Lecture 9

Multiple Linear Regression

DSC 40A, Spring 2024

Announcements

- Homework 4 is due on **Thursday, May 2nd**.
    - Some office hours are now in HDSI **355** – see the calendar for more details.
- Homework 2 scores are available on Gradescope.
    - Regrade requests are due on Monday.

The Midterm Exam is on Tuesday, May 7th!

- The Midterm Exam is on **Tuesday, May 7th in class**.
  - You must take it during your scheduled lecture session.
  - You will receive a randomized seat assignment over the weekend.
- 80 minutes, on paper, no calculators or electronics.
  - **You are allowed to bring one two-sided index card (4 inches by 6 inches) of notes that you write by hand (no iPad).**
- Content: Lectures 1-9, Homeworks 1-4, Groupworks 1-4.
- We will have a review session on **on Friday from 2-5PM in Center Hall 109** where we'll go over old homework and exam problems.
- Prepare by practicing with old exam problems at practice.dsc40a.com.
  - Problems are sorted by topic!

Agenda

- Multiple linear regression.
- Interpreting parameters.
- Feature engineering and transformations.

Question 🤔

Answer at q.dsc40a.com

Remember, you can always ask questions at q.dsc40a.com!

If the direct link doesn't work, click the "🤔 Lecture Questions" link in the top right corner of dsc40a.com.

Multiple linear regression

	departure_hour	day_of_month	minutes
0	10.816667	15	68.0
1	7.750000	16	94.0
2	8.450000	22	63.0
3	7.133333	23	100.0
4	9.150000	30	69.0
...	...	...	...

So far, we've fit **simple** linear regression models, which use only **one** feature (`'departure_hour'`) for making predictions.

Incorporating multiple features

- In the context of the commute times dataset, the simple linear regression model we fit was of the form:

$$\begin{aligned} \text{pred. commute} &= H(\text{departure hour}) \\ &= w_0 + w_1 \cdot \text{departure hour} \end{aligned}$$

- Now, we'll try and fit a multiple linear regression model of the form:

$$\begin{aligned} \text{pred. commute} &= H(\text{departure hour}) \\ &= w_0 + w_1 \cdot \text{departure hour} + w_2 \cdot \text{day of month} \end{aligned}$$

- Linear regression with **multiple** features is called **multiple linear regression**.
- How do we find w_0^*, w_1^*, and w_2^*?

Geometric interpretation

- The hypothesis function:

$$H(\text{departure hour}) = w_0 + w_1 \cdot \text{departure hour}$$

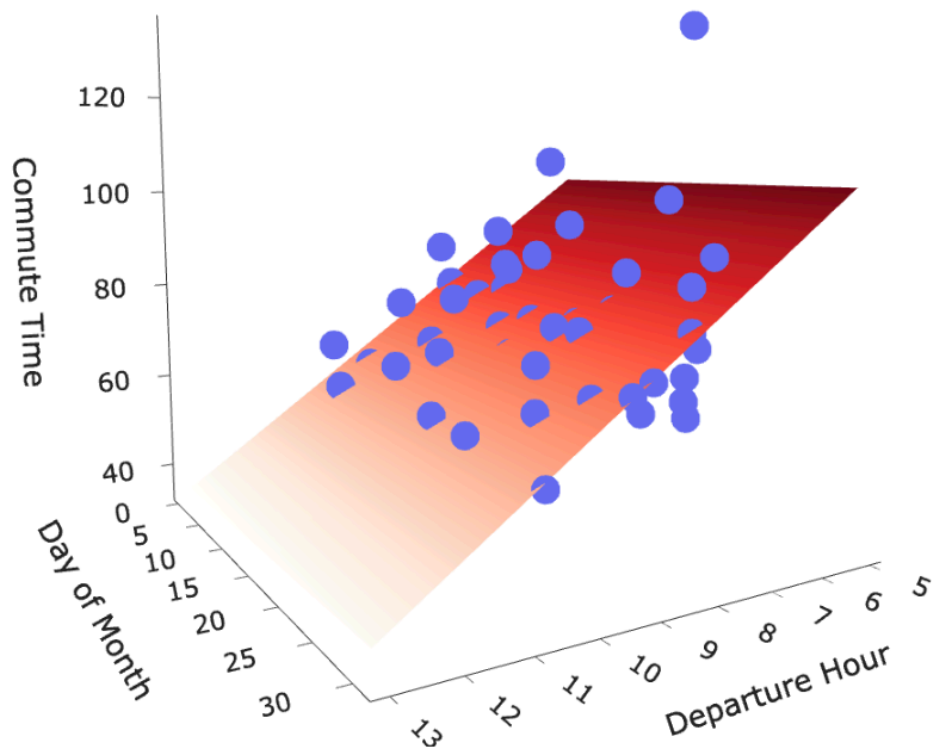
  looks like a **line** in 2D.

