UMD

UNIVERSITY OF MINNESOTA DULUTH

Driven to Discover℠

Lecture Notes for Topic 3: Linear Regression

Introduction to Data Mining , 2nd Edition

by

Tan, Steinbach, Karpatne, Kumar

Outline

- Introduction
- Linear Regression
- Normal Equations
- Gradient Descent
- Overfitting, Underfitting, and Regularization
- Linear Regression

Outline

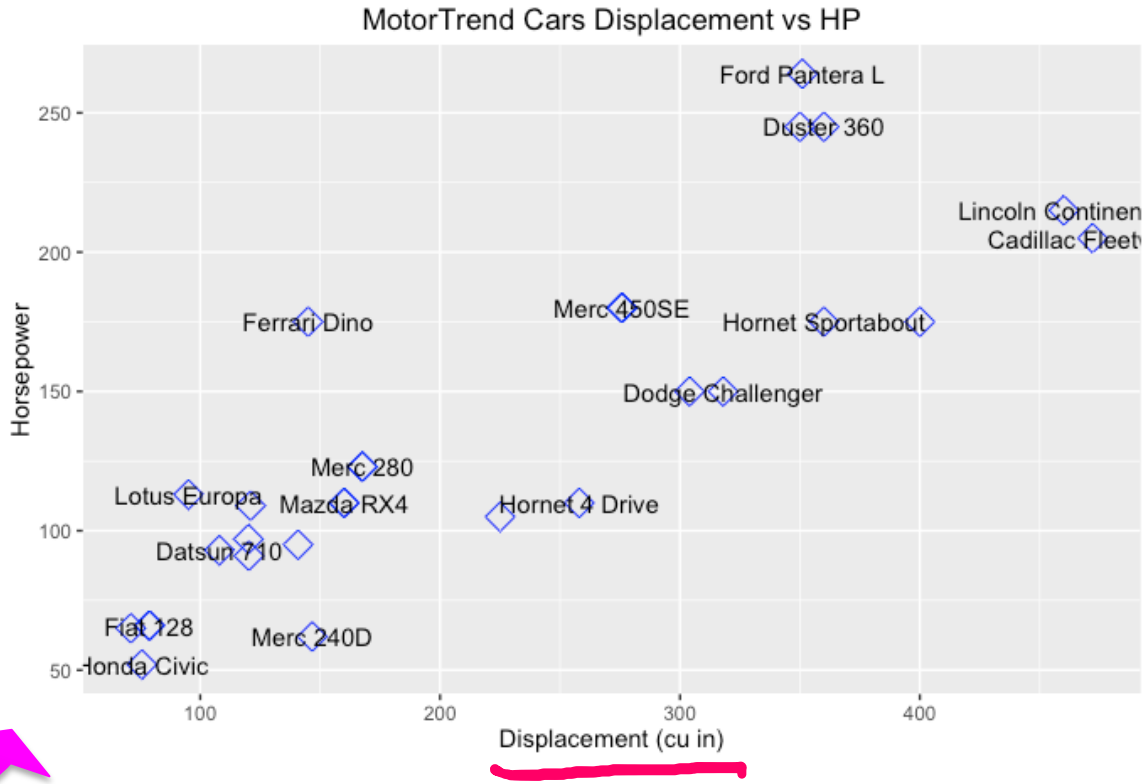
- Introduction
- Linear Regression
- Normal Equations
- Gradient Descent
- Overfitting, Underfitting, and Regularization
- Linear Regression

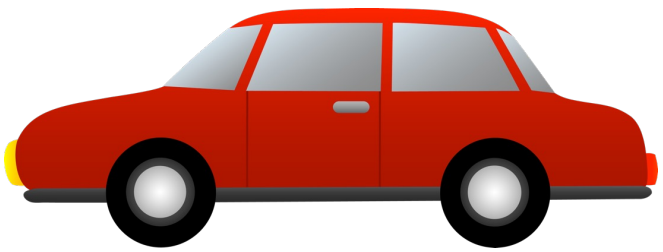
Regression

- Predict a value of a given continuous valued variable based on the values of other variables, assuming a linear or nonlinear model of dependency.
- Extensively studied in statistics, neural network fields
- Examples:
  - Predicting sales amounts of new product based on advertising expenditure.
  - Predicting wind velocities as a function of temperature, humidity, air pressure, etc.
  - Time series prediction of stock market indices.

A ”hello, World!” in Machine Learning

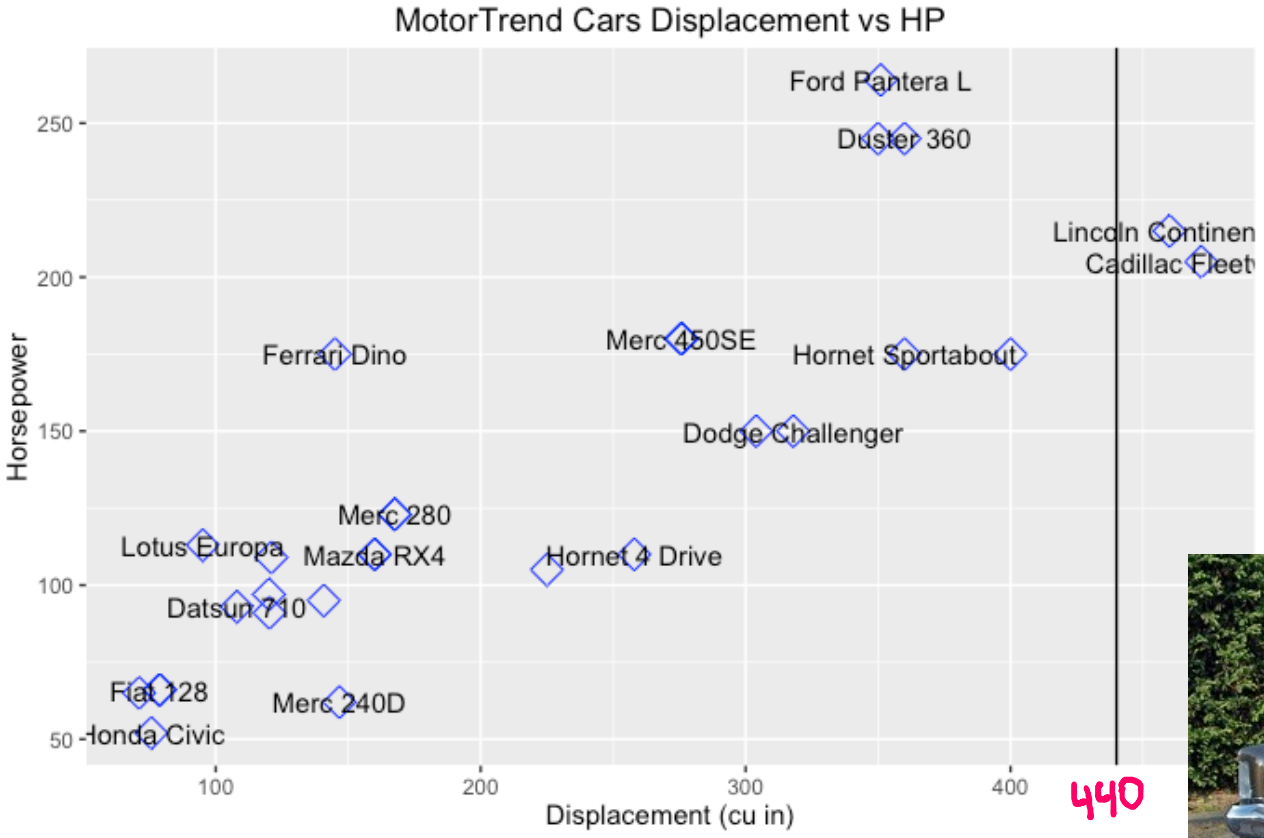
Horsepower is usually proportional to the size of the engine





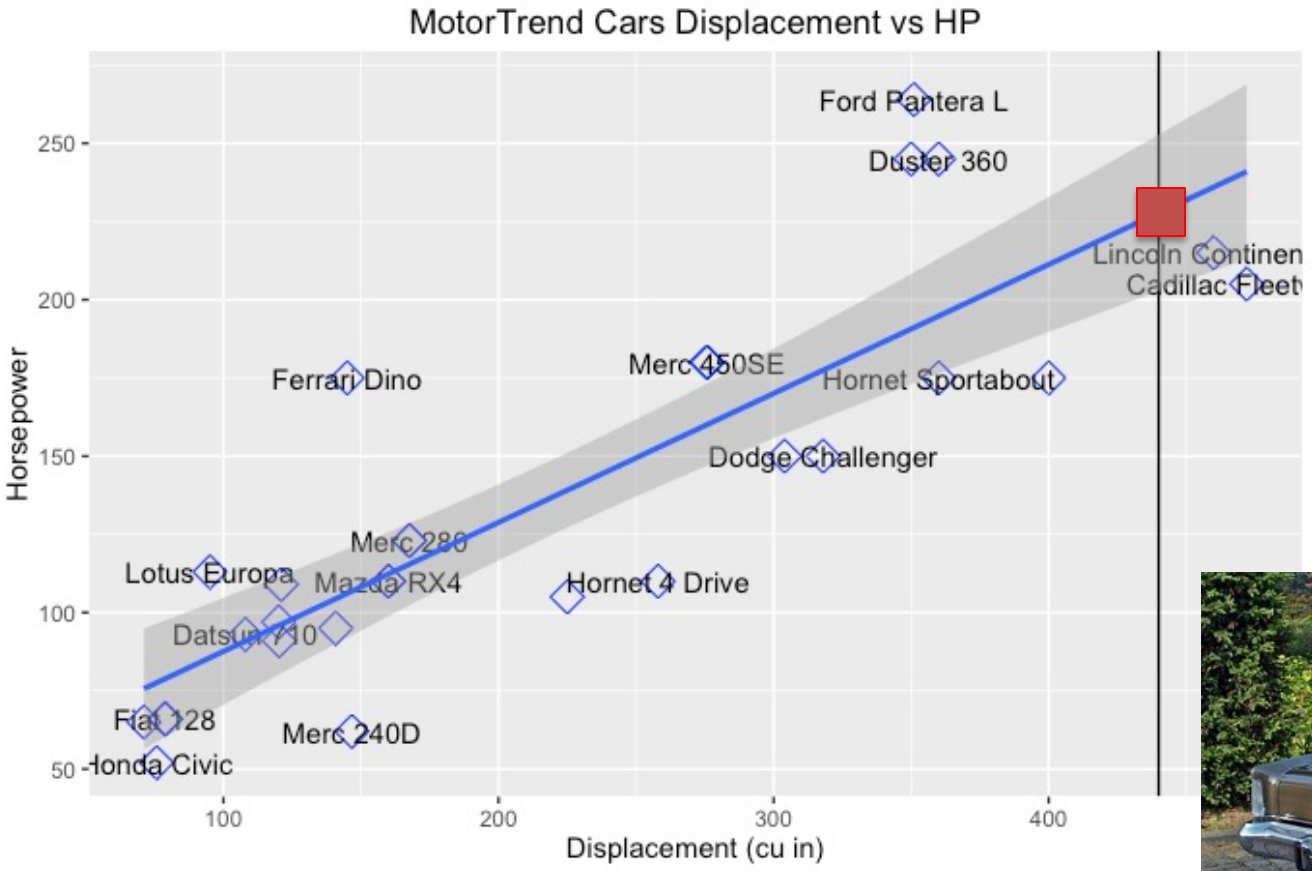
Little car credits:
https://clipartfox.com/categories/view/1043146d7fe5f5fda642973f60a4af14524ccf97/cartoon-car-free-clipart.html

