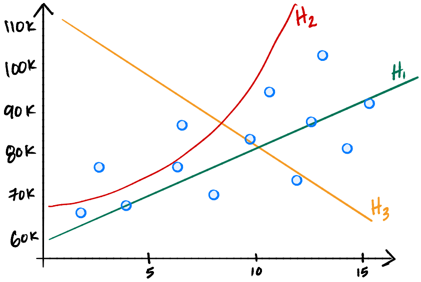


CSE 151A

Intro to Machine Learning

Lecture 06 – Part 01

Regression and Loss Functions

What influences salary?

- **Numerical**: age, height, years of experience
- **Categorical**: college, city, gender
- **Boolean**: knows Python?, had internship?

Regression

- **Given**: someone's age, experience, city, etc.
- **Predict**: their salary.
- Since the output space is $\mathbb{R}$, this is a **regression** problem.

Today

- **Goal**: Predict salary from years of experience.

Prediction Rules

- Salary is a function of experience.
- I.e., there is a function H so that:

$$\text{salary} \approx H(\text{years of experience})$$

- H is a **hypothesis function** or **prediction rule**.
- **Our goal**: find a good prediction rule, H.

Example Prediction Rules

$$H_1(\text{years of experience}) = \$50{,}000 + \$2{,}000 \times (\text{years of experience})$$

$$H_2(\text{years of experience}) = \$60{,}000 \times 1.05^{(\text{years of experience})}$$

$$H_3(\text{years of experience}) = \$100{,}000 - \$5{,}000 \times (\text{years of experience})$$

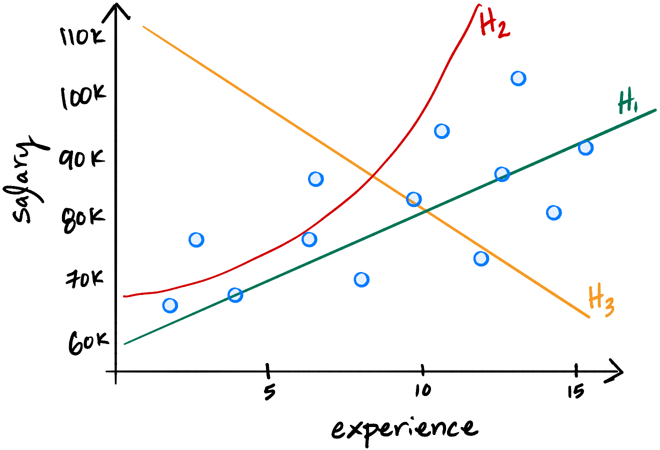
Comparing Predictions

- How do we know which is best: H_1, H_2, H_3?
- We gather data from n people. Let x_i be experience, y_i be salary:

$$\begin{array}{c} (\text{Experience}_1, \text{Salary}_1) \\ (\text{Experience}_2, \text{Salary}_2) \\ \ldots \\ (\text{Experience}_n, \text{Salary}_n) \end{array} \quad \rightarrow \quad \begin{array}{c} (x_1, y_1) \\ (x_2, y_2) \\ \ldots \\ (x_n, y_n) \end{array}$$

- See which rule works better on data.

Example



Quantifying the Error

- Our prediction for person i's salary is $H(x_i)$
- The **absolute error** in this prediction:

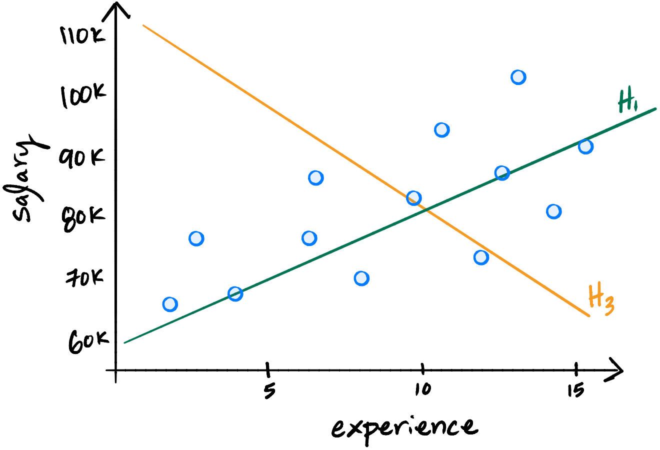
$$\left| H(x_i) - y_i \right|$$

- The **mean absolute error** of H:

$$R_{\text{abs}}(H) = \frac{1}{n} \sum_{i=1}^{n} \left| H(x_i) - y_i \right|$$

- Smaller the mean absolute error, the **better** the prediction rule.

