# tweets	# flu cases
x	y

$f()$

$y = f(x)$

We do not know $f()$

$x_1 = 10$ $y_1 = 7$

$x_2 = 49$ $y_2 = 41$

$x_3 = 3$ $y_3 = 1$

$x_4 = 450$ $y_4 = 304$

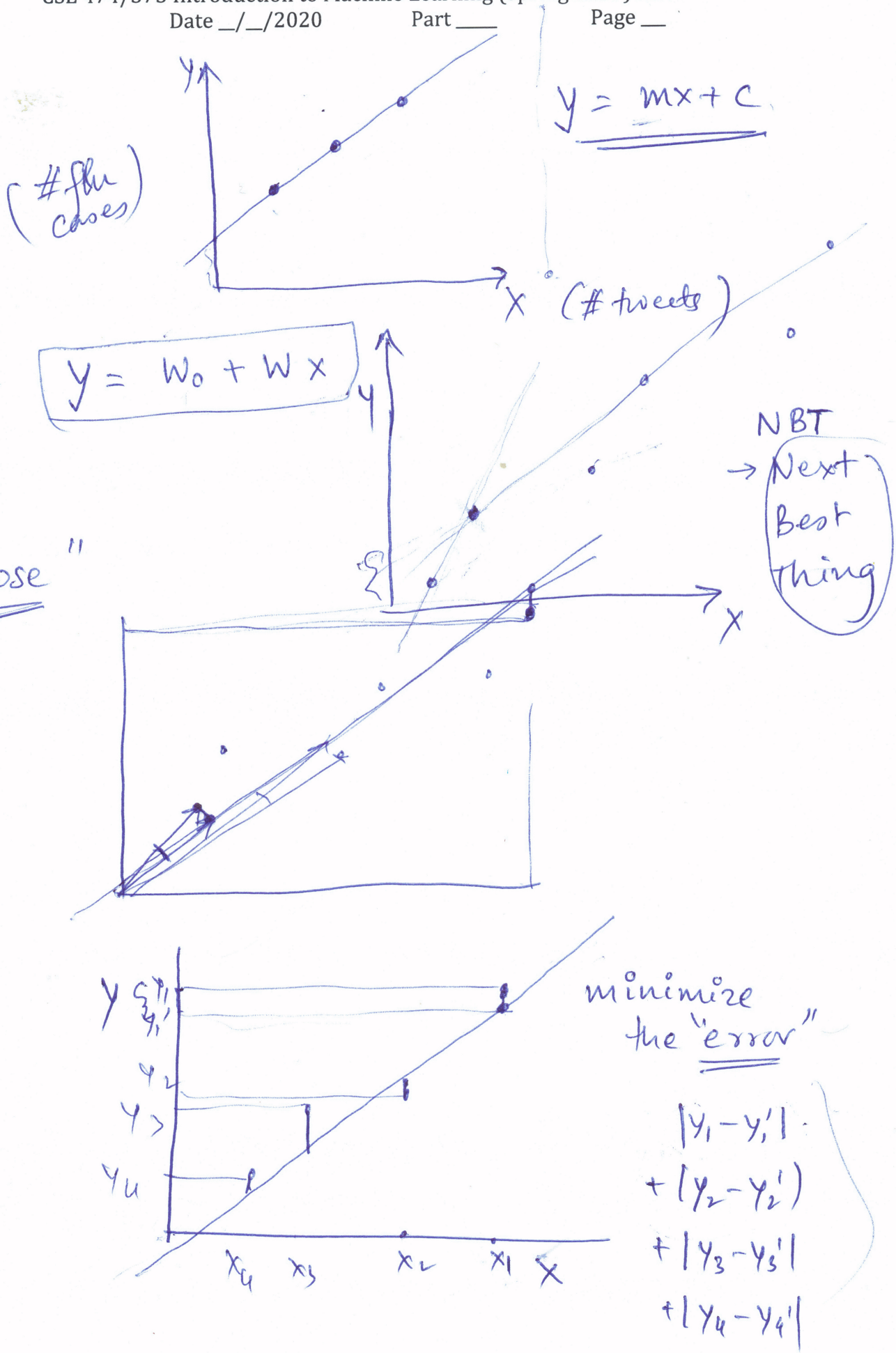
Training Data

$x_5 = 710$ $y_5 = ?$ → Inference

INDUCTIVE BIAS — Reduce the hypothesis space

NO INDUCTIVE BIAS

Linear Models. We assume that f is a straight line


y
x
(# flu cases)
(# tweets)
y = mx + c
y = w0 + w x
"close"
NBT
Next Best thing
minimize the "error"
|y1 − y1'|
+ |y2 − y2'|
+ |y3 − y3'|
+ |y4 − y4'|