A ”hello, World!” in Machine Learning



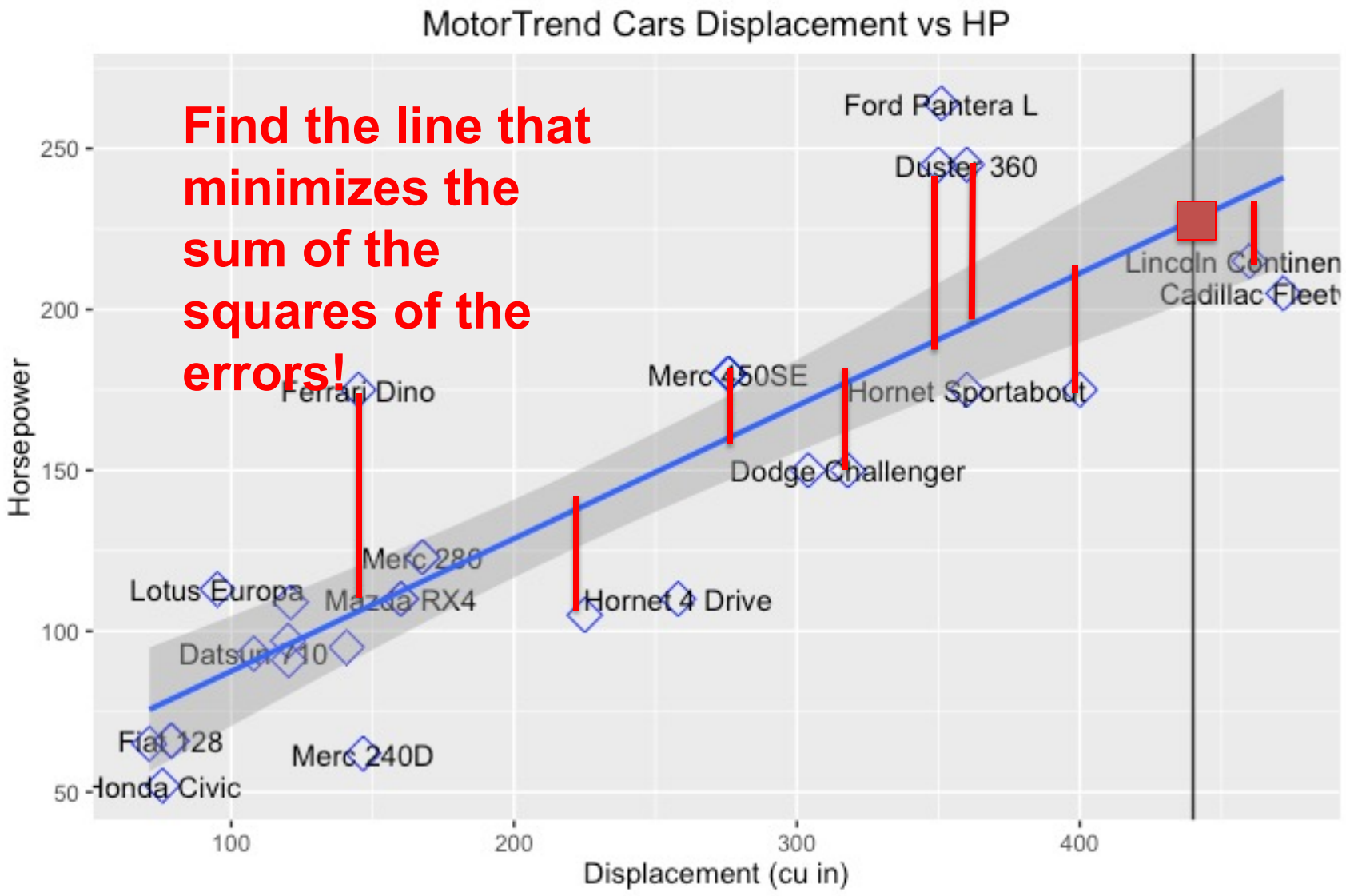
Can we predict the power of a Chrysler Imperial with an engine of 440 cu in?

A ”hello, World!” in Machine Learning

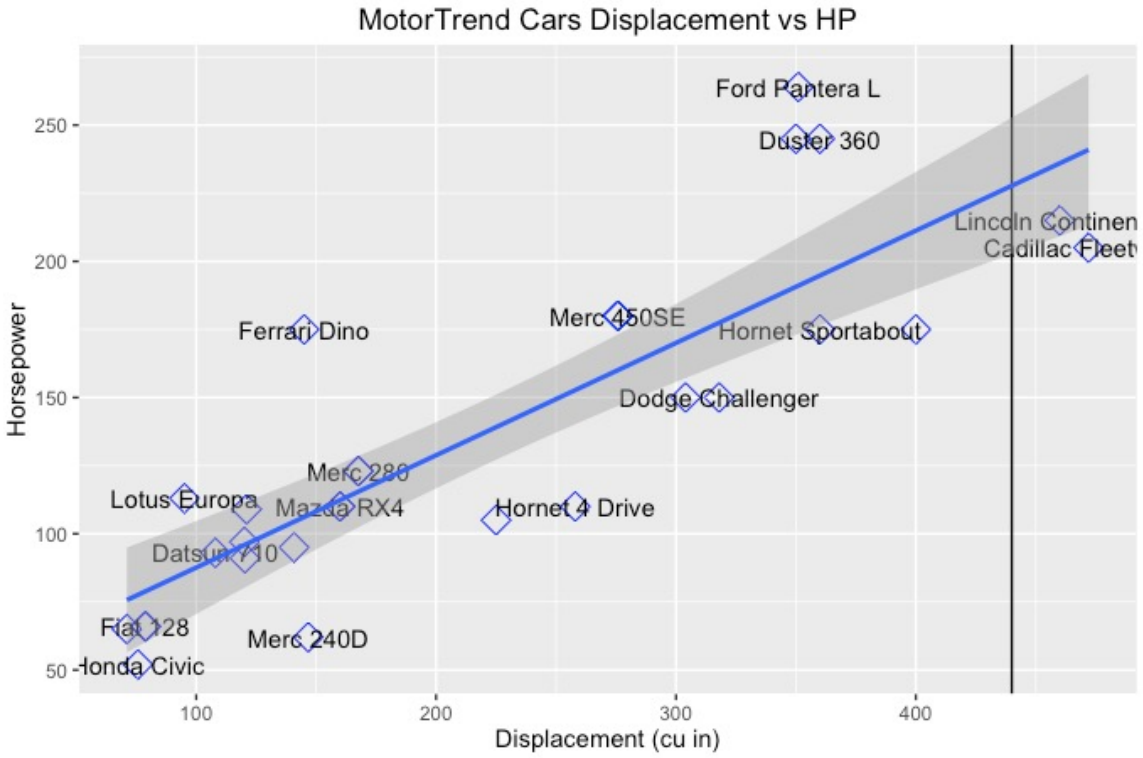


Can we predict the power of a Chrysler Imperial with an engine of 440 cu in?

A ”hello, World!” in Machine Learning



A ”hello, World!” in Machine Learning



Given a set of pairs

$$(disp_1, hp_1), \ldots, (disp_n, hp_n)$$

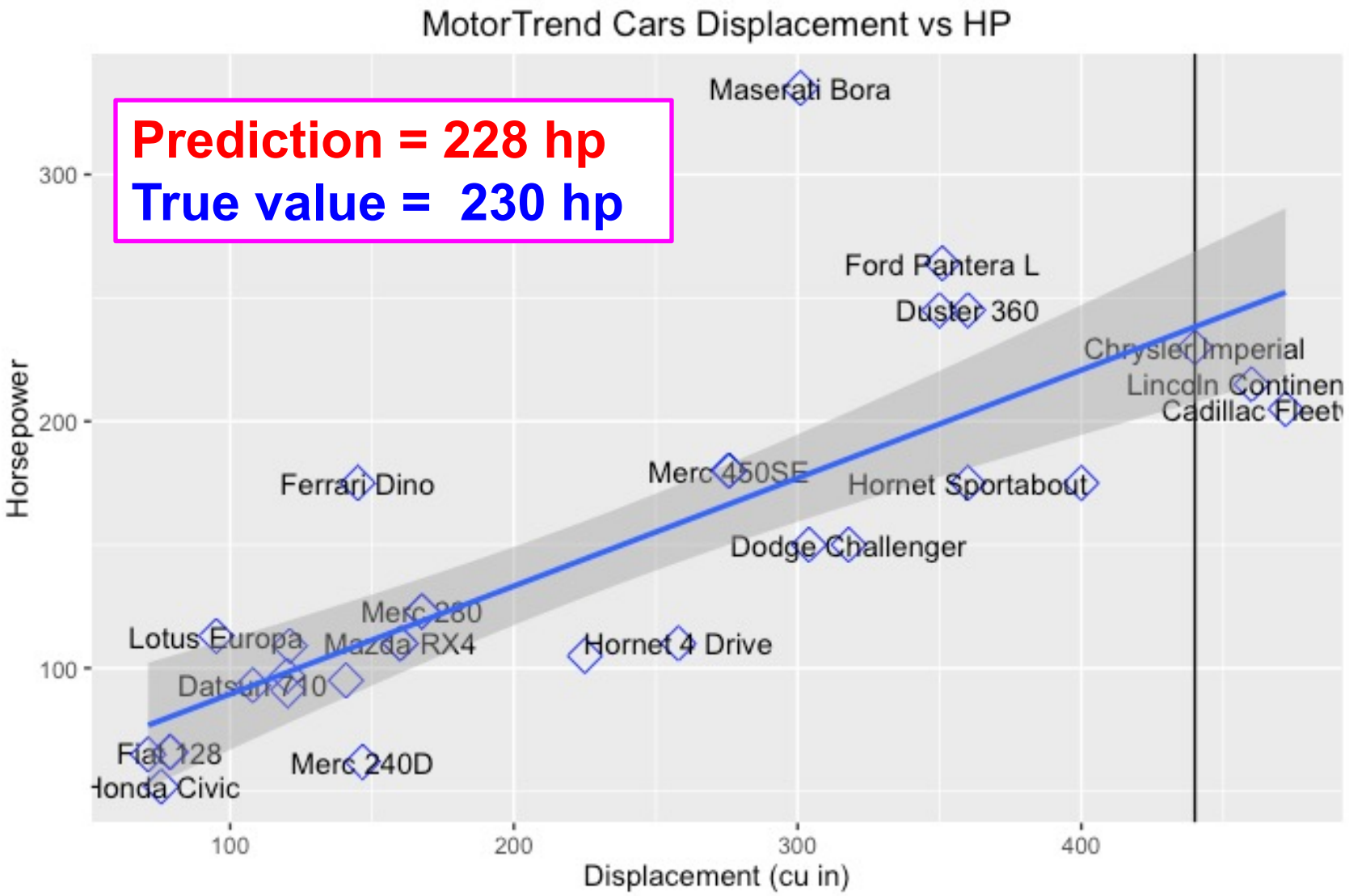
Find a line

$$\widehat{hp}_i = a \cdot disp_i + b$$

Minimize

$$\sum_{i=1}^{n} \left(hp_i - \widehat{hp}_i \right)^2$$

A ”hello, World!” in Machine Learning



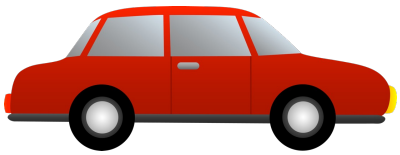
Example 2: Predicting Cognitive Test Scores of Children

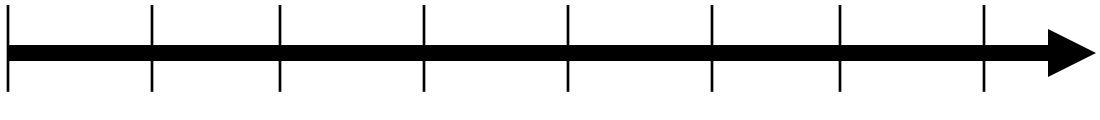
Outline

- Introduction
- Linear Regression
- Normal Equations
- Gradient Descent
- Overfitting, Underfitting, and Regularization
- Linear Regression in R

Motivation for Linear Regression

- Let's suppose that we want to model the movement of car that moves with constant acceleration (a) along a straight road





