DSC 140A

Probabilistic Modeling & Machine Learning

Lecture 02 | Part 1

News

News

- Lab 01 released. Due Sunday @ 11:59 pm.
- HW 01 released. Due Wednesday @ 11:59 pm.
  - LaTeX template available (optional).

DSC 140A

Probabilistic Modeling & Machine Learning

Lecture 02 | Part 2

Linear Models

Last Time: Nearest Neighbors

- Nearest neighbor methods are simple; can work well.

- However, they:
  1. "memorize" the training data (**inefficient**);
  2. do not learn relative important of features.

Example: Predicting Salary

- **Goal:** predict a data scientist's salary from three features:
  - x_1: years of experience
  - x_2: # of interview questions missed
  - x_3: favorite number

- **Observations:**
  - x_1 is **positively** associated with salary
  - x_2 is **negatively** associated with salary
  - x_3 is **not** associated with salary

Prediction Functions

- **Informally:** we think years of experience, etc., are predictive of salary.
- **Formally:** we think there is a function H that takes $\vec{x} = (x_1, x_2, x_3)$ and outputs a good prediction of salary.

$$H(\vec{x}) \to \text{prediction}$$

- H is called a **prediction function**.[1]

---

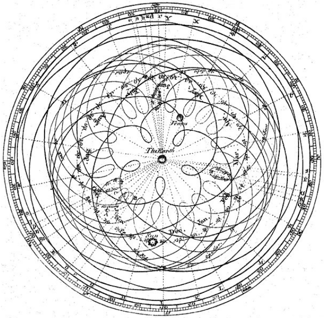
[1] Or, sometimes, a **hypothesis function**

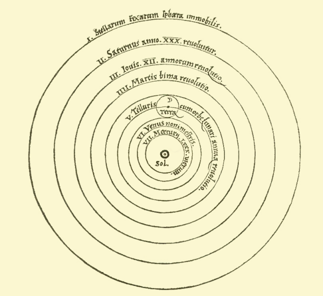
Prediction Functions

- **Goal:** find an accurate prediction function.
- What should our prediction function *look like*?
- That is, we must choose a **model**.
  - In context of prediction functions: a **hypothesis class**.

Occam's Razor

- **Occam's Razor**: when faced with two competing explanations (models), favor the simpler one.[2]





---

[2] As long as it works, of course.

Linear Functions

- **Idea:** model salary as a **weighted sum** of factors.
- That is, as a **linear function**:

$$H(\vec{x}) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

- $w_0, w_1, \ldots, w_3$ are the **parameters** or **weights**.
- **TODO:** how do we choose the weights?

Exercise

Recall:

- x_1: years of experience
- x_2: # of interview questions missed
- x_3: favorite number

What are reasonable values of the weights in the linear prediction function $H(\vec{x}) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$ if it is to be a good predictor of salary?

Parameter Vectors

- The parameters of a linear function can be packaged into a **parameter vector**, $\vec{w}$.
- **Example:** if $H(\vec{x}) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$ then $\vec{w} = (w_0, \ldots, w_3)^T$.

Parameterization

- A linear function $H(\vec{x})$ is **completely determined** by its parameter vector.
  - Can work either with the function, H, or vector, $\vec{w}$.

- Sometimes write $H(\vec{x}; \vec{w})$.

- Example: $\vec{w} = (8, 3, 1, 5, -2, -7)^T$ specifies

$$H(\vec{x}; \vec{w}) = 8 + 3x_1 + 1x_2 + 5x_3 - 2x_4 - 7x_5$$

Number of Parameters

- If a linear predictor $H(\vec{x}; \vec{w})$ takes in d-dimensional feature vectors, it has $d + 1$ parameters.

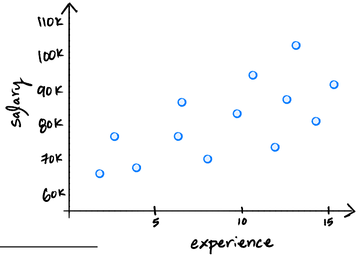
$$\begin{aligned} H(\vec{x}; \vec{w}) &= w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_d x_d \\ &= w_0 + \sum_{i=1}^{d} w_i x_i \end{aligned}$$

- That is, if $\vec{x} \in \mathbb{R}^d$, then $\vec{w} \in \mathbb{R}^{d+1}$.

Visualization

- Linear prediction rules have linear graphs.[3]
- **Example:** A linear prediction function for salary.

$$H_1(\vec{x}) = \$50{,}000 + (\text{experience}) \times \$8{,}000$$



[3] When visualized in feature space.

Visualization ($d > 1$)

- The **surface** of a prediction function H is made by plotting $H(\vec{x})$ for all $\vec{x}$.

