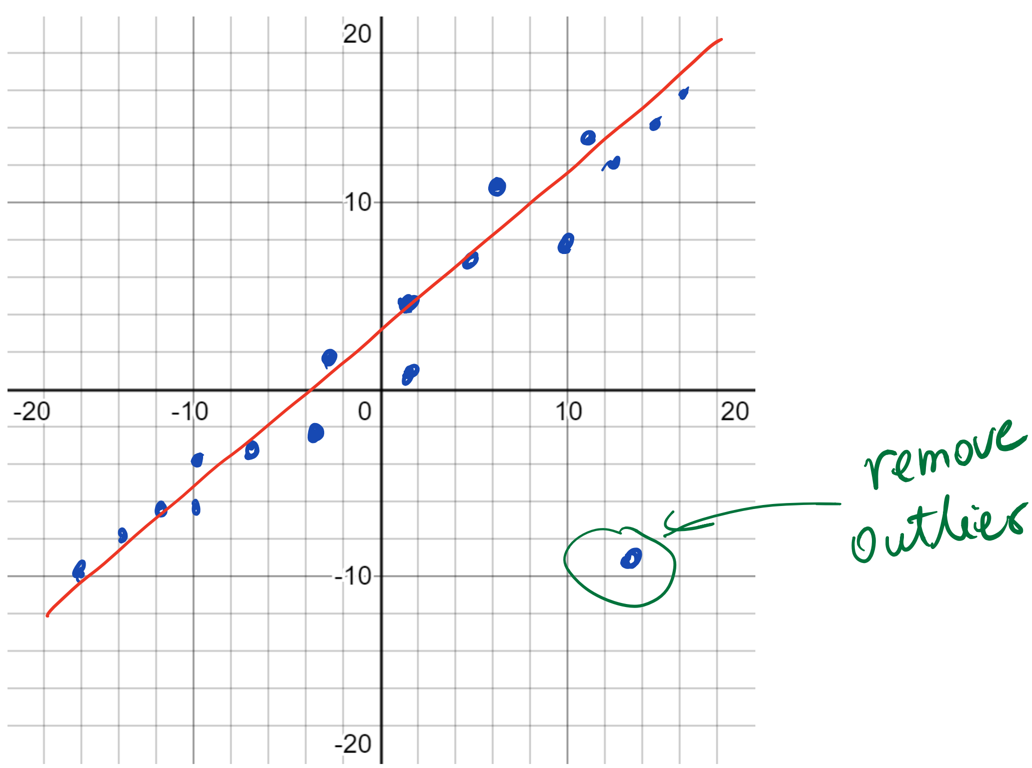
Lecture 11: Linear Regression

- Estimate the *unknown* target function $f : \mathcal{X} \to \mathcal{Y}$
- Let $\mathcal{D}$ denote the data set used for training our model.
- Every pair in $\mathcal{D}$ must satisfy $\boxed{f(input) = output.}$
- Suppose the *output* depends on *input* almost **linearly**
  - by doing exploratory data analysis
  - computing parameters such as correlation
  - additional subject knowledge expects linear relationship



- If the *input* is one-dimensional, we use simple regression, otherwise, we use multiple regression $\Rightarrow$ generalized setup

<u>Setup:</u>

Data set $\mathcal{D}$ has d-dimensional input vectors $[x_1, x_2, \cdots, x_d]^T$.

vectors $\vec{x}$ are column vectors

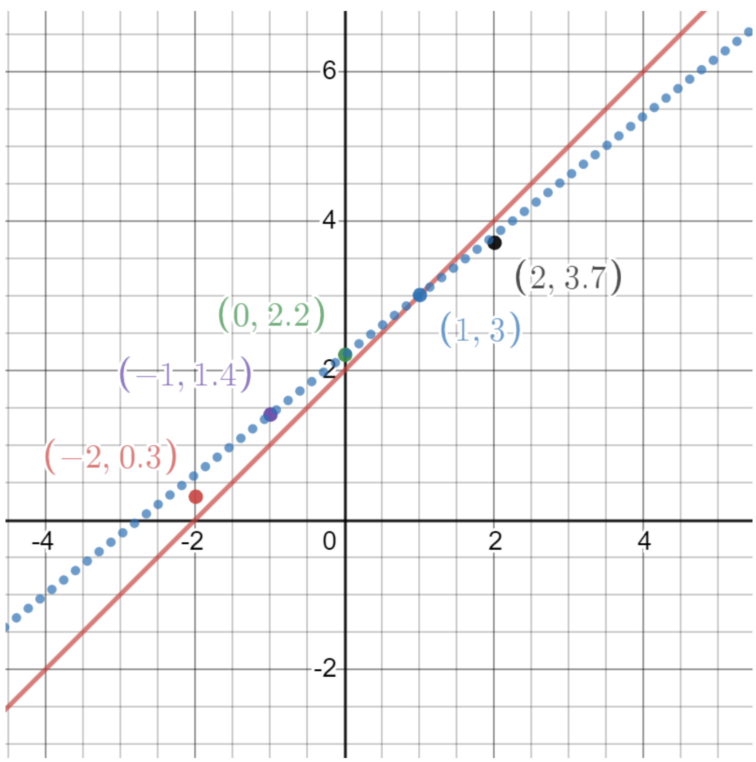
We are searching for a function

free coeff.

