An Introduction to Stochastic Gradient Descent

RMDS Workshop 5

Jacob Munson

jacobmunson1@montana.edu

October 7, 2020

Overview

Optimization

Gradient Descent Introduction

Error Terms

Gradient Descent

Optimization in a slide

- Optimization is used in essentially all computational field
- There are a *lot* of optimization algorithms
- We want high/low values in spaces and the associated parameters
- Sometimes closed-form solutions exist, such as Ordinary Least Squares (OLS) Regression
    - Often, we have to approximate a solution

Some intuition

- Recall: $y = m \times x + b$
- Assume we don't have OLS solution
- Want (m, b) that minimize Sums of Squared Residuals (SSR)
- How do we find (m, b)?
- Brute force
  - All $m \in \{-1M, ..., 1M\}$
  - All $b \in \{-1M, ..., 1M\}$
- "Shotgun" approach
- $m \in \{-1M, -100k, 250k, 1M\}$
- $b \in \{-50k, -25k, 0, 50k, 200k\}$
- Surely we can do something more intelligent

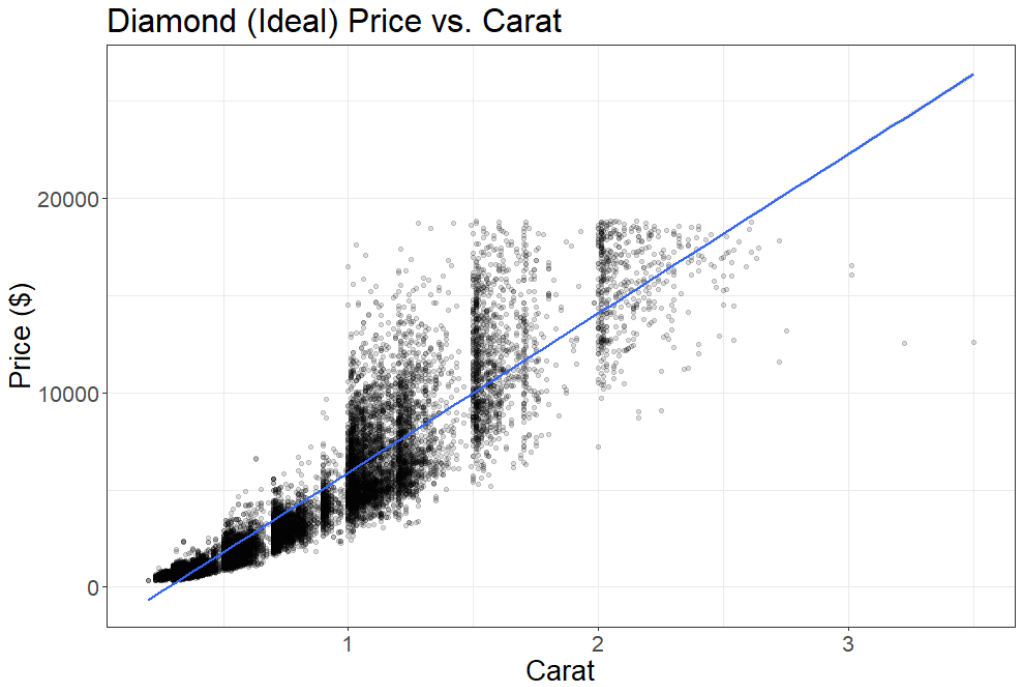

Figure 2: Diamond dataset

Gradient Descent

- Take a guess at parameter
- Define *error*
- Move down gradient of error (decrease error)
- Rinse and repeat
- Until
  - fixed number of iterations
  - step size is very small
  - Train error is small