$y = w_0 + wx$

x_1	y_1	$y_1' = w_0 + wx_1$
x_2	y_2	$y_2' = w_0 + wx_2$
x_3	y_3	$y_3' = w_0 + wx_3$
x_4	y_4	$y_4' = w_0 + wx_4$

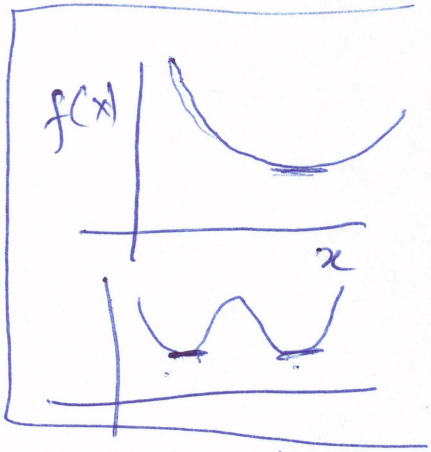
Parameters of the model

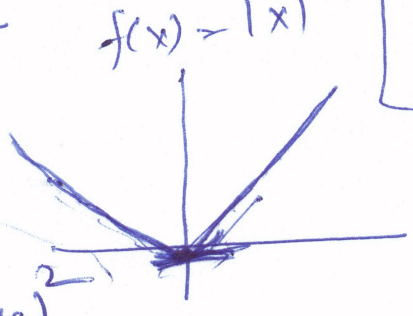
$$|(w_0 + wx_1) - y_1| + |(w_0 + wx_2) - y_2| + \cdots$$

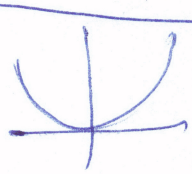
$$J = \frac{1}{N}\sum_{i=1}^{N} |(w_0 + wx_i) - y_i| \longrightarrow \text{Min. Abs. Dev.}$$

$J(w_0, w)$

Optimization Problem







$$J = \frac{1}{N}\sum_{i=1}^{N}((w_0 + wx_i) - y_i)^2 \leftarrow \text{Squared Loss}$$

(Linear regression Model)

$$J = \frac{1}{2}\sum \left(y_i - (w_0 + w x_i)\right)^2$$ ← objective functi'n

$f(u)$	$\frac{d}{du} f(u)$	$f'(u)$

$$\frac{\partial J}{\partial w} = 0$$

$$\frac{\partial J}{\partial w_0} = 0$$

$$\frac{\partial}{\partial w}\left[\frac{1}{2}\sum \left(y_i - (w_0 + w x_i)\right)^2\right]$$

$$\frac{\partial}{\partial w} z^2 = \frac{\partial z^2}{\partial z}\frac{\partial z}{\partial w}$$

$$= \frac{1}{2}\sum\left[\frac{\partial}{\partial w}\left(y_i - (w_0 + w x_i)\right)^2\right]$$

$$= \frac{1}{2}\sum\left[2\left(y_i - (w_0 + w x_i)\right)^{2-1}(-x_i)\right]$$

$$= -\sum \left(y_i - (w_0 + w x_i)\right) x_i$$

$$\frac{\partial}{\partial w_0}\left[\frac{1}{2}\sum\left(y_i - (w_0 + w x_i)\right)^2\right]$$

$$= -\sum\left(y_i - (w_0 + w x_i)\right)$$

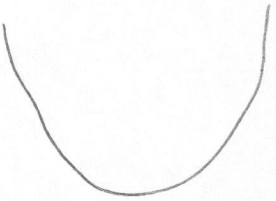
for w, w_0 to be the minima points.

$$\frac{\partial J}{\partial w} = 0 \quad \Big| \quad -\sum (y_i - (w_0 + w x_i)) x_i = 0$$

$$\frac{\partial J}{\partial w_0} = 0 \quad \Big| \quad -\sum (y_i - (w_0 + w x_i)) = 0$$

Solve these to get w and w_0

Do at home



What if we have multiple input features.