Mean Absolute Error



Finding the best prediction rule

- **Goal:** out of all functions $\mathbb{R} \to \mathbb{R}$, find the function H^* with the smallest mean absolute error.

- That is, find:

$$H^* = \arg\min_H \frac{1}{n} \sum_{i=1}^{n} \left| H(x_i) - y_i \right|$$

Finding the best prediction rule

- **Goal:** out of all functions $\mathbb{R} \to \mathbb{R}$, find the function H^* with the smallest mean absolute error.

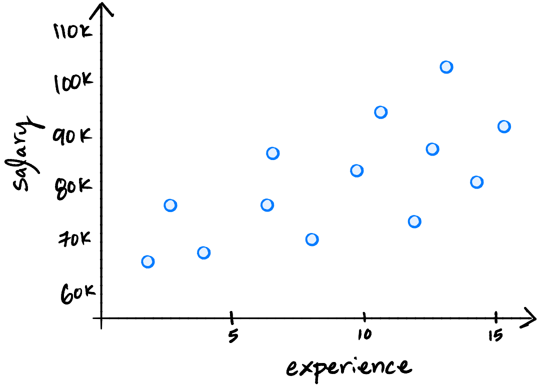
- That is, find:

$$H^* = \arg\min_{H} \frac{1}{n} \sum_{i=1}^{n} \left| H(x_i) - y_i \right|$$

- **There are two problems with this.**

Question

Is there a prediction rule H which has **zero** mean absolute error?



Problem #1

- We can make mean absolute error very small, even zero!
- But the function will be weird.
- This is called **overfitting**.
- Remember our real goal: make good predictions on data **we haven't seen**.

Solution

- Don't allow H to be just any function.
- Require that it has a certain form.
- Examples:
  - Linear: $H(x) = w_1 x + w_0$
  - Quadratic: $H(x) = w_2 x^2 + w_1 x + w_0$
  - Exponential: $H(x) = w_0 e^{w_1 x}$
  - Constant: $H(x) = w_0$

Finding the best linear rule

- **Goal:** out of all linear functions $\mathbb{R} \to \mathbb{R}$, find the function H^* with the smallest mean absolute error.

- That is, find:

$$H^* = \operatorname*{arg\,min}_{\text{linear } H} \frac{1}{n} \sum_{i=1}^{n} \left| H(x_i) - y_i \right|$$

Finding the best linear rule

- **Goal:** out of all **linear** functions $\mathbb{R} \to \mathbb{R}$, find the function H^* with the smallest mean absolute error.

- That is, find:

$$H^* = \underset{\text{linear } H}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \left| H(x_i) - y_i \right|$$

- **There is still a problem with this.**

Problem #2

- It is hard to minimize the mean absolute error:[1]

$$\frac{1}{n}\sum_{i=1}^{n} |H(x_i) - y_i|$$

- **Not differentiable!**
- What can we do?

[1] Though it can be done with linear programming.

Quantifying the Error

- Instead of absolute error, use the **squared error** of a prediction:

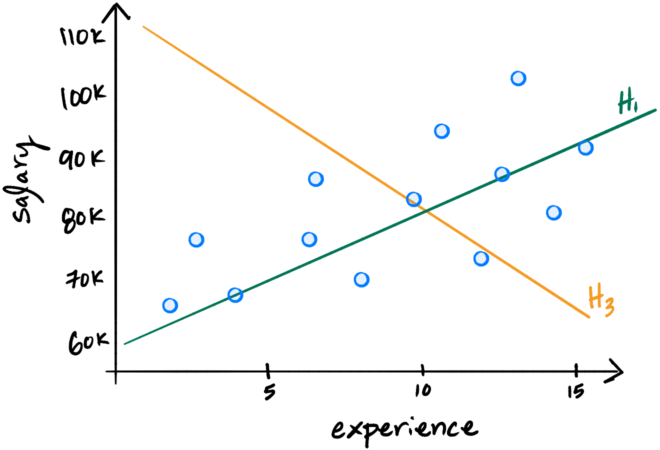
$$\left(H(x_i) - y_i\right)^2$$

- The **mean squared error** (MSE) of H:

$$R_{\text{sq}}(H) = \frac{1}{n}\sum_{i=1}^{n}\left(H(x_i) - y_i\right)^2$$

- **Is differentiable!**

Mean Squared Error



Our Goal

- Out of all **linear** functions $\mathbb{R} \to \mathbb{R}$, find the function H^* with the smallest **mean squared error**.