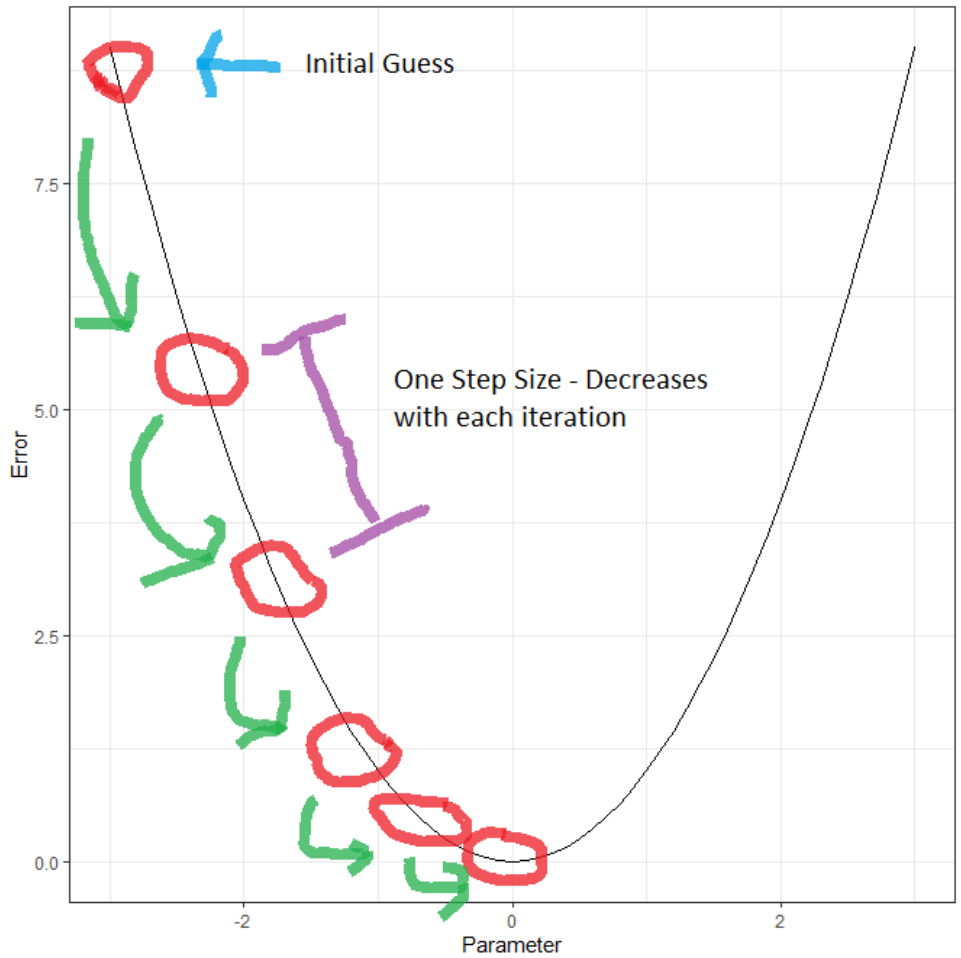

Figure 3: Dataset Snippet

Gradient Descent for Determining Intercept

	carat	price
	<dbl>	<int>
1	0.36	1215
2	0.33	814
3	0.79	2944

Figure 4: Dataset

- We have data in Fig. 4
- Want to fit...
  - $y = mx + b$
  - $\text{price} = m \times \text{carat} + b$
- Fig. 5 gives fit
- So how would we find that?
- Let's assume we know slope m
- Want to find intercept term b