$x_0 = ?$

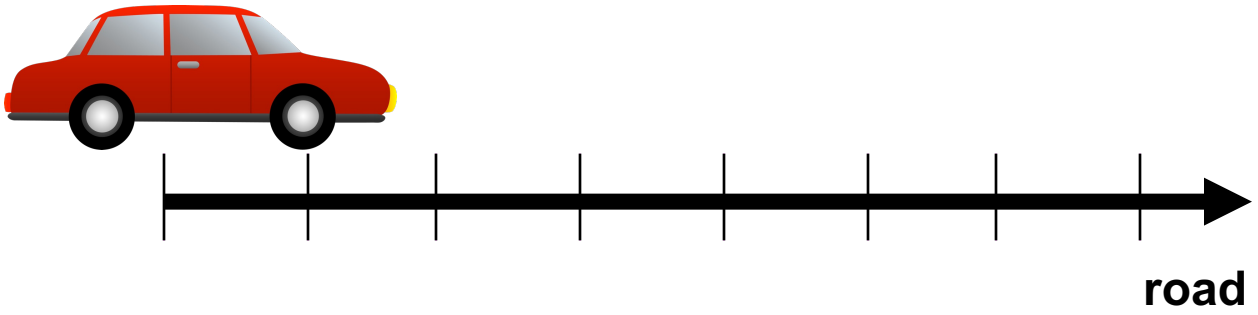
$v_0 = ?$

$a = ?$

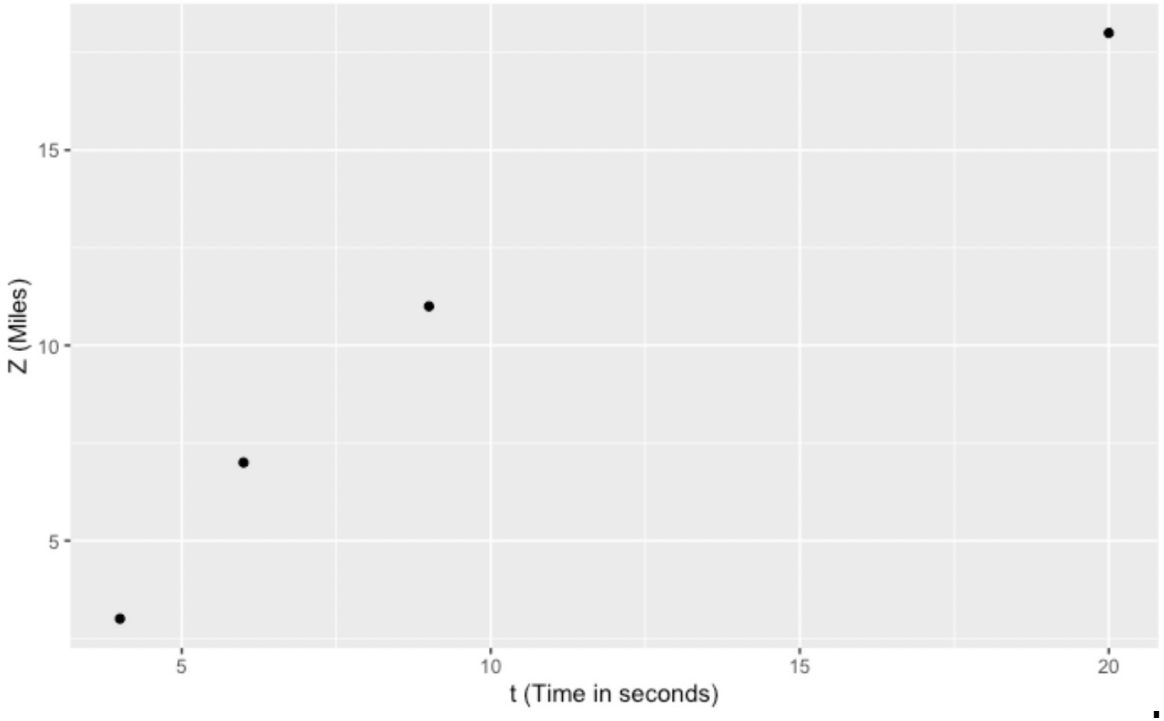
- We want to predict the initial position (x_0) and the initial velocity (v_0)
- We don't know the initial position or the velocity
- Assume that the GPS of the car is very noisy

Motivation for Linear Regression

- a is constant, x_0 = ?, v_0 = ?

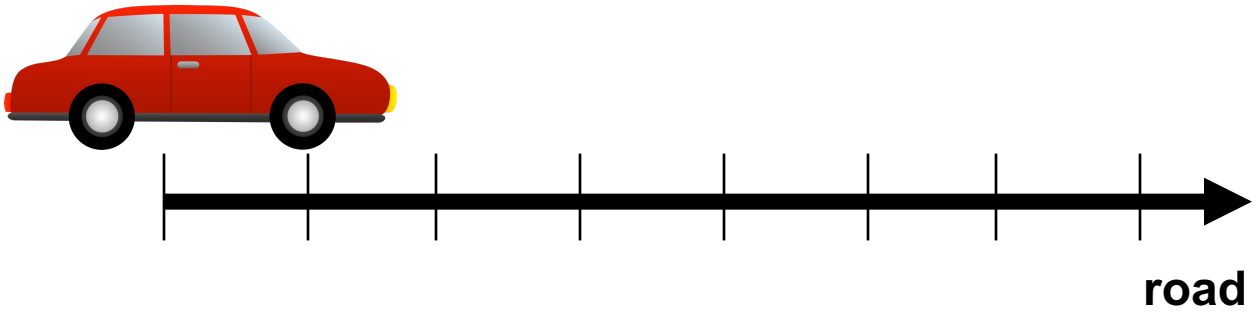


z (position)	t (time)
3	4s
7	6s
11	9s
18	20s



Motivation for Linear Regression

- a is constant, x_0 = ?, v_0 = ?



z (position)	t (time)
3	4s
7	6s
11	9s
18	20s

- We can postulate the following model for the movement of the car:

$$z = x_0 + v_0 t + \frac{at^2}{2} + \epsilon \qquad \epsilon \sim N(0, \sigma^2)$$

$$= \begin{pmatrix} 1 & t & t^2/2 \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ v_0 \\ a \end{pmatrix} + \epsilon$$

$$= H \cdot x + \epsilon$$

MODEL MATRIX → DESIGN MATRIX

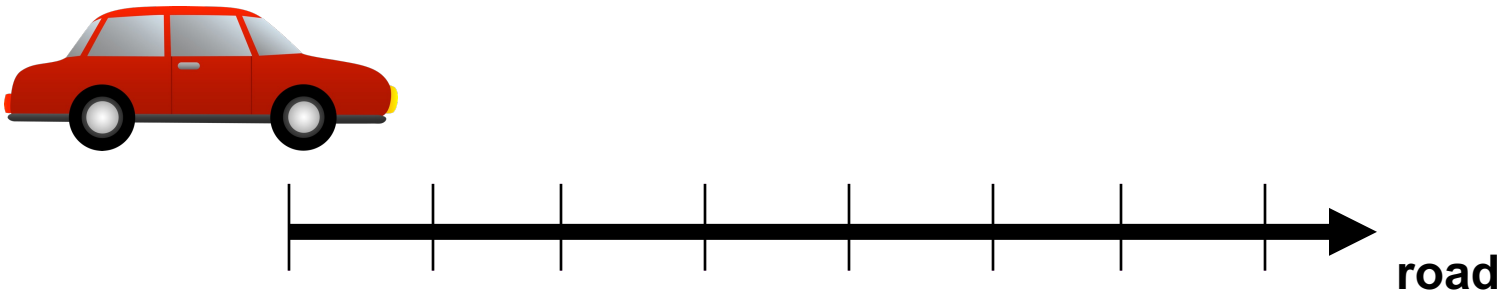
Notation:

- ***H*** **the model matrix**
- ***x*** **is the state vector**
- ***z*** **is the observation vector**
- ***epsilon*** **is the noise**

Motivation for Linear Regression Example

a is constant, x_0 = ?, v_0 = ?

$$\epsilon \sim N(0, \sigma^2)$$



- Suppose that we use this **noisy GPS** and find the following observations for the position of the car

z (position)	t (time)
3	4s
7	6s
11	9s
18	20s

$$3 = x_0 + v_0(4) + \frac{a(4)^2}{2} + \epsilon_1$$

$$7 = x_0 + v_0(6) + \frac{a(6)^2}{2} + \epsilon_2$$

$$11 = x_0 + v_0(9) + \frac{a(9)^2}{2} + \epsilon_3$$

$$18 = x_0 + v_0(20) + \frac{a(20)^2}{2} + \epsilon_4$$

$$e_i \sim N(0, \sigma^2)$$

Linear Regression Problem

a is constant,
x_0 = ?, v_0 = ?

- Fitting the **linear least squares regression** model is solved if we solve the following problem:

- Find x_0, v_0 and a that minimize L(x), called the ***loss function***, the sum of the squares of the differences between observations and predictions

$$L(x_0, v_0, a) = RSS(x_0, v_0, a)$$

$$= \left(3 - \left(x_0 + v_0(4) + \frac{a(4)^2}{2}\right)\right)^2 + \left(7 - \left(x_0 + v_0(6) + \frac{a(6)^2}{2}\right)\right)^2$$

$$+ \left(\underbrace{11}_{\text{obs}} - \underbrace{\left(x_0 + v_0(9) + \frac{a(9)^2}{2}\right)}_{\text{prediction}}\right)^2 + \left(\underbrace{18}_{\text{obs}} - \underbrace{\left(x_0 + v_0(20) + \frac{a(20)^2}{2}\right)}_{\text{prediction}}\right)^2$$

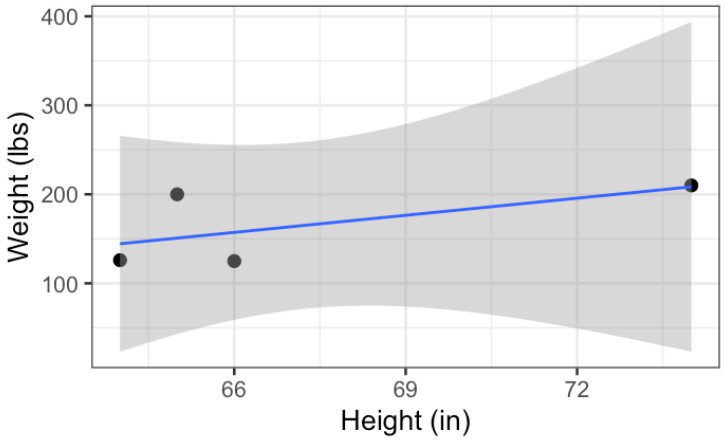
Exercise

- Given this dataset, write the loss function for the following linear least squares regression model:

	height	weight
1	74	210
2	66	125
3	64	126
4	65	200

$$\text{weight} = c_0 + c_1 \cdot \text{height}$$

$$L(c_0, c_1) = RSS(c_0, c_1) = \sum_{i=1}^{4} \text{observed}_i - \text{predicted}_i$$



Linear Regression Problem (Matrix Form)

- Now, we are going to re-write the previous optimization problem using matrix notation.

Review: Norms and Euclidean Product

- Remember that for any vector x

$$\|x\|_2^2 = x^T \cdot x$$

- For example, x = (1, 3, 5)^T, we know that

$$\|x\|_2^2 = 1^2 + 3^2 + 5^2 = \begin{pmatrix} 1 & 3 & 5 \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 3 \\ 5 \end{pmatrix} = x^T \cdot x$$

Review: Transpose of a Vector or Matrix

- Remember that for any vector

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

- We have that $x^T = \begin{pmatrix} x_1 & x_2 & x_3 \end{pmatrix}$

- And that for any matrix $H = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \end{pmatrix}$

- We have that

$$H^T = \begin{pmatrix} h_{11} & h_{21} \\ h_{12} & h_{22} \\ h_{13} & h_{23} \end{pmatrix}$$

Linear Regression Problem (Matrix Form)

- Let's write the same problem as before, but in matrix form.
- We had the data:

$$3 = x_0 + v_0(4) + \frac{a(4)^2}{2} + \epsilon_1$$

$$7 = x_0 + v_0(6) + \frac{a(6)^2}{2} + \epsilon_2$$

$$11 = x_0 + v_0(9) + \frac{a(9)^2}{2} + \epsilon_3$$

$$18 = x_0 + v_0(20) + \frac{a(20)^2}{2} + \epsilon_4$$

$$\epsilon_i \sim N(0, \sigma^2)$$

a is constant, x_0 = ?, v_0 = ?