$$h(x_1, x_2, \cdots, x_d) = \underbrace{w_0 + w_1 x_1 + \cdots + w_d x_d}_{\text{linear}} \approx y = f(x_1, x_2, \ldots, x_d).$$

For each input vector, add a 0^{th} coordinate, and set it equal to 1, as we did in the PLA algorithm, so the *input* is $(d+1)$-dimensional: $\mathbf{x} = [1, x_1, x_2, \cdots, x_d]^T$. $x_0 = 1$

- sampled points with reading error
- remove red line
- look for a line to fit the data points



target fn:

$f(x) = x+2$

$\boxed{y = 2+x}$

$\boxed{h(x) = 2.2 + 0.8x}$

approximates $f(x)$

Find an $\boxed{\textit{optimal}}$ coordinate vector $\mathbf{w}_{\text{lin}} = [w_0, w_1, \ldots, w_d]^T$ so that

$$y \approx h(\mathbf{x}) = \mathbf{w}_{\text{lin}}^T \mathbf{x} = \vec{w}_{lin} \cdot \vec{x}$$

Q: when are $f(x)$ and $h(x)$ "close"?

Model error

This model is probabilistic in nature, that is, each pair $(\mathbf{x}, y)$ occurs with joint probability $P(\mathbf{x}, y)$, a probability with unknown distribution.

The (least squares) error resulting from approximation of the target function with a hyperplane given by coordinate vector $\mathbf{w}$:

predicted output

$$E_{out}(\mathbf{w}) = \mathbb{E}\left[(\mathbf{w}^T\mathbf{x} - y)^2\right].$$

expected output

Without knowing the probability distribution P, we cannot compute the expectation.

Error resulting from the approximation of sample data points from $\mathcal{D}$:

$$y_k \approx \mathbf{w}^T\mathbf{x}_k,$$

so we define the *in* error as the average error of square distances from data points to the approximating hyperplane:

predicted

$$E_{in}(\mathbf{w}) = \frac{1}{N}\sum_{k=1}^{N}(\mathbf{w}^T\mathbf{x}_k - y_k)^2.$$

expected

Goal: minimize this error $\Rightarrow$ find minimum of a multi-variable function

$$\boxed{\mathbf{w}_{\text{lin}} = \underset{\mathbf{w}\in\mathbb{R}^{d+1}}{\text{argmin}}\, E_{in}(\mathbf{w}),}$$

that is, $\mathbf{w}_{\text{lin}}$ is the argument that minimizes the function E_{in}, $\vec{w}_{lin}$ is where $E_{in}(\vec{w})$ has global min $\Rightarrow$ critical point.

$\#$ points $\quad$ dim of $\vec{x}$

Notation: $\mathcal{D} = \{(\vec{\mathbf{x}}_1, y_1), \ldots, (\vec{\mathbf{x}}_N, y_N)\}$. Construct the $N \times (d+1)$-dimensional (input) matrix $X = \begin{bmatrix} \vec{\mathbf{x}}_1^T \\ \vec{\mathbf{x}}_2^T \\ \cdots \\ \vec{\mathbf{x}}_N^T \end{bmatrix}$ and output vector $\vec{\mathbf{y}} = \begin{bmatrix} y_1 \\ y_2 \\ \cdots \\ y_N \end{bmatrix}$.

X has 1^{st} column of 1's.

The error function, using this notation:

$$
\begin{aligned}
E_{in}(\mathbf{w}) &= \frac{1}{N} \sum_{k=1}^{N} (\mathbf{w}^T \mathbf{x}_k - y_k)^2 \\
&\stackrel{\checkmark}{=} \frac{1}{N} \|X\mathbf{w} - \mathbf{y}\|^2 \\
&\stackrel{\checkmark}{=} \frac{1}{N} (X\mathbf{w} - \mathbf{y})^T (X\mathbf{w} - \mathbf{y}) \\
&\stackrel{\checkmark}{=} \frac{1}{N} \left(\mathbf{w}^T X^T - \mathbf{y}^T\right) (X\mathbf{w} - \mathbf{y}) \\
&= \frac{1}{N} \left(\mathbf{w}^T X^T X\mathbf{w} - \mathbf{w}^T X^T \mathbf{y} - \mathbf{y}^T X\mathbf{w} + \mathbf{y}^T \mathbf{y}\right) \\
&= \frac{1}{N} \left(\mathbf{w}^T X^T X\mathbf{w} - 2\mathbf{y}^T X\mathbf{w} + \mathbf{y}^T \mathbf{y}\right)
\end{aligned}
$$

