Data, Environment and Society:
Lecture 13: Gradient Descent

* a bit of EJ?

Instructor: Duncan Callaway
GSI: Seigi Karasaki

October 4, 2018

Announcements

Today

- Gradient descent
- Environmental Justice

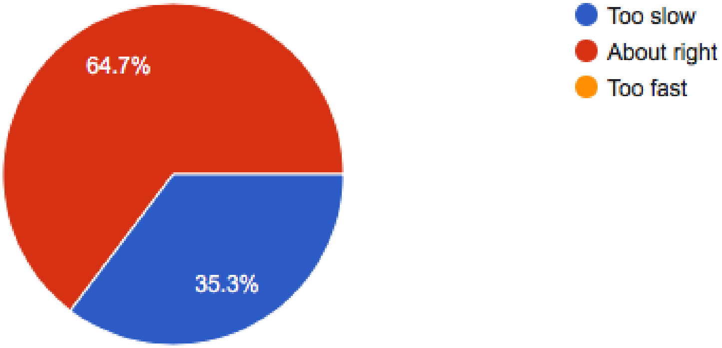
Reading

- Today: Ch 11 DS100
- Next *thursday:* Clark *et al* (using LUR data for EJ questions)

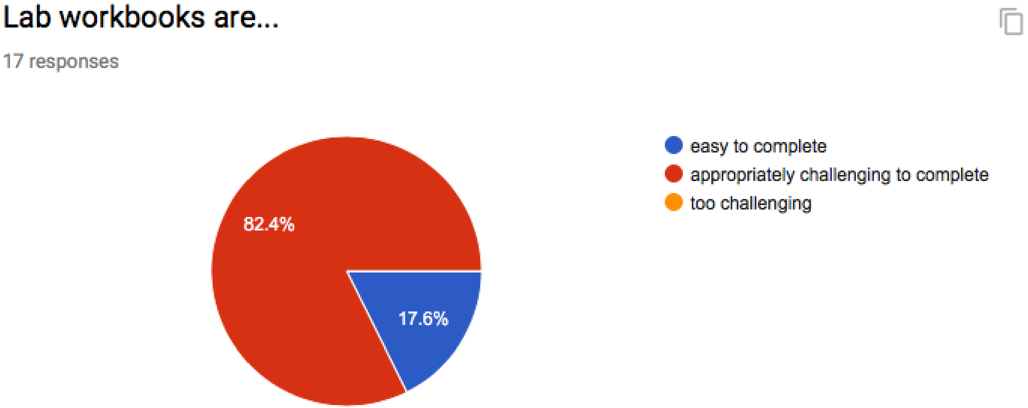
Survey results

Lecture pace is

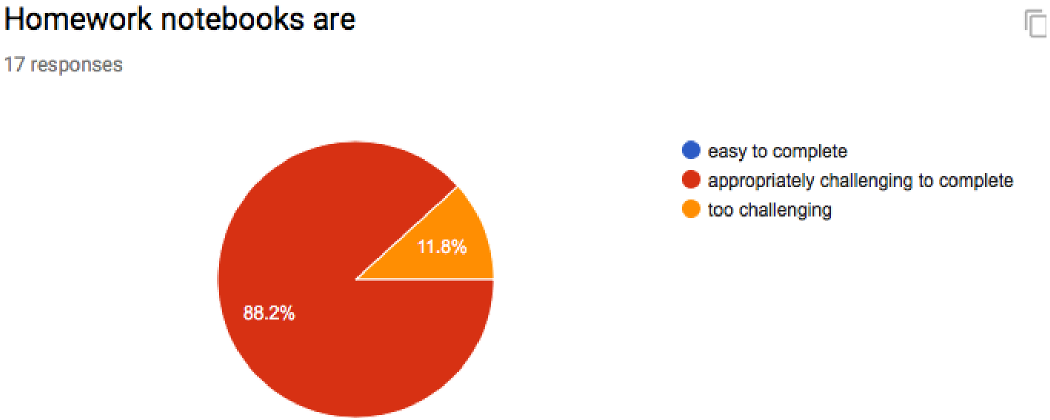
17 responses



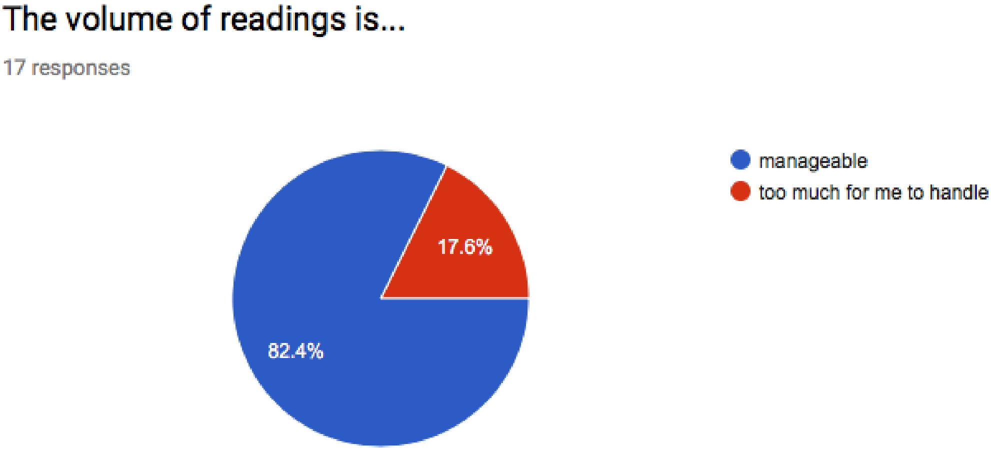
Survey results



Survey results



Survey results



Survey results: A few key takeaways

- Students asked for more time for discussion and interaction
- A few students suggested I assume background reading is done...
- Request for more board work
- Requests for more energy-enviro applications
- Students are struggling to find a way to take notes
- Grading rubric, more clarity on questions in HW and Labs
- Lots of positive feedback for Seigi

Survey results: A few key takeaways

- Students asked for more time for discussion and interaction
  - Will work to make this change
- A few students suggested I assume background reading is done...
- Request for more board work
- Requests for more energy-enviro applications
- Students are struggling to find a way to take notes
- Grading rubric, more clarity on questions in HW and Labs
- Lots of positive feedback for Seigi

Survey results: A few key takeaways

- Students asked for more time for discussion and interaction
  - Will work to make this change
- A few students suggested I assume background reading is done...
  - I'm trying to do this, then reinforcing what I feel are key points from ISLR.
- Request for more board work
- Requests for more energy-enviro applications
- Students are struggling to find a way to take notes
- Grading rubric, more clarity on questions in HW and Labs