	#tweets	#Commuters in public station			#flu cases
~~x_1~~	x_{11}	x_{12}	- - - -	x_{1D}	y_1
x_2	x_{21}	x_{22}	- - - -	x_{2D}	y_2
	⋮				⋮
	x_{N1}	x_{N2}	- - - -	x_{ND}	y_N
	w_0	w_1	w_2	w_D	

Prediction at x_1 = $(w_0 + w_1 x_{11} + w_2 x_{12} + \cdots + w_D x_{1D})$

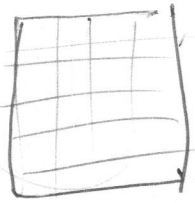
or Prediction at any x_i $= w_0 + w_1 x_{i1} + \cdots + w_D x_{iD}$

$$= w_0 + \sum_{j=1}^{D} w_j x_{ij}$$

$$J(w_0, w_1, \ldots w_D) = \frac{1}{2}\sum_{i=1}^{N}\left(y_i - \left(w_0 + \sum_{j=1}^{D} w_j x_{ij}\right)\right)^2$$

Matrix

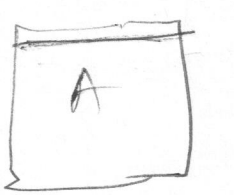
$A_{m \times n}$

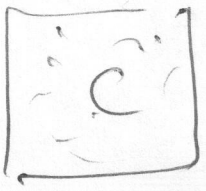


A B
$m \times n$ $n \times k$

Transpose

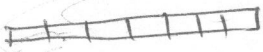
$n \times m$ A^T





Vector m

$\mathbf{a}$ (m)



dot-product or inner-product

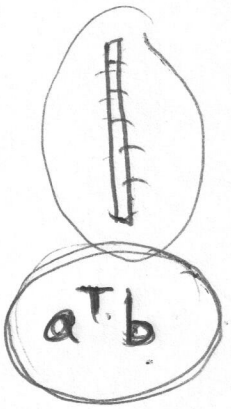
$$\mathbf{a} \cdot \mathbf{b} = \sum_{i=1}^{m} a_i b_i$$

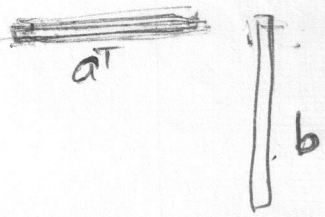
$m \rightarrow$ length of vectors a & b

Vector is a 1-column matrix.

$\mathbf{a}$ is a $(m \times 1)$ matrix

A dot-product between $\mathbf{a}$ $(m \times 1)$ and $\mathbf{b}$ $(m \times 1)$





$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_D \end{bmatrix}$$

weight vector

$$\mathbf{x}_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{iD} \end{bmatrix}$$

I can now write the prediction as

$$y_i' = (w_0 + w^T x_i)$$

$$w_0 + \mathbf{w}^T \mathbf{x}_i \equiv w_0 + \sum_{j=1}^{D} w_j x_{ij}$$

What if $x_i \equiv \begin{bmatrix} 1 \\ x_{i1} \\ x_{i2} \\ \vdots \\ x_{iD} \end{bmatrix}$ and $\mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ \vdots \\ w_D \end{bmatrix}$

$$w_0 + w^T x_i \equiv W^T x_i$$

Now w ~~is a $(D+1) \times 1$~~ is a $(D+1) \times 1$ vector.