- **Questions**:
  - How many dimensions do we need to graph the hypothesis function:

  $$H(\text{departure hour}) = w_0 + w_1 \cdot \text{departure hour} + w_2 \cdot \text{day of month}$$

  - What is the shape of the hypothesis function?

Commute Time vs. Departure Hour and Day of Month



Our new hypothesis function is a **plane** in 3D!

Our goal is to find the **plane** of best fit that pierces through the cloud of points.

The setup

- Suppose we have the following dataset.

	departure_hour	day_of_month	minutes
row			
1	8.45	22	63.0
2	8.90	28	89.0
3	8.72	18	89.0

- We can represent each day with a **feature vector**, $\vec{x}$:

The hypothesis vector

- When our hypothesis function is of the form:

$$H(\text{departure hour}) = w_0 + w_1 \cdot \text{departure hour} + w_2 \cdot \text{day of month}$$

the hypothesis vector $\vec{h} \in \mathbb{R}^n$ can be written as:

$$\vec{h} = \begin{bmatrix} H(\text{departure hour}_1, \text{day}_1) \\ H(\text{departure hour}_2, \text{day}_2) \\ \dots \\ H(\text{departure hour}_n, \text{day}_n) \end{bmatrix} = \begin{bmatrix} 1 & \text{departure hour}_1 & \text{day}_1 \\ 1 & \text{departure hour}_2 & \text{day}_2 \\ \dots & \dots & \dots \\ 1 & \text{departure hour}_n & \text{day}_n \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \\ w_2 \end{bmatrix}$$

Finding the optimal parameters

- To find the optimal parameter vector, $\vec{w}^*$, we can use the **design matrix** $X \in \mathbb{R}^{n \times 3}$ and **observation vector** $\vec{y} \in \mathbb{R}^n$:

$$X = \begin{bmatrix} 1 & \text{departure hour}_1 & \text{day}_1 \\ 1 & \text{departure hour}_2 & \text{day}_2 \\ \ldots & \ldots & \ldots \\ 1 & \text{departure hour}_n & \text{day}_n \end{bmatrix} \qquad \vec{y} = \begin{bmatrix} \text{commute time}_1 \\ \text{commute time}_2 \\ \vdots \\ \text{commute time}_n \end{bmatrix}$$

- Then, all we need to do is solve the **normal equations**:

$$X^T X \vec{w}^* = X^T \vec{y}$$

If $X^T X$ is invertible, we know the solution is:

$$\vec{w}^* = (X^T X)^{-1} X^T \vec{y}$$

Notation for multiple linear regression

- We will need to keep track of multiple features for every individual in our dataset.
    - In practice, we could have hundreds or thousands of features!
- As before, subscripts distinguish between individuals in our dataset. We have n individuals, also called **training examples**.
- Superscripts distinguish between **features**. We have d features.

$$\text{departure hour:} \quad x^{(1)}$$

$$\text{day of month:} \quad x^{(2)}$$

Think of $x^{(1)}$, $x^{(2)}$, ... as new variable names, like new letters.

