Prediction and Linear Regression

Big Data y Machine Learning para Economía Aplicada

Ignacio Sarmiento-Barbieri

Universidad de los Andes

Agenda

Agenda

1. Best Predictor
   - Statistical Properties
   - Numerical Properties
2. Calculating the OLS coefficients
   - Traditional Computation
   - Gradient Descent
3. Review

Best Predictor

- In the population, the best predictor of Y given W

$$g(W) = E[Y|W] \tag{1}$$

Best Predictor

- The CEF solves the best prediciton problem

$$min_{m(W)}E[(Y-m(W))^2] = \int (Y-m(W))^2 Pr(dW,dY) \tag{2}$$

- conditioning on W we have that

$$E_W E_{Y|X}[(Y-m(W))^2|W] \tag{3}$$

Agenda

1. Best Predictor
   - Statistical Properties
   - Numerical Properties
2. Calculating the OLS coefficients
   - Traditional Computation
   - Gradient Descent
3. Review

The Best Linear Prediction Problem in Finite Samples

$$(Y_i, X_i)_{i=1}^{n} = ((Y_1, X_1), \ldots, (Y_n, X_n)) \tag{4}$$

- We assume that this sample is a random sample from the distribution of (Y, X)

The Best Linear Prediction Problem in Finite Samples

- Replace E with $\mathbb{E}$

$$\sum_{j=1}^{k} \hat{\beta}_j X_j = \hat{\beta}' X \tag{5}$$

- $\hat{\beta}$ is any solution to the Best Linear Prediction Problem in the Sample, also known as Ordinary Least Squares (OLS)

Statistical Properties

Under certain assumptions HW Review the Assumption from Econometrics

- Small Sample (Gauss-Markov Theorem)
  - Unbiased: $E(\hat{\beta}) = \beta$
  - Minimum Variance: $Var(\tilde{\beta}) - Var(\hat{\beta})$ is positive semidefinite matrix Proof: HW. Remember: a matrix $M_{p\times p}$ is positive semi-definite iff $c'Mc \geq 0 \; \forall c \in \mathbb{R}^p$
- Large Sample
  - Consistency: $\hat{\beta} \to_p \beta$
  - Asymptotically Normal: $\sqrt{N}(\hat{\beta} - \beta) \sim_a N(0, S)$

Gauss Markov Theorem

- Gauss Markov Theorem that says OLS is BLUE is perhaps one of the most famous results in statistics.
  - $E(\hat{\beta}) = \beta$
  - $Var(\hat{\beta}) = \sigma^2(X'X)^{-1}$
- and implies that $\hat{y}$ is an unbiased predictor and minimum variance, from the class of unbiased linear predictors (BLUP) H.W. proof

Gauss Markov Theorem

- Gauss Markov Theorem that says OLS is BLUE is perhaps one of the most famous results in statistics.
  - $E(\hat{\beta}) = \beta$
  - $Var(\hat{\beta}) = \sigma^2(X'X)^{-1}$
- and implies that $\hat{y}$ is an unbiased predictor and minimum variance, from the class of unbiased linear predictors (BLUP) H.W. proof
- However, it is essential to note the limitations of the theorem.
  - Correctly specified with exogenous Xs,
  - The term error is homoscedastic
  - No serial correlation.
  - Nothing about the OLS estimator being the more efficient than any other estimator one can imagine.

Statistical Properties

- The fundamental statistical issue is that we are trying to estimate k parameters
- We need many observations per parameter
- n/p should be large, or, equivalently that p/n should be small

Analysis of Variance

- Involves the decomposition of the variation of Y into explained and unexplained parts.
- Explained variation is a measure of the predictive performance of a model.
- Can be conducted both in the population and in the sample.

