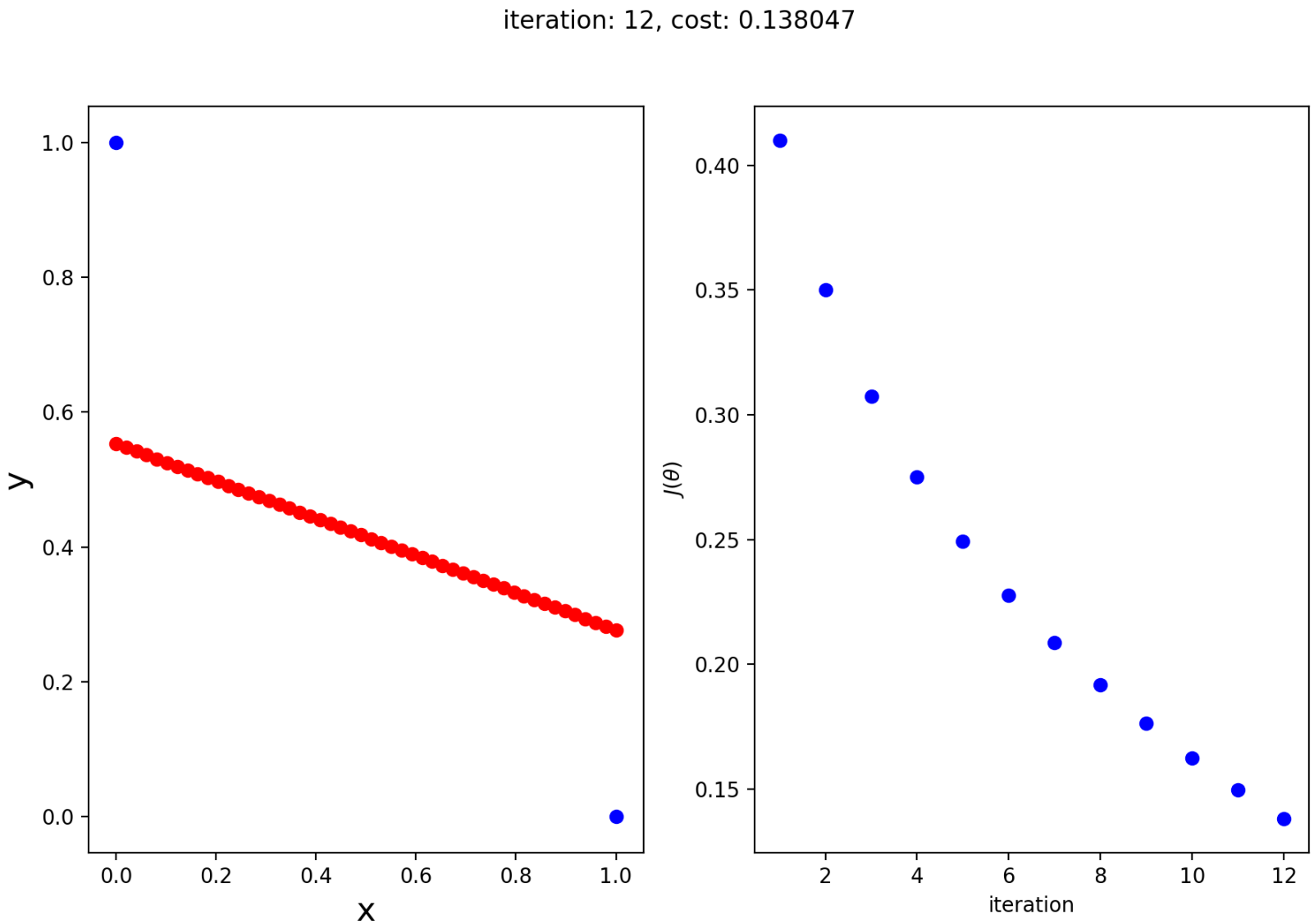
This is what you should have obtained in Handout 5!