- That is, find:

$$H^* = \underset{\text{linear } H}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \left(H(x_i) - y_i \right)^2$$

- This problem is called **least squares regression**.

By the way...

- Prediction functions will play a large role.
- **Absolute error** and **squared error** are **loss functions**.

$$|H(x) - y_i| \qquad (H(x_i) - y_i)^2$$

By the way...

- The average loss on the training data is called the **empirical risk**

- Example: the mean squared error is the empirical risk of the square loss:

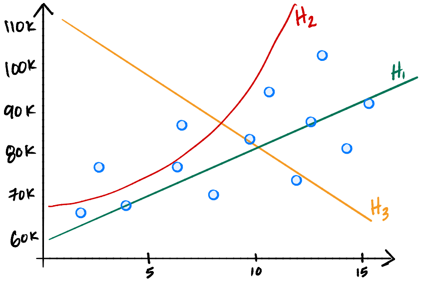
$$R_{\text{sq}}(H) = \frac{1}{n} \sum_{i=1}^{n} \left(H(x_i) - y_i \right)^2$$

By the way...

- A major paradigm in ML: find the prediction function which minimizes risk.
- Called **Empirical Risk Minimization**, or **ERM**.

Up next...

...minimizing the MSE.



CSE 151A
Intro to Machine Learning

Lecture 06 – Part 02
Minimizing the MSE

Our Goal

- Out of all **linear** functions $\mathbb{R} \to \mathbb{R}$, find the function H^* with the smallest **mean squared error**.
- That is, find:

$$H^* = \underset{\text{linear } H}{\arg\min} \frac{1}{n} \sum_{i=1}^{n} \left(H(x_i) - y_i \right)^2$$

- This problem is called **least squares regression**.

Minimizing the MSE

- The MSE is a function of a function:

$$R_{\text{sq}}(H) = \frac{1}{n}\sum_{i=1}^{n}\left(H(x_i) - y_i\right)^2$$

- But since H is linear, $H(x) = w_1 x + w_0$.

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

- Now it's a function of w_1, w_0.

Updated Goal

- Find slope w_1 and intercept w_0 which minimize the MSE, $R_{\text{sq}}(w_1, w_0)$:

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

- Strategy: multivariate calculus.

Recall: the **gradient**

- If $f(x, y)$ is a function of two variables, the **gradient** of f at the point (x_0, y_0) is a **vector** of **partial derivatives**:

$$\nabla f(x_0, y_0) = \begin{pmatrix} \frac{\partial f}{\partial x}(x_0) \\ \\ \frac{\partial f}{\partial y}(y_0) \end{pmatrix}$$

- **Key Fact**: gradient is zero at critical points.

Strategy

To minimize $R(w_1, w_0)$: compute the gradient, set equal to zero, solve.

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

$$\frac{\partial R_{\text{sq}}}{\partial w_1} =$$

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

$$\frac{\partial R_{\text{sq}}}{\partial w_0} =$$

Strategy

$$0 = \frac{2}{n}\sum_{i=1}^{n}\big((w_1 x_i + w_0) - y_i\big)x_i \quad 0 = \frac{2}{n}\sum_{i=1}^{n}\big((w_1 x_i + w_0) - y_i\big)$$

1. Solve for w_0 in second equation.
2. Plug solution for w_0 into first equation, solve for w_1.

Solve for w_0

$$0 = \frac{2}{n} \sum_{i=1}^{n} \left((w_1 x_i + w_0) - y_i \right)$$

Solve for w_0

$$0 = \frac{2}{n} \sum_{i=1}^{n} \left((w_1 x_i + w_0) - y_i \right)$$

Key Fact

- Define

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

- Then

$$\sum_{i=1} (x_i - \bar{x}) = 0 \qquad \sum_{i=1} (y_i - \bar{y}) = 0$$

Solve for w_1

$$0 = \frac{2}{n} \sum_{i=1}^{n} \left((w_1 x_i + w_0) - y_i \right) x_i \qquad w_0 = \bar{y} - w_1 \bar{x}$$

Solve for w_1

$$0 = \frac{2}{n}\sum_{i=1}^{n}\Big((w_1 x_i + w_0) - y_i\Big)x_i \qquad w_0 = \bar{y} - w_1\bar{x}$$