Analysis of Variance

$$Y = \beta' X + \epsilon$$

$$E[\epsilon \mid X] = 0,$$

Idea: decompose the variation in Y into the sum of explained variation and residual variation.

$$E[Y^2] = E[(\beta' X)^2] + E[\epsilon^2]$$

Agenda

Numerical Properties

- Numerical properties have nothing to do with how the data was generated
- These properties hold for every data set, just because of the way that $\hat{\beta}$ was calculated
- Davidson & MacKinnon, Greene y Ruud have nice geometric interpretations

Projection

OLS Residuals:

$$\hat{\epsilon} = y - \hat{y} \tag{6}$$
$$= y - X\hat{\beta} \tag{7}$$

replacing $\hat{\beta}$

$$\hat{\epsilon} = y - X(X'X)^{-1}X'y \tag{8}$$
$$= (I - X(X'X)^{-1}X')y \tag{9}$$

Define two matrices

- Projection matrix $P_X = X(X'X)^{-1}X'$
- Annihilator (residual maker) matrix $M_X = (I - P_X)$

Projection

- Both are symmetric
- Both are idempotent $(A'A) = A$
- $P_X X = X \Rightarrow$ projection matrix
- $M_X X = 0 \Rightarrow$ annihilator matrix

Frisch-Waugh-Lovell (FWL) Theorem

- Lineal Model: $Y = X\beta + u$
- Split it: $Y = X_1\beta_1 + X_2\beta_2 + u$

Theorem

1. The OLS estimates of β_2 from these equations

$$y = X_1\beta_1 + X_2\beta_2 + u \tag{10}$$

$$M_{X_1}y = M_{X_1}X_2\beta_2 + residuals \tag{11}$$

   are numerically identical

2. the OLS residuals from these regressions are also numerically identical

Applications

- Why FWL is useful in the context of big volume of data?
- An computationally inexpensive way of
  - Removing nuisance parameters
    - E.g. the case of multiple fixed effects. The traditional way is either apply the within transformation with respect to the FE with more categories then add one dummy for each category for all the subsequent FE
    - Not feasible in certain instances.
  - Computing certain diagnostic statistics: Leverage, R^2, LOOCV.
  - Helps with online updating

Applications: Fixed Effects

- For example: Carneiro, Guimarães, & Portugal (2012) *AEJ: Macroeconomics*

$$\ln w_{ijft} = x_{it}\beta + \lambda_i + \theta_j + \gamma_f + u_{ijft} \tag{12}$$

$$Y = X\beta + D_1\lambda + D_2\theta + D_3\gamma + u \tag{13}$$

  - Data set 31.6 million observations, with 6.4 million individuals (i), 624 thousand firms (f), and 115 thousand occupations (j), 11 years (t).
  - Storing the required indicator matrices would require 23.4 terabytes of memory
  - From their paper
    "In our application, we first make use of the Frisch-Waugh-Lovell theorem to remove the influence of the three high- dimensional fixed effects from each individual variable, and, in a second step, implement the final regression using the transformed variables. With a correction to the degrees of freedom, this approach yields the exact least squares solution for the coefficients and standard errors"

Applications: Outliers and High Leverage Data

- App

$$\hat{\beta} = (X'X)^{-1}X'y \quad (14)$$

Applications: Outliers and High Leverage Data

Consider a dummy variable e_j which is an $n - vector$ with element j equal to 1 and the rest is 0. Include it as a regressor

$$y = X\beta + \alpha e_j + u \tag{15}$$

using FWL we can do

$$M_{e_j} y = M_{e_j} X\beta + r \tag{16}$$

- β and *residuals* from both regressions are identical
- Same estimates as those that would be obtained if we deleted observation j from the sample. We are going to denote this as $\beta^{(j)}$

Agenda

Agenda

1. Best Predictor
   - Statistical Properties
   - Numerical Properties
2. Calculating the OLS coefficients
   - Traditional Computation
   - Gradient Descent
3. Review

Linear Regression

- Using matrix algebra, the loss function:

$$\tilde{\epsilon}'\tilde{\epsilon} = (y - X\tilde{\beta})'(y - X\tilde{\beta}) \tag{17}$$

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

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

$$\hat{\beta} = \underset{\tilde{\beta}}{argmin}\, SSR(\tilde{\beta}) \tag{18}$$

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

$$X = \begin{bmatrix} 1 & 2 \\ 1 & 2 \\ 1 & 3 \end{bmatrix} \; y = \begin{pmatrix} 1 \\ 4 \\ 2 \end{pmatrix} \tag{19}$$

1. QR factorization X=QR

$$Q = \begin{bmatrix} -0.57 & -0.41 \\ -0.57 & -0.41 \\ -0.57 & 0.82 \end{bmatrix} \; R = \begin{bmatrix} -1.73 & -4.04 \\ 0 & 0.81 \end{bmatrix} \tag{20}$$