- Lots of positive feedback for Seigi

Survey results: A few key takeaways

- Students asked for more time for discussion and interaction
  - Will work to make this change
- A few students suggested I assume background reading is done...
  - I'm trying to do this, then reinforcing what I feel are key points from ISLR.
- Request for more board work
  - I will work to do more on the iPad – is that working?
- Requests for more energy-enviro applications
- Students are struggling to find a way to take notes
- Grading rubric, more clarity on questions in HW and Labs
- Lots of positive feedback for Seigi

Survey results: A few key takeaways

- Students asked for more time for discussion and interaction
  - Will work to make this change
- A few students suggested I assume background reading is done...
  - I'm trying to do this, then reinforcing what I feel are key points from ISLR.
- Request for more board work
  - I will work to do more on the iPad – is that working?
- Requests for more energy-enviro applications
  - Trying! Also need to make sure we cover methods...but we'll try to get more in.
- Students are struggling to find a way to take notes
- Grading rubric, more clarity on questions in HW and Labs
- Lots of positive feedback for Seigi

Survey results: A few key takeaways

- Students asked for more time for discussion and interaction
  - Will work to make this change
- A few students suggested I assume background reading is done...
  - I'm trying to do this, then reinforcing what I feel are key points from ISLR.
- Request for more board work
  - I will work to do more on the iPad – is that working?
- Requests for more energy-enviro applications
  - Trying! Also need to make sure we cover methods...but we'll try to get more in.
- Students are struggling to find a way to take notes
  - Any suggestions?
- Grading rubric, more clarity on questions in HW and Labs

