$X$ design matrix

departure hour$_1$

departure hour$_2$

$\vdots$

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix}_{n \times 2}$$

commute time$_2$

$$\vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}_{n \times 1}$$

observation vector

$$\vec{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix}$$

parameter vector

$$\vec{h} = X\vec{w} = \begin{bmatrix} H(x_1) \\ H(x_2) \\ \vdots \\ H(x_n) \end{bmatrix}$$

hypothesis vector

mean squared error!

**Goal:** Choose the $\vec{w}^*$ that minimizes $\overbrace{\frac{1}{n}\|\vec{y} - X\vec{w}\|^2}^{\text{mean squared error!}}$ $R_{sq}(\vec{w})$

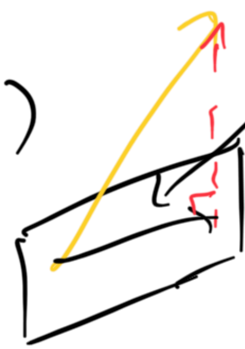
Equivalent to the other formulas:

$$\underbrace{X^T X \vec{w}^* = X^T \vec{y}}_{\text{normal equations}}$$

if $X^TX$ is invertible, unique solution

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

got this by projecting $\vec{y}$ onto span $(X)$



best predictions

the same!!!!

$$w_1^* = r\frac{\sigma_y}{\sigma_x}$$

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

?

$$X = \begin{bmatrix} 1 & dh_1 & dm_1 \\ 1 & dh_2 & dm_2 \\ \vdots & \vdots & \vdots \\ 1 & dh_n & dm_n \end{bmatrix}_{n \times 3}$$

departure hour$_n$

day of month$_n$

$$\vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

unchanged!

$$\vec{\omega} = \begin{bmatrix} \omega_0 \\ \omega_1 \\ \omega_2 \end{bmatrix}_{3 \times 1}$$

# Feature vectors

- 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 |

not a feature!

$$\vec{x}_1 = \begin{bmatrix} 8.45 \\ 22 \end{bmatrix}_{2\times 1} \qquad \vec{x}_2 = \begin{bmatrix} 8.90 \\ 28 \end{bmatrix} \qquad \vec{x}_3 = \begin{bmatrix} 8.72 \\ 18 \end{bmatrix}$$

?

# Augmented feature vector

$\text{Aug}(\vec{x})$ = append a 1 to the beginning of $\vec{x}$

$$\text{Aug}(\vec{x}_1) = \begin{bmatrix} 1 \\ 8.45 \\ 22 \end{bmatrix}$$

In general, if $\vec{x} = \begin{bmatrix} x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(d)} \end{bmatrix}_{d \times 1}$

then

$$\text{Aug}(\vec{x}) = \begin{bmatrix} 1 \\ x^{(1)} \\ x^{(2)} \\ \vdots \\ x^{(d)} \end{bmatrix}_{(d+1) \times 1}$$

http://localhost:8888/nbclassic/notebooks/lectures/lec17/lec17-blank.ipynb#/slide-5-0

Why make an augmented feature vector?

$\Rightarrow$ $\text{Aug}(\vec{x})$ and $\vec{w}$ have same dimensions!

e.g. two features

intercept term

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

now, our predictions for a single data point are a dot product!

$$H(\vec{x}) = \underbrace{w_0 + w_1 x^{(1)} + w_2 x^{(2)}}_{\text{upgraded } y = w_0 + w_1 x}$$

$$= \vec{w} \cdot \text{Aug}(\vec{x})$$

6.1

$$X = \begin{bmatrix} 1 & x_1^{(1)} & x_1^{(2)} & \cdots & x_1^{(d)} \\ 1 & x_2^{(1)} & x_2^{(2)} & \cdots & x_2^{(d)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n^{(1)} & x_n^{(2)} & \cdots & x_n^{(d)} \end{bmatrix}_{n \times (d+1)} \qquad \vec{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix}_{(d+1) \times 1}$$

$$\vec{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

Who cares???

THE NORMAL EQUATIONS are still relevant!

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

http://localhost:8888/nbclassic/notebooks/lectures/lec17/lec17-blank.ipynb#/slide-5-0

In general, with $d$ features, $X$ is of shape $n \times d+1$!

$$X = \begin{bmatrix} 1 & x_1 & x_1 & \cdots & x_1 \\ 1 & x_2 & x_2 & \cdots & x_2 \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & x_n & x_n & \cdots & x_n \end{bmatrix}_{n \times (d+1)}$$

?

http://localhost:8888/nbclassic/notebooks/lectures/lec17/lec17-blank.ipynb#/slide-48-0

## Linear in the parameters

no parameters!

- **Using linear regression**, 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.

$w_0 + w_1 \boxed{\phantom{x}} + w_2 \boxed{\phantom{x}} + \dots + w_d \boxed{\phantom{x}}$

should not have $w$!

- For any of the above examples, we **could** express our model as a product of a design matrix and parameter vector, and that's all that `LinearRegression` in `sklearn` needs.

What we put in the `X` argument to `model.fit` is up to us!

$w_1$ is part of $e$!

$\mathbb{X}\vec{w}^*$

- Using linear regression, 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.

$w_1, w_2$ are in the sin!

49.1