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

$$\begin{bmatrix} -1.73 & -4.04 \\ 0 & 0.81 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} = \begin{bmatrix} -4.04 \\ -0.41 \end{bmatrix} \tag{21}$$

Solution is $(3.5, -0.5)$

QR decomposition

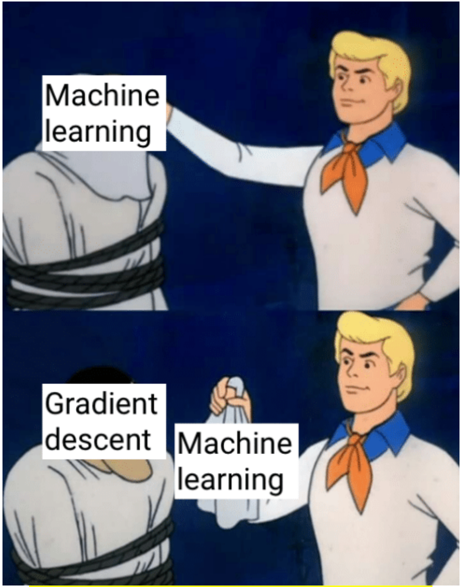
This is actually what R does under the hood

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

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.

$$\min_f E[L(y_i, f(\boldsymbol{X_i}))] \tag{22}$$

Gradient Descent

Linear regression

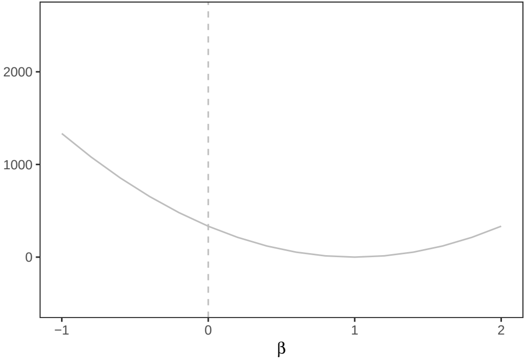
- The problem boils down to estimating the coefficients of vector β which minimize an objective function:

$$\arg\min_{\beta} \sum_{i=1}^{n} \frac{1}{n} \left(y_i - \beta_0 + \sum_{k=1}^{K} X_k \beta_k \right)^2 \tag{23}$$

Gradient Descent

Linear regression

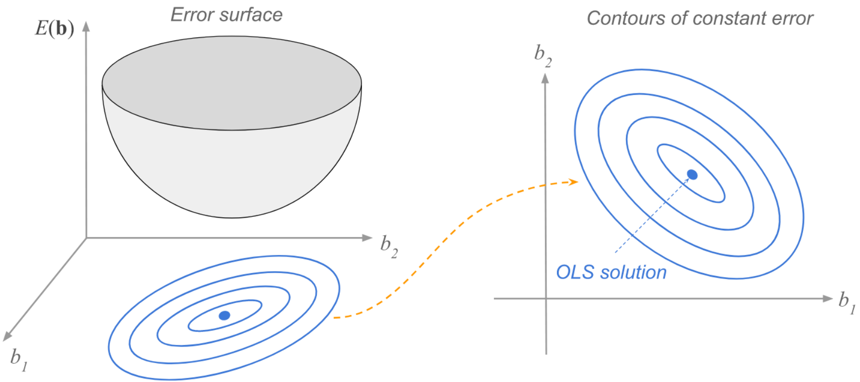
- Intuition: Loss Function 1 dimension



Gradient Descent

Linear regression

- Intuition: Loss Function 2 dimensions



Gradient Descent

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

$$L(\beta^{(j)} + \delta\beta) < L(\beta^{(j)}) \tag{24}$$

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

$$\beta^{(t+1)} = \beta^{(j)} - \epsilon\nabla_\beta L(\beta^{(j)}) \tag{25}$$

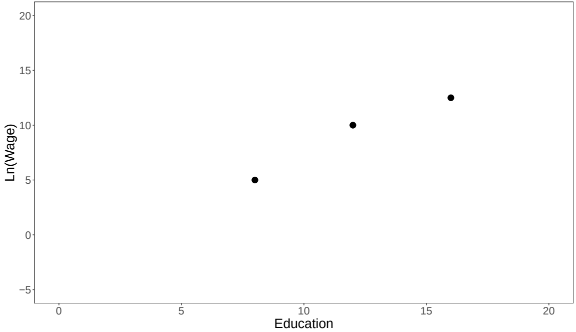
- In other words, you need to calculate how much the cost function will change if you change β just a little bit.