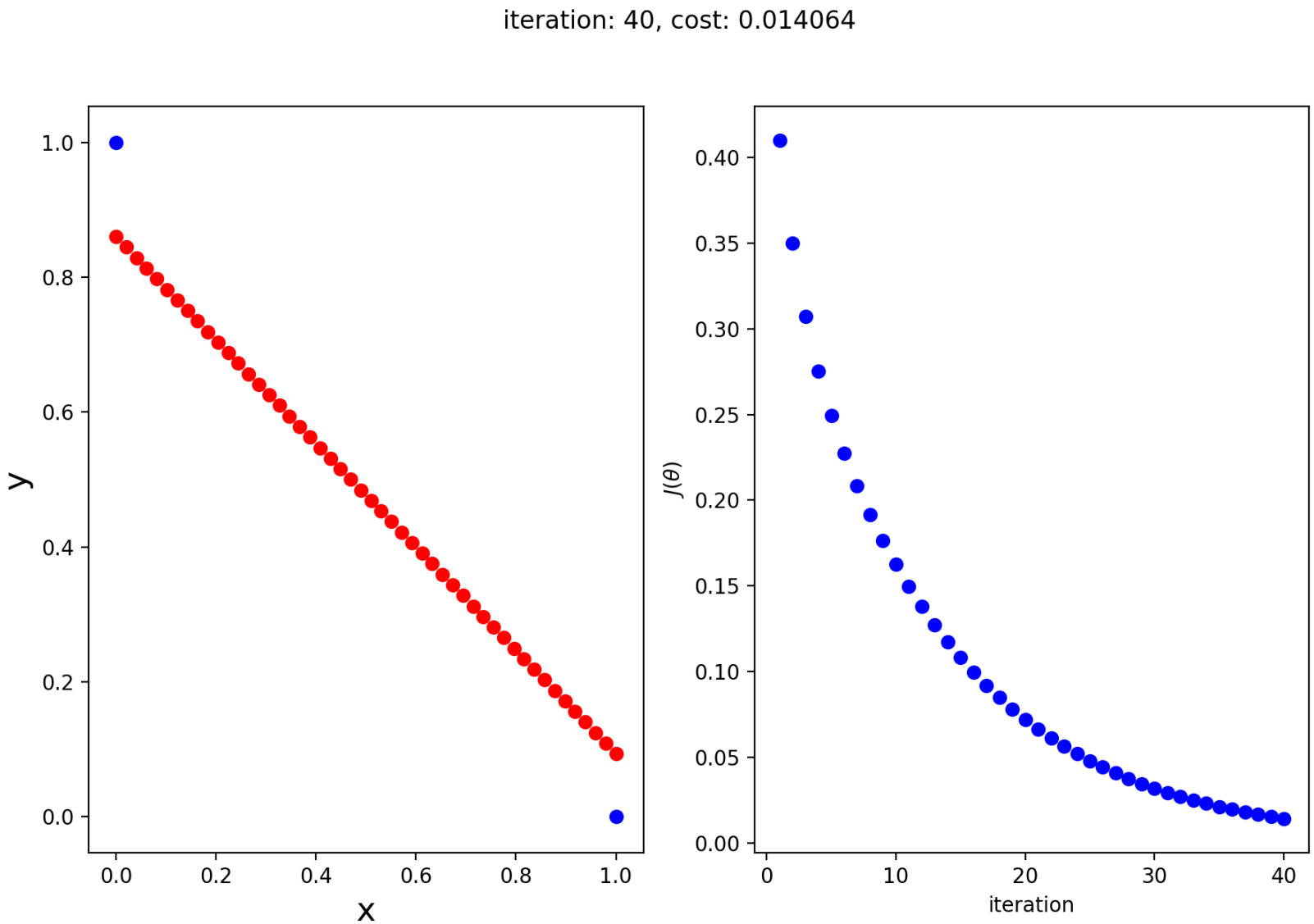
Toy example, iteration 2



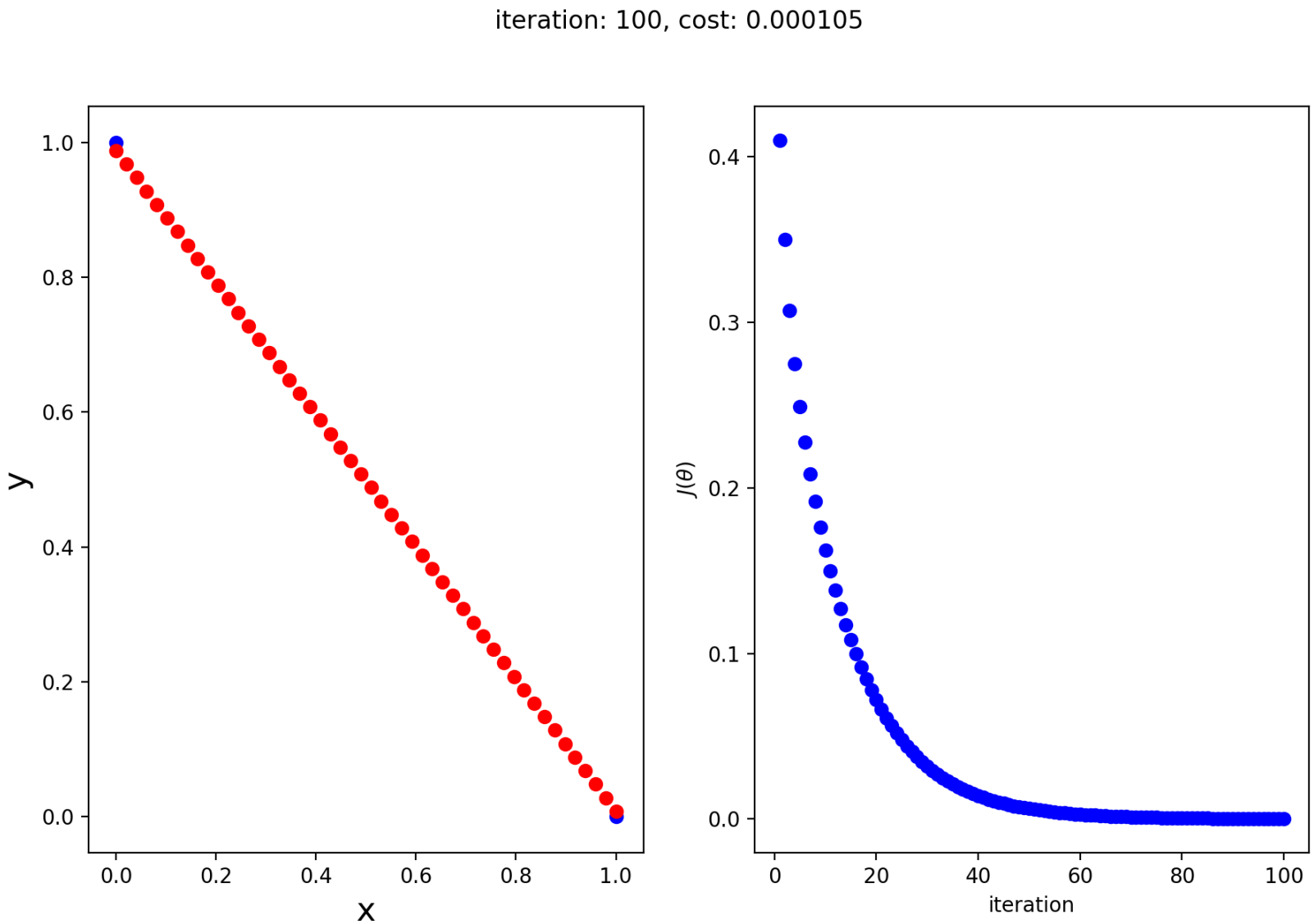
Toy example, iteration 12



Toy example, iteration 40

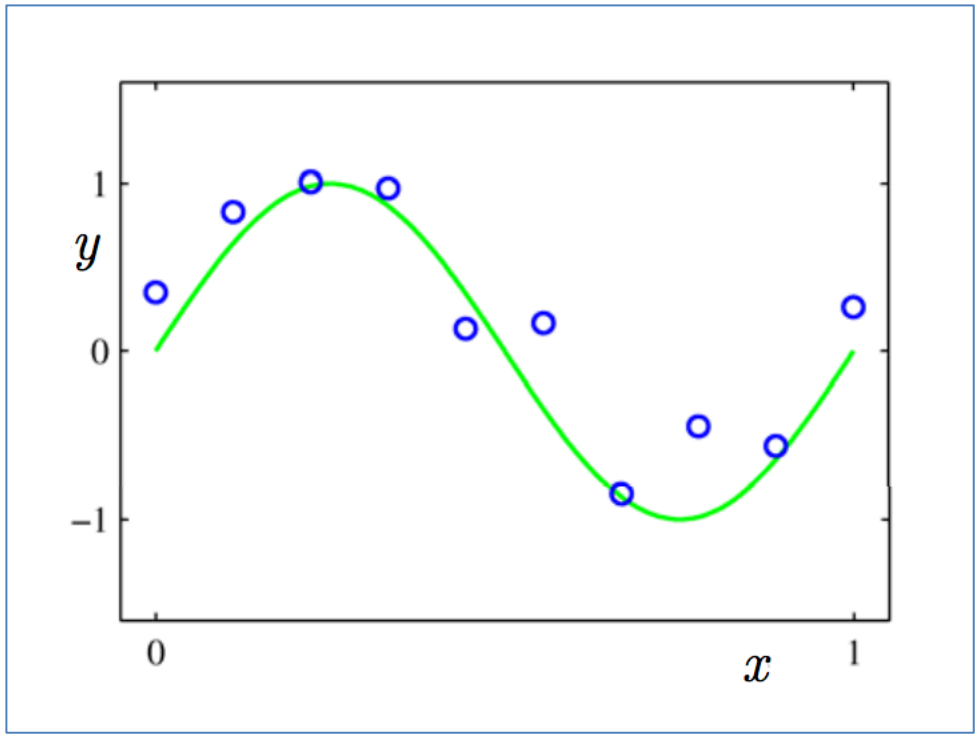


Toy example, iteration 100



Polynomial Regression

- Can be thought of as regular linear regression with a change of basis



Plot: Jessica Wu

Recap

Normal

Equations

Closed form

$$J(\vec{w}) = \frac{1}{2}\sum_{i=1}^{n}\left(\underbrace{\vec{w}^T\vec{x}_i}_{\hat{y}_i} - y_i\right)^2$$