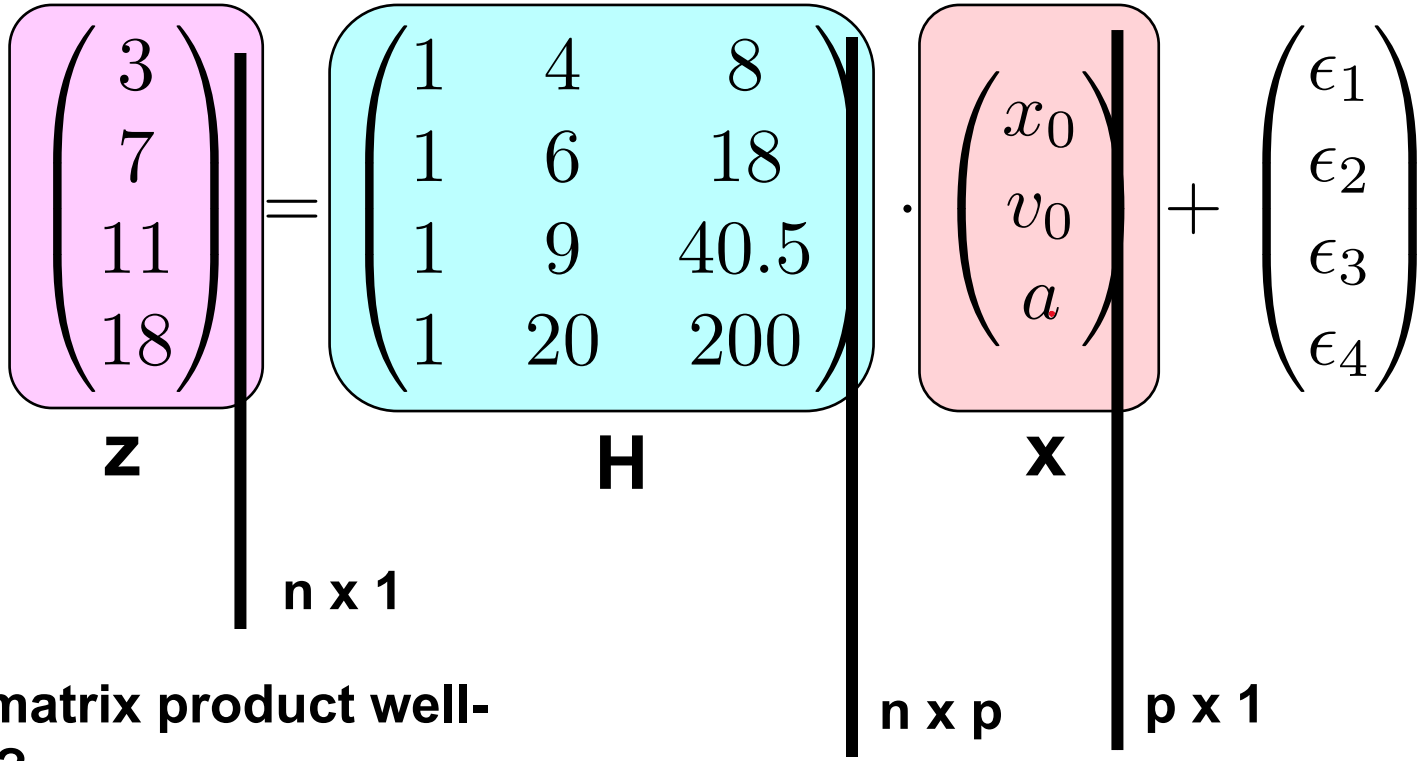
- Which we can write as:

$$\begin{pmatrix} 3 \\ 7 \\ 11 \\ 18 \end{pmatrix} = \begin{pmatrix} 1 & 4 & 8 \\ 1 & 6 & 18 \\ 1 & 9 & 40.5 \\ 1 & 20 & 200 \end{pmatrix} \cdot \begin{pmatrix} x_0 \\ v_0 \\ a \end{pmatrix} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \end{pmatrix}$$

z **H** **x**

Linear Regression Problem (Matrix Form)

- Let's check if equation matrix makes sense.
- Suppose that our dataset has n rows. Here n = 4.
- Call p the number of predictors. Here p = 3.



Is this matrix product well-formed?

Linear Regression Problem (Matrix Form)

- Given this system,

a is constant,
x_0 = ?, v_0 = ?

$$\epsilon_i \sim N(0, \sigma^2)$$

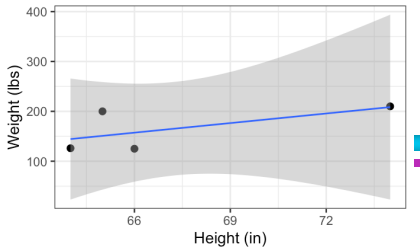
$$\underbrace{\begin{pmatrix} 3 \\ 7 \\ 11 \\ 18 \end{pmatrix}}_{\mathbf{z}} = \underbrace{\begin{pmatrix} 1 & 4 & 8 \\ 1 & 6 & 18 \\ 1 & 9 & 40.5 \\ 1 & 20 & 200 \end{pmatrix}}_{\mathbf{H}} \cdot \underbrace{\begin{pmatrix} x_0 \\ v_0 \\ a \end{pmatrix}}_{\mathbf{x}} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \end{pmatrix}$$

Find x such that it minimizes the following loss or penalty function:

$$L(x) = RSS(x) = \frac{1}{2}\|z - H \cdot x\|_2^2$$

$$= \frac{1}{2}(z - H \cdot x)^T (z - H \cdot x)$$

Exercise



- Write the expanded loss function for the following linear least squares regression model in matrix form:

$$\text{weight} = c_0 + c_1 \cdot \text{height}$$

	height	weight
1	74	210
2	66	125
3	64	126
4	65	200

$$L(x) = (z - H \cdot x)^T (z - H \cdot x)$$

$$= \frac{1}{2} \left(\begin{pmatrix} 210 \\ 125 \\ 126 \\ 200 \end{pmatrix} - \begin{pmatrix} 1 & 74 \\ 1 & 66 \\ 1 & 64 \\ 1 & 65 \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} \right)^T \cdot \left(\begin{pmatrix} 210 \\ 125 \\ 126 \\ 200 \end{pmatrix} - \begin{pmatrix} 1 & 74 \\ 1 & 66 \\ 1 & 64 \\ 1 & 65 \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \end{pmatrix} \right)$$

Outline

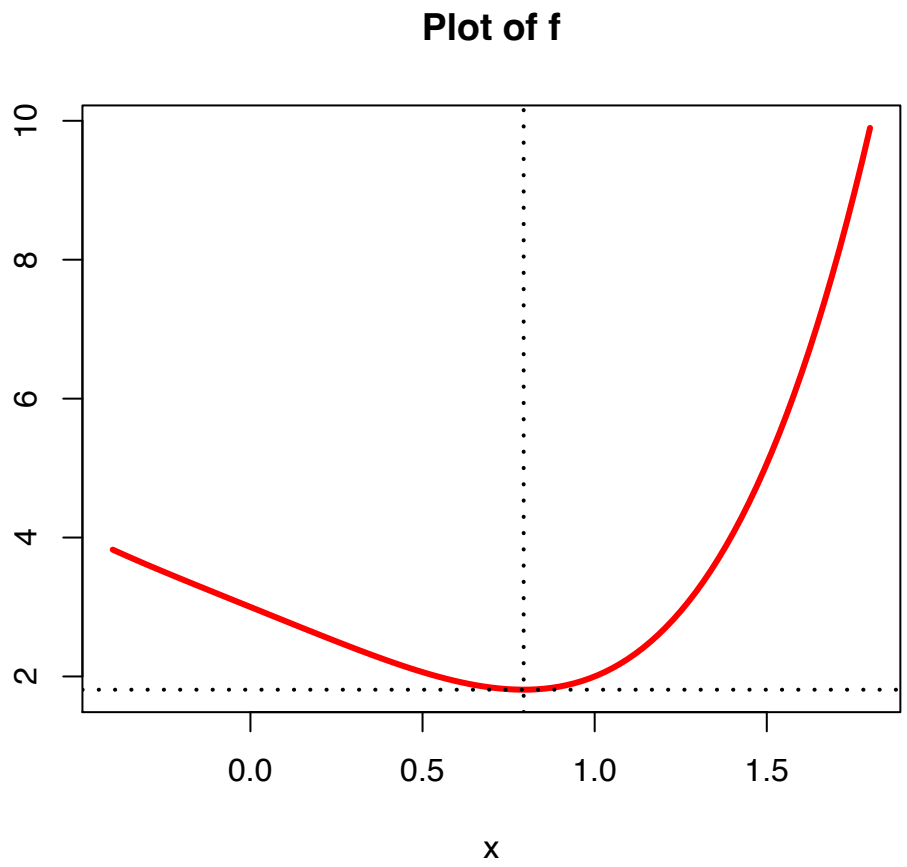
- Introduction
- Linear Regression
- Normal Equations
- Gradient Descent
- Overfitting, Underfitting, and Regularization
- Linear Regression in R

Review: Calculus

- Suppose we have the scalar-valued function of a scalar

$$f : \mathbb{R} \longrightarrow \mathbb{R}$$
$$f(x) = x^4 - 2x + 3$$

- We wish to find where the minimum of f is attained
- What do we do?



Review: Calculus

$f: \mathbb{R} \rightarrow \mathbb{R}$ (SCALAR. → SCALAR)

- What do we do?

$$f(x) = x^4 - 2x + 3$$

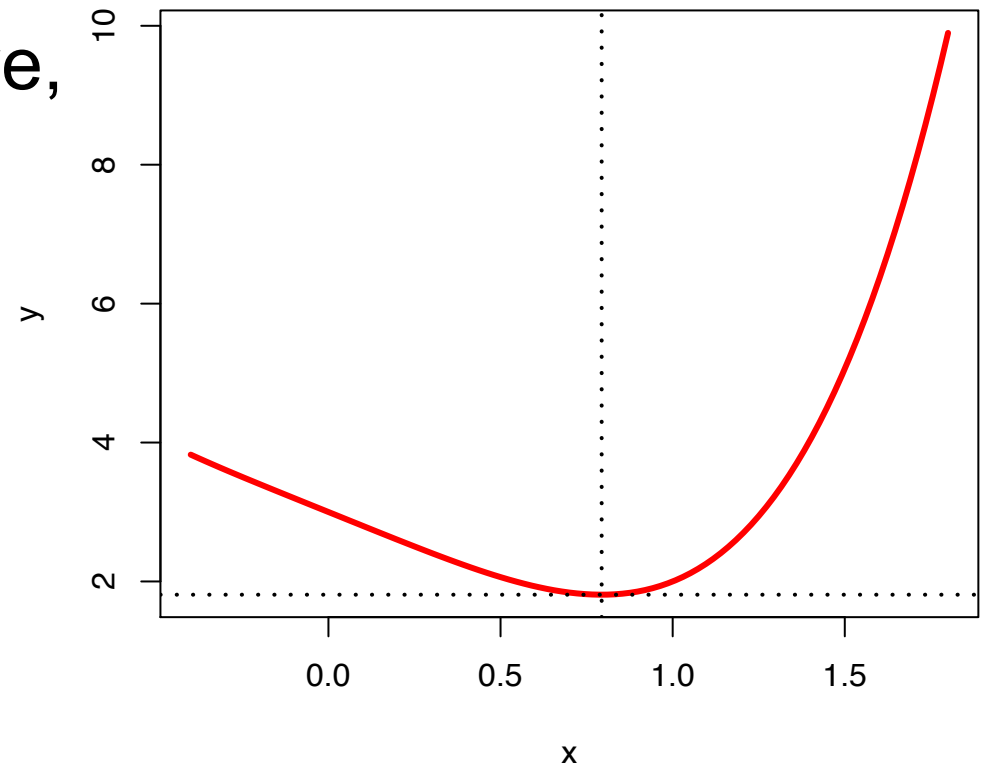
- Compute the derivative, set it to 0, solve for x

$$\begin{aligned}\frac{d}{dx} f(x) &= \frac{d}{dx}(x^4 - 2x + 3) \\ &= 4x^3 - 2 \\ &= 0\end{aligned}$$