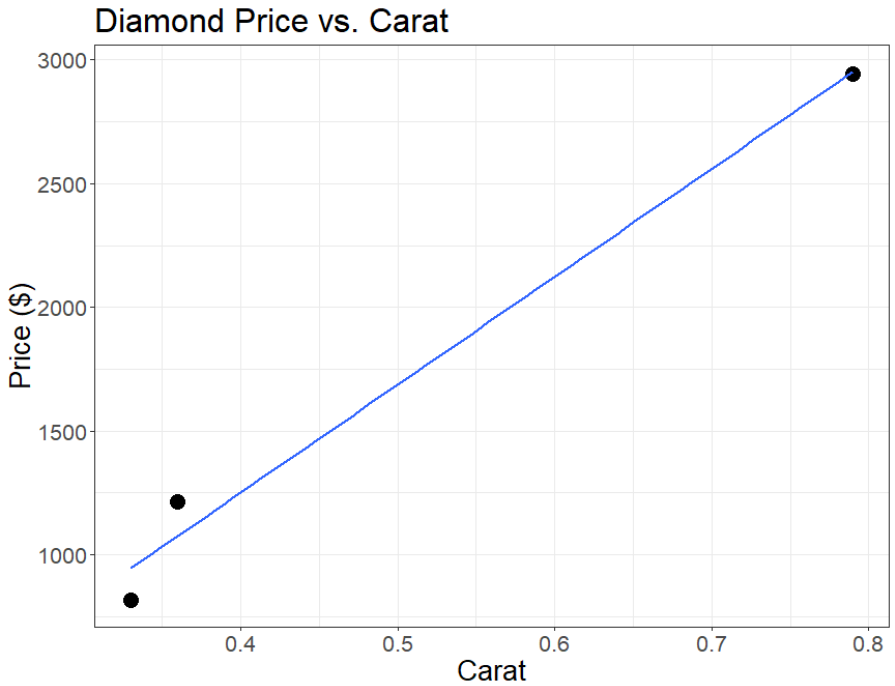

Figure 5: Regression on data

Gradient Descent for Determining Intercept

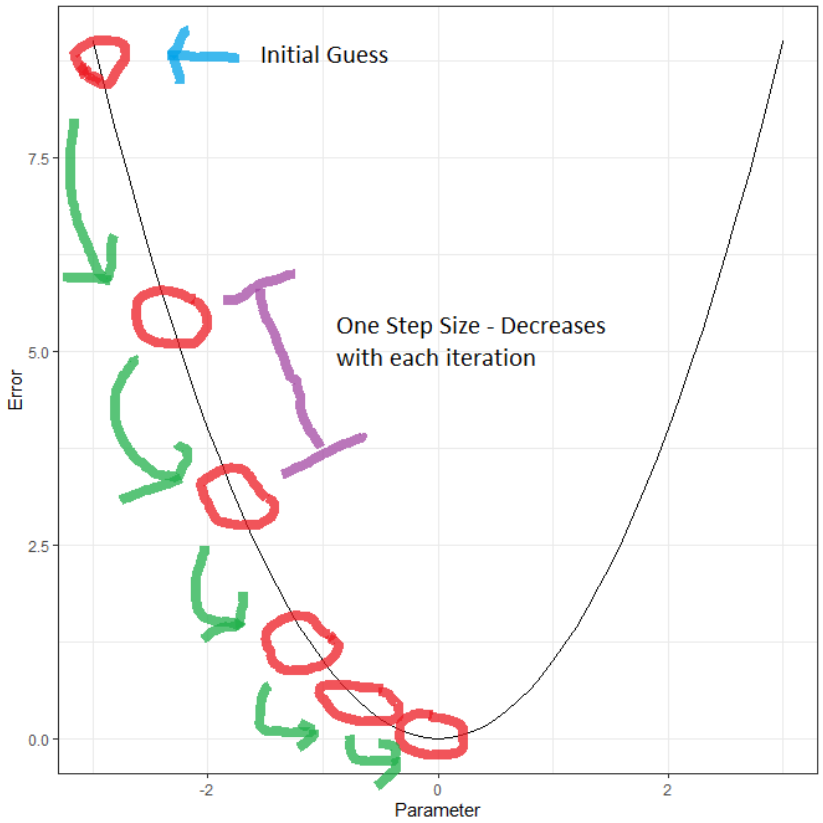

Figure 6: Rolling down the gradient

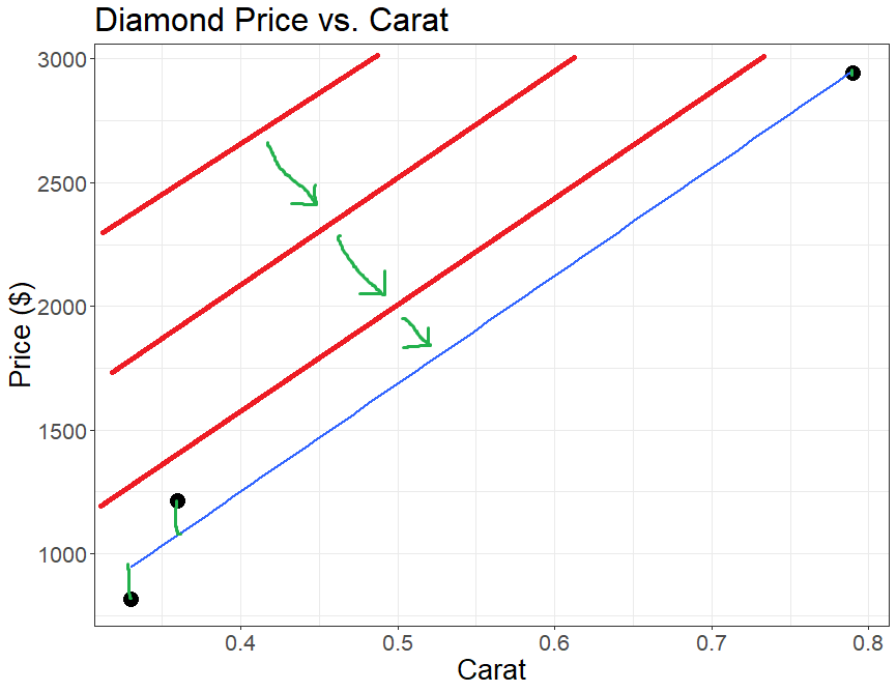

Figure 7: Various intercepts

Derivation of Error

- $\text{error}_i = \text{price} - \hat{\text{price}}_i$
- $\text{SSR} = (\text{price}_1 - \hat{\text{price}}_1)^2 + (\text{price}_2 - \hat{\text{price}}_2)^2 + (\text{price}_3 - \hat{\text{price}}_3)^2$
  - Where $\hat{\text{price}}_i = \text{b} + m \times \text{carat}_i$
  - carat_i and m are fixed values
  - b gets updated with each iteration
  - $\hat{\text{price}}_i$ changes with each iteration
- $\text{SSR} = (\text{price}_1 - (\text{b} + m \times \text{carat}_1)^2$
  $+ (\text{price}_2 - (\text{b} + m \times \text{carat}_2)^2$
  $+ (\text{price}_3 - (\text{b} + m \times \text{carat}_3)^2$

Derivation of Error, Continued...

- Want: $\frac{\partial \text{SSR}}{\partial b}$
