Data, Environment and Society: Lecture 12: Gradient Descent

Instructor: Duncan Callaway
GSI: Salma Elmallah

October 8, 2019

Announcements

Today

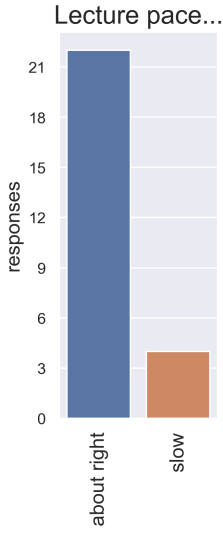
- Gradient descent, a.k.a. how to use computers to do better than the normal equations
- Discuss Novotny *et al*

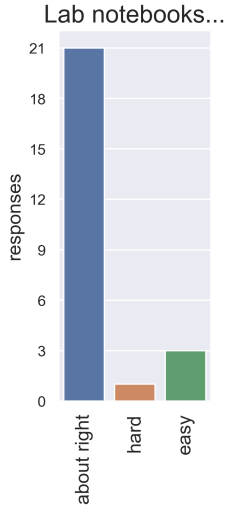
Next time

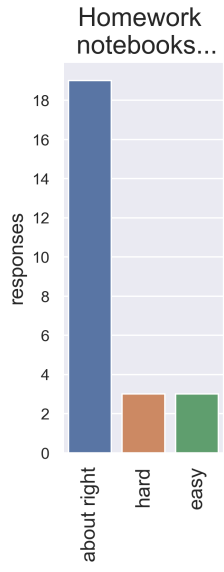
- Resampling, a.k.a. how to use computers to do better than AIC.
- Bring your laptop!

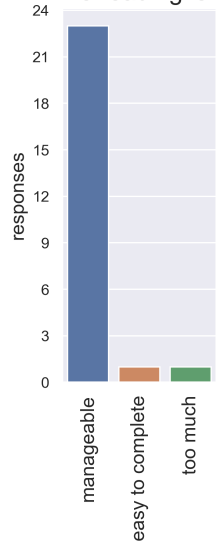
Reading

- Today: Novotny *et al*; Ch 11 DS100
- Next time: ISLR 5.1-5.2


Lecture pace...
responses
0
3
6
9
12
15
18
21
about right
slow


Lab notebooks...
responses
0
3
6
9
12
15
18
21
about right
hard
easy


Homework notebooks...
responses
0
2
4
6
8
10
12
14
16
18
about right
hard
easy


The reading is...
responses
0
3
6
9
12
15
18
21
24
manageable
easy to complete
too much

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 “constant” model

On the next few slides we’re be talking about the so-called constant model.

Consider a basic linear model. We’re familiar with this one:

$$\hat{y}_i = \theta + \beta x_i$$

The constant model just forces $\beta = 0$:

$$\hat{y}_i = \theta$$

In words: The model is that all observations are actually the same value.

Of course in a lot of cases this will be a terrible model, but it has interesting properties, as we’ll see.

The “constant” model

On the next few slides we’re be talking about the so-called constant model.

Consider a basic linear model. We’re familiar with this one:

$$\hat{y}_i = \theta + \beta x_i$$

The constant model just forces $\beta = 0$:

$$\hat{y}_i = \theta$$

In words: The model is that all observations are actually the same value.

Of course in a lot of cases this will be a terrible model, but it has interesting properties, as we’ll see.

The loss function

Mean squared error, aka the 'L2' norm

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

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

Mean absolute error, aka the 'L1' norm

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

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

The loss function

Mean squared error, aka the 'L2' norm

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

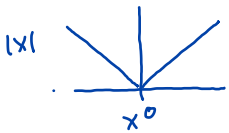
$$\text{Constant model} \rightarrow \text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \theta)^2$$