- Therefore, the min. is attained at

$$4x^3 - 2 = 0$$

$$x = \sqrt[3]{2/4} = 0.7937005$$

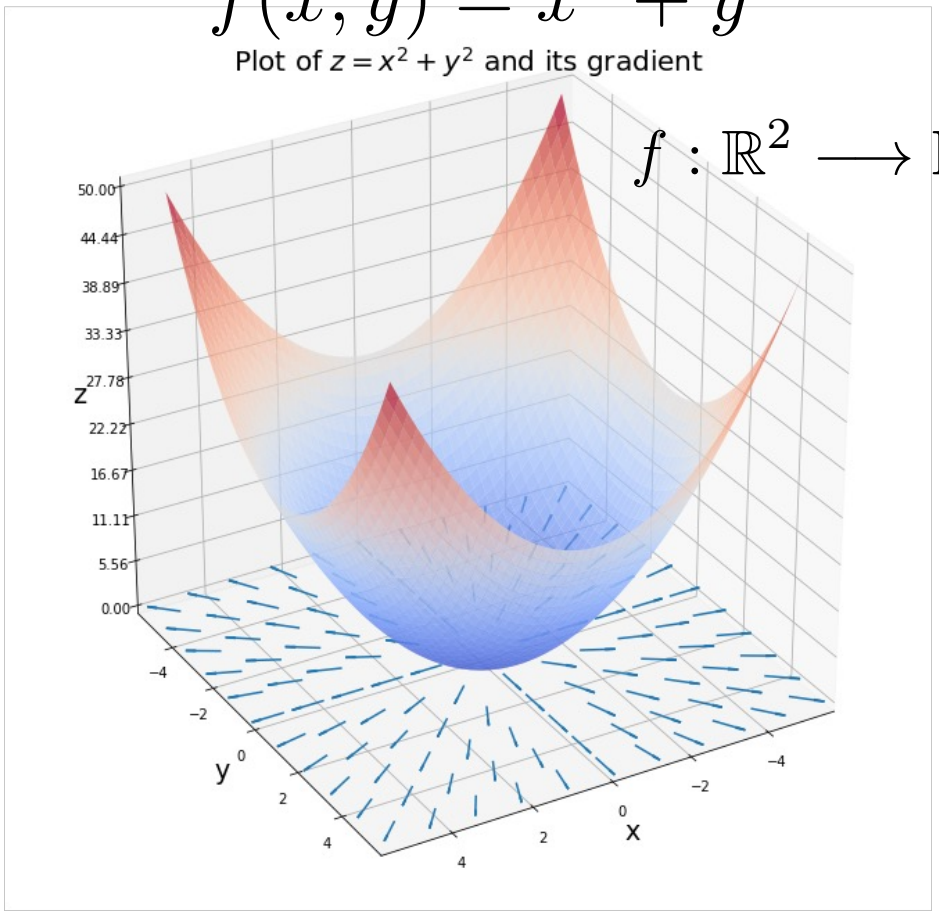


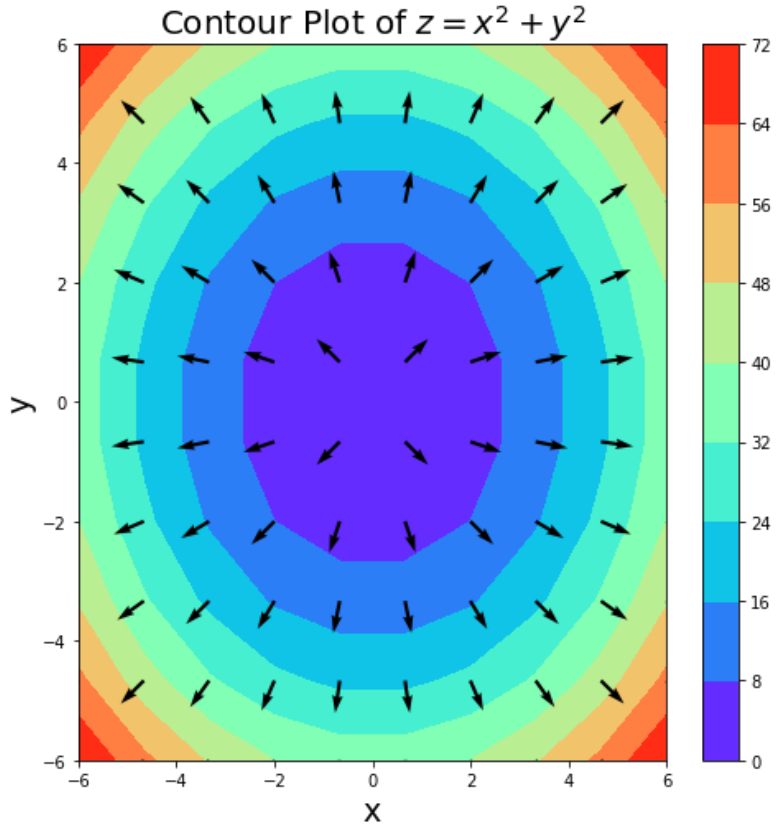
Gradient of a Scalar-Valued Function

- Remember the gradient is

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right)$$

$$f(x, y) = x^2 + y^2$$





Gradient of a Scalar-Valued Function

- Remember the gradient is $\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right)$

- Example: Suppose that

$$f(x, y) = \log(x)y + y^2$$

Let's compute the gradient of f:

$$\begin{aligned} \nabla f(x, y) &= \left(\frac{\partial f}{\partial x} \left(\log(x)y + y^2 \right), \frac{\partial f}{\partial y} \left(\log(x)y + y^2 \right) \right) \\ &= \left(\frac{y}{x}, \log(x) + 2y \right) \end{aligned}$$

Loss Function

Scalar Form

$$L(x_0, v_0, a) = RSS(x_0, v_0, a)$$

$$= \left(3 - \left(x_0 + v_0(4) + \frac{a(4)^2}{2}\right)\right)^2 + \left(7 - \left(x_0 + v_0(6) + \frac{a(6)^2}{2}\right)\right)^2$$

$$+ \left(\underbrace{11}_{\text{obs}} - \underbrace{\left(x_0 + v_0(9) + \frac{a(9)^2}{2}\right)}_{\text{prediction}}\right)^2 + \left(\underbrace{18}_{\text{obs}} - \underbrace{\left(x_0 + v_0(20) + \frac{a(20)^2}{2}\right)}_{\text{prediction}}\right)^2$$

Matrix Form $L(x) = (z - H \cdot x)^T (z - H \cdot x)$

$$\underbrace{\begin{pmatrix} 3 \\ 7 \\ 11 \\ 18 \end{pmatrix}}_{\mathbf{z}} = \underbrace{\begin{pmatrix} 1 & 4 & 8 \\ 1 & 6 & 18 \\ 1 & 9 & 40.5 \\ 1 & 20 & 200 \end{pmatrix}}_{\mathbf{H}} \cdot \underbrace{\begin{pmatrix} x_0 \\ v_0 \\ a \end{pmatrix}}_{\mathbf{x}} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \end{pmatrix}$$

Idea of the Derivation of the Normal Equations

- We are going to take derivatives with respect to x of

$$L(x) = (z - H \cdot x)^T (z - H \cdot x)$$

- But x is here a vector, H is a matrix. So, we take the gradient instead.

$$\begin{aligned} \nabla L(x) &= \nabla \left(\frac{1}{2} (z - H \cdot x)^T (z - H \cdot x) \right) \\ &= H^T \cdot H \cdot x - H^T \cdot z \\ &= 0 \end{aligned}$$

Normal Equations

- To fit a linear regression model, you only need to solve for x in the normal equations are

$$H^T \cdot H \cdot x = H^T z$$

- We know H and z, we wish to find x

Using the Normal Equations

- If we are solving by hand, it is fine to find the solution by doing

$$x = (H^T \cdot H)^{-1} \cdot H^T z$$

- But for serious purposes, we never do that. We instead use Gaussian Elimination:

$$x = (H^T \cdot H) \backslash\ (H^T z)$$

In R to find x we can do:
solve(t(H) %*% H, t(H) %*% z)

Dimensions of the Normal Equations

- Are the normal equations well-formed?

$$\underset{p \times n}{H^T} \cdot \underset{n \times p}{H} \cdot \underset{p \times 1}{x} = \underset{p \times n}{H^T} \underset{n \times 1}{z}$$

$$\underbrace{\begin{pmatrix} 3 \\ 7 \\ 11 \\ 18 \end{pmatrix}}_{z} = \underbrace{\begin{pmatrix} 1 & 4 & 8 \\ 1 & 6 & 18 \\ 1 & 9 & 40.5 \\ 1 & 20 & 200 \end{pmatrix}}_{H} \cdot \underbrace{\begin{pmatrix} x_0 \\ v_0 \\ a \end{pmatrix}}_{x} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \end{pmatrix}$$

Final Comments About the Normal Equation

- The normal equations are

$$H^T \cdot H \cdot x = H^T z$$

This makes some assumptions about the existence of the solution x

- This system has a solution if:
  - No predictors are linearly dependent on 1 or more other predictors
  - If there are more observations than predictors, i.e., the number of rows of H is larger than the number of columns. If this is not the case, we can use regularization.

Solving the Normal Equations of the Example

$$H = \begin{pmatrix} 1 & 4 & 8 \\ 1 & 6 & 18 \\ 1 & 9 & 40.5 \\ 1 & 20 & 200 \end{pmatrix} \qquad z = \begin{pmatrix} 3 \\ 7 \\ 11 \\ 18 \end{pmatrix}$$

- Finding x by hand would give

$$x = (H^T \cdot H)^{-1} \cdot H^T z$$

$$\begin{aligned} x &= \left(\begin{pmatrix} 1 & 4 & 8 \\ 1 & 6 & 18 \\ 1 & 9 & 40.5 \\ 1 & 20 & 200 \end{pmatrix}^T \cdot \begin{pmatrix} 1 & 4 & 8 \\ 1 & 6 & 18 \\ 1 & 9 & 40.5 \\ 1 & 20 & 200 \end{pmatrix} \right)^{-1} \cdot \begin{pmatrix} 1 & 4 & 8 \\ 1 & 6 & 18 \\ 1 & 9 & 40.5 \\ 1 & 20 & 200 \end{pmatrix}^T \cdot \begin{pmatrix} 3 \\ 7 \\ 11 \\ 18 \end{pmatrix} \\ &= \left(\begin{pmatrix} 1 & 1 & 1 & 1 \\ 4 & 6 & 9 & 20 \\ 8 & 18 & 40.5 & 200 \end{pmatrix} \cdot \begin{pmatrix} 1 & 4 & 8 \\ 1 & 6 & 18 \\ 1 & 9 & 40.5 \\ 1 & 20 & 200 \end{pmatrix} \right)^{-1} \cdot \begin{pmatrix} 1 & 4 & 8 \\ 1 & 6 & 18 \\ 1 & 9 & 40.5 \\ 1 & 20 & 200 \end{pmatrix}^T \cdot \begin{pmatrix} 3 \\ 7 \\ 11 \\ 18 \end{pmatrix} \\ &= \begin{pmatrix} -5.44 \\ 2.39 \\ -0.12 \end{pmatrix} = \begin{pmatrix} x_0 \\ v_0 \\ a/2 \end{pmatrix} \end{aligned}$$

Design Matrix

DESIGN MATRIX (H)

- DEPENDS ON THE MODEL AND THE DATASET.

AGE	HEIGHT	INCOME (Ks)
17	66 in	20
20	70 in	31
25	72 in	50.

EXAMPLE 1:

INCOME $= C_0 + C_1 \text{age} + C_2 \text{height}$

$$H = \begin{pmatrix} 1 & 17 & 66 \\ 1 & 20 & 70 \\ 1 & 25 & 72 \end{pmatrix} \qquad Z = \begin{pmatrix} 20 \\ 31 \\ 50 \end{pmatrix}$$

PARAMS $= (C_0, C_1, C_2)$

Design Matrix

DESIGN MATRIX (H)

- DEPENDS ON THE MODEL AND THE DATASET.

AGE	HEIGHT	INCOME (Ks)
17	66 in	20
20	70 in	31
25	72 in	50.

EXAMPLE 1:

INCOME $= C_0 + C_1 \text{ age} + C_2 \text{ height}$

$$H = \begin{pmatrix} 1 & 17 & 66 \\ 1 & 20 & 70 \\ 1 & 25 & 72 \end{pmatrix} \qquad Z = \begin{pmatrix} 20 \\ 31 \\ 50 \end{pmatrix}$$

PARAMS $= (C_0, C_1, C_2)$

Write the loss function in matrix form

$$L(x) = (z - H \cdot x)^T (z - H \cdot x)$$

Design Matrix

DESIGN MATRIX (H)

- DEPENDS ON THE MODEL AND THE DATASET.

AGE	HEIGHT	INCOME (Ks)
17	66 in	20
20	70 in	31
25	72 in	50.