same

$$
\underbrace{\begin{bmatrix} \vec{X}_1^T \\ \vdots \\ \vec{X}_N^T \end{bmatrix}}_{N \times (d+1)} \underbrace{\begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix}}_{(d+1)\times 1} = \begin{bmatrix} x_{10} w_0 + x_{11} w_1 + \ldots + x_{1d} w_d \\ \vdots \\ x_{N0} w_0 + \ldots \quad x_{Nd} w_d \end{bmatrix} = \underbrace{\begin{bmatrix} \vec{X}_1 \cdot \vec{w} \\ \vdots \\ \vec{X}_N \cdot \vec{w} \end{bmatrix}}_{(d+1)\times 1} = \begin{bmatrix} \vec{w}^T \vec{X}_1 \\ \vdots \\ \vec{w}^T \vec{X}_N \end{bmatrix}
$$

Recall: $\| [z_1, \ldots, z_N]^T \|^2 = z_1^2 + \ldots + z_N^2$

$\|\vec{z}\|^2 = \vec{z} \cdot \vec{z} = \vec{z}^T \vec{z}$

$(A - B)^T = A^T - B^T$. $\quad (AB)^T = B^T A^T$

$\vec{u} \cdot \vec{v} = \vec{u}^T \vec{v} = \vec{v}^T \vec{u}$

Recall: To find the minimum or maximum, we set the derivative equal to zero. But since here we have $(d+1)$ variables $\cdots$ we set all *partial* derivatives equal to zero $\Rightarrow$ set the gradient to zero.

Gradients

The idea behind partial derivatives is simple. When taking the partial derivative of a function with respect to a variable, you treat all other variables as constants.

Example: take partial derivatives of $f(x, y) = x^2y + \cos(x + y)$.

$$f_x = \frac{\partial f}{\partial x} = 2xy - \sin(x+y)(1+0)$$
$$f_y = \frac{\partial f}{\partial y} = x^2 - \sin(x+y)\cdot(0+1)$$

We collect all partial derivatives into a vector called the **gradient.**

$$\nabla f = \left[2xy - \sin(x+y),\ x^2 - \sin(x+y)\right]^T$$

Since all partial derivatives have to be zero at the minimum, the gradient has to be the zero vector.

Properties:

- ✓ If c is a constant scalar, $\nabla_{\mathbf{w}}(c) = \mathbf{0}$

scalar

- If $\mathbf{b}$ is a constant column vector, $\nabla_{\mathbf{w}}(\mathbf{b}^T\mathbf{w}) = \mathbf{b}$
- For a matrix A, using the product rule,

$$\nabla_{\mathbf{w}}(\mathbf{w}^T A\mathbf{w}) = A\mathbf{w} + A^T\mathbf{w} = (A + A^T)\mathbf{w}$$

scalar

Thus, we can minimize the error E_{in}: $\vec{w} = \begin{bmatrix} w_0 \\ \vdots \\ w_d \end{bmatrix}$

$$
\begin{aligned}
0 &= \nabla_{\mathbf{w}} E_{in}(\mathbf{w}) \\
0 &= \nabla_{\mathbf{w}} \left[\frac{1}{N} \left(\mathbf{w}^T X^T X \mathbf{w} - 2\mathbf{y}^T X \mathbf{w} + \mathbf{y}^T \mathbf{y} \right) \right]
\end{aligned}
$$

$$0 = \nabla_{\vec{w}} \left[\vec{w}^T (X^T X) \vec{w} - 2 (X^T \vec{y})^T \vec{w} + \vec{y}^T \vec{y} \right]$$

$$0 = \left(X^T X + (X^T X)^T \right) \vec{w} - 2 X^T \vec{y} + \vec{0}$$

$$0 = (X^T X + X^T X) \vec{w} - 2 X^T \vec{y}$$

$$0 = 2 X^T X \vec{w} - 2 X^T \vec{y}$$

$$\boxed{X^T X \vec{w} = X^T \vec{y}} \quad (*)$$

$E_{in}(\vec{w})$ is minimized at $\vec{w}$ that satisfies $(*)$

Remarks:

- $X^T X$ is a square matrix. If it is invertible, we can solve for $\mathbf{w}$:

$$\boxed{\mathbf{w} = (X^T X)^{-1} X^T \mathbf{y}.} \quad (**)$$

- If $X^T X$ is not invertible, one may still be able to find a solution for $\mathbf{w}$ in $X^T X \mathbf{w} = X^T \mathbf{y}$, but the solution may not be unique.