Least Squares Solutions

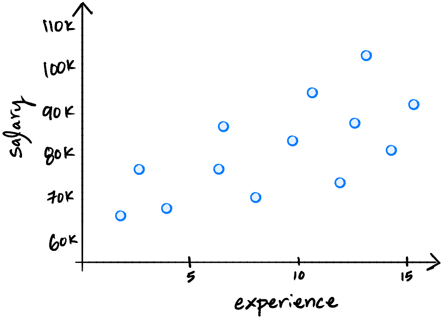
- The **least squares solutions** for the slope w_1 and intercept w_0 are:

$$w_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} \qquad w_0 = \bar{y} - w_1\bar{x}$$

where $\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$

Interpretation of Slope

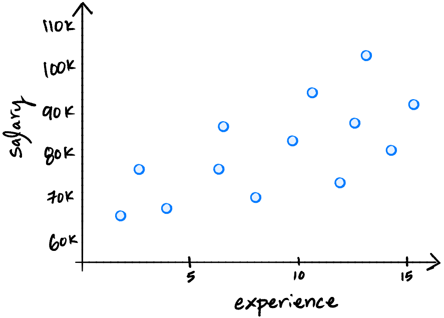
$$w_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$



- What is the sign of $(x_i - \bar{x})(y_i - \bar{y})$ when:
  - $x_i > \bar{x}$ and $y_i > \bar{y}$?

Interpretation of Slope

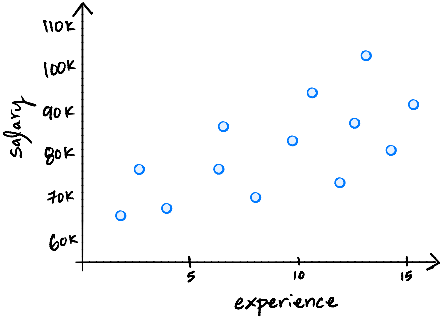
$$w_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$



- What is the sign of $(x_i - \bar{x})(y_i - \bar{y})$ when:
  - $x_i < \bar{x}$ and $y_i < \bar{y}$?

Interpretation of Slope

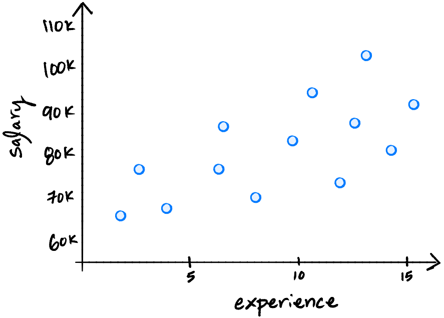
$$w_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$



- What is the sign of $(x_i - \bar{x})(y_i - \bar{y})$ when:
  - $x_i > \bar{x}$ and $y_i < \bar{y}$?

Interpretation of Slope

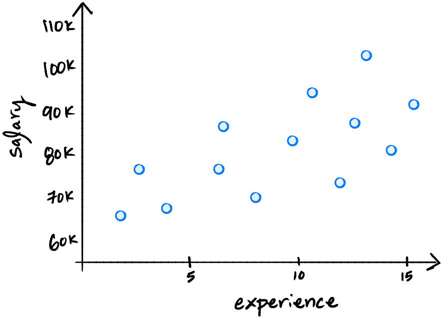
$$w_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$



- What is the sign of $(x_i - \bar{x})(y_i - \bar{y})$ when:
  - $x_i < \bar{x}$ and $y_i > \bar{y}$?

Interpretation of Intercept

$$w_0 = \bar{y} - w_1 \bar{x}$$

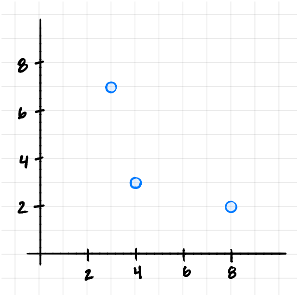


- What is $H(\bar{x})$?

Question

We fit a linear prediction rule for salary given years of experience. Then everyone gets a $5,000 raise. What happens to slope/intercept?