- If H is a linear prediction function, and
  - $\vec{x} \in R^1$, then $H(x)$ is a straight line.
  - $\vec{x} \in \mathbb{R}^2$, then $H(\vec{x})$ is a plane.
  - $\vec{x} \in \mathbb{R}^d$, then $H(\vec{x})$ is a d-dimensional **hyperplane**.

Note: Compact Form

- Recall the **dot product** of vectors $\vec{a}$ and $\vec{b}$:

$$\vec{a} = (a_1, a_2, \dots, a_d)^T \qquad \vec{b} = (b_1, b_2, \dots, b_d)^T$$

$$\vec{a} \cdot \vec{b} = a_1 b_1 + a_2 b_2 + \dots + a_d b_d$$

- Observe:

$$\begin{aligned} H(\vec{x}; \vec{w}) &= w_0 + w_1 x_1 + \dots + w_d x_d \\ &= \underbrace{(w_0, w_1, \dots, w_d)^T}_{\vec{w}} \cdot \underbrace{(1, x_1, \dots, x_d)^T}_{?} \end{aligned}$$

Note: Compact Form

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

$$\vec{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix} \qquad \text{Aug}(\vec{x}) = \begin{pmatrix} 1 \\ x_1 \\ x_2 \\ \vdots \\ x_d \end{pmatrix}$$

- With augmentation, we can write:

$$\begin{aligned} H(\vec{x}) &= w_0 + w_1 x_1 + w_2 x_2 + \ldots + w_d x_d \\ &= \vec{w} \cdot \text{Aug}(\vec{x}) \end{aligned}$$

Classification?

- We have been focusing on **regression**.
- Linear prediction functions can be used for **classification**, too.
- We will come back to this.

DSC 140A

Probabilistic Modeling & Machine Learning

Lecture 02 | Part 3

Empirical Risk Minimization

Picking a Prediction Function

- Suppose we model salary as a linear function:

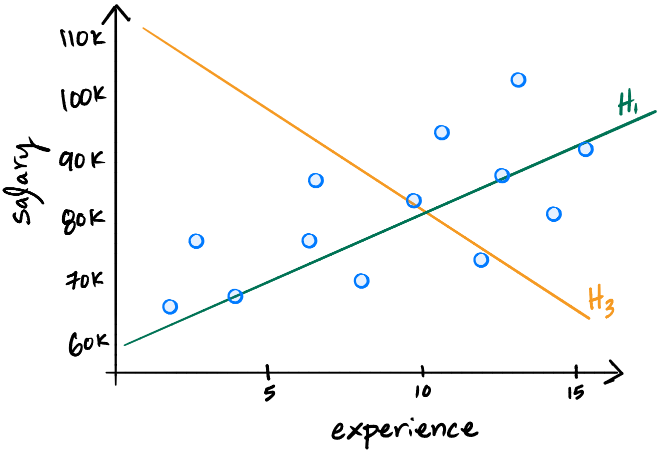
$$H(\vec{x}; \vec{w}) = w_0 + w_1 x_1 + w_2 x_2 + x_3 x_3$$

- **Question:** how do we choose weights $w_0, \ldots, w_3$ so that H makes good predictions?

Learning

- **Assumption:** the future will look like the past.
- *If so*, we should pick a prediction function that worked well on past data.
- That is, we should **learn** a function from data.

Example

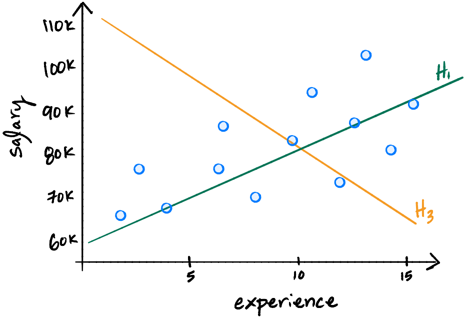


Training Data

- To learn, we gather **training data**.
- A set $\mathcal{D}$ of n pairs: $(\vec{x}^{(i)}, y_i)$
  - $\vec{x}^{(i)}$ is the ith **feature vector**
  - y_i is its **label** (the correct answer)
- In regression, y_i is a continuous number; in classification, it is discrete.
- This regime is called **supervised learning**.

An Optimization Problem

- Some prediction functions "fit" the data better than others.
- **Idea:** find the function that "fits best"



Quantifying Fit

- How do we measure "fit"?
- Formally: measure difference between our prediction $H(\vec{x}^{(i)})$ and the "right answer", y_i.
- A **loss function** quantifies how wrong a single prediction is.
- **Example:** the **absolute loss** $\ell_{\text{abs}}(H(\vec{x}^{(i)}), y_i) = |H(\vec{x}^{(i)}) - y_i|$

Quantifying Overall Fit

- **Idea:** a good H makes good predictions *on average* over entire data set.