Augmented feature vectors

- The **augmented feature vector** $\text{Aug}(\vec{x})$ is the vector obtained by adding a 1 to the front of feature vector $\vec{x}$:

$$\vec{x} = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(d)} \end{bmatrix} \qquad \text{Aug}(\vec{x}) = \begin{bmatrix} 1 \\ x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(d)} \end{bmatrix} \qquad \vec{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix}$$

- Then, our hypothesis function is:

$$\begin{aligned} H(\vec{x}) &= w_0 + w_1 x^{(1)} + w_2 x^{(2)} + \ldots + w_d x^{(d)} \\ &= \vec{w} \cdot \text{Aug}(\vec{x}) \end{aligned}$$

The general problem

- We have n data points, $(\vec{x}_1, y_1), (\vec{x}_2, y_2), \ldots, (\vec{x}_n, y_n)$, where each $\vec{x}_i$ is a feature vector of d features:

$$\vec{x}_i = \begin{bmatrix} x_i^{(1)} \\ x_i^{(2)} \\ \vdots \\ x_i^{(d)} \end{bmatrix}$$

- We want to find a good linear hypothesis function:

$$\begin{aligned} H(\vec{x}) &= w_0 + w_1 x^{(1)} + w_2 x^{(2)} + \ldots + w_d x^{(d)} \\ &= \vec{w} \cdot \text{Aug}(\vec{x}) \end{aligned}$$

The general solution

- Define the **design matrix** $X \in \mathbb{R}^{n \times (d+1)}$ and **observation vector** $\vec{y} \in \mathbb{R}^n$:

$$X = \begin{bmatrix} 1 & x_1^{(1)} & x_1^{(2)} & \dots & x_1^{(d)} \\ 1 & x_2^{(1)} & x_2^{(2)} & \dots & x_2^{(d)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n^{(1)} & x_n^{(2)} & \dots & x_n^{(d)} \end{bmatrix} = \begin{bmatrix} \text{Aug}(\vec{x}_1)^T \\ \text{Aug}(\vec{x}_2)^T \\ \vdots \\ \text{Aug}(\vec{x}_n)^T \end{bmatrix} \qquad \vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

- Then, solve the **normal equations** to find the optimal parameter vector, $\vec{w}^*$:

$$X^T X \vec{w}^* = X^T \vec{y}$$

Terminology for parameters

- With d features, $\vec{w}$ has $d + 1$ entries.
- w_0 is the **bias**, also known as the **intercept**.
- $w_1, w_2, \ldots, w_d$ each give the **weight**, or **coefficient**, or **slope**, of a feature.

$$H(\vec{x}) = w_0 + w_1 x^{(1)} + w_2 x^{(2)} + \ldots + w_d x^{(d)}$$

Interpreting parameters

Example: Predicting sales

- For each of 26 stores, we have:
    - net sales,
    - square feet,
    - inventory,
    - advertising expenditure,
    - district size, and
    - number of competing stores.
- **Goal**: Predict net sales given the other five features.
- To begin, we'll start trying to fit the hypothesis function to predict sales:

$$H(\text{square feet}, \text{competitors}) = w_0 + w_1 \cdot \text{square feet} + w_2 \cdot \text{competitors}$$

Question 🤔

Answer at q.dsc40a.com

$$H(\text{square feet}, \text{competitors}) = w_0 + w_1 \cdot \text{square feet} + w_2 \cdot \text{competitors}$$

What will be the signs of w_1^* and w_2^*?

- A. w_1^* $+$ w_2^* $+$
- B. w_1^* $+$ w_2^* $-$
- A. w_1^* $-$ w_2^* $+$
- A. w_1^* $-$ w_2^* $-$

Let's find out! Follow along in this notebook.

Question 🤔

Answer at q.dsc40a.com

Which feature is most "important"?

- A. square feet: $w_1^* = 16.202$
- B. competitors: $w_2^* = -5.311$
- C. inventory: $w_2^* = 0.175$
- D. advertising: $w_3^* = 11.526$
- E. district size: $w_4^* = 13.580$

Which features are most "important"?

- The most important feature is **not necessarily** the feature with largest magnitude weight.
- Features are measured in different units, i.e. different scales.
  - Suppose I fit one hypothesis function, H_1, with sales in US dollars, and another hypothesis function, H_2, with sales in Japanese yen (1 USD $\approx$ 157 yen).
  - Sales is just as important in both hypothesis functions.
  - But the weight of sales in H_1 will be 157 times larger than the weight of sales in H_2.
- **Solution**: If you care about the interpretability of the resulting weights, **standardize** each feature before performing regression, i.e. convert each feature to standard units.

