The Linear Algebra Behind Machine Learning: Under the Hood with Regression Analysis

Steven Slezak

Data Analytics Lead, Insurance Industry
Professor, Agribusiness Department (retired)
College of Agriculture, Cal Poly
https://github.com/seslezak/R-Code
https://www.linkedin.com/in/steven-slezak-a3243914/

SOUTHERN CALIFORNIA R USERS ALL-HANDS MEETUP

WARNER BROS

BURBANK, CALIFORNIA

23 NOVEMBER 2019

First, a Fun Reminder of Matrix Algebra Simplicity

Create a simple vector (1 through 12) and multiply by its transpose

```
v <- matrix(1:12)
v %*% t(v)
```

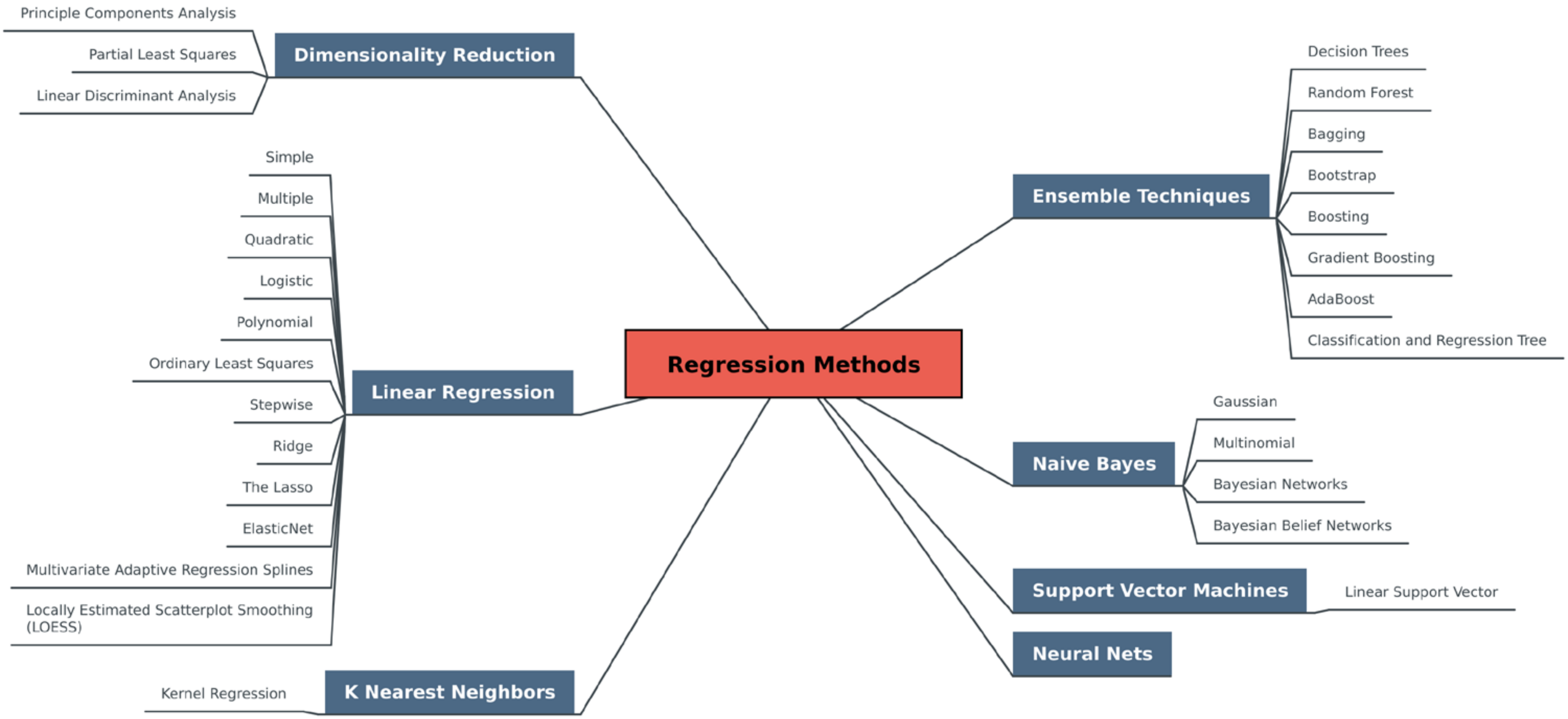
Who remembers what you get?

```
##       [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
##  [1,]    1    2    3    4    5    6    7    8    9    10    11    12
##  [2,]    2    4    6    8   10   12   14   16   18    20    22    24
##  [3,]    3    6    9   12   15   18   21   24   27    30    33    36
##  [4,]    4    8   12   16   20   24   28   32   36    40    44    48
##  [5,]    5   10   15   20   25   30   35   40   45    50    55    60
##  [6,]    6   12   18   24   30   36   42   48   54    60    66    72
##  [7,]    7   14   21   28   35   42   49   56   63    70    77    84
##  [8,]    8   16   24   32   40   48   56   64   72    80    88    96
##  [9,]    9   18   27   36   45   54   63   72   81    90    99   108
## [10,]   10   20   30   40   50   60   70   80   90   100   110   120
## [11,]   11   22   33   44   55   66   77   88   99   110   121   132
## [12,]   12   24   36   48   60   72   84   96  108   120   132   144
```