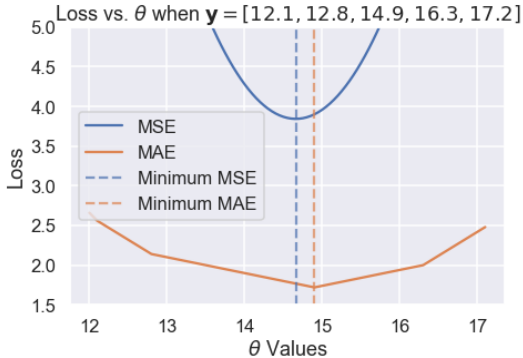
Mean absolute error, aka the 'L1' norm

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|$$

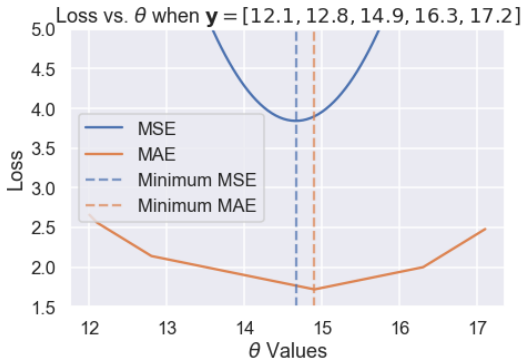
$$\text{Constant model} \rightarrow \text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \theta|$$

Advantages and disadvantages to MAE and MSE?





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?

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

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

Mean absolute error, solution

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

$$\frac{dL}{d\theta} = \frac{1}{n}\sum_{i=1}^{n} 2(y_i - \theta)(-1) = \frac{1}{n}\sum_{i=1}^{n} 2y_i(-1) - \frac{2}{n}\sum\theta(-1)$$

$$\frac{dL}{d\theta} = 0 \Rightarrow -\frac{2}{n}\sum y_i = -\frac{2}{n}\sum\theta^*$$

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

$\Rightarrow$ that's the mean!

$\rightarrow$ It's the mean!

Mean absolute error, solution

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

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

$$\frac{\partial L}{\partial \theta} = 0 \rightarrow 2\theta = \frac{2}{n}\left(\sum_{i=1}^{n} y_i\right)$$

$$\rightarrow \theta = \frac{1}{n}\left(\sum_{i=1}^{n} y_i\right) = \bar{y}$$

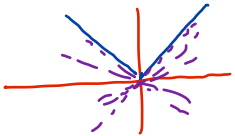
$\rightarrow$ It's the mean!

Absolute deviation loss, solution

(MAE)

$$
\begin{aligned}
L &= \frac{1}{n}\sum_{i=1}^{n}|y_i-\theta| \\
&= \frac{1}{n}\left(\sum_{y_i<\theta}|y_i-\theta| + \sum_{y_i=\theta}|y_i-\theta| + \sum_{y_i>\theta}|y_i-\theta|\right) \\
\frac{\partial L}{\partial \theta} &= \frac{1}{n}\left(\sum_{y_i<\theta}(-1) + \sum_{y_i=\theta}(a) + \sum_{y_i>\theta}(1)\right)
\end{aligned}
$$

Absolute deviation loss, solution



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

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

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

Can you see that the optimal value is the median?

Though the derivative of $|\cdot|$ is not defined at zero, we know it's bounded by -1 and 1. So here, $-1 < a < 1$.

Absolute deviation loss, solution

$$
\begin{aligned}
L &= \frac{1}{n}\sum_{i=1}^{n}|y_i-\theta| \\
&= \frac{1}{n}\left(\sum_{y_i<\theta}|y_i-\theta| + \sum_{y_i=\theta}|y_i-\theta| + \sum_{y_i>\theta}|y_i-\theta|\right) \\
\frac{\partial L}{\partial \theta} &= \frac{1}{n}\left(\sum_{y_i<\theta}(-1) + \sum_{y_i=\theta}(a) + \sum_{y_i>\theta}(1)\right)
\end{aligned}
$$

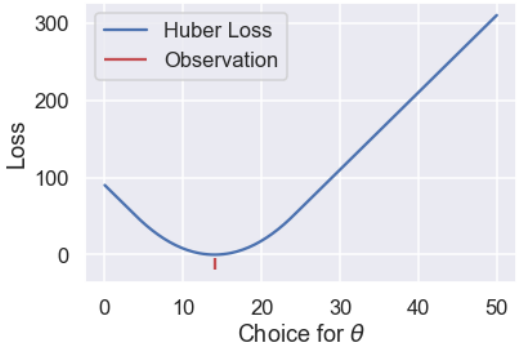
Can you see that the optimal value is the median?

