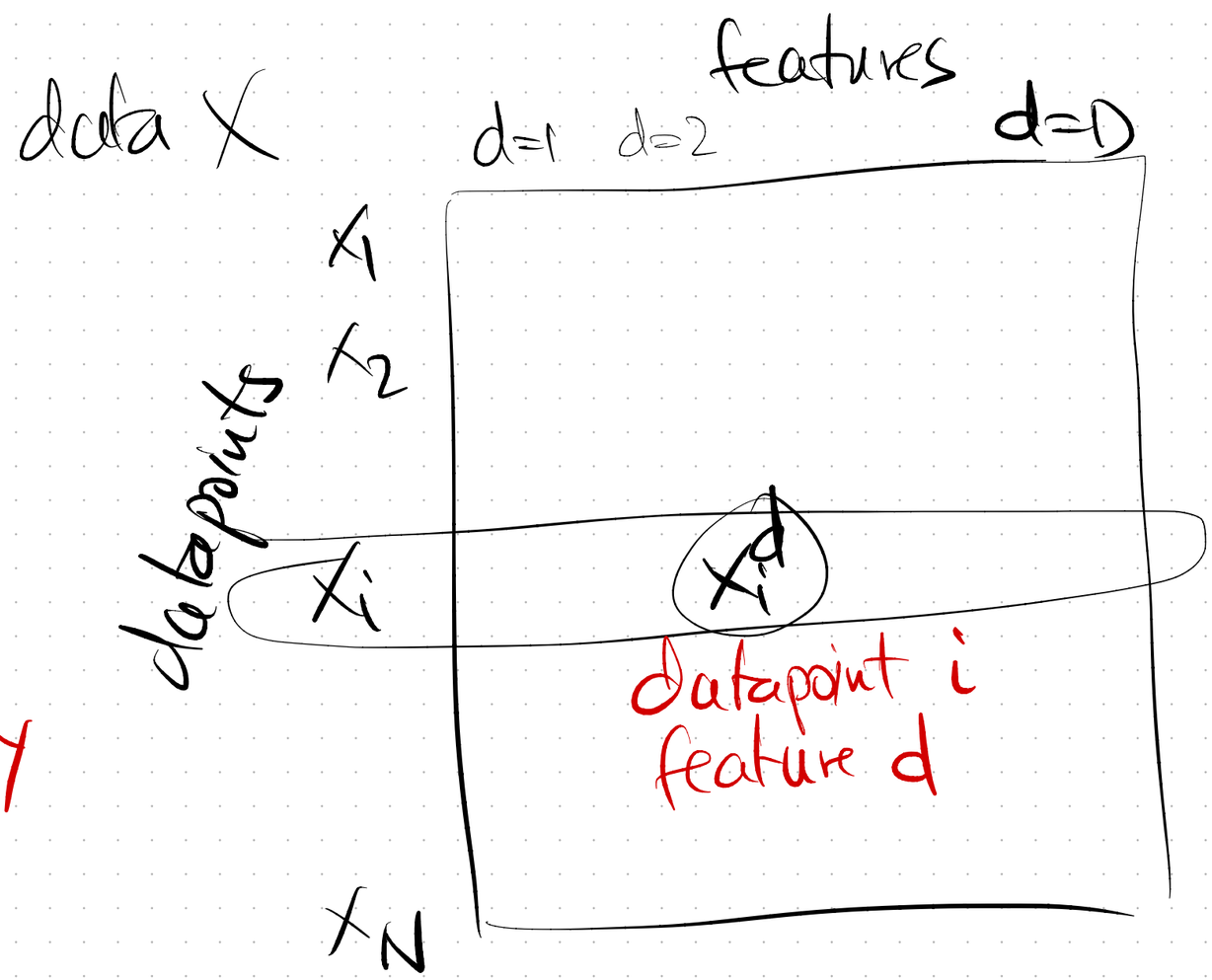
Lecture 5|21-23

- Gradient Descent
- Linear Regression with GD
- Logistic Regression
- HW2A (released later today) due 6|3
- ROC curve ≃ AUC geometrically
- Perceptron w/GD or geometry



Logistic Regression : deal with classification classes $Y \rightleftarrows$ yes / no

ex. Spambase dataset

Lin. Regression $h(x) = xw \triangleq Y$ = quantity label (ex. house price)