- Lots of positive feedback for Seigi

Survey results: A few key takeaways

- Students asked for more time for discussion and interaction
  - Will work to make this change
- A few students suggested I assume background reading is done...
  - I'm trying to do this, then reinforcing what I feel are key points from ISLR.
- Request for more board work
  - I will work to do more on the iPad – is that working?
- Requests for more energy-enviro applications
  - Trying! Also need to make sure we cover methods...but we'll try to get more in.
- Students are struggling to find a way to take notes
  - Any suggestions?
- Grading rubric, more clarity on questions in HW and Labs
  - We may not get a rubric, but we will work to clarify and ensure fair grading
- Lots of positive feedback for Seigi

Survey results: A few key takeaways

- Students asked for more time for discussion and interaction
  - Will work to make this change
- A few students suggested I assume background reading is done...
  - I'm trying to do this, then reinforcing what I feel are key points from ISLR.
- Request for more board work
  - I will work to do more on the iPad – is that working?
- Requests for more energy-enviro applications
  - Trying! Also need to make sure we cover methods...but we'll try to get more in.
- Students are struggling to find a way to take notes
  - Any suggestions?
- Grading rubric, more clarity on questions in HW and Labs
  - We may not get a rubric, but we will work to clarify and ensure fair grading
- Lots of positive feedback for Seigi
  - Your GSI rocks!

Basic estimation process, so far

1. Define a loss function
2. Set derivatives of loss function equal to zero and solve for parameters

The challenge:

- Setting loss function derivatives to zero not always easy.
- This doesn't scale well for big problems (e.g. many different nonlinear transformations of the Novotny data)

The loss function

Mean squared error, aka the 'L2' norm

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

constant model $\Rightarrow \quad \hat{y}_i = \theta \quad \Rightarrow \quad MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - \theta)^2$

Mean absolute error, aka the 'L1' norm

$$MAE = \frac{1}{n} \sum_{i=1}^{n} | y_i - \theta |$$

The loss function

Mean squared error, aka the ‘L2’ norm

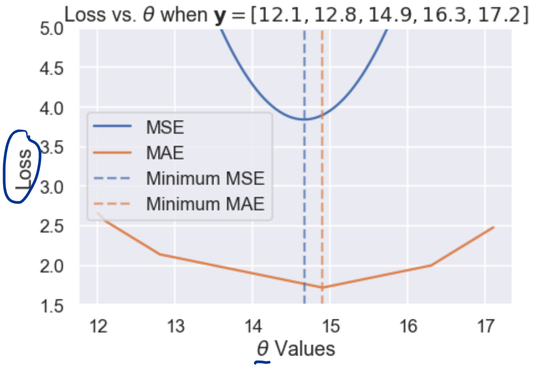
$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

$$\text{Constant model,} \hat{y} = \theta \rightarrow \text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \theta)^2$$

Mean absolute error, aka the ‘L1’ norm

$$\begin{aligned}\text{MAE} &= \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \\ &= \frac{1}{n}\sum_{i=1}^{n}|y_i - \theta|\end{aligned}$$

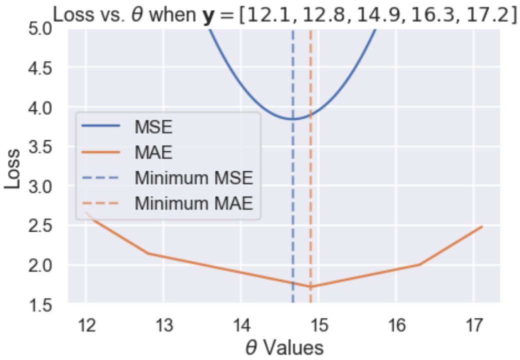
Advantages and disadvantages to MAE and MSE?



← data set

MSE : Continuously dif'able

Advantages and disadvantages to MAE and MSE?



- MSE is differentiable $\rightarrow$ can solve directly for coefficients
- MAE is less impacted by extreme values

Aside: what do these cost functions provide with the "constant" model?