  Solve $A \vec{w} = \vec{b}$

- In our applications, since the number of data points in the training data set is much larger than the number of dimensions for the input ($N >> d$), the column vectors in X will be linearly independent.

  we can use $(**)$

Linear Regression Algorithm:

Step 1. Construct the $N \times (d+1)$-dimensional matrix X with rows $\mathbf{x}_k^T$, and the vector $\mathbf{y} = [y_1, \ldots, y_N]^T$

Step 2. Compute the matrix $A = (X^T X)^{-1} X^T$ (if possible)

Step 3. Find $\mathbf{w}_{\text{lin}} = A\mathbf{y}$.

Remark: $X\mathbf{w}_{\text{lin}}$ only approximates $\mathbf{y}$ due to sampling error.

Example 1: Consider the following selling data from sample of 10 Corvette cars, aged 1-6 years, available from the *Kelley Blue Book*. Here x denotes the age of the car and y denotes the selling price, in hundreds of dollars:

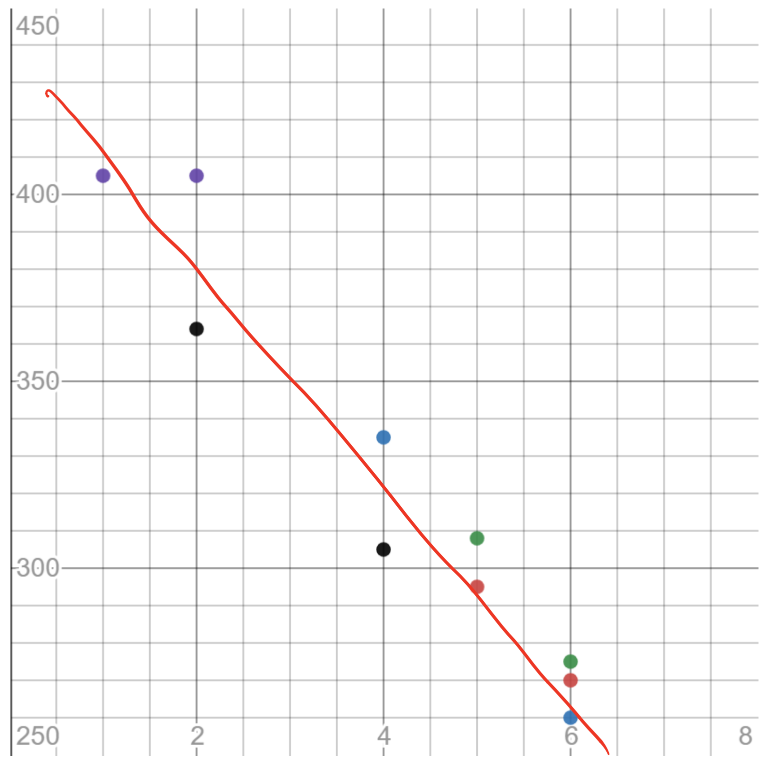
x	6	6	6	2	2	5	4	5	1	4
y	270	260	275	405	364	295	335	308	405	305

(a) Draw a scatter plot to determine if linear regression should be used.

(b) Run through the Linear Regression Algorithm to find $\vec{\mathbf{w}}_{\text{lin}}$

(c) Price a 3 year old Corvette, using your result in (b).

$$X = \begin{bmatrix} 1 & 6 \\ 1 & 6 \\ 1 & 6 \\ 1 & 2 \\ 1 & 2 \\ 1 & 5 \\ 1 & 4 \\ 1 & 5 \\ 1 & 1 \\ 1 & 4 \end{bmatrix} \qquad \vec{y} = \begin{bmatrix} 270 \\ 260 \\ 275 \\ 405 \\ 364 \\ 295 \\ 335 \\ 308 \\ 405 \\ 305 \end{bmatrix}$$



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

typo in data input

$$X^T X = \begin{bmatrix} 10 & 40 \\ 40 & 190 \end{bmatrix}$$

$$X^T \vec{y} = \begin{bmatrix} 3222 \\ 12040 \end{bmatrix}$$

$$\vec{w} = (X^T X)^{-1} (X^T \vec{y}) = \begin{bmatrix} 435.267 \\ -28.267 \end{bmatrix}$$

c) $\vec{w} \cdot \begin{bmatrix} 1 \\ 3 \end{bmatrix} = 350.47$

CORRECT ANSWERS:

$$X^T X = \begin{bmatrix} 10 & 41 \\ 41 & 199 \end{bmatrix}, \quad X^T \vec{y} = \begin{bmatrix} 3222 \\ 12348 \end{bmatrix}$$

$$\vec{w} = \begin{bmatrix} 436.602 \\ -27.903 \end{bmatrix} \qquad \text{(c) } \vec{w} \cdot \begin{bmatrix} 1 \\ 3 \end{bmatrix} = 352.89$$