- Find H minimizing the **expected loss**, also called the **empirical risk**:

$$R(H) = \sum_{i=1}^{n} \ell(H(\vec{x}^{(i)}), y_i)$$

- Note: R depends on both H and the data!

Empirical Risk Minimization

- This strategy is called **empirical risk minimization (ERM)**.
- Step 1: choose a **hypothesis class**
  - Let's assume we've chosen linear predictors
- Step 2: choose a **loss function**
- Step 3: minimize **expected loss (empirical risk)**

ERM for Regression

- We have chosen as our hypothesis class the set of **linear functions** $\mathbb{R}^d \to \mathbb{R}$.

- Suppose we choose **absolute loss**:

$$\ell_{\text{abs}}(H(\vec{x}^{(i)}), y_i) = |H(\vec{x}^{(i)}) - y_i|$$

- **Goal:** find H minimizing **mean absolute error**:

$$R_{\text{abs}}(H) = \sum_{i=1}^{n} |H(\vec{x}^{(i)}) - y_i|$$

Minimizing Mean *Absolute* Error

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

- That is, find:

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

Minimizing Mean *Absolute* Error

- Assume for now that $d = 1$ (one feature). Then $w \in \mathbb{R}^2$ and:

$$H(x; \vec{w}) = w_0 + w_1 x$$

- Recall that H is completely determined by w_0, w_1.

- Equivalent goal: find w_0 and w_1 minimizing

$$\frac{1}{n} \sum_{i=1}^{n} \left| H(x; w_0, w_1) - y_i \right|$$

Minimizing Mean *Absolute* Error

- To find optimal w_0 and w_1, might use calculus.
  - Set $\partial R / \partial w_0 = 0$ and $\partial R / \partial w_1 = 0$ and solve.
- Problem: absolute value is **not differentiable!**
- It is hard to minimize the mean absolute error.[4]
- What can we do?

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

Minimizing Mean *Squared* Error

- The **square loss** *is* differentiable:

$$\ell_{\text{sq}}(H(\vec{x}), y) = (H(\vec{x}) - y)^2$$

- Let's try minimizing the mean squared error instead.

Main Idea

We often choose a loss function out of practical considerations.

DSC 140A

Probabilistic Modeling & Machine Learning

Lecture 02 | Part 4

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**.

For now...

- For simplicity, assume that there is only one feature (predictor variable).
  - $H(x; \vec{w}) = w_0 + w_1 x$
  - I.e., one-dimensional linear regression.

- We will come back to multi-dimensional case in the next lecture.

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, $H(x) = w_1 x + w_0$.

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

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

Updated Goal

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

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

- Strategy: multivariate calculus.

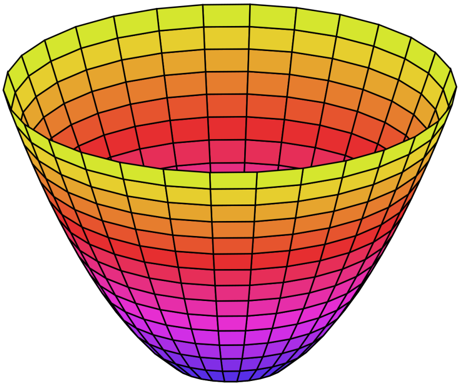
Exercise

Suppose we plotted $R_{sq}(w_1, w_0)$. What would it look like?

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

- Can R_{sq} be negative?
- Can it be zero?
- How many minima / maxima?

Answer



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.

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

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

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

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

Strategy

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

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

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

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

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

Least Squares Solutions

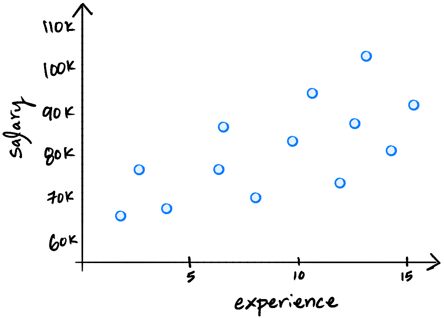
- The **least squares solutions** for the slope w_1 and intercept w_0 are:

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

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

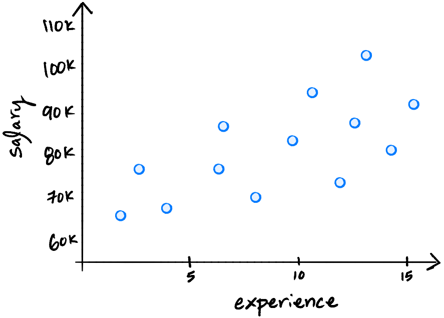
$$w_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$