What well-known values minimize these loss functions?

$$\theta^*_{\text{MSE}} = \arg\min_\theta \frac{1}{n}\sum_{i=1}^{n}(y_i - \theta)^2$$

$$\theta^*_{\text{MAE}} = \arg\min_\theta \frac{1}{n}\sum_{i=1}^{n}|y_i - \theta|$$

$$\rightarrow \frac{\partial}{\partial\theta}\frac{1}{n}\sum(y_i-\theta)^2 = 0$$

$$\frac{1}{n}2(y_i-\theta)\frac{\partial(-\theta)}{\partial\theta} = 0$$

$$-\frac{2}{n}\sum(y_i-\theta) = 0$$

$$-\frac{2}{n}\left[\sum_{i=1}^{n}y_i - \sum_{i=1}^{n}\theta\right] = 0$$

$$\sum_{i=1}^{n}y_i = \sum_{i=1}^{n}\theta$$

$$\sum_{i=1}^{n}y_i = n\theta$$

$$\Rightarrow \theta = \frac{1}{n}\sum_{i=1}^{n}y_i$$

$\Rightarrow$ Mean!

Aside: what do these cost functions provide with the "constant" model?

What well-known values minimize these loss functions?

$$\theta^*_{\text{MSE}} = \arg\min_{\theta} \frac{1}{n}\sum_{i=1}^{n}(y_i - \theta)^2$$

$$\theta^*_{\text{MAE}} = \arg\min_{\theta} \frac{1}{n}\sum_{i=1}^{n}|y_i - \theta|$$

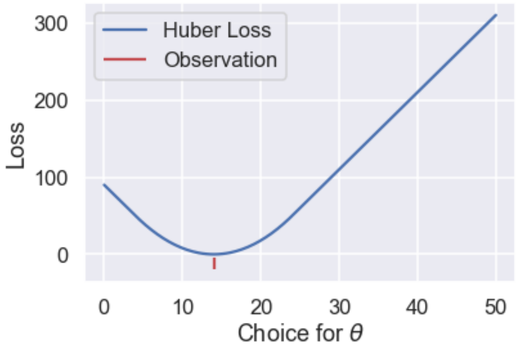
- MSE returns the mean value of a sequence
- MAE returns the *median*

Huber loss

What does this buy us?

$$L_\delta(\theta, \mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n} \begin{cases} \frac{1}{2}(y_i - \theta)^2 & |y_i - \theta| \le \delta \\ \delta(|y_i - \theta| - \frac{1}{2}\delta) & \text{otherwise} \end{cases}$$

Huber loss

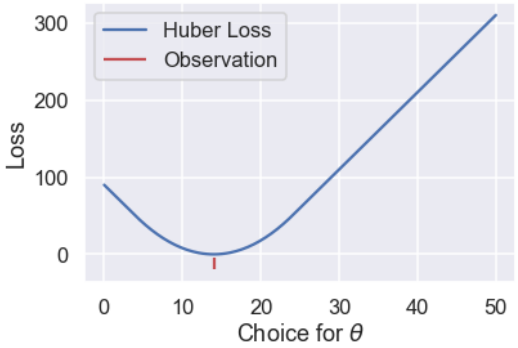


What does this buy us?

- Differentiable
- Absolute value at extremes – not dominated by outlier.

$$L_\delta(\theta, \mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n} \begin{cases} \frac{1}{2}(y_i - \theta)^2 & |y_i - \theta| \le \delta \\ \delta(|y_i - \theta| - \frac{1}{2}\delta) & \text{otherwise} \end{cases}$$

Huber loss



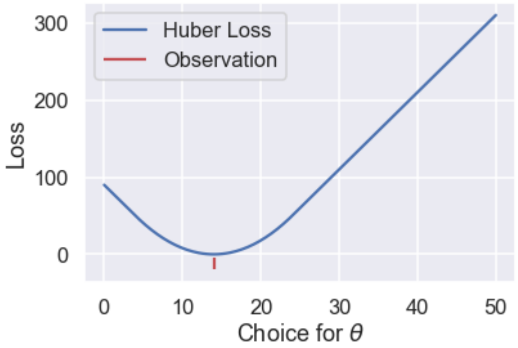
What does this buy us?