Simple Regression: Basis of Machine Learning

- Our Simple Line ($y = mx + b$) Requires Setting of Parameters
  - ✓ slope and intercept
- Algorithm Used to Determine Parameters
  - ✓ sum of the square errors
  - ✓ minimize error between true outputs and predicted data
- All Machine Learning Involves:
  - ✓ model with parameters
  - ✓ data
  - ✓ algorithm for optimizing parameters
- Neural Nets Pass Multilinear Inputs Through a Network of Non-Linear Activation Functions
- We Will Do All This with Linear Algebra (in Five Minutes!)

Regression: Gateway to Machine Learning



What is Batting Average for All Major League Players Through History?

X – Independent Variable

Number of At Bats (AB)

Y – Dependent Variable

Number of Hits (H)

Batting Average = Hits ÷ At Bats

$$BA = \frac{H}{AB}$$

The Data Set:

Lahman Package in *R*

Batting Table

105,861 Rows and 22 Columns

Our Problem:

Create a Linear Algebra Solution in *R*

A Quick Review of the Math

Just a Teensy Bit of Matrix Algebra; *R* Makes it Easy!

The Equations

$y = mx + b$	the equation for a line
$Y = X\beta + \varepsilon$	the OLS regression equation
$\hat{\beta} = (X^T X)^{-1} X^T Y$	calculates the estimated coefficients
$VCV = Var(\hat{\beta} \mid X) = \frac{1}{n-k} \hat{\varepsilon}^T \hat{\varepsilon}\, (X^T X)^{-1}$	the variance-covariance matrix
$SSR = \hat{\varepsilon}^T \hat{\varepsilon} = \sum_{i=1}^{n} \hat{\varepsilon}_i^2$	sum of the squared residuals
$TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2$	total sum of squares
$R^2 = 1 - \frac{SSR}{TSS}$	coefficient of determination