Though the derivative of $|\cdot|$ is not defined at zero, we know it's bounded by -1 and 1. So here, $-1 < a < 1$.

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

In this case we can find the solution without explicitly setting the derivative equal to zero.

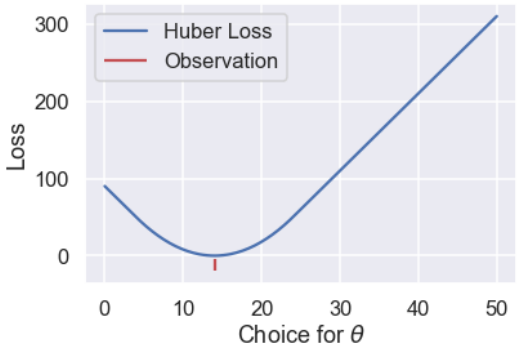
Huber loss



What does this buy us?

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

Huber loss

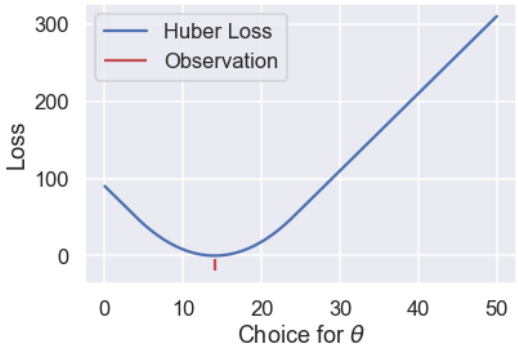


What does this buy us?

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

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

Huber loss



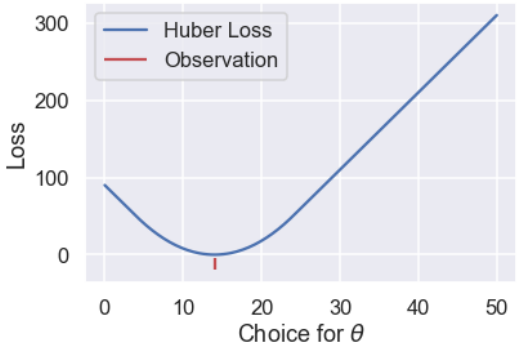
What does this buy us?

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

What does this cost us?

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

Huber loss



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

What does this buy us?

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

What does this cost us?

- There is no “closed form” solution (e.g. no equivalent to normal equations).

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:

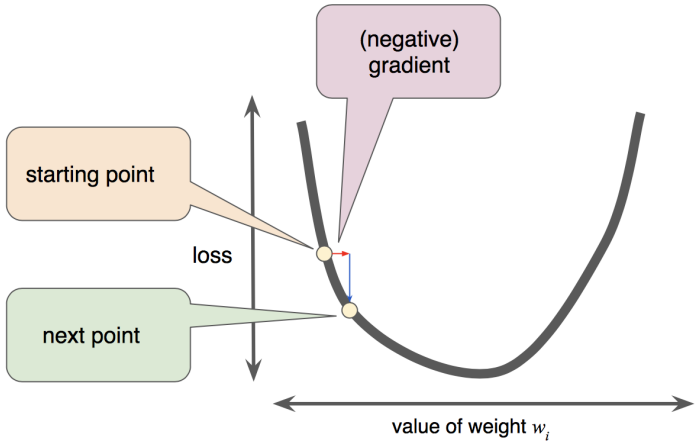
A separate issue: In situations where the normal equations (or something like them) can be used to solve for parameters:

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

It can be very difficult computationally to invert a large $X^T X$ (On my computer, Python can't deal with with 50,000 by 50,000).

Gradient descent – sketch

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

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:

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

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

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:

$\theta_1 = 0$

$\theta_2 = \theta_1 - \alpha \left. \frac{\partial L}{\partial \theta} \right|_{\theta_1}$

$= \theta_1 - \alpha \left(-2 \sum (y_i - \theta_1) \right)$

$\theta_3 = \theta_2 - \alpha \left. \frac{\partial L}{\partial \theta} \right|_{\theta_2}$