cost function, want to minimize!

$$= \frac{1}{2}\left(\underbrace{X\vec{w} - \vec{y}}_{1\times n}\right)^T\left(\underbrace{X\vec{w} - \vec{y}}_{n\times 1}\right)$$

$$= \frac{1}{2}\left(\underbrace{\vec{w}^T X^T X \vec{w}}_{\underbrace{(X^TX)}_{\text{const}}\text{ "}w^2\text{"}} - \vec{w}^T X^T \vec{y} - \vec{y}^T X \vec{w} + \underbrace{\vec{y}^T\vec{y}}_{\text{const}}\right)$$

equal

$$-2\vec{w}^T X^T \vec{y}$$

$$\underbrace{-2X^T\vec{y}}_{\text{const}}\text{ "}w\text{"}$$

$$\underbrace{[\ \cdot\ ]}_{1\times n}\underbrace{\begin{bmatrix}\cdot\\ \vdots\\ \cdot\end{bmatrix}}_{n\times 1}$$

sum of squares!

$$\underbrace{(A}_{n\times k}\underbrace{B)^T}_{k\times m} = \underbrace{B^T}_{m\times k}\underbrace{A^T}_{k\times n}$$

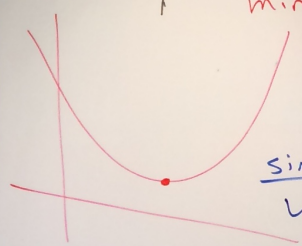
$m\times n$

$$(A-B)^T = A^T - B^T$$

$$\nabla J = X^T X \vec{w} - X^T \vec{y} = 0$$

$$\Rightarrow \boxed{\vec{w} = \underbrace{(X^T X)^{-1}}_{\text{"}\frac{1}{Var(x)}\text{"}} \underbrace{X^T \vec{y}}_{\text{"}Cov(x,y)\text{"}}}$$

minimum (check with second derivative)