- $$\frac{\partial \text{SSR}}{\partial b} = \frac{\partial((\text{price}_1 - (\text{b} + m \times \text{carat}_1)^2)}{\partial b} + \frac{\partial((\text{price}_2 - (\text{b} + m \times \text{carat}_2)^2)}{\partial b} + \frac{\partial((\text{price}_3 - (\text{b} + m \times \text{carat}_3)^2)}{\partial b}$$
- $$\frac{\partial \text{SSR}}{\partial b} = -2(\text{price}_1 - (\text{b} + m \times \text{carat}_1)) + -2(\text{price}_2 - (\text{b} + m \times \text{carat}_2)) + -2(\text{price}_3 - (\text{b} + m \times \text{carat}_3))$$
- Update rule: $\text{new-intercept} = \text{old-intercept} - \text{step-size}$
  - Where $\text{step-size} = \frac{\partial \text{SSR}}{\partial b} \times \alpha$; α is learning rate
  - Plain language: "updated intercept is the old intercept adjusted in the direction of rolling down the gradient"

Gradient Descent

- Two step process
  - Step 1: Compute $\frac{\partial \text{SSR}}{\partial b}$
  - Step 2: Update intercept with step size found in (1)
  - This is one iteration
  - For more iterations...
  - Step 1 (again): Compute $\frac{\partial \text{SSR}}{\partial b}$ with new intercept found in (Step 2)
  - Step 2 (again): Update intercept with step size found in (Step 1 (again))
  - Repeat until satisfied