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:

$$\Rightarrow \theta_1 = 0$$

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

$$\vdots$$

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

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

Gradient descent, in code

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

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

        theta = new_theta
```

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 $\alpha = 0.01$.

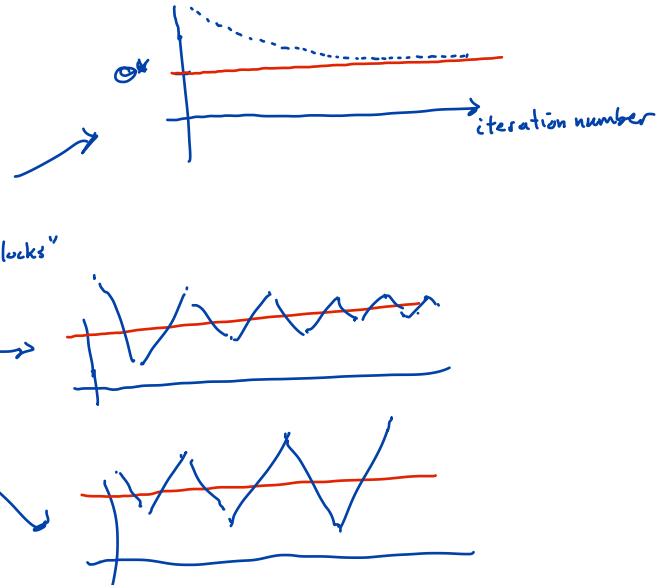
Questions to answer together: How does the rate change on each iteration...

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

Types of gradient descent trajectories

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

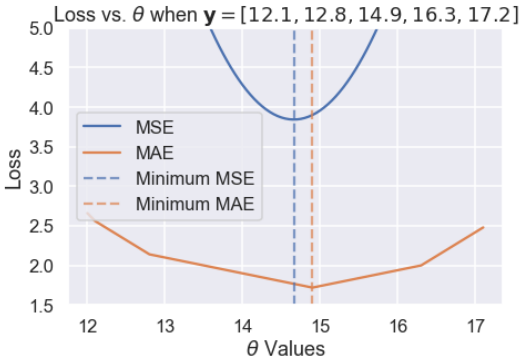
What do you think the point of a “dynamic learning rate" might be?

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

Advantages:

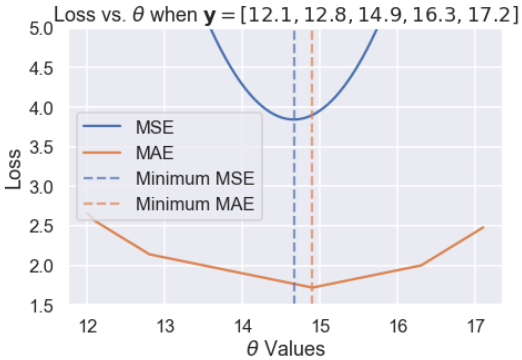
- cover a lot of ground when you’re far from the optimal value
- refined steps when you get close, so you don’t miss the optimal value.

Gradient descent – absolute deviation loss, ctd.



What's the problem with doing gradient descent here?

Gradient descent – absolute deviation loss, ctd.

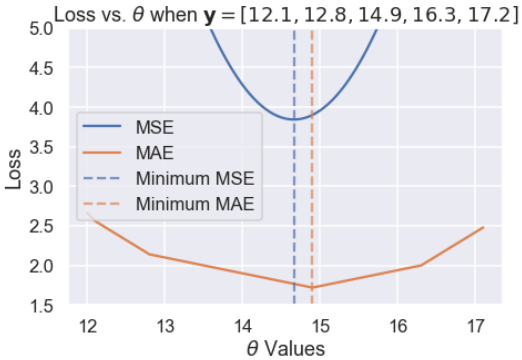


What's the problem with doing gradient descent here?

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.

Novotny Questions

1. Describe the basic model selection process and how it differs from AIC (which we learned briefly in class on Thursday). This question can be answered if you read Section 2.2 and 2.3 carefully and do a little background research to understand unfamiliar terms.
2. The authors compare R2 values for the test data versus R2 for the training data. What do they obeserve? What does this imply about their data and model? Read Section 3.3 to answer this question.
3. Also in section 3.3 the authors state "We also investigated the extent to which monitor locations span the (independent) variable space, an important issue for any LUR". What does this mean? Why is spanning the variable space important?
4. The authors discuss several limitations in the Discussion section. Review these. How important are their limitations? Are there others you think are worth considering?