- Differentiable
- Absolute value at extremes – not dominated by outlier.

What does this cost us?

$$L_\delta(\theta, \mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n}\begin{cases}\frac{1}{2}(y_i - \theta)^2 & |y_i - \theta| \le \delta \\ \delta(|y_i - \theta| - \frac{1}{2}\delta) & \text{otherwise}\end{cases}$$

Huber loss



$$L_\delta(\theta, \mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n}\begin{cases}\frac{1}{2}(y_i - \theta)^2 & |y_i - \theta| \le \delta \\ \delta(|y_i - \theta| - \frac{1}{2}\delta) & \text{otherwise}\end{cases}$$

What does this buy us?

- Differentiable
- Absolute value at extremes – not dominated by outlier.

What does this cost us?

- Optimal solution requires derivative w.r.t. θ *and* derivative w.r.t. δ equal zero.
- That can be tricky.

Estimation takeaway # 1:

Analytical solutions for parameters (e.g. by setting partial derivatives equal to zero) not always available for some of the types of loss functions we'd like to use.

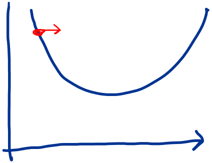
Estimation takeaway # 2:

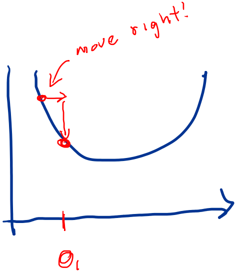
A separate issue: In situations where the normal equations (or something like them) can be used to solve for parameters:

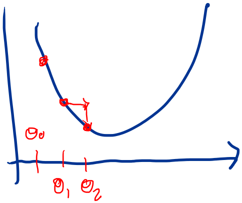
$$\Theta = (X^T X)^{-1} X^T Y$$

It can be very difficult computationally to invert a large $X^T X$ (I crashed my computer with 50,000 by 50,000).

Gradient descent – sketch

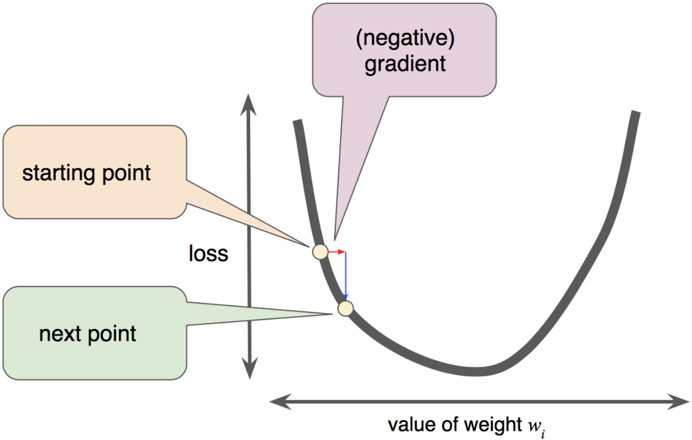






How big should the move be?

Gradient descent – sketch



https://developers.google.com/machine-learning/crash-course/reducing-loss/gradient-descent

Gradient descent – math

What's the gradient? For our purposes, it is the slope of the loss function at a given point *with respect to a particular parameter.*

The gradient is $\nabla_\theta L(\theta, \mathbf{y}) =$ $\frac{\partial}{\partial \theta} L(\theta, y)$

Gradient descent process:

1. Choose a value for the "learning rate", α
2. Choose a starting value of θ (0 is a common choice).
3. Compute $\theta - \alpha \cdot \frac{\partial}{\partial \theta} L(\theta, \mathbf{y})$ and store this as the new value of θ.
4. Repeat until θ doesn't change (much) between iterations.

Gradient descent – math

What's the gradient? For our purposes, it is the slope of the loss function at a given point *with respect to a particular parameter.*

The gradient is $\nabla_\theta L(\theta, \mathbf{y}) = \frac{\partial}{\partial\theta} L(\theta, \mathbf{y})$.