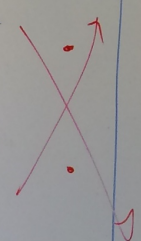


simple linear regression

$$w_1 = \frac{Cov(x,y)}{Var(x)}$$

$$A^{-1}A_x = A^{-1}B$$

$$x = A^{-1}B$$



SGD

usually set $\vec{w} = \vec{0}$ zero vector.

shuffle

for i in range(n):

$$\vec{w} \leftarrow \vec{w} - \alpha \underbrace{(\vec{w}^T \vec{x}_i - y_i)\vec{x}_i}_{\text{gradient of } J \text{ wrt one } \vec{x}_i}$$

updating all w's at the same time.

$$\begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \leftarrow \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} - \underbrace{\alpha(\vec{w}^T \vec{x}_i - y_i)}_{\text{scalar.}} \begin{bmatrix} 1 \\ x_i \end{bmatrix}$$

w_j high (pos) $\Rightarrow$ feature positively correlated w/ output

w_j large (neg) $\Rightarrow$ " negatively " "

$w_j \approx 0 \Rightarrow$ feature is uninformative.

Runtime

T = # of SGD iterations

go through all data

is cost not changing?

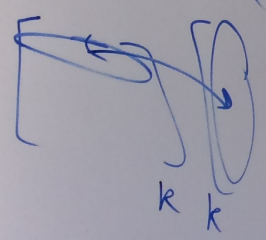
n = # of examples

p = # features

inverse : k dim matrix:

$\Rightarrow O(k^3)$

matrix mult : $(n \times k) \times (k \times m)$



$O(nkm)$

SGD = $O(Tnp)$

p very high => SGD!

o.w. $\Downarrow$

$$\boxed{O(np^2 + p^3)}$$

Normal Eqs =

$\underset{(p \times n)(n \times p)}{X^T X}$ $\}$ $O(np^2)$

$(X^T X)^{-1}$ $\}$ $O(p^3)$

$(X^T X)^{-1} X^T$ $\}$ $O(p^2 n)$

$(p \times p)$ $(p \times n)$

$\cdot \vec{Y}$ $\}$ $O(pn)$

Outline for September 24

- Recap two ways of solving linear regression
  - SGD
  - Normal equations (closed-form solution)

- Regularization

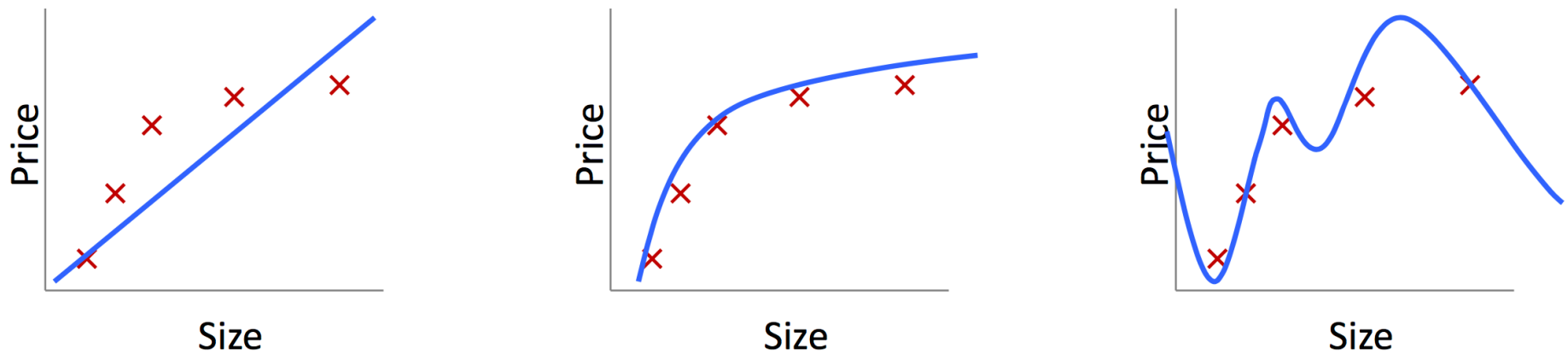
- Introduction to conditional probability
  - Clinical trials example

- Begin: Naïve Bayes

Adapted from slide by Jessica Wu
[example by Andrew Ng]