Standard units

- Recall: to convert a feature $x_1, x_2, \ldots, x_n$ to standard units, we use the formula:

$$x_{i\ (\text{su})} = \frac{x_i - \bar{x}}{\sigma_x}$$

- Example: 1, 7, 7, 9.
  - Mean: $\frac{1+7+7+9}{4} = \frac{24}{4} = 6$.
  - Standard deviation:

$$\text{SD} = \sqrt{\frac{1}{4}\left((1-6)^2 + (7-6)^2 + (7-6)^2 + (9-6)^2\right)} = \sqrt{\frac{1}{4} \cdot 36} = 3$$

  - Standardized data:

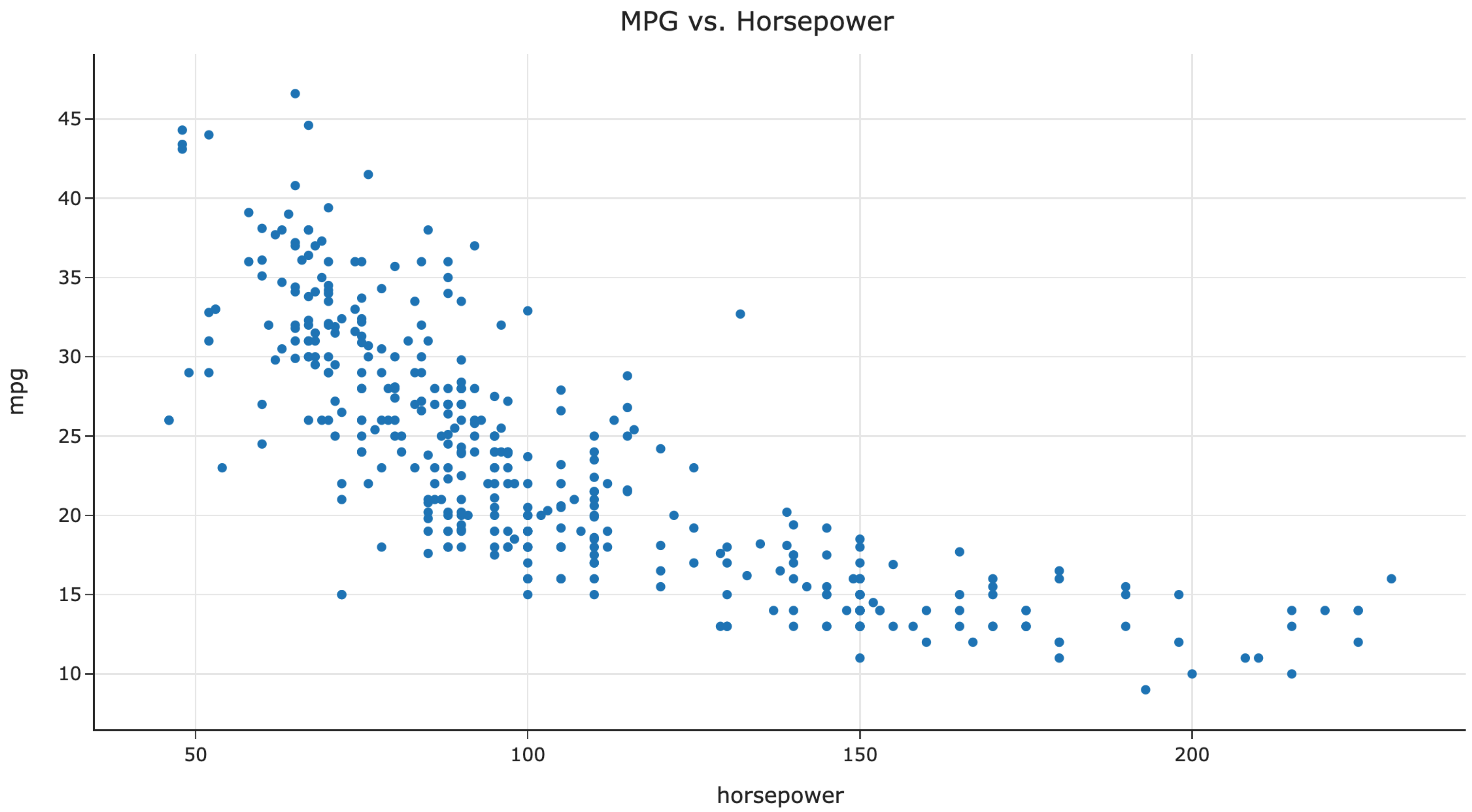
$$1 \mapsto \frac{1-6}{3} = \boxed{-\frac{5}{3}} \qquad 7 \mapsto \frac{7-6}{3} = \boxed{\frac{1}{3}} \qquad 7 \mapsto \boxed{\frac{1}{3}} \qquad 9 \mapsto \frac{9-6}{3} = \boxed{1}$$