Gradient descent process:

1. Choose a value for the "learning rate", α
2. Choose a starting value of θ (0 is a common choice).
3. Compute $\theta - \alpha \cdot \frac{\partial}{\partial\theta} L(\theta, \mathbf{y})$ and store this as the new value of θ.
4. Repeat until θ doesn't change (much) between iterations.

Gradient descent for quadratic loss

Let's derive the gradient:

Gradient descent for quadratic loss

Let's derive the gradient:

$$L = \sum_{i=1}^{n} (y_i - \theta)^2$$

$$\frac{\partial L}{\partial \theta} = -2 \sum_{i=1}^{n} (y_i - \theta)$$

...and then write a few iterations:

$$\theta_0 = 0$$

$$\theta_1 = \theta_0 - \alpha \left(-2 \sum_{i=1}^{n} (y_i - \theta_0)\right)$$

$$\theta_2 = \theta_1 - \alpha \left(-2 \sum_{i=1}^{n} (y_i - \theta_1)\right)$$

$$\vdots$$

$$\theta_{t+1} = \theta_t - \alpha \left(-2 \sum (y_i - \theta_t)\right)$$

$$|\theta_{t+1} - \theta_t| < \text{tolerance}$$

$\Rightarrow$ stop!

Gradient descent for quadratic loss

Let's derive the gradient:

$$L = \sum_{i=1}^{n} (y_i - \theta)^2$$

$$\frac{\partial L}{\partial \theta} = -2 \sum_{i=1}^{n} (y_i - \theta)$$

...and then write a few iterations:

$$\Rightarrow \theta_1 = 0$$

$$\theta_2 = \theta_1 - \alpha\left(-2 \sum_{i=1}^{n} (y_i - \theta_1)\right)$$

$$\vdots$$

$$\theta_{t+1} = \theta_t - \alpha\left(-2 \sum_{i=1}^{n} (y_i - \theta_t)\right)$$

Stop when $|\theta_{t+1} - \theta_t| < \text{tol}$, where "tol" is a small tolerance parameter.

Gradient descent, in code

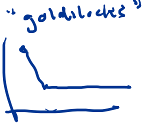
```
def minimize(loss_fn, grad_loss_fn, dataset, alpha=0.2, progress=True):
    '''
    Uses gradient descent to minimize loss_fn. Returns the minimizing value of
    theta_hat once theta_hat changes less than 0.001 between iterations.
    '''
    theta = 0
    while True:
        if progress:
            print(f'theta: {theta:.2f} | loss: {loss_fn(theta, dataset):.2f}')
        gradient = grad_loss_fn(theta, dataset)
        new_theta = theta - alpha * gradient

        if abs(new_theta - theta) < 0.001:
            return new_theta

        theta = new_theta
```

https://www.textbook.ds100.org/ch/11/gradient_descent_define.html

Gradient descent – what does the learning rate do?



Get in small groups and play with this Google tool: https://goo.gl/JNPhUv.

Set α to a higher value than the default – it'll take forever at $\alpha = 0.01$.

Questions to answer together: How does the rate change on each iteration...

1. ...when the learning rate is really small?
2. ...when the learning rate is really big?

Gradient descent – what does the learning rate do?

Get in small groups and play with this Google tool: https://goo.gl/JNPhUv.

Set α to a higher value than the default – it'll take forever at $\alpha = 0.01$.

Questions to answer together: How does the rate change on each iteration...

1. ...when the learning rate is really small?
2. ...when the learning rate is really big?

There are four qualitatively different behaviors:

1. Monotonically decreasing loss
2. One step to optimal parameter
3. Loss declines in periodic oscillations
4. Loss grows out of control

What do you think the point of a “dynamic learning rate" might be?

What do you think the point of a “dynamic learning rate" might be?

Basic idea: Start with a big learning rate, then make it smaller and smaller as you approach the optimal value

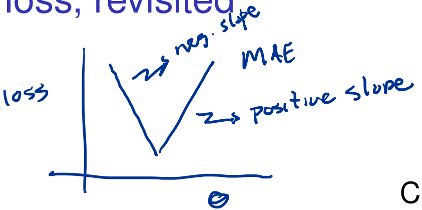
What do you think the point of a “dynamic learning rate" might be?