Generalization error

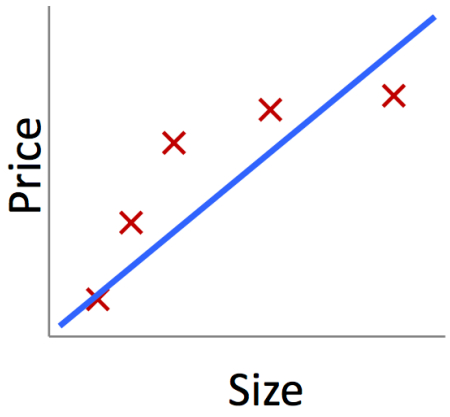
- Example: price vs. size (i.e. of a house or car)



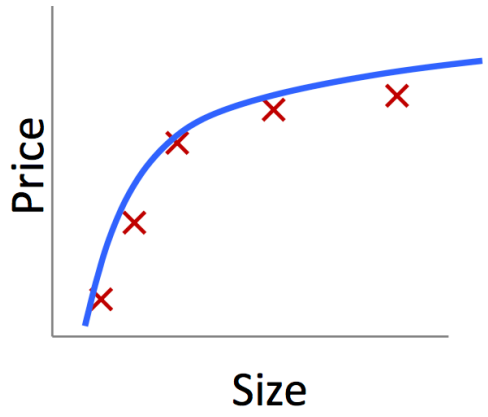
Adapted from slide by Jessica Wu
[example by Andrew Ng]

Generalization error

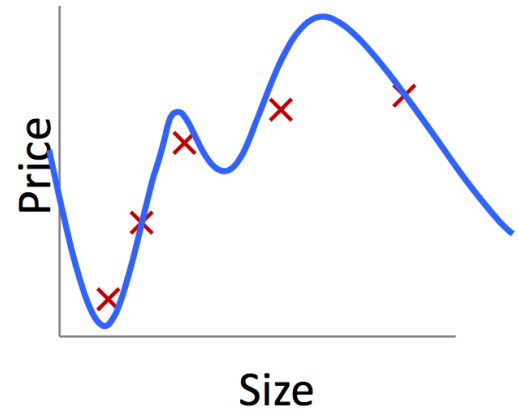
- Example: price vs. size (i.e. of a house or car)



underfitting
(high bias)



correct fit

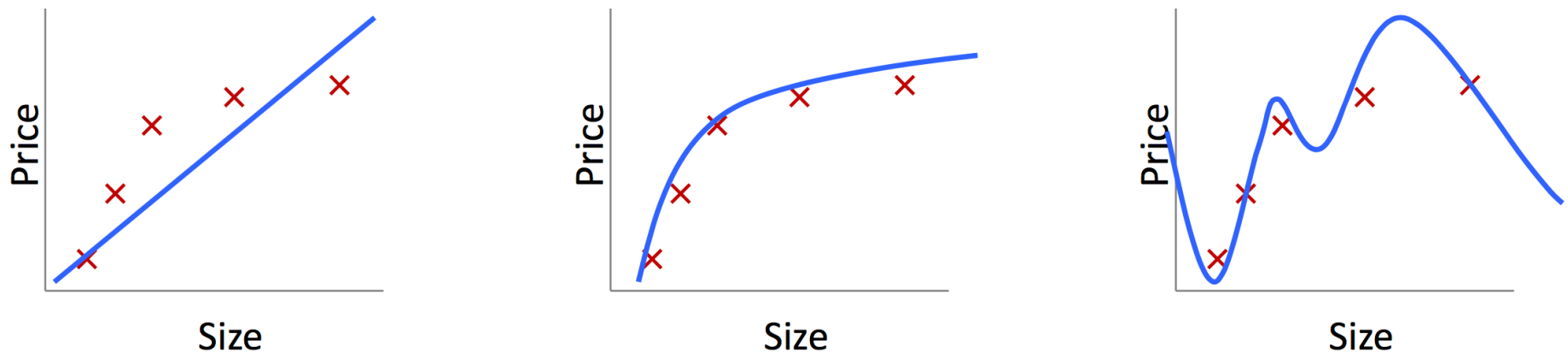


overfitting
(high variance)

Adapted from slide by Jessica Wu
[example by Andrew Ng]

Generalization error

- Example: price vs. size (i.e. of a house or car)



underfitting
(high bias)

correct fit

overfitting
(high variance)

<u>Structural error:</u>
Hypothesis space cannot model true relationship

-More data doesn't help
-Need a more flexible model

balance
⇔

<u>Estimation (approximation) error:</u>
Hypothesis space *can* model true relationship, BUT hard to identify correct model due to large hypothesis space, small *n*, or noise
-Reduce hypothesis space
-Add more data

Regularization

What if ...