Each x_i is also a $(D+1) \times 1$ vector.

$$J(w) = \frac{1}{2} \sum_{i=1}^{N} (y_i - w^T x_i)^2$$

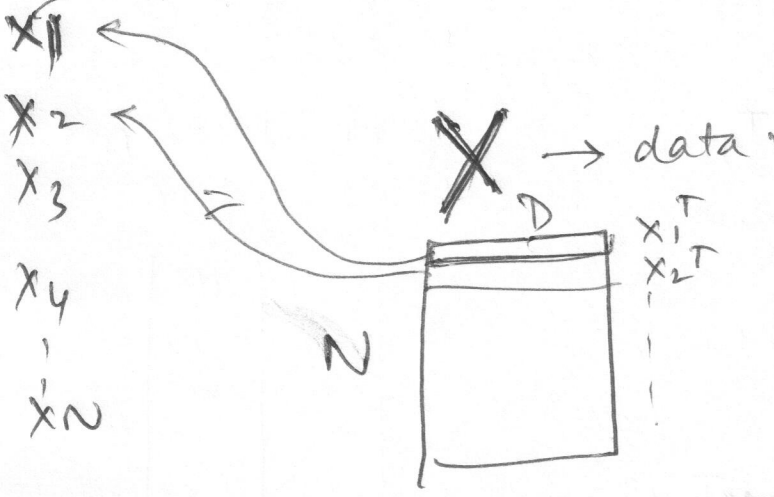
$f(x) = x^2$

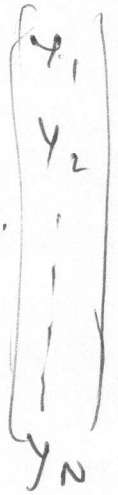
$\frac{d}{dx} f(x) = 2x$

$f(x) = \frac{1}{2} x^2$

$\frac{d}{dx} f(x) = x$

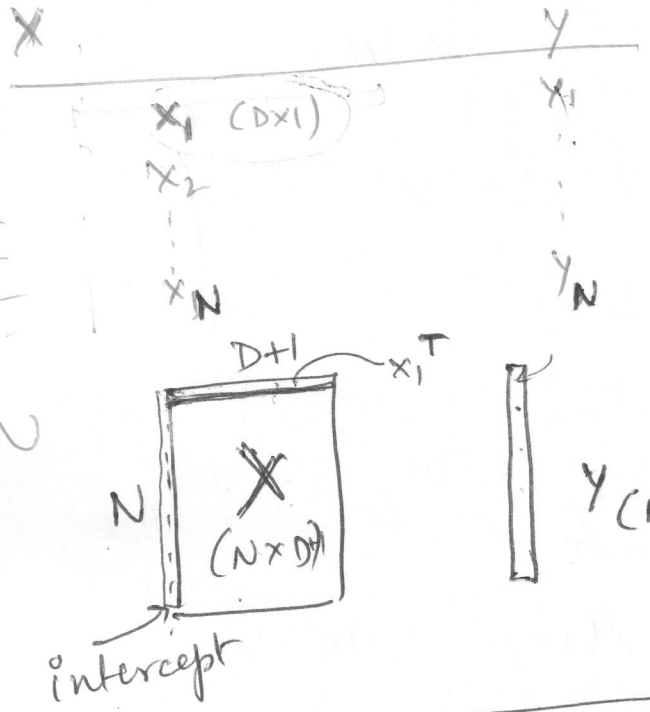
Training data is $\rightarrow$ is a $(D+1 \times 1)$ matrix

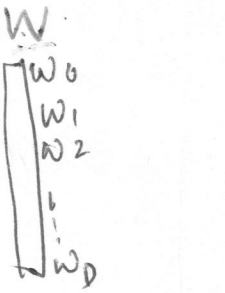




$$J(w) = \frac{1}{2} (y - Xw)^T (y - Xw)$$

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





$((D+1)\times 1)$

$a^T b = b^T a$

$a^T a = \sum_{j=1}^{D} a_j^2$

$(AB)^T = B^T A^T$

$$\frac{1}{2}\sum_{i=1}^{N}(y_i - x_i^T w)^2$$

$x_i^T w = w^T x_i$

$$Xw = \begin{bmatrix} w^T x_1 \\ w^T x_2 \\ \vdots \\ w^T x_N \end{bmatrix}$$

$x_1^T w \rightarrow$ Prediction at x_1

~~$(D+1)\times 1$~~

$N \times 1$

$$y - Xw = \begin{bmatrix} y_1 - w^T x_1 \\ y_2 - w^T x_2 \\ \vdots \\ y_N - w^T x_N \end{bmatrix}$$

