Prediction and Linear Regression

Big Data y Machine Learning para Economía Aplicada

Ignacio Sarmiento-Barbieri

Universidad de los Andes

Agenda

1. Machine learning is all about prediction
2. Prediction vs Causality
3. Getting serious about prediction
   - The basic logic of prediction
   - Prediction error and its components
4. Prediction and linear regression
   - Traditional Computation
   - Gradient Descent
5. Review

Agenda

Machine learning is all about prediction

- Machine learning is a branch of computer science and statistics, tasked with developing algorithms to predict outcomes y from observable variables x.
- The learning part comes from the fact that we don't specify how exactly the computer should predict y from x. This is left as an empirical problem that the computer can "learn".
- In general, this means that we abstract from the underlying model, the approach is pragmatic

Machine learning is all about prediction

- Machine learning is a branch of computer science and statistics, tasked with developing algorithms to predict outcomes y from observable variables x.
- The learning part comes from the fact that we don't specify how exactly the computer should predict y from x. This is left as an empirical problem that the computer can "learn".
- In general, this means that we abstract from the underlying model, the approach is pragmatic

"Whatever works, works...."

“Whatever works, works....”

“Whatever works, works....”????

- In many applications, ML techniques can be successfully applied by data scientists with little knowledge of the problem domain.
- For example, the company Kaggle hosts prediction competitions (`www.kaggle.com/competitions`) in which a sponsor provides a data set, and contestants around the world can submit entries, often predicting successfully despite limited context about the problem.

“Whatever works, works....”????

- However, much less attention has been paid to the limitations of pure prediction methods.
- When ML applications are used “off the shelf” without understanding the underlying assumptions or ensuring that conditions like stability are met, then the validity and usefulness of the conclusions can be compromised.
- A deeper question concerns whether a given problem can be solved using only techniques for prediction, or whether statistical approaches to estimating the causal effect of an intervention are required.

Agenda

Policy Prediction Problems

- Empirical policy research often focuses on causal inference.
- Since policy choices seem to depend on understanding the counterfactual— what happens with and without a policy—this tight link of causality and policy seems natural.
- While this link holds in many cases, there are also many policy applications where causal inference is not central, or even necessary.

The Causal Paradigm

$$y = f(X) + u \tag{1}$$

- Interest lies on inference
- "Correct" $f()$ to understand how y is affected by X
- Model: Theory, experiment
- Hypothesis testing (std. err., tests)

The Predictive Paradigm

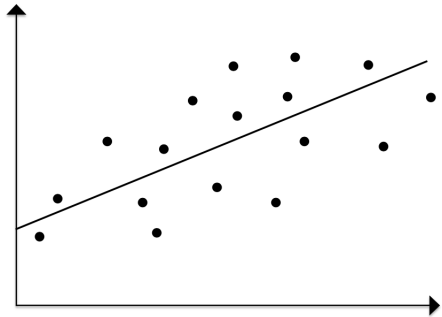
$$y = f(X) + u \tag{2}$$

- Interest on predicting y
- "Correct" $f()$ to be able to predict (no inference!)
- Model? We treat $f()$ as a black box, and any approximation $\hat{f}()$ that yields a good prediction is good enough (*Whatever works, works.*).

Prediction vs. Causality: Target

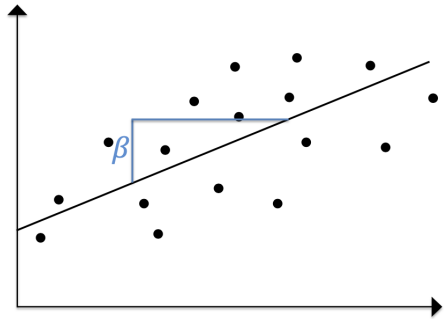
$$y = f(x) + \epsilon \tag{3}$$

$$y = \alpha + \beta x + \epsilon \tag{4}$$



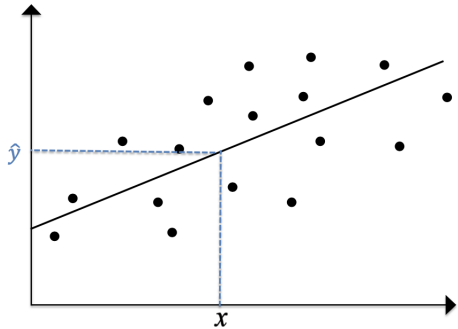
Prediction vs. Causality: Target

$$y = \alpha + \beta x + \epsilon \tag{5}$$



Prediction vs. Causality: Target

$$y = \underbrace{\alpha + \beta x}_{\hat{y}} + \epsilon \tag{6}$$



Prediction vs. Causality: The garden of the parallel paths?