- 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

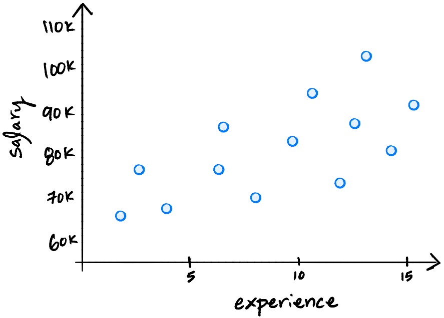
$$w_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$



- 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

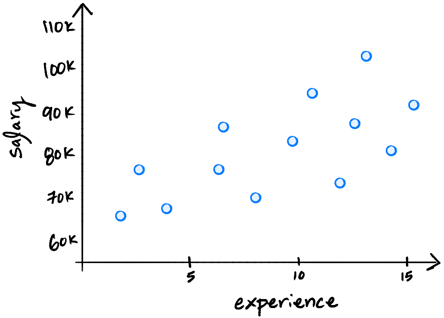
$$w_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$



- 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

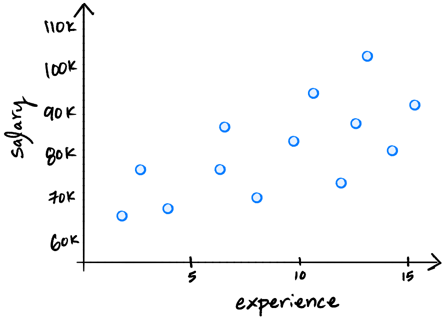
$$w_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$



- 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

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

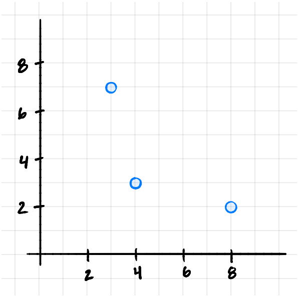


- What is $H(\bar{x})$?

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				

DSC 140A

Probabilistic Modeling & Machine Learning

Lecture 02 | Part 5

Fitting Non-Linear Trends

Non-Linear Trends

- We have fit a straight line of the form:

$$H(x) = w_0 + w_1 x$$

- What if we believe, e.g., salary grows with the **square** of experience?

- I.e., how do we fit a function of the form:

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

“Linear” Models

- The **linear** in **linear prediction function** refers to the weights, not the features.

- These are all **linear** prediction functions:
  - $H(x) = w_0 + w_1 x + w_2 x^2$
  - $H(x) = w_0 + w_1 e^x$
  - $H(x) = w_0 + w_1 \sqrt{x} + w_2 \sin x$

- These are **not**:
  - $H(x) = w_0 + w_1 e^{w_2 x}$
  - $H(x) = w_0 + w_1 \sin(w_2 x)$

In General

- $H(x) = w_0 + w_1\phi(x)$ is a linear model, no matter what ϕ is.[5]
- ϕ is called a **basis function** (or **feature map**).
- Example: $\phi(x) = x^2$

[5] Provided ϕ does not involve w_0 and w_1

Minimizing Mean Squared Error

- Fix a basis function $\phi(x)$.
- **Goal:** pick w_0 and w_1 so as to minimize the mean squared error of H:

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

Minimizing Mean Squared Error

- Notation: define $z_i = \phi(x_i)$.
- Strategy: compute $\partial R_{\text{sq}} / \partial w_0$ and $\partial R_{\text{sq}} / \partial w_1$, set to zero, solve.

Solution

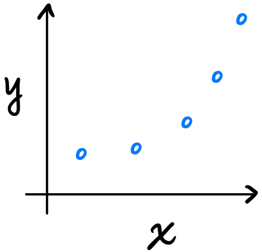
- **Observation:** This is the **exact same** calculation we've done, but with x_i replaced by z_i.

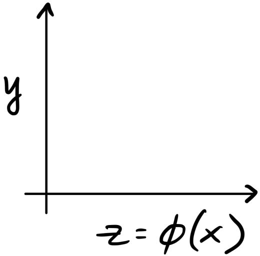
- The **least squares solutions**:

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

- where $\bar{z} \equiv \frac{1}{n}\sum_{i=1}^{n}\phi(x_i)$

Intuition





x	1	2	3	4
y	2	8	18	32
$z = x^2$	1	4	9	16

Interpretation

- To fit a function $H(x) = w_0 + w_1\phi(x)$:

1. Create new data set $\{(z_i, y_i)\}$, where $z_i = \phi(x_i)$.
2. Fit a straight line $H(z) = w_0 + w_1 z$ on this new data.
3. Use w_0 and w_1 in $H(x) = w_0 + w_1\phi(x)$

Summary

- We have seen how to fit linear prediction functions of the form:

$$H(x) = w_0 + w_1 \phi(x)$$

- **Next time**: how do we fit functions of the form:

$$H(x_1, x_2, \ldots) = w_0 + w_1 \phi(x_1) + w_2 \phi(x_2) + \ldots$$

- How does this compare to nearest neighbor methods?