Example



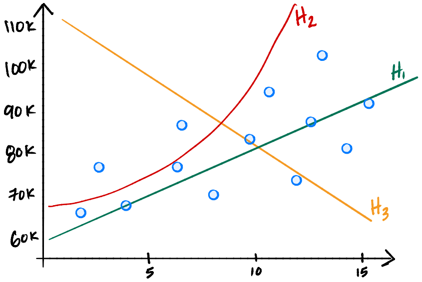
$\bar{x} =$

$\bar{y} =$

$$w_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} =$$

$$w_0 = \bar{y} - w_1\bar{x}$$

x_i	y_i	$(x_i - \bar{x})$	$(y_i - \bar{y})$	$(x_i - \bar{x})(y_i - \bar{y})$	$(x_i - \bar{x})^2$
3	7				
4	3				
8	2				



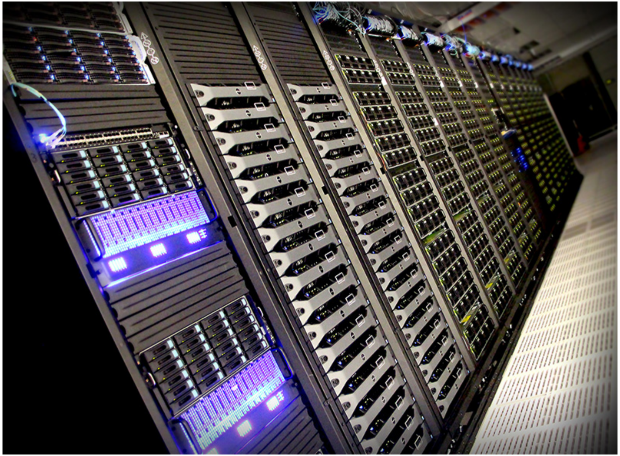
CSE 151A

Intro to Machine Learning

Lecture 06 – Part 03

Fitting Non-Linear Trends

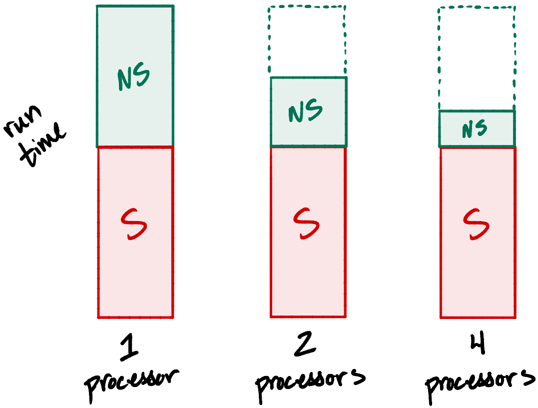
Example: Parallel Processing



Problem

- Some parts of a program are necessarily **sequential**.
- E.g., downloading the data must happen before analysis.
- More processors do not speed up **sequential** code.
- But they do speed up **non-sequential** code.

Speedup



Amdahl's Law

The time T it takes to run a program on p processors is:

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

where t_S and t_{NS} are the time it takes the sequential and non-sequential parts to run on one processor, respectively.

Amdahl's Law

The time T it takes to run a program on p processors is:

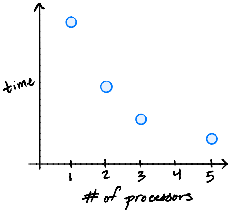
$$T(p) = t_S + \frac{t_{NS}}{p}$$

where t_S and t_{NS} are the time it takes the sequential and non-sequential parts to run on one processor, respectively.

Problem: we don't know t_S and t_{NS}.

Fitting Amdahl's Law

- **Solution**: we will learn t_S and t_{NS} from data.
- Run with varying number of processors:



- Find prediction rule $H(p) = \dfrac{t_{NS}}{p} + t_S$ by minimizing MSE.

General Problem

- Given data $(x_1, y_1), \dots, (x_n, y_n)$.
- Fit a **non-linear** rule $H(x) = w_1 \cdot \frac{1}{x} + w_0$ by minimizing MSE:

$$R_{\text{sq}}(H) = \frac{1}{n} \sum_{i=1}^{n} \left(H(x_i) - y_i\right)^2$$

Using definition of H:

Minimizing MSE

- Take derivatives, you'll find:

$$\frac{\partial R_{\text{sq}}}{\partial w_1}(w_1, w_0) = \frac{2}{n} \sum_{i=1}^{n} \left[\left(w_1 \cdot \frac{1}{x_i} + w_0 \right) - y_i \right] \frac{1}{x_i}$$

$$\frac{\partial R_{\text{sq}}}{\partial w_0}(w_1, w_0) = \frac{2}{n} \sum_{i=1}^{n} \left[\left(w_1 \cdot \frac{1}{x_i} + w_0 \right) - y_i \right]$$