- we have a limited # of training examples ($n<p$), or
- we want to automatically control the complexity of the learned hypothesis?

Regularization

What if ...

- we have a limited # of training examples ($n<p$), or
- we want to automatically control the complexity of the learned hypothesis?

Idea: penalize large values of w_j

Why prefer small weights?

- if large weights, small change in feature can result in large change in prediction
- prevent giving too much weight to any one feature
- might prefer zero weight for useless features

Slide by Jessica Wu

Common Regularizers

$$||\vec{w}||_0 = \sum_{j:w_j \neq 0} 1$$

L_0 norm

- Number of non-zero entries
- Minimizing L_0 norm is NP hard

Slide adapted from Jessica Wu

Common Regularizers

$$||\vec{w}||_0 = \sum_{j:w_j \neq 0} 1$$

$$||\vec{w}||_1 = \sum_{j=1}^{p} |w_j|$$

L_0 norm

- Number of non-zero entries
- Minimizing L_0 norm is NP hard

L_1 norm

- Sum of magnitude of weights
- Not differentiable

Slide adapted from Jessica Wu

Common Regularizers

$$||\vec{w}||_0 = \sum_{j:w_j \neq 0} 1$$

$$||\vec{w}||_1 = \sum_{j=1}^{p} |w_j|$$

$$||\vec{w}||_2 = \sqrt{\sum_{j=1}^{p} w_j^2}$$

L_0 norm

- Number of non-zero entries
- Minimizing L_0 norm is NP hard

L_1 norm

- Sum of magnitude of weights
- Not differentiable

L_2 norm

- Sum of squared weights
- Differentiable

Slide adapted from Jessica Wu

Adding regularization to our cost function

$$J(\vec{w}) = \frac{1}{2}\sum_{i=1}^{n}(\vec{w}^T\vec{x}_i - y_i)^2 + \frac{\lambda}{2}\sum_{j=1}^{p} w_j^2$$

don't regularize "bias"!

shift away from 0

"w_0"

$\boxed{\text{SGD}}$

$$\vec{w} \leftarrow \vec{w} - \alpha\left[(\vec{w}^T\vec{x}_i - y_i)\vec{x}_i + \lambda\vec{w}\right]$$

$$\leftarrow \vec{w}\underbrace{(1-\alpha\lambda)}_{\text{pulling weights down}} \ldots$$

$\boxed{\text{normal}}$

$$\boxed{\vec{w} = (X^TX + \lambda I^*)^{-1}X^T\vec{y}}$$

identity

don't penalize w_0

$$\lambda\begin{bmatrix} \cancel{1}\,0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \rightarrow I^*$$

λ is small and positive

Outline for September 24

- Recap two ways of solving linear regression
  - SGD
  - Normal equations (closed-form solution)
- Regularization
- Introduction to conditional probability
  - Clinical trials example
- Begin: Naïve Bayes

Probability

and "joint"

$$\boxed{P(A, B) = P(A)P(B|A)}$$

$$= P(A \underbrace{\cap}_{\text{and}} B)$$

Bayes Rule.

$$= P(B)P(A|B)$$

$$\boxed{P(A|B) = \frac{P(A)P(B|A)}{P(B)}}$$

$$P(\underbrace{R}_{\text{rain}}) = 20\%$$

$$P(R, \underbrace{U}_{\text{umbrella}}) = 15\%$$

$$P(U|R) = ?$$

"given" "given it is raining"

$$P(U|R) = \frac{P(U,R)}{P(R)}$$

$$= \frac{15\%}{20\%}$$

$$= 75\%$$