Gradient Descent on Our Data

- OLS estimates in Fig. 14
  - *That* is our groundtruth
- Assume we know $m = 4366.6$
- So let's look at some results!

```
Call:
lm(formula = df_sample$price ~ df_sample$carat)

Coefficients:
    (Intercept)  df_sample$carat
         -496.5           4366.6
```

Figure 8: OLS Parameter Estimates

Gradient Descent Results

- Let $\alpha = 0.001$, $b = -1000$, and run for 150 iterations

```
[1] "Iteration =  136 | Intercept =  -718.62  | SSR =  186327.92  | Step size =  -1.34"
[1] "Iteration =  137 | Intercept =  -717.29  | SSR =  184536.27  | Step size =  -1.33"
[1] "Iteration =  138 | Intercept =  -715.96  | SSR =  182766.06  | Step size =  -1.32"
[1] "Iteration =  139 | Intercept =  -714.65  | SSR =  181017.02  | Step size =  -1.32"
[1] "Iteration =  140 | Intercept =  -713.34  | SSR =  179288.91  | Step size =  -1.31"
[1] "Iteration =  141 | Intercept =  -712.04  | SSR =  177581.48  | Step size =  -1.3"
[1] "Iteration =  142 | Intercept =  -710.75  | SSR =  175894.47  | Step size =  -1.29"
[1] "Iteration =  143 | Intercept =  -709.46  | SSR =  174227.65  | Step size =  -1.29"
[1] "Iteration =  144 | Intercept =  -708.18  | SSR =  172580.76  | Step size =  -1.28"
[1] "Iteration =  145 | Intercept =  -706.91  | SSR =  170953.59  | Step size =  -1.27"
[1] "Iteration =  146 | Intercept =  -705.65  | SSR =  169345.87  | Step size =  -1.26"
[1] "Iteration =  147 | Intercept =  -704.4  | SSR =  167757.4  | Step size =  -1.25"
[1] "Iteration =  148 | Intercept =  -703.15  | SSR =  166187.93  | Step size =  -1.25"
[1] "Iteration =  149 | Intercept =  -701.91  | SSR =  164637.23  | Step size =  -1.24"
[1] "Iteration =  150 | Intercept =  -700.68  | SSR =  163105.09  | Step size =  -1.23"
> print(paste0("Estimated intercept = ", round(last(b),2), " | Actual intercept (of sub
sample$coefficients[1],2)))
[1] "Estimated intercept = -700.68 | Actual intercept (of subset) = -496.54"
```

Figure 9: OLS Parameter Estimates

Gradient Descent Results

- Let $\alpha = 0.001$, $b = -1000$, and run for 250 iterations

```
[1] "Iteration =  236 | Intercept =  -618.2  | SSR =  81516.64  | Step size =  -0.73"
[1] "Iteration =  237 | Intercept =  -617.47  | SSR =  80978.95  | Step size =  -0.73"
[1] "Iteration =  238 | Intercept =  -616.74  | SSR =  80447.7  | Step size =  -0.73"
[1] "Iteration =  239 | Intercept =  -616.02  | SSR =  79922.8  | Step size =  -0.72"
[1] "Iteration =  240 | Intercept =  -615.31  | SSR =  79404.18  | Step size =  -0.72"
[1] "Iteration =  241 | Intercept =  -614.59  | SSR =  78891.77  | Step size =  -0.71"
[1] "Iteration =  242 | Intercept =  -613.89  | SSR =  78385.48  | Step size =  -0.71"
[1] "Iteration =  243 | Intercept =  -613.18  | SSR =  77885.26  | Step size =  -0.7"
[1] "Iteration =  244 | Intercept =  -612.48  | SSR =  77391.02  | Step size =  -0.7"
[1] "Iteration =  245 | Intercept =  -611.79  | SSR =  76902.69  | Step size =  -0.7"
[1] "Iteration =  246 | Intercept =  -611.09  | SSR =  76420.2  | Step size =  -0.69"
[1] "Iteration =  247 | Intercept =  -610.41  | SSR =  75943.49  | Step size =  -0.69"
[1] "Iteration =  248 | Intercept =  -609.72  | SSR =  75472.48  | Step size =  -0.68"
[1] "Iteration =  249 | Intercept =  -609.04  | SSR =  75007.1  | Step size =  -0.68"
[1] "Iteration =  250 | Intercept =  -608.37  | SSR =  74547.29  | Step size =  -0.68"
> print(paste0("Estimated intercept = ", round(last(b),2), " | Actual intercept (of sub
sample$coefficients[1],2)))
[1] "Estimated intercept = -608.37 | Actual intercept (of subset) = -496.54"
```