Standard units for multiple linear regression

- The result of standardizing each feature (separately!) is that the units of each feature are on the same scale.
    - There's no need to standardize the outcome (net sales), since it's not being compared to anything.
    - Also, we can't standardize the column of all 1s.
- Then, solve the normal equations. The resulting $w_0^*, w_1^*, \ldots, w_d^*$ are called the **standardized regression coefficients**.
- Standardized regression coefficients can be directly compared to one another.
- Note that standardizing each feature **does not** change the MSE of the resulting hypothesis function!

Once again, let's try it out! Follow along in this notebook.

Feature engineering and transformations



Question: Would a linear hypothesis function work well on this dataset?

A quadratic hypothesis function

- It looks like there's some sort of quadratic relationship between horsepower and MPG in the last scatter plot. We want to try and fit a hypothesis function of the form:

$$H(x) = w_0 + w_1 x + w_2 x^2$$

  - Note that while this is quadratic in horsepower, it is **linear in the parameters**!
  - That is, it is a **linear combination of features**.
- We can do that, by choosing our two "features" to be x_i and x_i^2, respectively.
  - In other words, $x_i^{(1)} = x_i$ and $x_i^{(2)} = x_i^2$.
  - More generally, we can create new features out of existing features.

A quadratic hypothesis function

- Desired hypothesis function: $H(x) = w_0 + w_1 x + w_2 x^2$.
- The resulting design matrix looks like:

$$X = \begin{bmatrix} 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \\ \dots \\ 1 & x_n & x_n^2 \end{bmatrix}$$

- To find the optimal parameter vector $\vec{w}^*$, we need to solve the **normal equations**!

$$X^T X \vec{w}^* = X^T \vec{y}$$

More examples

- What if we want to use a hypothesis function of the form:
  $H(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3$?

- What if we want to use a hypothesis function of the form:
  $H(x) = w_1 \frac{1}{x^2} + w_2 \sin x + w_3 e^x$?

Feature engineering

- The process of creating new features out of existing information in our dataset is called **feature engineering**.
- In this class, feature engineering will mostly be restricted to creating non-linear functions of existing features (as in the previous example).
- In the future you'll learn how to do other things, like encode categorical information.
  - You'll be exposed to this in Homework 4, Problem 5!

Non-linear functions of multiple features

- Recall our earlier example of predicting sales from square footage and number of competitors. What if we want a hypothesis function of the form:

$$\begin{aligned} H(\text{sqft}, \text{comp}) &= w_0 + w_1 \cdot \text{sqft} + w_2 \cdot \text{sqft}^2 + w_3 \cdot \text{comp} + w_4 \cdot (\text{sqft} \cdot \text{comp}) \\ &= w_0 + w_1 s + w_2 s^2 + w_3 c + w_4 sc \end{aligned}$$

- The solution is to choose a design matrix accordingly:

$$X = \begin{bmatrix} 1 & s_1 & s_1^2 & c_1 & s_1 c_1 \\ 1 & s_2 & s_2^2 & c_2 & s_2 c_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & s_n & s_n^2 & c_n & s_n c_n \end{bmatrix}$$