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

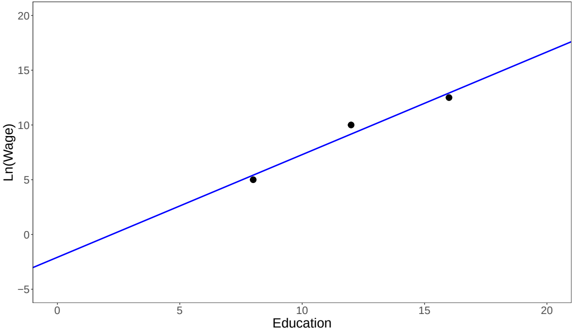


Gradient Descent: Example

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

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

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



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

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

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

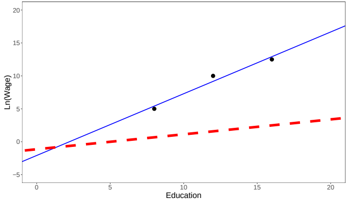
Gradient Descent: Example

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

$$\alpha^1 = -1.1384$$
$$\beta^1 = 0.2266$$



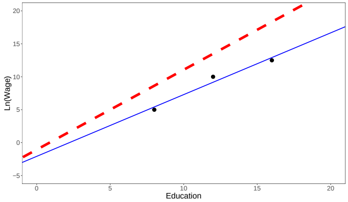
Gradient Descent: Example

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

$$\alpha^2 = -1.0624$$
$$\beta^2 = 1.212689$$



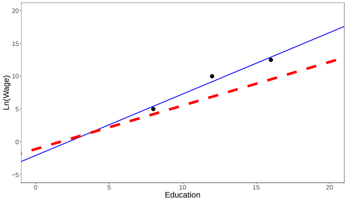
Gradient Descent: Example

Third Iteration

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

$$\alpha^3 = -1.0624$$

$$\beta^3 = 1.212689$$



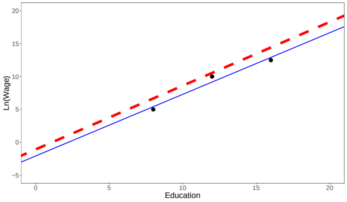
Gradient Descent: Example

Fourth Iteration

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

$$\alpha^4 = -1.082738$$

$$\beta^4 = 0.9693922$$



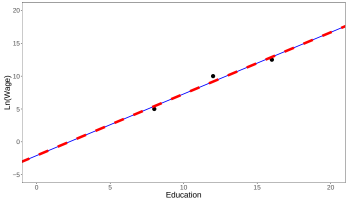
Gradient Descent: Example

7211 Iteration

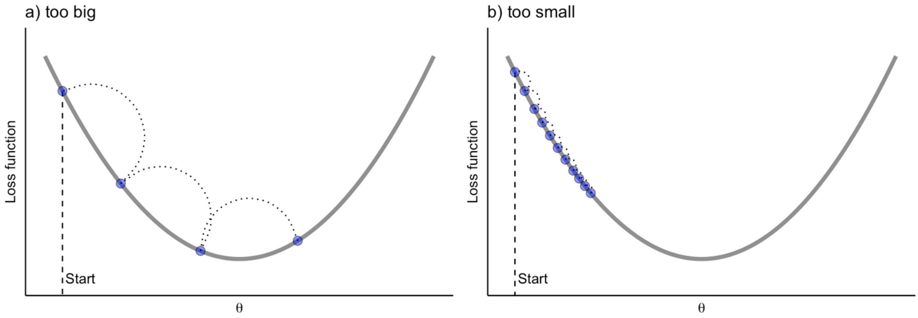
log(wage)	Education (years)
5	8
10	12
12.5	16

$$\alpha^{7211} = -2.076246$$
$$\beta^{7211} = 0.9369499$$

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



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.

Table 1: Comparison of algorithms for Linear Regression

Algorithm	**Large** k	**Large** n	**Hyperparams**
QR	Fast	Slow	0
SVD	Fast	Slow	0
Batch GD	Slow	Fast	2

Agenda

Review

- These two Weeks: The predictive paradigm and linear regression
  - BLP: $E(y|X)$
  - Inner workings of linear regression
- Next Module: Uncertainty