Figure 10: OLS Parameter Estimates

Gradient Descent Results

- Let $\alpha = 0.01$, $b = -1000$, and run for 150 iterations

```
[1] "Iteration =  136 | Intercept =  -496.65  | SSR =  36574.56  | Step size =  -0.01"
[1] "Iteration =  137 | Intercept =  -496.64  | SSR =  36574.56  | Step size =  -0.01"
[1] "Iteration =  138 | Intercept =  -496.64  | SSR =  36574.55  | Step size =  -0.01"
[1] "Iteration =  139 | Intercept =  -496.63  | SSR =  36574.55  | Step size =  -0.01"
[1] "Iteration =  140 | Intercept =  -496.63  | SSR =  36574.54  | Step size =  -0.01"
[1] "Iteration =  141 | Intercept =  -496.62  | SSR =  36574.54  | Step size =  -0.01"
[1] "Iteration =  142 | Intercept =  -496.62  | SSR =  36574.54  | Step size =  0"
[1] "Iteration =  143 | Intercept =  -496.61  | SSR =  36574.54  | Step size =  0"
[1] "Iteration =  144 | Intercept =  -496.61  | SSR =  36574.53  | Step size =  0"
[1] "Iteration =  145 | Intercept =  -496.6   | SSR =  36574.53  | Step size =  0"
[1] "Iteration =  146 | Intercept =  -496.6   | SSR =  36574.53  | Step size =  0"
[1] "Iteration =  147 | Intercept =  -496.6   | SSR =  36574.53  | Step size =  0"
[1] "Iteration =  148 | Intercept =  -496.59  | SSR =  36574.53  | Step size =  0"
[1] "Iteration =  149 | Intercept =  -496.59  | SSR =  36574.53  | Step size =  0"
[1] "Iteration =  150 | Intercept =  -496.59  | SSR =  36574.53  | Step size =  0"
> print(paste0("Estimated intercept = ", round(last(b),2), " | Actual intercept (of sub
sample$coefficients[1],2)))
[1] "Estimated intercept = -496.59 | Actual intercept (of subset) = -496.54"
```