Basic idea: Start with a big learning rate, then make it smaller and smaller as you approach the optimal value

Advantages:

- cover a lot of ground when you’re far from the optimal value
- refined steps when you get close, so you don’t miss the optimal value.

Absolute deviation loss, revisited



$$L = \frac{1}{n}\sum_{i=1}^{n} |y_i - \theta|$$

$$= \frac{1}{n}\left(\sum_{y_i<\theta} |y_i - \theta| + \sum_{y_i=\theta} |y_i - \theta| + \sum_{y_i>\theta} |y_i - \theta|\right)$$

$$\frac{\partial L}{\partial \theta} = \frac{1}{n}\left(\sum_{y_i<\theta}(-1) + \sum_{y_i=\theta}(0) + \sum_{y_i>\theta}(1)\right)$$

Can you see why the optimal value is the median?

Absolute deviation loss, revisited

$$L = \frac{1}{n}\sum_{i=1}^{n} |y_i - \theta|$$

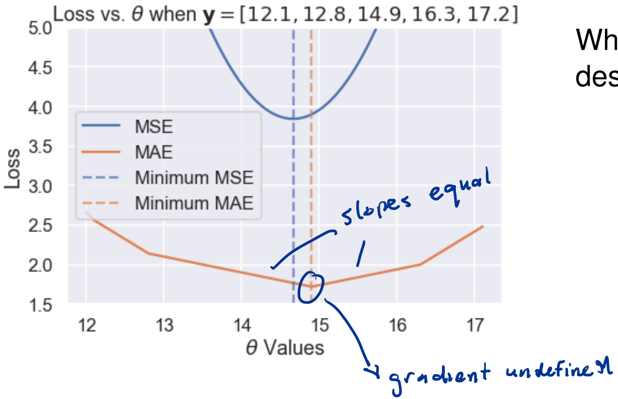
$$= \frac{1}{n}\left(\sum_{y_i<\theta} |y_i - \theta| + \sum_{y_i=\theta} |y_i - \theta| + \sum_{y_i>\theta} |y_i - \theta|\right)$$

$$\frac{\partial L}{\partial \theta} = \frac{1}{n}\left(\sum_{y_i<\theta}(-1) + \sum_{y_i=\theta}(0) + \sum_{y_i>\theta}(1)\right)$$

Can you see why the optimal value is the median?

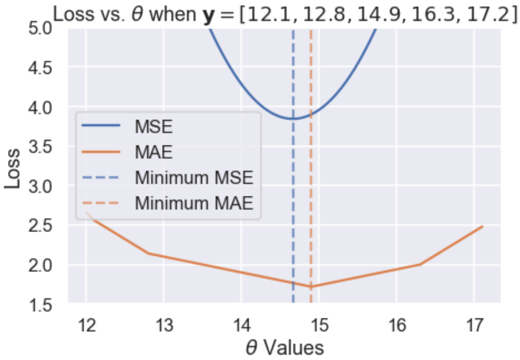
The right solution just “counts” the number of observations on each side of the optimal value

Gradient descent – absolute deviation loss, ctd.



What's the problem with doing gradient descent here?

Gradient descent – absolute deviation loss, ctd.

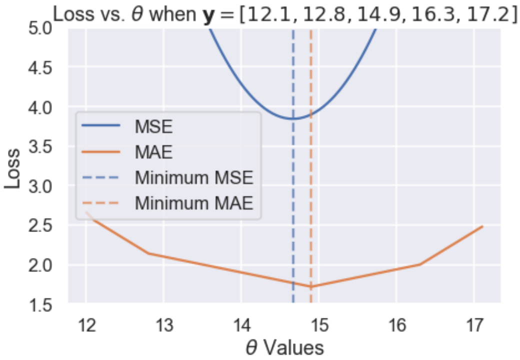


What's the problem with doing gradient descent here?

The derivative does not go to zero at the optimal value.

So once the solution is close, it won't converge, unless...

Gradient descent – absolute deviation loss, ctd.



What's the problem with doing gradient descent here?

The derivative does not go to zero at the optimal value.

So once the solution is close, it won't converge, unless...we use a dynamic learning rate.