- We've seen that prediction and causality
  - Answer different questions
  - Serve different purposes
  - Seek different targets
- Different strokes for different folks, or complementary tools in an applied economist's toolkit?

Agenda

1. Machine learning is all about prediction
2. Prediction vs Causality
3. Getting serious about prediction
   - The basic logic of prediction
   - Prediction error and its components
4. Prediction and linear regression
   - Traditional Computation
   - Gradient Descent
5. Review

Agenda

1. Machine learning is all about prediction
2. Prediction vs Causality
3. Getting serious about prediction
   - The basic logic of prediction
   - Prediction error and its components
4. Prediction and linear regression
   - Traditional Computation
   - Gradient Descent
5. Review

The basic logic of prediction

- We have data $\{y_i, X_i\}$

The basic logic of prediction

Mathematically

The basic logic of prediction

Example: the linear regression

photo from https://www.dailydot.com/parsec/batman-1966-labels-tumblr-twitter-vine/

Agenda

1. Machine learning is all about prediction
2. Prediction vs Causality
3. Getting serious about prediction
   - The basic logic of prediction
   - Prediction error and its components
4. Prediction and linear regression
   - Traditional Computation
   - Gradient Descent
5. Review

Prediction error

$$e_j = y_j - \hat{y}_j \tag{7}$$

Minimizing our losses

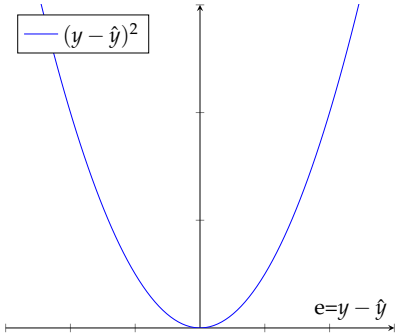
- When making a prediction we want to minimize the prediction errors

$$L(\hat{y}, y) \tag{8}$$

Minimizing our losses

- When making a prediction we want to minimize the prediction errors

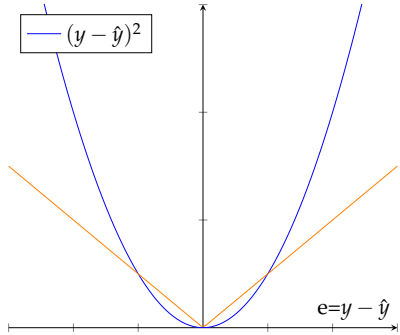
$$L(\hat{y}, y) \tag{8}$$



Minimizing our losses

- When making a prediction we want to minimize the prediction errors

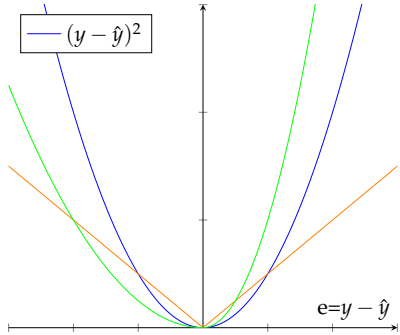
$$L(\hat{y}, y) \tag{9}$$



Minimizing our losses

- When making a prediction we want to minimize the prediction errors

$$L(\hat{y}, y) \tag{10}$$



Minimizing our losses

- A very common loss function in a regression setting is the squared loss $L(d) = d^2$
- When we have multiple observations, we agregate those losses

Minimizing our losses

- We are interested in the accuracy of the predictions on previously unseen data.
- Can we find the function f^* within a function class $\mathcal{F}$ that has a low expected prediction loss?