Figure 11: OLS Parameter Estimates

Gradient Descent Results

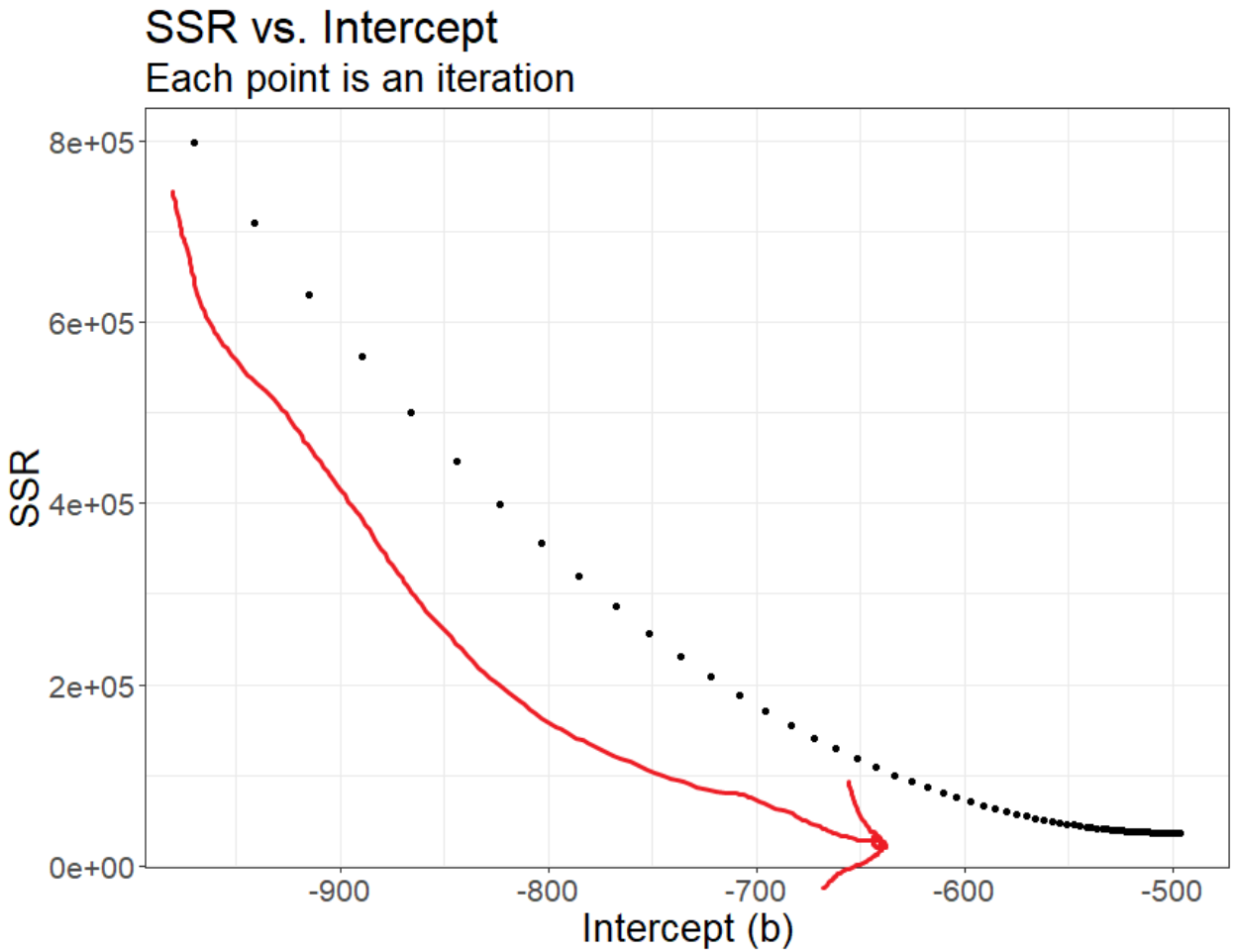

Figure 12: OLS Parameter Estimates

But, what's stochastic?

- So far, we've had a dataset with 3 values
- We've done Gradient Descent
- Pretend we have 1 million data points
- Processing is expensive
  - Especially if there are many parameters
- We can randomly sample a subset, say 25%, and perform Gradient Descent on the subset
- *Stochastic* Gradient Descent is just Gradient Descent on a subset of your data
- We've been doing Stochastic Gradient Descent all along since our 3 points are randomly sampled from a larger set
- The larger the sample, the closer to the "actual" parameters we get