Finding the optimal parameter vector, $\vec{w}^*$

- As long as the form of the hypothesis function permits us to write $\vec{h} = X\vec{w}$ for some X and $\vec{w}$, the mean squared error is:

$$R_{\text{sq}}(\vec{w}) = \frac{1}{n} \|\vec{y} - X\vec{w}\|^2$$

- Regardless of the values of X and $\vec{y}$, the value of $\vec{w}^*$ that minimizes $R_{\text{sq}}(\vec{w})$ is the solution to the **normal equations**:

$$X^T X\vec{w}^* = X^T\vec{y}$$

Linear in the parameters

- We can fit rules like:

$$w_0 + w_1 x + w_2 x^2 \qquad w_1 e^{-{x^{(1)}}^2} + w_2 \cos(x^{(2)} + \pi) + w_3 \frac{\log 2x^{(3)}}{x^{(2)}}$$

  - This includes arbitrary polynomials.
  - These are all linear combinations of (just) features.

- We can't fit rules like:

$$w_0 + e^{w_1 x} \qquad w_0 + \sin(w_1 x^{(1)} + w_2 x^{(2)})$$

  - These are **not** linear combinations of just features!

- We can have any number of parameters, as long as our hypothesis function is **linear in the parameters**, or linear when we think of it as a function of the parameters.

Determining function form

- How do we know what form our hypothesis function should take?
- Sometimes, we know from *theory*, using knowledge about what the variables represent and how they should be related.
- Other times, we make a guess based on the data.
- Generally, start with simpler functions first.
    - Remember, the goal is to find a hypothesis function that will generalize well to unseen data.

Example: Amdahl's Law

- Amdahl's Law relates the runtime of a program on p processors to the time to do the sequential and nonsequential parts on one processor.

$$H(p) = t_{\mathrm{S}} + \frac{t_{\mathrm{NS}}}{p}$$

- Collect data by timing a program with varying numbers of processors:

Processors	Time (Hours)
1	8
2	4
4	3

Example: Fitting $H(x) = w_0 + w_1 \cdot \frac{1}{x}$

Processors	Time (Hours)
1	8
2	4
4	3

How do we fit hypothesis functions that aren't linear in the parameters?

- Suppose we want to fit the hypothesis function:

$$H(x) = w_0 e^{w_1 x}$$

- This is **not** linear in terms of w_0 and w_1, so our results for linear regression don't apply.
- **Possible solution**: Try to apply a **transformation**.

Transformations

- **Question**: Can we re-write $H(x) = w_0 e^{w_1 x}$ as a hypothesis function that **is** linear in the parameters?

Transformations

- **Solution**: Create a new hypothesis function, $T(x)$, with parameters b_0 and b_1, where $T(x) = b_0 + b_1 x$.
- This hypothesis function is related to $H(x)$ by the relationship $T(x) = \log H(x)$.
- $\vec{b}$ is related to $\vec{w}$ by $b_0 = \log w_0$ and $b_1 = w_1$.
- Our new observation vector, $\vec{z}$, is $\begin{bmatrix} \log y_1 \\ \log y_2 \\ \ldots \\ \log y_n \end{bmatrix}$.
- $T(x) = b_0 + b_1 x$ is linear in its parameters, b_0 and b_1.
- Use the solution to the normal equations to find $\vec{b}^*$, and the relationship between $\vec{b}$ and $\vec{w}$ to find $\vec{w}^*$.

Once again, let's try it out! Follow along in this notebook.

Non-linear hypothesis functions in general

- Sometimes, it's just not possible to transform a hypothesis function to be linear in terms of some parameters.
- In those cases, you'd have to resort to other methods of finding the optimal parameters.
    - For example, $H(x) = w_0 \sin(w_1 x)$ **can't** be transformed to be linear.
    - But, there are other methods of minimizing mean squared error:

$$R_{\text{sq}}(w_0, w_1) = \frac{1}{n} \sum_{i=1}^{n} (y_i - w_0 \sin(w_1 x))^2$$

    - One method: **gradient descent**, the topic of the next lecture!
- Hypothesis functions that are linear in the parameters are much easier to work with.

Roadmap

- This is the end of the content that's in scope for the Midterm Exam.
- On Thursday, we'll introduce **gradient descent**, a technique for minimizing functions that can't be minimized directly using calculus or linear algebra.
- After the Midterm Exam, we'll:
  - Look at a technique for identifying patterns in data when there is no "right answer" $\vec{y}$, called **clustering**.
  - Switch gears to **probability**.