Minimizing our losses

- We are interested in the accuracy of the predictions on previously unseen data.
- Can we find the function f^* within a function class $\mathcal{F}$ that has a low expected prediction loss?
- By conditioning on X, it suffices to minimize the $EMSE(f)$ point wise

$$f(x) = argmin_{f^*} E_{Y|X}[(y - f^*)^2 | X = x) \tag{11}$$

Minimizing our losses

- f^* a random variable and we can treat f^* as a constant (predictor)

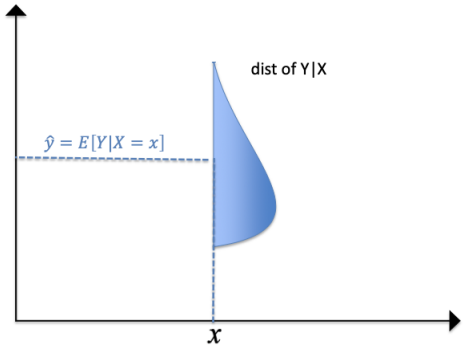
$$min_{f^*} E(y - f^*)^2 = \int (y - f^*)^2 f(y) dy \tag{12}$$

- **Result**: The best prediction of Y at any point $X = x$ is the conditional mean, when best is measured using a square error loss

Minimizing our losses

- **Result**: The best prediction of y at any point $X = x$ is the conditional mean, when best is measured using a square error loss

$$f^* = E[y|X = x] \tag{13}$$



Minimizing our losses

- Prediction problem solved if we knew $f^* = E[y|X = x]$

Minimizing our losses

- Prediction problem solved if we knew $f^* = E[y|X = x]$
- But we have to settle for an estimate: $\hat{f}(x)$
- The EMSE of this

$$E(y - \hat{y})^2 = E(f(X) + u - \hat{f}(X))^2 \tag{14}$$

Reducible and irreducible error

$$E(y - \hat{y})^2 = \underbrace{[f(X) - \hat{f}(X)]^2}_{Reducible} + \underbrace{Var(u)}_{Irreducible} \tag{15}$$

- The focus the is on techniques for estimating f with the aim of minimizing the reducible error
- It is important to keep in mind that the irreducible error will always provide a bound on the accuracy of our prediction for y
- This bound is almost always unknown in practice

Bias/Variance Decomposition

Recall that

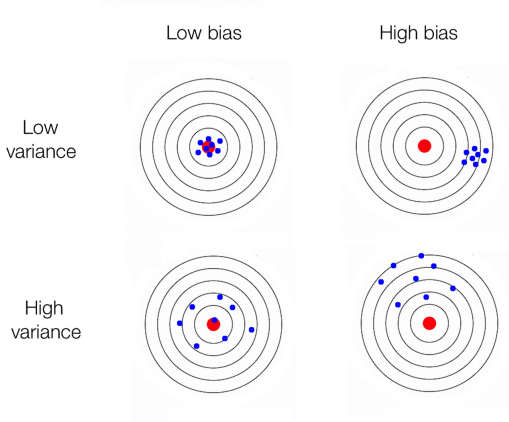
- $Bias(\hat{f}(X)) = E(\hat{f}(X)) - f = E(\hat{f}(X) - f(X))$
- $Var(\hat{f}) = E(\hat{f} - E(\hat{f}))^2$

Result (very important!)

$$EMSE = Bias^2(\hat{f}(X)) + V(\hat{f}(X)) + \underbrace{Var(u)}_{Irreducible} \tag{16}$$

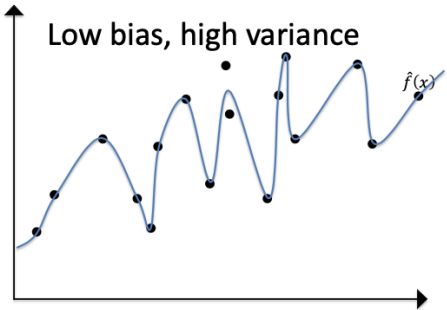
Proof: as an exercise

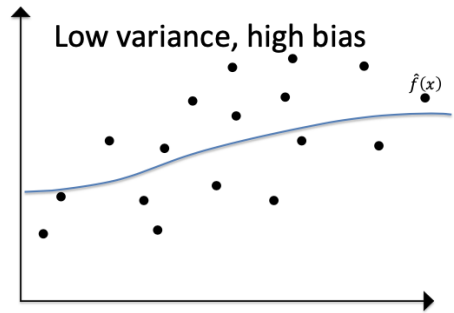
Bias/Variance Decomposition



Source: https://tinyurl.com/y4lvjxpc

Bias/Variance Decomposition





The Task of Finding the Best Model

Prediction and linear regression

- The goal is to predict y given another variables X.
- We assume that the link between y and X is given by the simple model:

$$y = f(X) + u \tag{17}$$

- we just learned that under a squared loss we need to approximate $E[y|X = x]$

Agenda

Prediction and linear regression

- As economists we know that we can approximate $E[y|X = x]$ with a linear regression

$$f(X) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p \tag{18}$$

- The problem boils down to choosing the right complexity
- and estimating β

photo from https://www.dailydot.com/parsec/batman-1966-labels-tumblr-twitter-vine/