EXAMPLE 1:

INCOME $= c_0 + c_1 \text{ age} + c_2 \text{ height}$

$$H = \begin{pmatrix} 1 & 17 & 66 \\ 1 & 20 & 70 \\ 1 & 25 & 72 \end{pmatrix} \qquad z = \begin{pmatrix} 20 \\ 31 \\ 50 \end{pmatrix}$$

PARAMS $= (c_0, c_1, c_2)$

Write the loss function in matrix form

$$L(x) = (z - H \cdot x)^T (z - H \cdot x)$$

Minimize the Loss function L(c0, c1, c2)

$$H^T \cdot H \cdot x = H^T z \qquad x = (H^T \cdot H)^{-1} \cdot H^T z$$

Design Matrix

DESIGN MATRIX (H)

- DEPENDS ON THE MODEL AND THE DATASET.

AGE	HEIGHT	INCOME (Ks)
17	66 in	20
20	70 in	31
25	72 in	50.

EXAMPLE 1:

INCOME $= c_0 + c_1 \text{ age} + c_2 \text{ height}$

$$H = \begin{pmatrix} 1 & 17 & 66 \\ 1 & 20 & 70 \\ 1 & 25 & 72 \end{pmatrix} \qquad z = \begin{pmatrix} 20 \\ 31 \\ 50 \end{pmatrix}$$

PARAMS $= (c_0, c_1, c_2)$

Write the loss function in matrix form

$$L(x) = (z - H \cdot x)^T (z - H \cdot x)$$

Minimize the Loss function L(c0, c1, c2)

$$x = (H^T \cdot H)^{-1} \cdot H^T z$$

$$x = \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix} \begin{pmatrix} -36.14 \\ 3.86 \\ -0.14 \end{pmatrix}$$

Design Matrix

DESIGN MATRIX (H)

- DEPENDS ON THE MODEL AND THE DATASET.

AGE	HEIGHT	INCOME (Ks)
17	66 in	20
20	70 in	31
25	72 in	50.

EXAMPLE 2:

$INCOME = c_0 + c_1\, age^2 + c_2\, height$

$PARAMS = (c_0, c_1, c_2)$

$$H = \begin{pmatrix} 1 & 17^2 & 66 \\ 1 & 20^2 & 70 \\ 1 & 25^2 & 72 \end{pmatrix} \qquad Z = \begin{pmatrix} 20 \\ 31 \\ 50 \end{pmatrix}$$

$PARAMS = (c_0, c_1, c_2)$

Write the loss function in matrix form

$$L(x) = (z - H \cdot x)^T (z - H \cdot x)$$

Minimize the Loss function L(c0, c1, c2)

$$H^T \cdot H \cdot x = H^T z$$

Design Matrix

DESIGN MATRIX (H)

- DEPENDS ON THE MODEL AND THE DATASET.

AGE	HEIGHT	INCOME (Ks)
17	66 in	20
20	70 in	31
25	72 in	50.

EXAMPLE 2:

$INCOME = c_0 + c_1\, age^2 + c_2\, height$

$PARAMS = (c_0, c_1, c_2)$

$$H = \begin{pmatrix} 1 & 17^2 & 66 \\ 1 & 20^2 & 70 \\ 1 & 25^2 & 72 \end{pmatrix} \qquad Z = \begin{pmatrix} 20 \\ 31 \\ 50 \end{pmatrix}$$

$PARAMS = (c_0, c_1, c_2)$

Write the loss function in matrix form

$$L(x) = (z - H \cdot x)^T (z - H \cdot x)$$

Minimize the Loss function L(c0, c1, c2)

$$H^T \cdot H \cdot x = H^T z \qquad x = (H^T \cdot H)^{-1} \cdot H^T z$$

Design Matrix

EXAMPLE 3

$$\text{INCOME} = C_0 + C_1 \cdot (\text{age} * \text{height}) + C_2\, \text{age}^2 + C_3 \cdot \log(\text{height})$$

$$H = \begin{pmatrix} 1 & 17.66 & 17^2 & \log(66) \\ 1 & 20.70 & 20^2 & \log(70) \\ 1 & 25.72 & 25^2 & \log(72) \end{pmatrix} \qquad Z = \begin{pmatrix} 20 \\ 31 \\ 50 \end{pmatrix}$$

$$\text{PARAMS} = (C_0, C_1, C_2, C_3)$$

$$Z = \text{INCOME} = H \times$$

$$= \begin{pmatrix} 1 & 17.66 & 17^2 & \log(66) \\ 1 & 20.70 & 20^2 & \log(70) \\ 1 & 25.72 & 25^2 & \log(72) \end{pmatrix} \begin{pmatrix} C_0 \\ C_1 \\ C_2 \\ C_3 \end{pmatrix}$$

EXAMPLE 4

$$\text{INCOME} = C_0 + C_1\, \text{age} + C_2\, \text{AGE}^3 + C_3\, \text{AGE}^{1000} + C_4 \log(\text{height} * \text{age}^2)$$

Dealing with Categorical Predictors

- So far, our predictors have been quantitative, but we can also have categorical predictors.

Education Level	Capital Gains	Age	Height	Income
High School	N	43	66	20,000
Grad School	Y	31	70	31,000
College	Y	58	72	50,000
Grad School	N	36	74	46,000

Dealing with Categorical Predictors

Education Level	Capital Gains	Age	Height	Income
High School	N	43	66	20,000
Grad School	Y	31	70	31,000
College	Y	58	72	50,000
Grad School	N	36	74	46,000

$$\text{Model 1:}\quad \text{income} = c_0 + c_1 \cdot \text{Capital Gains} + c_2 \cdot \text{age}$$

$$\begin{pmatrix} 1 & 0 & 43 \\ 1 & 1 & 31 \\ 1 & 1 & 58 \\ 1 & 0 & 36 \end{pmatrix}$$

Dealing with Categorical Predictors

Education Level	Capital Gains	Age	Height	Income
High School	N	43	66	20,000
Grad School	Y	31	70	31,000
College	Y	58	72	50,000
Grad School	N	36	74	46,000

Model 2: $\text{income} = c_0 + c_1 \cdot \text{Education Level} + c_2 \cdot \text{age}$

$$\begin{pmatrix} 1 & 1 & 0 & 43 \\ 1 & 0 & 0 & 31 \\ 1 & 0 & 1 & 58 \\ 1 & 0 & 0 & 36 \end{pmatrix}$$

Dealing with Categorical Predictors

- Suppose we want to predict the average horsepower for a car and we only now the car's transmission type.
- We define

$$t_i \equiv \begin{cases} 1 & \text{if the ith car is automatic} \\ 0 & \text{if not} \end{cases}$$

- Then the model becomes

$$hp_i = \alpha_0 + \alpha_1 t_i + \epsilon_i$$

- Where hp_i is the power of the ith car in the dataset

Dealing with Categorical Predictors

- But since that categorical attribute is binary, it is really two separate models

$$hp_i = \alpha_0 + \alpha_1 t_i + \epsilon_i = \begin{cases} \alpha_0 + \alpha_1 + \epsilon_i & \text{if the ith car is automatic} \\ \alpha_0 + \epsilon_i & \text{if not} \end{cases}$$

- Where alpha0 is the average horsepower of non-automatic cars
- Alpha1 is the difference between the average horsepower of automatic cars and the average for non-automatic cars.

Outline

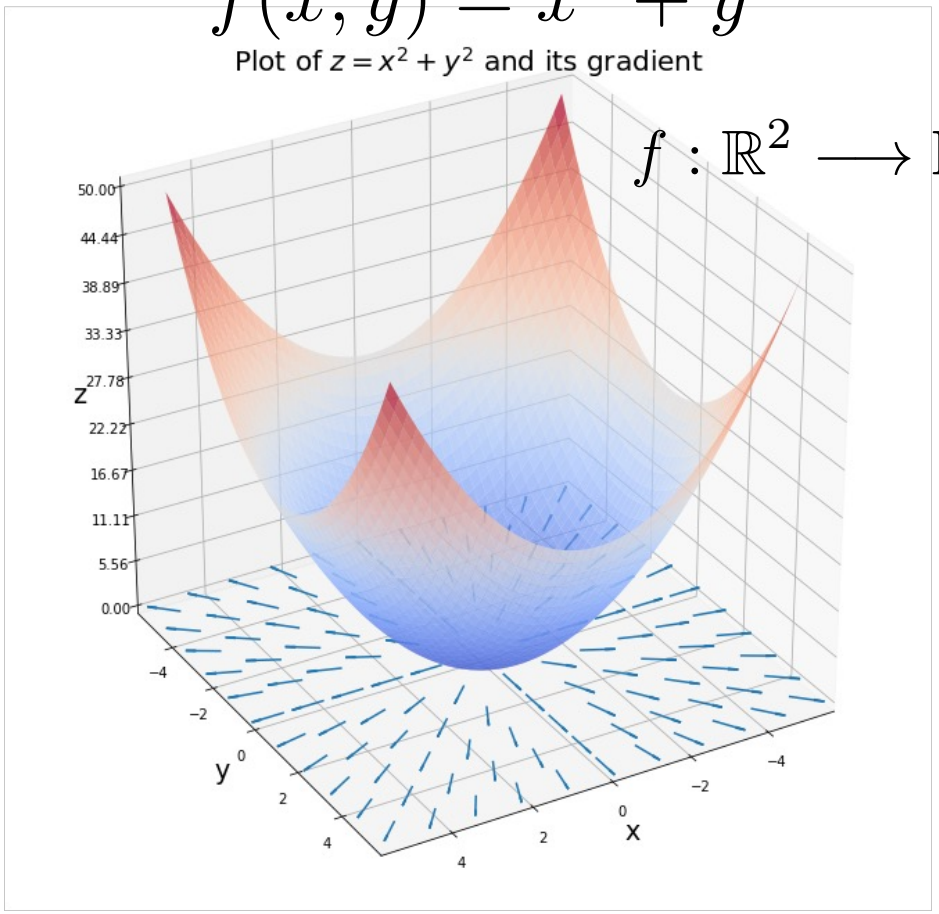
- Introduction
- Linear Regression
- Normal Equations
- Gradient Descent
- Overfitting, Underfitting, and Regularization
- Linear Regression in R

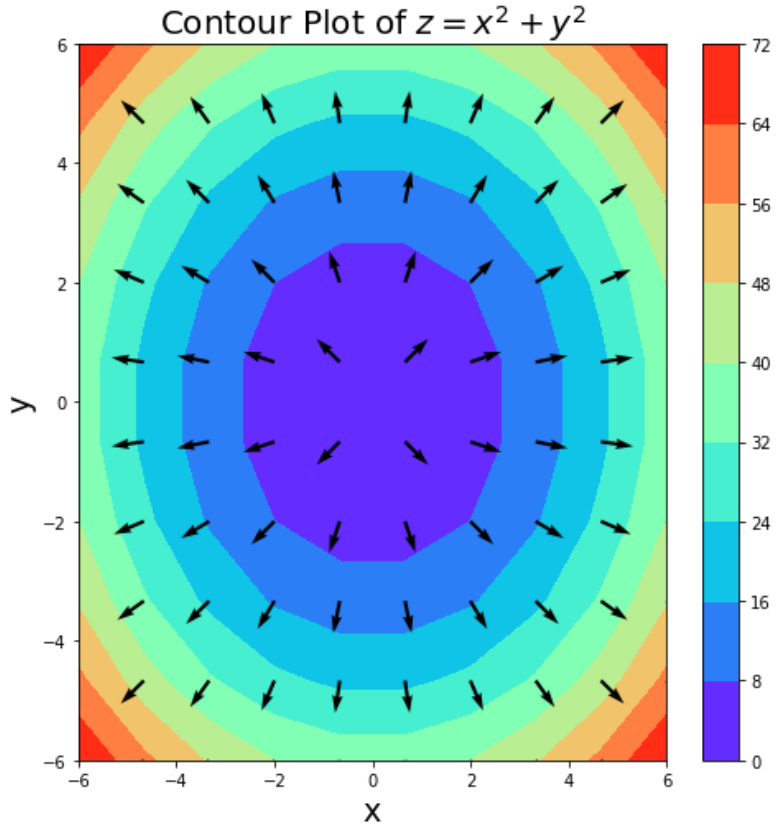
Gradient of a Scalar-Valued Function

- Remember the gradient is

$$\nabla f = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n} \right)$$

$$f(x, y) = x^2 + y^2$$





(General) Gradient Descent Algorithm

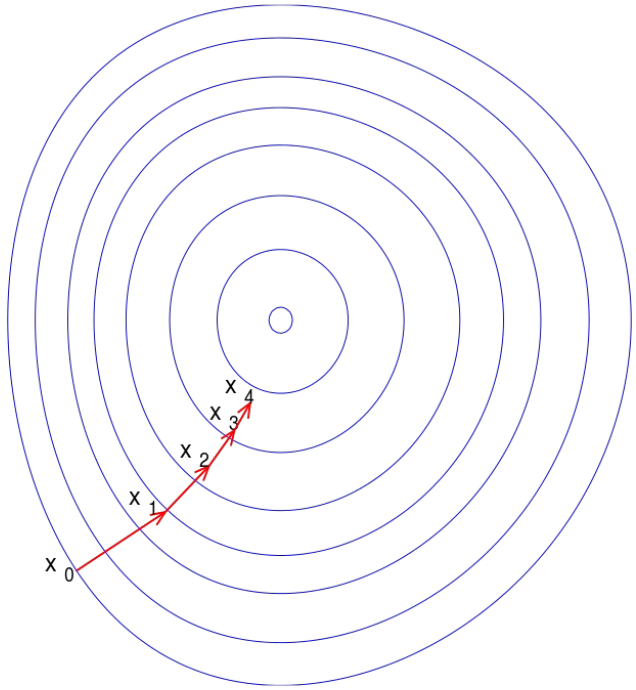
- **Input**: $f(x) : \mathbb{R}^n \longrightarrow \mathbb{R}, x_0 \in \mathbb{R}^n, \lambda > 0 \in \mathbb{R}, \epsilon > 0 \in \mathbb{R}$ (Initial guess; Learning rate; Tolerance)
- **Output**: A local minimum of f