Matrix Operators in *R*

as.matrix() – coerces object to matrix class

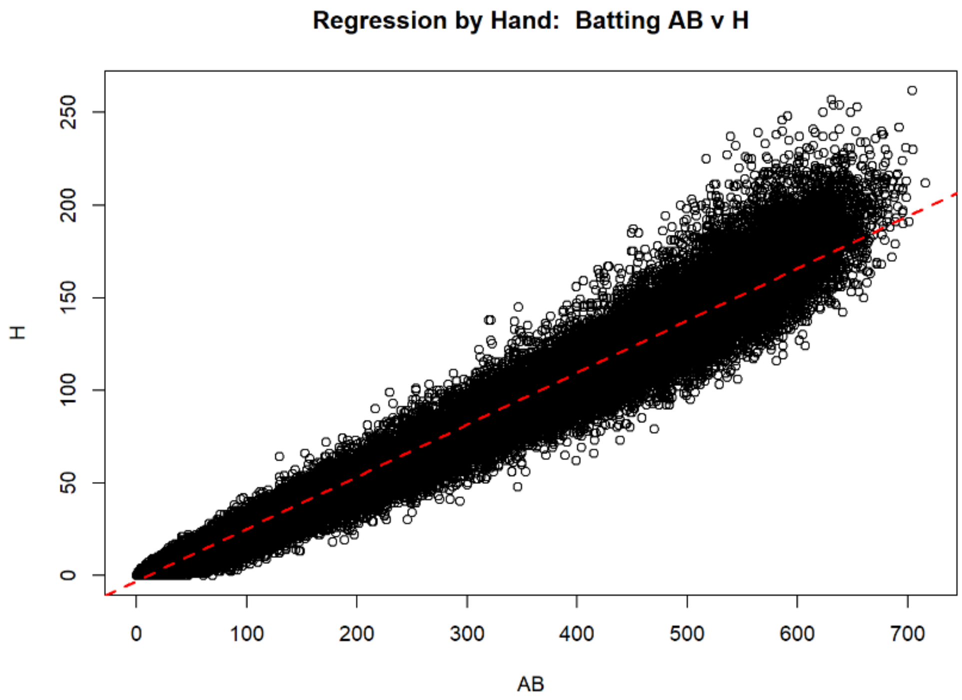
t() – transposes matrix

%%* – matrix multiplication operator

solve() – takes inverse of matrix

Visualize the Results

```
plot(Batting$AB, Batting$H, xlab = 'AB', ylab = 'H',
     main = 'Regression by Hand:  Batting AB v H')
abline(a = bh[1], b = bh[2], col = 'red', lwd = 2, lty = 'dashed')
```



Load the Libraries

```
library(readr)
library(tidyverse)
library(GGally)
library(gridExtra)
library(scales)
library(Lahman)
```

Run Analysis Using Base *R* and Output Results

```
reg1 <- lm(H ~ AB, Batting)
stargazer::stargazer(reg1, type = 'text')
```

```
##
## =====================================================
##                          Dependent variable:
##                     ---------------------------------
##                                   H
## -----------------------------------------------------
## AB                             0.281***
##                               (0.0001)
##
## Constant                      -2.747***
##                                (0.032)
##
## -----------------------------------------------------
## Observations                   105,861
## R2                              0.975
## Adjusted R2                     0.975
## Residual Std. Error      8.220 (df = 105859)
## F Statistic         4,208,123.000*** (df = 1; 105859)
## =====================================================
## Note:                  *p<0.1; **p<0.05; ***p<0.01
```

Create X and Y Matrices

```
X <- as.matrix(cbind(1, Batting$AB))
Y <- as.matrix(Batting$H)
```

Calculate the Beta Hat and Residuals

```
bh <- round(solve(t(X) %*% X) %*% t(X) %*% Y, digits = 3)
beta.hat <- as.data.frame(cbind(c('Intercept', 'AB'), bh))
names(beta.hat) <- c('Coeff', 'Est')
beta.hat
```

```
##         Coeff    Est
## 1 Intercept -2.747
## 2          AB  0.281
```

```
res <- as.matrix(Batting$H - bh[1] - bh[2] * Batting$AB)
```

Calculate the Variance-Covariance Matrix, Standard Error, and P-Value

```
n <- nrow(Batting)
k <- ncol(X)
VCV <- 1/(n - k) * as.numeric(t(res) %*% res) * solve(t(X) %*% X)
```

```
StdErr <- sqrt(diag(VCV))
P.Val <- rbind(2 * pt(abs(bh[1] / StdErr[1]), df = n - k, lower.tail = FALSE),
               2 * pt(abs(bh[2] / StdErr[2]), df = n - k, lower.tail = FALSE))
```

Combine this With Beta Hat

```
beta.hat2 <- cbind(beta.hat, StdErr, P.Val)
beta.hat
```

```
##          Coeff    Est
## 1 Intercept -2.747
## 2          AB  0.281
```

```
beta.hat2
```

```
##          Coeff    Est        StdErr P.Val
## 1 Intercept -2.747 0.0317964070     0
## 2          AB  0.281 0.0001369756     0
```

Return the Base R and Output Results

```r
reg1 <- lm(H ~ AB, Batting)
stargazer::stargazer(reg1, type = 'text')
```

```
##
## =====================================================
##                          Dependent variable:
##                     ---------------------------------
##                                  H
## -----------------------------------------------------
## AB                           0.281***
##                              (0.0001)
##
## Constant                     -2.747***
##                               (0.032)
##
## -----------------------------------------------------
## Observations                  105,861
## R2                             0.975
## Adjusted R2                    0.975
## Residual Std. Error      8.220 (df = 105859)
## F Statistic         4,208,123.000*** (df = 1; 105859)
## =====================================================
## Note:                     *p<0.1; **p<0.05; ***p<0.01
```

Plot the Regression Line

```
plot(Batting$AB, Batting$H, xlab = 'AB', ylab = 'H',
     main = 'Regression by Hand:  Batting AB v H')
abline(a = bh[1], b = bh[2], col = 'red', lwd = 2, lty = 'dashed')
```