Let's just do the whole dataset

- Let $\alpha = 0.001$, $b = -1000$, and run for 150 iterations
- α seems to be a problem

```
[1] "Iteration =  82 | Intercept =  2.02717750950434e+136  | SSR =  4.99626576544772e+273  | Step size =  -2.07
[1] "Iteration =  83 | Intercept =  -8.53482275051517e+137  | SSR =  8.85627279647717e+276  | Step size =  8.73
[1] "Iteration =  84 | Intercept =  3.5933310744219e+139  | SSR =  1.56984378989681e+280  | Step size =  -3.678
[1] "Iteration =  85 | Intercept =  -1.51286424895311e+141  | SSR =  2.7826711996246e+283  | Step size =  1.548
[1] "Iteration =  86 | Intercept =  6.36946106094237e+142  | SSR =  4.93250287388734e+286  | Step size =  -6.52
[1] "Iteration =  87 | Intercept =  -2.68167049587796e+144  | SSR =  8.74324807192063e+289  | Step size =  2.74
[1] "Iteration =  88 | Intercept =  1.12903691217454e+146  | SSR =  1.54980927131012e+293  | Step size =  -1.15
[1] "Iteration =  89 | Intercept =  -4.75347120763724e+147  | SSR =  2.74715844464329e+296  | Step size =  4.86
[1] "Iteration =  90 | Intercept =  2.00130644783943e+149  | SSR =  4.86955373134092e+299  | Step size =  -2.04
[1] "Iteration =  91 | Intercept =  -8.42590040669356e+150  | SSR =  8.63166578129253e+302  | Step size =  8.62
[1] "Iteration =  92 | Intercept =  3.54747258922612e+152  | SSR =  1.53003043544649e+306  | Step size =  -3.63
[1] "Iteration =  93 | Intercept =  -1.49355690951598e+154  | SSR =  Inf  | Step size =  1.52903163540824e+154"
[1] "Iteration =  94 | Intercept =  6.28817330044419e+155  | SSR =  Inf  | Step size =  -6.43752899139579e+155"
```

Figure 13: OLS Parameter Estimates

Let's try again

- Let $\alpha = 0.00001$, $b = -1000$, and run for 150 iterations
- Much better

```
[1] "Iteration =  133 | Intercept =  -2300.37  | SSR =  41542684938.25  | Step size =  0"
[1] "Iteration =  134 | Intercept =  -2300.37  | SSR =  41542684938.25  | Step size =  0"
[1] "Iteration =  135 | Intercept =  -2300.37  | SSR =  41542684938.25  | Step size =  0"
[1] "Iteration =  136 | Intercept =  -2300.37  | SSR =  41542684938.25  | Step size =  0"
[1] "Iteration =  137 | Intercept =  -2300.37  | SSR =  41542684938.25  | Step size =  0"
[1] "Iteration =  138 | Intercept =  -2300.37  | SSR =  41542684938.25  | Step size =  0"
[1] "Iteration =  139 | Intercept =  -2300.37  | SSR =  41542684938.25  | Step size =  0"
[1] "Iteration =  140 | Intercept =  -2300.37  | SSR =  41542684938.25  | Step size =  0"
[1] "Iteration =  141 | Intercept =  -2300.37  | SSR =  41542684938.25  | Step size =  0"
[1] "Iteration =  142 | Intercept =  -2300.37  | SSR =  41542684938.25  | Step size =  0"
[1] "Iteration =  143 | Intercept =  -2300.37  | SSR =  41542684938.25  | Step size =  0"
[1] "Iteration =  144 | Intercept =  -2300.37  | SSR =  41542684938.25  | Step size =  0"
[1] "Iteration =  145 | Intercept =  -2300.37  | SSR =  41542684938.25  | Step size =  0"
[1] "Iteration =  146 | Intercept =  -2300.37  | SSR =  41542684938.25  | Step size =  0"
[1] "Iteration =  147 | Intercept =  -2300.37  | SSR =  41542684938.25  | Step size =  0"
[1] "Iteration =  148 | Intercept =  -2300.37  | SSR =  41542684938.25  | Step size =  0"
[1] "Iteration =  149 | Intercept =  -2300.37  | SSR =  41542684938.25  | Step size =  0"
[1] "Iteration =  150 | Intercept =  -2300.37  | SSR =  41542684938.25  | Step size =  0"
> print(paste0("Estimated intercept = ", round(last(b),2), " | Actual intercept (of subset)
[1] "Estimated intercept = -2300.37 | Actual intercept (of subset) = -2300.37"
```

Figure 14: OLS Parameter Estimates

Takeaways

- We could repeat this work but estimate both slope and intercept parameters (only intercept was estimate here, for ease of introduction)
- SGD is *great* if we can describe the errors and differentiate them
- There are some hyperparameters to tune
- The surface may not be nice to play with