Minimizing MSE

- Set to zero, solve. You'll find:

$$w_1 = \frac{\sum_{i=1}^{n} \left(\frac{1}{x_i} - \frac{1}{n} \sum_{i=1}^{n} \frac{1}{x_i} \right) (y_i - \bar{y})}{\sum_{i=1}^{n} \left(\frac{1}{x_i} - \frac{1}{n} \sum_{i=1}^{n} \frac{1}{x_i} \right)^2} \qquad w_0 = \bar{y} - w_1 \cdot \frac{1}{n} \sum_{i=1}^{n} \frac{1}{x_i}$$

Minimizing MSE

- Set to zero, solve. You'll find:

$$w_1 = \frac{\sum_{i=1}^{n} \left(\frac{1}{x_i} - \frac{1}{n} \sum_{i=1}^{n} \frac{1}{x_i} \right)(y_i - \bar{y})}{\sum_{i=1}^{n} \left(\frac{1}{x_i} - \frac{1}{n} \sum_{i=1}^{n} \frac{1}{x_i} \right)^2} \qquad w_0 = \bar{y} - w_1 \cdot \frac{1}{n} \sum_{i=1}^{n} \frac{1}{x_i}$$

- Define $z_i = \frac{1}{x_i}$, $\bar{z} = \frac{1}{n} \sum_{i=1}^{n} z_i = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{x_i}$. Then:

$$w_1 = \qquad\qquad w_0 =$$

Fitting Non-Linear Trends

To fit a prediction rule of the form $H(x) = w_1 \cdot \frac{1}{x} + w_0$:

1. Create a new data set $(z_1, y_1), \dots, (z_n, y_n)$, where $z_i = \frac{1}{x_i}$.

Fitting Non-Linear Trends

To fit a prediction rule of the form $H(x) = w_1 \cdot \frac{1}{x} + w_0$:

2. Fit $H(z) = w_1 z + w_0$ using familiar least squares solutions:

$$w_1 = \frac{\sum_{i=1}^{n}(z_i - \bar{z})(y_i - \bar{y})}{\sum_{i=1}^{n}(z_i - \bar{z})^2} \qquad w_0 = \bar{y} - w_1 \cdot \bar{z}$$

Fitting Non-Linear Trends

To fit a prediction rule of the form $H(x) = w_1 \cdot \frac{1}{x} + w_0$:

3. Use w_1 and w_0 in original prediction rule, $H(x)$.

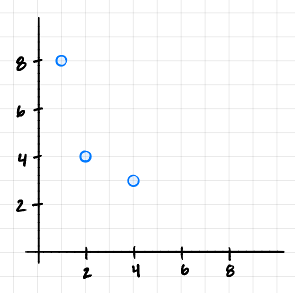
Example: Amdahl's Law

- We have timed our program:

Processors	Time (Hours)
1	8
2	4
4	3

- Fit prediction rule: $H(p) = \frac{t_{\mathrm{NS}}}{p} + t_{\mathrm{S}}$

Example: fitting $H(x) = w_1 \cdot \frac{1}{x_i} + x_0$



$\bar{z} =$

$\bar{y} =$

$$w_1 = \frac{\sum_{i=1}^{n}(z_i - \bar{z})(y_i - \bar{y})}{\sum_{i=1}^{n}(z_i - \bar{z})^2} =$$

$w_0 = \bar{y} - w_1\bar{z}$

x_i	z_i	y_i	$(z_i - \bar{z})$	$(y_i - \bar{y})$	$(z_i - \bar{z})(y_i - \bar{y})$	$(z_i - \bar{z})^2$
3	7					
4	3					
8	2					

Example: Amdahl's Law

- We found: $t_{\text{NS}} = \frac{48}{7} \approx 6.88, \quad t_{\text{S}} = 1$
- Our prediction rule:

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

$$= \frac{6.88}{p} + 1$$

Fitting Non-Linear Trends

- We can fit rules like:

$$w_1 x + w_0 \qquad w_1 \cdot \frac{1}{x} + w_0 \qquad w_1 x^2 + w_0 \qquad w_1 e^x + w_0$$

- We can't fit rules like:

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

- Can fit as long as **linear** function of w_1, w_0.

What's Left?

- How do we make predictions with lots of features?

- E.g., experience, age, GPA, number of internships, etc.