SCALAR 0.000001 10^{-12}

- For k = 0, 1, 2, …:
  - Compute the gradient ∇f
  - Take the step

$$x_{k+1} = x_k - \lambda(\nabla f)(x_k)$$

  - Test for convergence. If yes, then exit.



The gradient is always normal to the level sets

Gradient Descent: Step-by-Step

$$f : \mathbb{R}^2 \longrightarrow \mathbb{R} \qquad \nabla f : \mathbb{R}^2 \longrightarrow \mathbb{R}^2$$

$$f(x, y) = x^2 + y^2 \qquad \nabla f(x, y) = (2x, 2y)$$

$$x_0 = (1, 1) \in \mathbb{R}^2 \qquad \epsilon = 10^{-12} \qquad \lambda = 0.1$$

Iteration 1:

1) Compute the gradient at x_0 = (1,1)

$$\nabla f(1, 1) = (2 \cdot 1, 2 \cdot 1) = (2, 2)$$

2) Take the step

$$\begin{aligned} x_1 &= x_0 - \lambda \nabla f(x_0) \\ &= (1, 1) - (0.1) \cdot (2, 2) \\ &= (0.8, 0.8) \end{aligned}$$

Gradient Descent: Step-by-Step

3) Test for convergence

$$\|x_1 - x_0\|_2 < \epsilon = 10^{-12}$$

$$\begin{aligned}\|x_1 - x_0\|_2 &= \|(0.8, 0.8) - (1, 1)\|_2 \\ &= \|(-0.2, -0.2)\|_2 = \sqrt{(-0.2)^2 + (-0.2)^2} \\ &= \sqrt{0.08} \\ &< 10^{-12}\end{aligned}$$

Iteration 2:

4) Compute the gradient at x_1 = (0.8, 0.8)

$$\begin{aligned}\nabla f(x_1) = \nabla f(0.8, 0.8) &= (2 \cdot 0.8, 2 \cdot 0.8) \\ &= (1.6, 1.6)\end{aligned}$$

Gradient Descent: Step-by-Step

5) Take the step

$$\begin{aligned} x_2 &= x_1 - \lambda \nabla f(x_1) \\ &= (0.8, 0.8) - (0.1) \cdot (1.6, 1.6) \\ &= (0.8, -0.8) \end{aligned}$$

6) Test for convergence

$$\begin{aligned} \|x_2 - x_1\|_2 &= \|(-0.8, -0.8) - (0.8, 0.8)\|_2 \\ &= \|(-1.6, -1.6)\|^2 \\ &= \sqrt{(-1.6)^2 + (-1.6)^2} \\ &< 10^{-12} = \epsilon \end{aligned}$$

Gradient Descent for Linear Regression

- **Input**: $f(x) : \mathbb{R}^n \longrightarrow \mathbb{R}, x_0 \in \mathbb{R}^n, \lambda > 0 \in \mathbb{R}, \epsilon > 0 \in \mathbb{R}$ (Initial guess; Learning rate; Tolerance)
- **Output**: A local minimum of f

- For k = 0, 1, 2, …:
  - Compute the gradient at x_k

$$\nabla f(x_k) = (H^T \cdot H) \cdot x_k - H^T z$$

  - Take the step

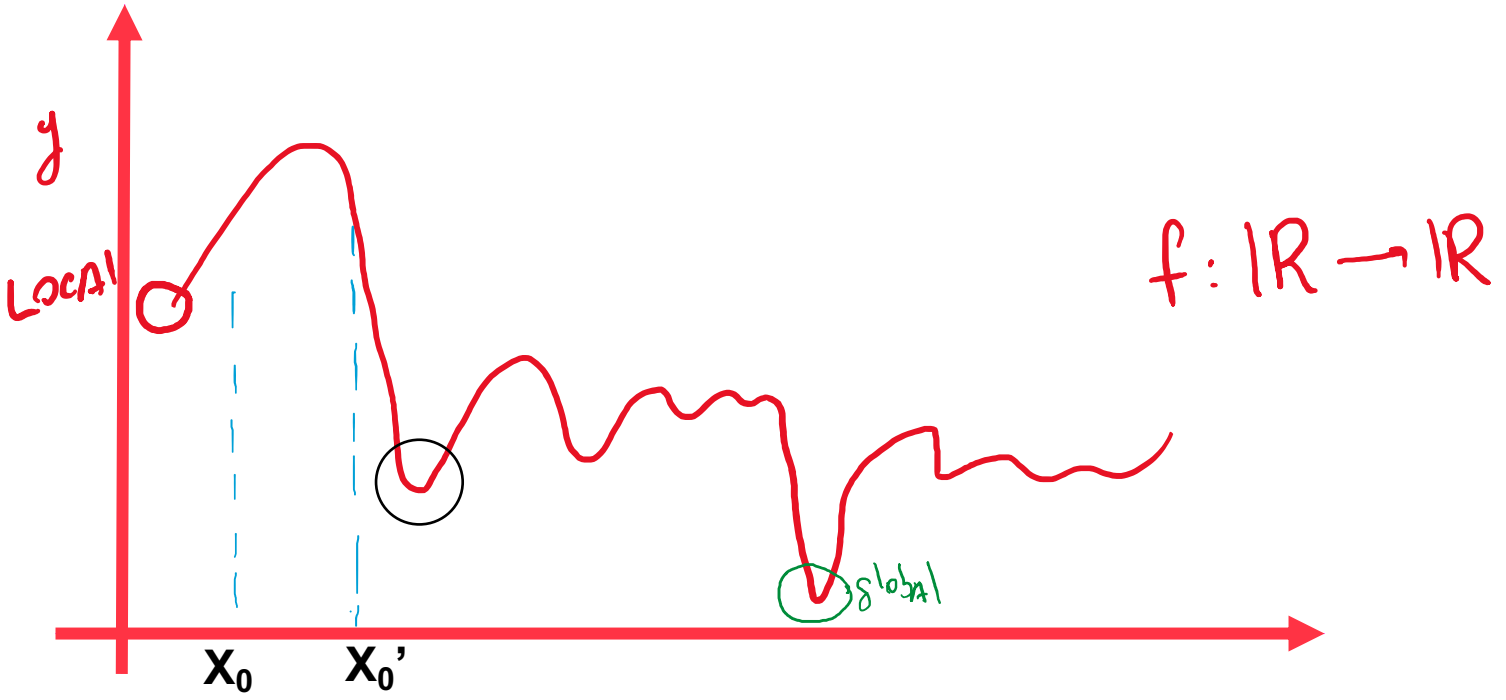
$$x_{k+1} = x_k - \lambda(\nabla f)(x_k)$$

  - Test for convergence:
    - If $\|x_{k+1} - x_k\| < \epsilon$ for a small fixed epsilon, then exit.

Gradient Descent for Linear Least Squares

```
function [x] = grad_regr(H, z, x0,lambda, max_iter)
%grad_reg Finds the linear regression coefficients
%   Finds the linear regression coefficients by using the
%   gradient method to solve H'*H*x = H'z, using a step of lambda
%   and running no more than max_iter iterations.

% Set the initial point
x        = x0;
tol      = 1e-5;
% The error can be whatever as long as it is > 1e-5
error    = 1;
% Store the errors for plotting
errors   = zeros(max_iter,1);
num_iter = 0;

while error > tol && num_iter < max_iter
    % Compute the gradient at x = H'*H*x - H'*z
    gradient = H' * H * x - H' * z;

    % Take a step
    prev = x;
    x = x - lambda * gradient;

    % Compute the error
    error = norm(prev - x, 2);%(z - H*x)' * (z - H*x);

    % Increase the number of iterations
    num_iter = num_iter + 1;

    % Store the error
    errors(num_iter) = error;
end
```

Getting Stuck at Local Minima



- If we start gradient descent at X_0 or X_0' we will likely end at a local minima.
- If we start gradient descent at X_1, we might find the global minimum

Comments about Gradient Descent for LLS

- The normal equations provide a closed-form solution (a formula) to linear least squares.
- This means that one does not need to use gradient descent to fit the linear regression model.
- However, it can be the case that solving with the normal equations can be too expensive, and that is when GD can be useful.

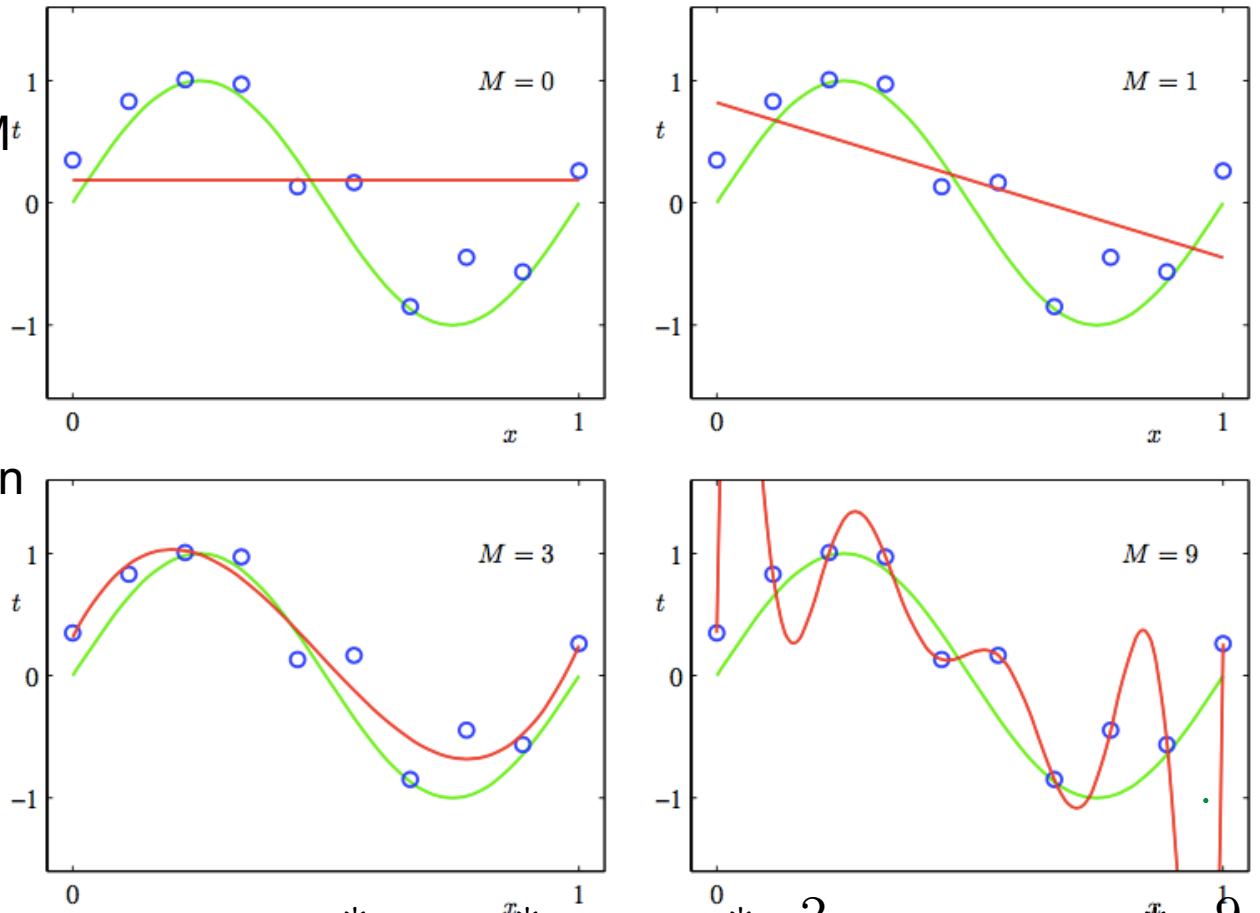
Outline

- Introduction
- Linear Regression
- Derivation of the Normal Equations
- Gradient Descent
- Overfitting, Underfitting, and Regularization
- Linear Regression in R

Underfitting and Overfitting

Plot that shows under and overfitting of polynomials of degree M (shown in red) with respect to a dataset of points in blue. The true function that generates the points is shown in green, and in unknown in practice.