$$\frac{1}{2}(y - Xw)^T (y - Xw) = \frac{1}{2}\sum_{i=1}^{N}(y_i - x_i^T w)^2$$

$$J(w) = \frac{1}{2}(y - Xw)^T (y - Xw)$$

Find $\mathbf{w}$ that minimizes $J(w)$

$(A+B)^T = A^T + B^T$

$$J(w) = \frac{1}{2}\left(y^T - (Xw)^T\right)(y - Xw)$$

$$= \frac{1}{2}\left[y^T y - y^T Xw - (Xw)^T y + (Xw)^T Xw\right]$$

$$= \frac{1}{2}\left[y^T y - 2(y^T X)w + w^T X^T X w\right]$$

$$= \frac{1}{2}\left[y^T y - 2w^T (y^T X)^T + w^T X^T X w\right]$$

$$\nabla J = \frac{d}{dw}\frac{1}{2}\left[y^T y - 2w^T (y^T X)^T + w^T X^T X w\right]$$

$$= \frac{1}{2}\left[-2(y^T X)^T + 2X^T X w\right]$$

$$\nabla J = 0$$

$$-(y^T X)^T + X^T X w = 0$$

$$-X^T y + X^T X w = 0$$

$$(X^T X)w = X^T y$$

Multiplying both sides by $(X^T X)^{-1}$

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

$f(u,v) = 7u^2 + 3v^3 + 8u + 9$

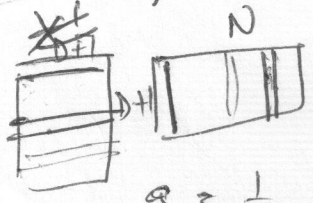
$$\begin{bmatrix} \frac{\partial}{\partial u} f(u,v) \\ \frac{\partial}{\partial v} f(u,v) \end{bmatrix}$$

gradient with respect to (u,v)

$$\frac{d}{da}(a^T b) = b$$

$$\frac{d}{da}(a^T M a) = (M + M^T)a = 2Ma \quad \text{[if } M \text{ is symmetric]}$$

$X^T X$ is symmetric



$a = \frac{1}{a}$

$x^2 - 7x$

$x(x-7)$

What is the complexity of this algorithm.

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

$(D*1) \times 1$

$(D \times 1)$

$O(D^3)$ → # features.

$(X^T X)^{-1}$

$O(N)$ $A^{-1}u$

A^{-1}

b

$\frac{1}{b}$

$$J(w) = (y - Xw)^T (y - Xw)$$

$$\nabla J(w) = -X^T y + X^T X w$$

Start at some w

while not converged:

Calculate $\nabla J(w)$

$$w = w - \eta \nabla J(w)$$

~~check~~

$J(w) \rightarrow$ calculate

subtract from $J(w_{previous})$

$$w = (X^T X)^{-1} X^T y \longrightarrow O(D^3)$$

if $D^3 > N$

x given, y infer

$y = f(x)$ ← parameters $\underline{w}$

X data matrix, y

$$J(w) = \frac{1}{2}(y - Xw)^T(y - Xw)$$

↑ linear regression

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

Stochastic Gradient Descent (SGD)

$$J(w) = \frac{1}{2}\sum_{i=1}^{N}(y_i - w^T x_i)^2 = \sum_{i=1}^{N} J(w_i)$$

If I had only one training data point:

$$J(w) = \frac{1}{2}(y_1 - w^T x_1)^2 \longrightarrow$$ can apply GD to find optimal w

$w \leftarrow w_0$ Initialize

For $i = 1 : N$

$J(w) = \frac{1}{2}(y_i - w^T x_i)^2 \; J(w_i)$

Run GD: $w = w - \eta \nabla J$

→ SGD:

$w = (1-\xi) w_{n-1} + \xi [w - \eta \nabla J]$

Instead of choosing one training example at a time, one can choose a small batch (mini-batch)

Momentum