Obtaining the coefficients

$$min_f E(y - f(X))^2 = min_\beta E(y - \beta_0 + \sum_{k=1}^{K} X_k \beta_k)^2 \tag{19}$$

Agenda

1. Machine learning is all about prediction
2. Prediction vs Causality
3. Getting serious about prediction
   - The basic logic of prediction
   - Prediction error and its components
4. Prediction and linear regression
   - Traditional Computation
   - Gradient Descent
5. Review

Linear Regression

- Using matrix algebra, the loss function:

$$\tilde{e}'\tilde{e} = (y - X\tilde{\beta})'(y - X\tilde{\beta}) \tag{20}$$

  - $SSR(\tilde{\beta})$ is the aggregation of squared errors if we choose $\tilde{\beta}$ as an estimator.

- The **least squares estimator** $\hat{\beta}$ will be

$$\hat{\beta} = \underset{\tilde{\beta}}{argmin}\, SSR(\tilde{\beta}) \tag{21}$$

Normal Equations

- FOC are

$$\frac{\partial \tilde{e}'\tilde{e}}{\partial \tilde{\beta}} = 0 \tag{22}$$

$$-2X'y + 2X'X\tilde{\beta} = 0 \tag{23}$$

- SOC (H.W.)

Normal Equations

- Let $\hat{\beta}$ be the solution. Then $\hat{\beta}$ satisfies the following normal equation

$$X'X\hat{\beta} = X'y \tag{24}$$

- If the inverse of $X'X$ exists, then

$$\hat{\beta} = (X'X)^{-1}X'y \tag{25}$$

- Pro
  - Closed solution (a bonus!!)
- Cons
  - Involves inverting a $K \times K$ matrix $X'X$
  - requires allocating $O(nk + k^2)$ if n is "big" we cannot store in memory

QR decomposition

- To avoid inverting $X'X$ we can use matrix decomposition: QR decomposition
- Most software use it

Theorem If $A \in \mathbb{R}^{n \times k}$ then there exists an orthogonal $Q \in \mathbb{R}^{n \times k}$ and an upper triangular $R \in \mathbb{R}^{k \times k}$ so that $A = QR$

- Orthogonal Matrices:
  - Def: $Q'Q = QQ' = I$ and $Q' = Q^{-1}$
  - Prop: product of orthogonal is orthogonal, e.g $A'A = I$ and $B'B = I$ then $(AB)'(AB) = B'(A'A)B = B'B = I$
- **(Thin QR)** If $A \in \mathbb{R}^{n \times k}$ has full column rank then $A = Q_1 R_1$ the QR factorization is unique, where $Q_1 \in \mathbb{R}^{n \times k}$ and R is upper triangular with positive diagonal entries

QR decomposition

- $\hat{\beta}$?

$$(X'X)\hat{\beta} = X'y \tag{26}$$
$$(R'Q'QR)\hat{\beta} = R'Q'y \tag{27}$$
$$(R'R)\hat{\beta} = R'Q'y \tag{28}$$
$$R\hat{\beta} = Q'y \tag{29}$$

- Solve by back substitution

QR decomposition

$$X = \begin{bmatrix} 1 & 2 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \quad y = \begin{pmatrix} 1 \\ 4 \\ 2 \end{pmatrix} \tag{30}$$

1. QR factorization X=QR

$$Q = \begin{bmatrix} -0.57 & -0.41 \\ -0.57 & -0.41 \\ -0.57 & 0.82 \end{bmatrix} \quad R = \begin{bmatrix} -1.73 & -4.04 \\ 0 & 0.81 \end{bmatrix} \tag{31}$$

2. Calculate $Q'y = [-4.04, -0.41]'$
3. Solve

$$\begin{bmatrix} -1.73 & -4.04 \\ 0 & 0.81 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} -4.04 \\ -0.41 \end{bmatrix} \tag{32}$$