Which models underfit and which models overfit? Why?



$$t = w_0^* + w_1^* x + w_2^* x^2 + \cdots + w_9^* x^9$$

Figure from Bishop, C. Pattern Recognition and Machine Learning. Springer, 2006

Dealing with Overfitting

Plot that shows under and overfitting of polynomials of degree M (shown in red) with respect to a dataset of points in blue. The true function that generates the points is shown in green, and in unknown in practice.

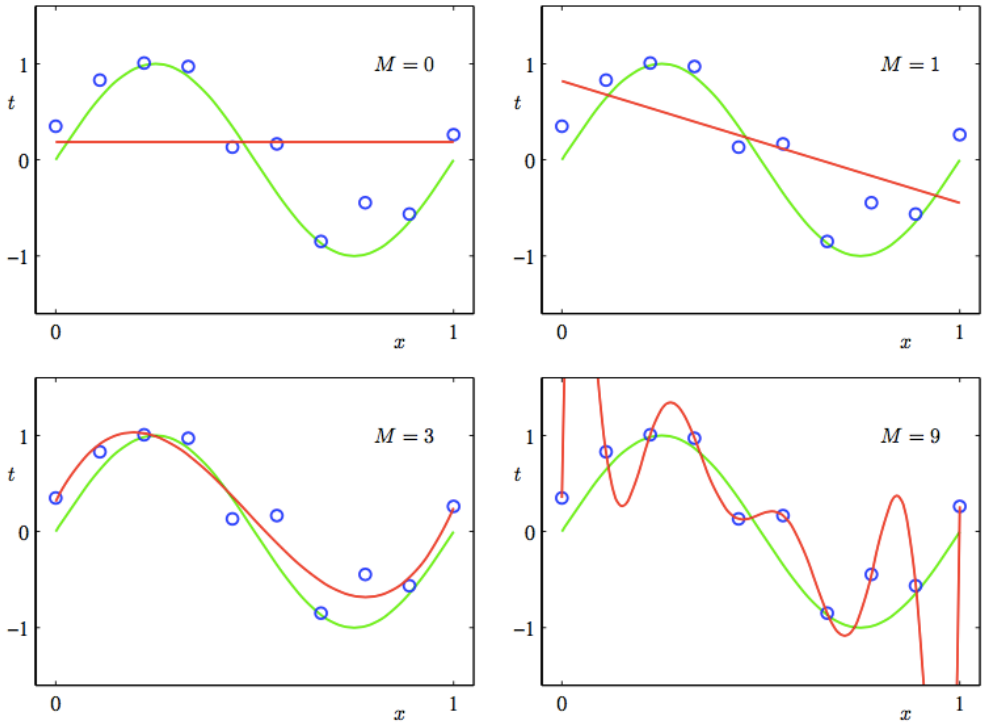


Table showing the coefficients of the linear regression models

	$M=0$	$M=1$	$M=6$	$M=9$
$w_0^\star$	0.19	0.82	0.31	0.35
$w_1^\star$		-1.27	7.99	232.37
$w_2^\star$			-25.43	-5321.83
$w_3^\star$			17.37	48568.31
$w_4^\star$				-231639.30
$w_5^\star$				640042.26
$w_6^\star$				-1061800.52
$w_7^\star$				1042400.18
$w_8^\star$				-557682.99
$w_9^\star$				125201.43

$$t = w_0^* + w_1^* x + w_2^* x^2 + \cdots + w_9^* x^9$$

What is going on with the coefficients when M is large?

Figure from Bishop, C. Pattern Recognition and Machine Learning. Springer, 2006

Regularization in Linear Regression

- Regularization is one strategy that can be used to deal with overfitting in many models.
- In many cases, it is usually convenient to add some regularization to your models

Regularization: Ridge Regression

- **Ridge Regression**, also called **Tikhonov Regularization**, solves the following optimization problem:

Find x such that it minimizes the following loss or penalty function:

$$L(x, \lambda) = \frac{1}{2}\|z - Hx\|_2^2 + \underbrace{\lambda\|x\|_2^2}$$

Regular linear regression doesn't have this term

Regularization: Ridge Regression

Find x such that it minimizes the following loss or penalty function:

$$L(x, \lambda) = \frac{1}{2}\|z - Hx\|_2^2 + \lambda\|x\|_2^2$$

- **Lambda is a number that you choose. The greater the lambda, the greater the regularization.**

Regularization: Ridge Regression

- **Ridge Regression** models are as easy to fit as your classical linear regression models. You only need to solve a system of linear equations for x.

$$(H^T \cdot H + \lambda I) \cdot x = H^T \cdot z$$

- Do:

$$x = (H^T \cdot H + \lambda I)^{-1} H^T \cdot z$$

Outline

- Introduction
- Linear Regression
- Normal Equations
- Gradient Descent
- Overfitting, Underfitting, and Regularization
- Linear Regression in R

Linear Regression in R

- To fit a model with linear regression in R,

```
df.name <- data.frame(z = c(3,7,11,18),
                      t = c(4,6,9,20))

model <- lm(z ~ 1 + t + I(t^2/2), data = df.name)
```

Notice that `t^2/2 is not part of the data frame, but computed from it`

Here the predictors are 1, t, and $t^2/2$, which are the columns of the H matrix

$$\underbrace{\begin{pmatrix} 3 \\ 7 \\ 11 \\ 18 \end{pmatrix}}_{\mathbf{z}} = \underbrace{\begin{pmatrix} 1 & 4 & 8 \\ 1 & 6 & 18 \\ 1 & 9 & 40.5 \\ 1 & 20 & 200 \end{pmatrix}}_{\mathbf{H}} \cdot \underbrace{\begin{pmatrix} x_0 \\ v_0 \\ a \end{pmatrix}}_{\mathbf{x}} + \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \epsilon_4 \end{pmatrix}$$

Linear Regression in R

```
df.name <- data.frame(z = c(3,7,11,18),
                      t = c(4,6,9,20))

model <- lm(z ~ 1 + t + I(t^2/2), data = df.name)
```

```{r}
model <- lm(data = my.data, z ~ 1 + t + I(t^2/2))
summary(model)
```

```
Call:
lm(formula = z ~ 1 + t + I(t^2/2), data = my.data)

Residuals:
       1        2        3        4
-0.149928  0.285578 -0.145385  0.009736

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.44430    0.94291  -5.774   0.1092
t            2.39277    0.20224  11.831   0.0537 .
I(t^2/2)    -0.12210    0.01604  -7.611   0.0832 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.3539 on 1 degrees of freedom
Multiple R-squared:  0.999,     Adjusted R-squared:  0.9969
F-statistic: 489.5 on 2 and 1 DF,  p-value: 0.03195
```

Linear Regression in R

```{r}
model <- lm(data = my.data, z ~ 1 + t + I(t^2/2))
summary(model)
```

```
Call:
lm(formula = z ~ 1 + t + I(t^2/2), data = my.data)

Residuals:
       1        2        3        4
-0.149928  0.285578 -0.145385  0.009736

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -5.44430    0.94291  -5.774   0.1092
t            2.39277    0.20224  11.831   0.0537 .
I(t^2/2)    -0.12210    0.01604  -7.611   0.0832 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3539 on 1 degrees of freedom
Multiple R-squared:  0.999,     Adjusted R-squared:  0.9969
F-statistic: 489.5 on 2 and 1 DF,  p-value: 0.03195
```

```
>> H = [1 4 8; 1 6 18; 1 9 40.5; 1 20 200]

H =

    1.0000    4.0000    8.0000
    1.0000    6.0000   18.0000
    1.0000    9.0000   40.5000
    1.0000   20.0000  200.0000

>> z = [3; 7; 11; 18]

z =

     3
     7
    11
    18

>> (H' * H)\(H'*z)

ans =

   -5.4443
    2.3928
   -0.1221

>>
```

MATLAB

The model found is then:

$$z = (-5.44) + (2.39)t + (-0.12)\frac{t^2}{2} + \epsilon$$

Linear Regression in R

- To use the model you just built to predict the horsepower of another car not in the dataset, we do

```r
predict(model, data.frame(disp = c(120, 119, 110, 147)))
```

```
       1        2        3        4
98.24085 97.80330 93.86532 110.05477
```

- And those are the predictions for the horsepower of cars with engines whose displacements are 120, 119, 110 and 147.

ADDITIONAL CONTENT

Regularization: Lasso

- **Lasso Regression** solves the following optimization problem:

Find x such that it minimizes the following loss or penalty function:

$$L(x, \lambda) = \frac{1}{2}\|z - Hx\|_2^2 + \lambda\|x\|_1^2$$

Manhattan Norm!!

- Lasso will drive many coefficients to 0, so it can be used for feature selection

About R^2

- According to C. Shalizy

$$
\begin{aligned}
R^2 &= \frac{\mathrm{Var}\left[m(X)\right]}{\mathrm{Var}\left[Y\right]} \\
&= \frac{\mathrm{Var}\left[\beta_0 + \beta_1 X\right]}{\mathrm{Var}\left[\beta_0 + \beta_1 X + \epsilon\right]} \\
&= \frac{\mathrm{Var}\left[\beta_1 X\right]}{\mathrm{Var}\left[\beta_1 X + \epsilon\right]} \\
&= \frac{\beta_1^2 \mathrm{Var}\left[X\right]}{\beta_1^2 \mathrm{Var}\left[X\right] + \sigma^2}
\end{aligned}
$$

https://www.reddit.com/r/statistics/comments/3ow1cd/my_stats_professor_just_went_on_a_rant_about_how/

About R^2

- According to C. Shalizy, these are the myths around this number:
  1. It does not measure the goodness of fit
     - It can be very low for a perfectly correct model: A small Var[X], large sigma^2, move R^2 to 0
     - It can be very high for a perfectly wrong model: A big Var[X] and a non-zero slope move R^2 to 1.
  - It is Useless as a measure for predictability
     - "Mean Squared Error is a *much* better measure of how good predictions are; better yet are estimates of out-of-sample error[…]"
  - R^2 cannot be compared across datasets:
     - The same model has different R^2 in different datasets

https://www.reddit.com/r/statistics/comments/3ow1cd/my_stats_professor_just_went_on_a_rant_about_how/

About R^2

- According to C. Shalizy, these are the myths around this number:
  - “Using adjusted R^2 instead of R^2 does absolutely nothing to fix any of this”
  - “At this point, you might be wondering just what R2 is good for — what job it does that isn’t better done by other tools. The only honest answer I can give you is that I have never found a situation where it helped at all. If I could design the regression curriculum from scratch, I would never mention it. Unfortunately, it lives on as a historical relic, so you need to know what it is, and what mis-understandings about it people suffer from”

https://www.reddit.com/r/statistics/comments/3ow1cd/my_stats_professor_just_went_on_a_rant_about_how/
https://www.stat.cmu.edu/~cshalizi/mreg/15/lectures/10/lecture-10.pdf
https://data.library.virginia.edu/is-r-squared-useless/