Solution is $(3.5, -0.5)$

QR decomposition

This is actually what R does under the hood

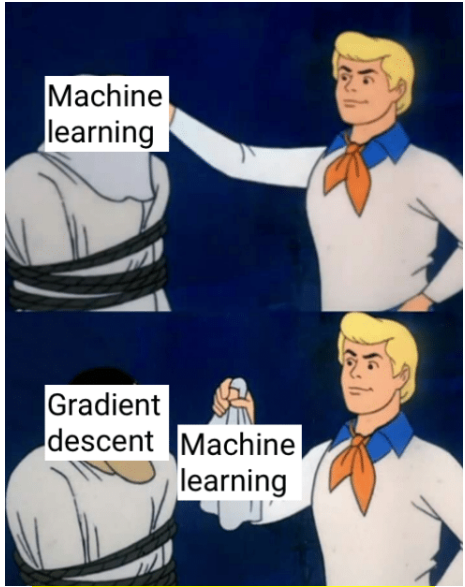
obj	list [12] (S3: lm)	List of length 12
coefficients	double [2]	-1.71e+08 3.01e+08
residuals	double [207607]	1.17e+09 -2.38e+08 -5.21e+08 -1.96e+08 -5.12e+07 -1.91e+08 ...
effects	double [207607]	-3.15e+11 2.10e+11 -5.24e+08 -1.98e+08 -5.34e+07 -1.93e+08 ...
rank	integer [1]	2
fitted.values	double [207607]	4.31e+08 4.31e+08 7.32e+08 4.31e+08 4.31e+08 4.31e+08 ...
assign	integer [2]	0 1
qr	list [5] (S3: qr)	List of length 5
df.residual	integer [1]	207605
xlevels	list [0]	List of length 0
call	language	lm(formula = price ~ bathrooms, data = dta0)
terms	formula	price ~ bathrooms
model	list [207607 x 2] (S3: data.fra	A data.frame with 207607 rows and 2 columns

Note that R's `lm` also returns many objects that have the same size as X and y

Agenda

1. Machine learning is all about prediction
2. Prediction vs Causality
3. Getting serious about prediction
   - The basic logic of prediction
   - Prediction error and its components
4. Prediction and linear regression
   - Traditional Computation
   - Gradient Descent
5. Review

Gradient Descent



Gradient Descent

- Gradient Descent is a very generic optimization algorithm capable of finding optimal solutions to a wide range of problems.
- The general idea of Gradient Descent is to tweak parameters iteratively in order to minimize a loss function.

$$\min_f E[L\left(y_i, f(\boldsymbol{X_i})\right)] \tag{33}$$

Gradient Descent

Linear regression

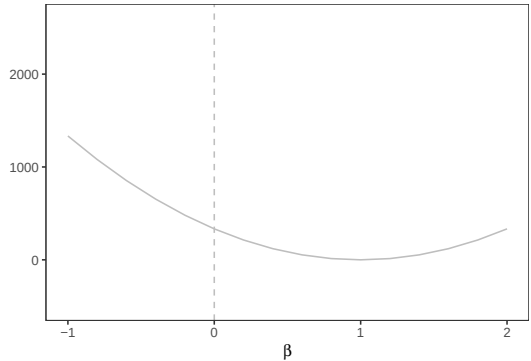
- The problem boils down to estimating the coefficients of vector β which minimize an objective function:

$$\arg\min_{\beta} \sum_{i=1}^{n} \frac{1}{n} \left(y_i - \beta_0 + \sum_{k=1}^{K} X_k \beta_k \right)^2 \tag{34}$$

Gradient Descent

Linear regression

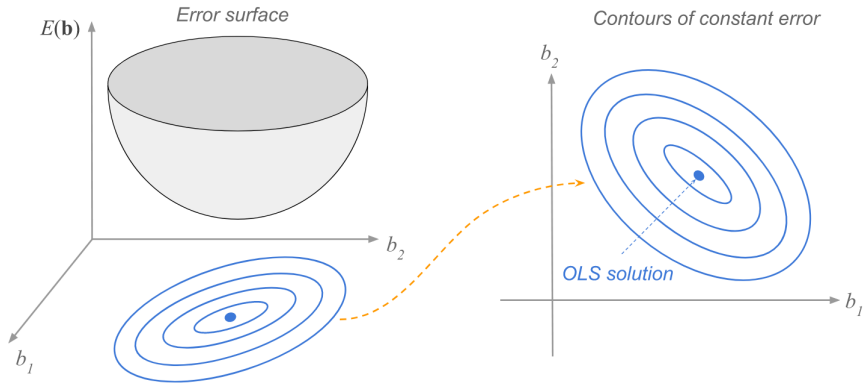
- Intuition: Loss Function 1 dimension



Gradient Descent

Linear regression

- Intuition: Loss Function 2 dimension



Gradient Descent

- In a more general context, when at a point $\beta \in \mathbb{R}^k$, at any step i, the gradient descent algorithm tries to move in a direction $\delta\beta$ such that:

$$L(\beta^{(t)} + \delta\beta) < L(\beta^{(t)}) \tag{35}$$

- The choice of $\delta\beta$ is made such that $\delta\beta = -\epsilon\nabla_\beta L(\beta^{(t)})$:

$$\beta^{(t+1)} = \beta^{(t)} - \epsilon\nabla_\beta L(\beta^{(t)}) \tag{36}$$

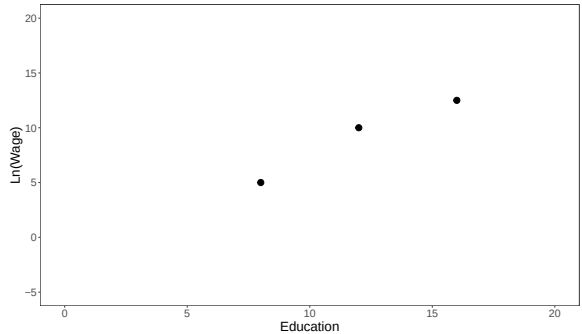
- In other words, you need to calculate how much the cost function will change if you change β just a little bit.

Gradient Descent

- Algorithm
  1. Randomly pick starting values for the parameters
  2. Compute the gradient of the objective function at the current value of the parameters using all the observations from the training sample
  3. Update the parameters
  4. Repeat from step 2 until a fixed number of iteration or until convergence.

Gradient Descent: Example

log(wage)	Education (years)
5	8
10	12
12.5	16



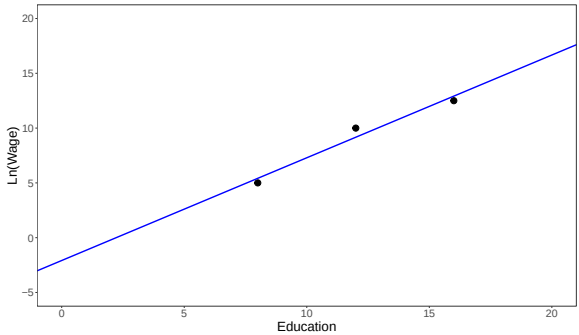
Gradient Descent: Example

log(wage)	Education (years)
5	8
10	12
12.5	16

$$\hat{\beta} = (X'X)^{-1}X'y$$

```
beta<-solve(t(X)%*%X)%*%t(X)%*%y

lm(y~x,data)
```



$$y = -2.0833 + 0.9375 \times Educ$$

Gradient Descent: Example

$$R(\alpha, \beta) = \frac{1}{n} \sum_{i=1}^{n} (y_i - (\alpha + \beta x_i))^2$$

The Gradient

$$\nabla R(\alpha, \beta) = \begin{pmatrix} \frac{\partial R}{\partial \alpha} \\ \frac{\partial R}{\partial \beta} \end{pmatrix} = \begin{pmatrix} -\frac{2}{n} \sum_{i=1}^{n} (y_i - \alpha - \beta x_i) \\ -\frac{2}{n} \sum_{i=1}^{n} x_i (y_i - \alpha - \beta x_i) \end{pmatrix}$$

Updating

$$\alpha^{(t+1)} = \alpha^{(t)} - \epsilon \frac{\partial R}{\partial \alpha}$$

$$\beta^{(t+1)} = \beta^{(t)} - \epsilon \frac{\partial R}{\partial \beta}$$

Gradient Descent: Example

First Iteration

log(wage)	Education (years)
5	8
10	12
12.5	16

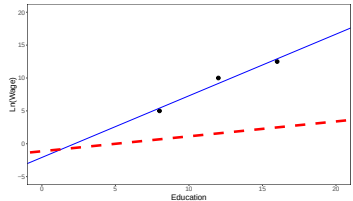
Start with an initial guess: $\alpha = -1; \beta = 2$, and a learning rate ($\epsilon = 0.005$). Then we have

$$\alpha' = (-1) - 0.005\,(-2/3 \times ((5 - (-1) - 2 \times 8) + (10 - (-1) - 2 \times 12) + (12.5 - (-1) - 2 \times 16))$$

$$\beta' = 2 + 0.005\,(-2/3 \times (8(5 - (-1) - 2 \times 8) + 12(10 - (-1) - 2 \times 12) + 16(12.5 - (-1) - 2 \times 16))$$

$$\alpha' = -1.1384$$

$$\beta' = 0.2266$$



Gradient Descent: Example

Second Iteration

log(wage)	Education (years)
5	8
10	12
12.5	16

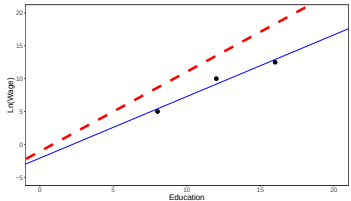
Start with an initial guess: $\alpha = -1; \beta = 2$, and a learning rate ($\epsilon = 0.005$). Then we have

$$\alpha^2 = (-1.1384) - 0.005\,(-2/3 \times ((5 - (-1.1384) - (0.2266) \times 8) + (10 - (-1.1384) - (0.2266) \times 12) + (12.5 - (-1.1384) - (0.2266) \times 16))$$

$$\beta^2 = (0.2266) + 0.005\,(-2/3 \times (8(5 - (-1.1384) - (0.2266) \times 8) + 12(10 - (-1.1384) - (0.2266) \times 12) + 16(12.5 - (-1.1384) - (0.2266) \times 16))$$

$$\alpha^2 = -1.0624$$

$$\beta^2 = 1.212689$$



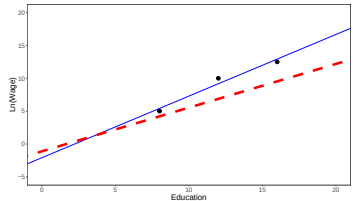
Gradient Descent: Example

Third Iteration

log(wage)	Education (years)
5	8
10	12
12.5	16

$$\alpha^3 = -1.0624$$

$$\beta^3 = 1.212689$$



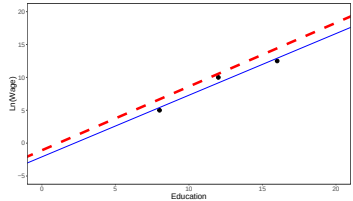
Gradient Descent: Example

Fourth Iteration

log(wage)	Education (years)
5	8
10	12
12.5	16

$$\alpha^4 = -1.082738$$

$$\beta^4 = 0.9693922$$



Gradient Descent: Example

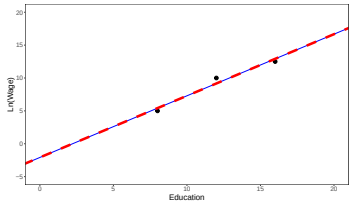
7211 Iteration

log(wage)	Education (years)
5	8
10	12
12.5	16

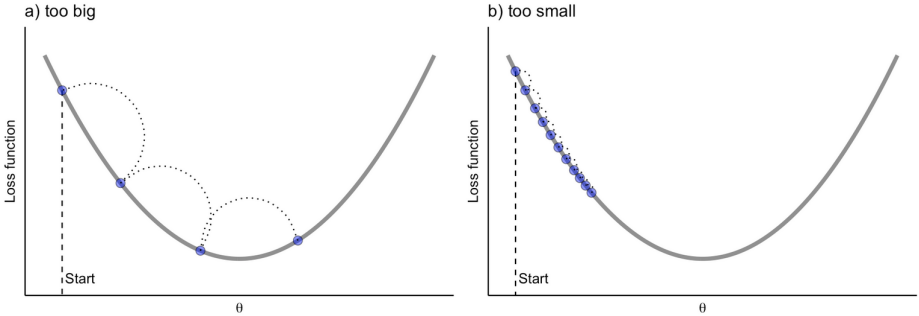
$$\alpha^{7211} = -2.076246$$

$$\beta^{7211} = 0.9369499$$

$$y^{ols} = -2.0833 + 0.9375 \times Educ$$



The learning rate



Source: Boehmke, B., & Greenwell, B. (2019)

- We can choose ϵ in several different ways:
  - Set ϵ to a small constant.
  - Use varying learning rates.

Agenda

Review

- This Week: The predictive paradigm and linear regression
  - Machine Learning is all about prediction
  - ML targets something different than causal inference, they can complement each other
  - Linear Regression can approximate $E(y|X)$
  - Inner workings of linear regression
- Next Module: Choosing choosing the right complexity, Out of sample prediction. Over-fit, Resampling Methods, Web-scrapping