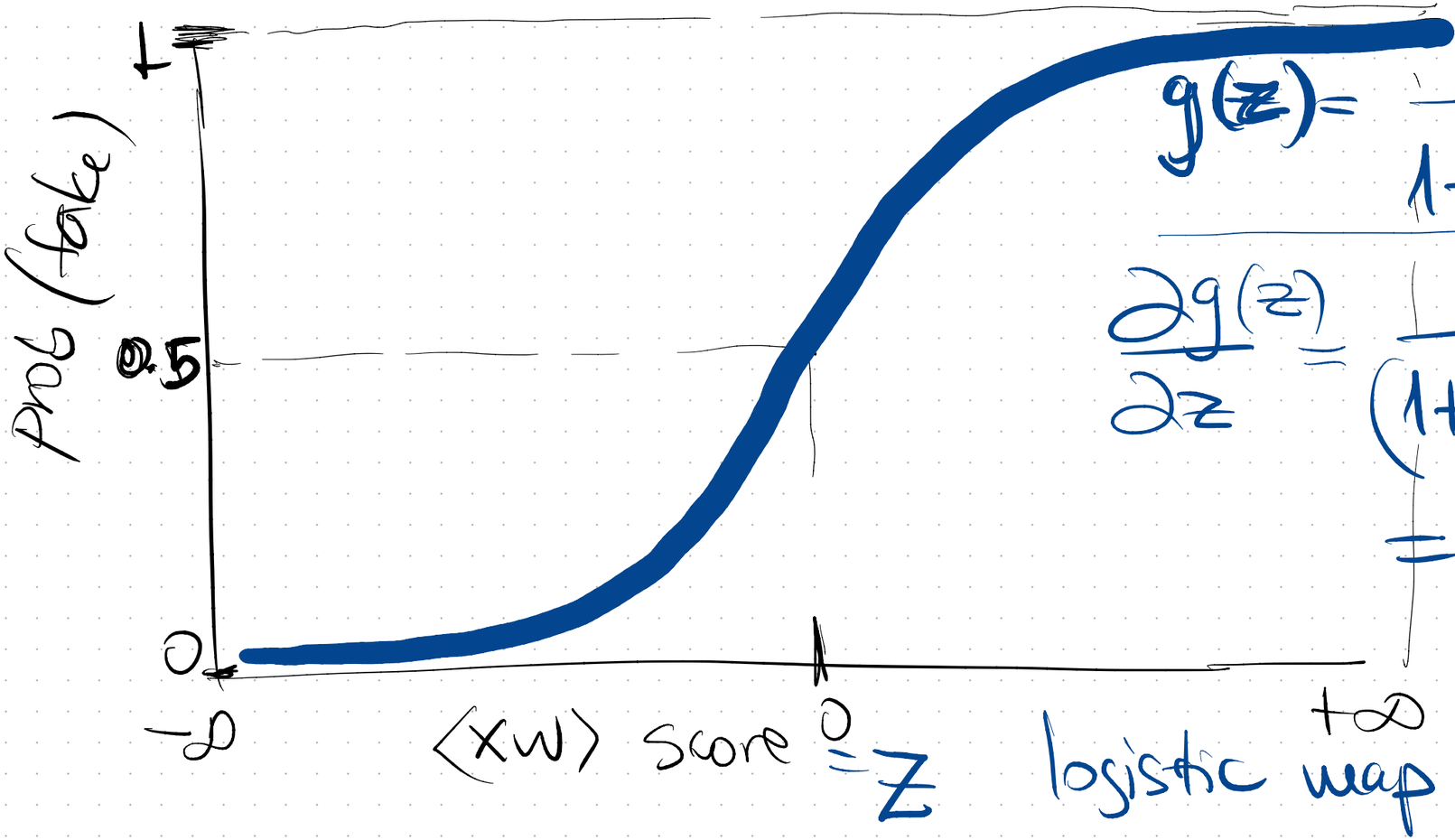
idea • xw = linear regression score

• $h(x) = \boxed{\text{map}}$ $\underbrace{\text{linear score } \langle xw \rangle}$ → probability of Y=yes } fake

prob(label = yes)

logistic map "g"

Z = score = xw



$$g(z) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-xw}}$$

$$\frac{\partial g(z)}{\partial z} = \frac{1}{(1+e^{-z})^2} \cdot e^{-z} =$$

$$= \frac{1}{1+e^{-z}} \left(1 - \frac{1}{1+e^{-z}}\right) =$$

$$= g(z)\left(1 - g(z)\right)$$

- key property: overshooting does not penalize ORN

Lin Regression $Y = \{0, 1\}$

want $xw \simeq Y$

if $xw \ll 0$: error $(xw-0)^2$

if $xw \gg 1$ error $(xw-1)^2$

Logistic Regression

want $\frac{1}{1+e^{-xw}} \simeq Y$

: xw very small $\to -\infty$

$\frac{1}{1+e^{\infty}} \simeq 0$

xw very large $\to +\infty$

$\frac{1}{1+e^{-\infty}} \simeq 1$

Logistic Regression (coef w)

$h(x) = \text{logistic}(\text{Regscore}(x)) = \text{logistic}(xw)$

$= \frac{1}{1+e^{-xw}} \overset{\text{interpret}}{\simeq}$ probab $(\hat{y} = \text{yes})$ ($\hat{y}$: class)

Logistic Regression uses log-likelihood OBJECTIVE (not sq error)

WANT $h(x_i) = g(x_i w) = \frac{1}{1+e^{-x_i w}} \simeq \text{prob}(Y_i = 1)$ 2 classes

$Y_i = \begin{cases} 1 \text{ yes} \\ 0 \text{ No} \end{cases}$

$P(y_i=1) \simeq h(x_i)$
$P(y_i=0) \simeq 1 - h(x_i)$
$\Rightarrow P(Y|x_i) = h(x_i)^{y_i} \cdot (1-h(x_i))^{1-y_i}$

y_i filters correct pred; $1-y_i$ filters

0/1 fillers

$h(x_i)^{y_i} \cdot (1-h(x_i))^{1-y_i} = \begin{cases} h(x_i) \text{ if } y_i = 1 \\ 1-h(x_i) \text{ if } y_i = 0 \end{cases}$

chance of being correct

if $y_i = 1$ (yes) $\Rightarrow$ want $h(x_i)$ = high ; $1-h(x_i)$ low

$y_0 = 0$ (no) $\Rightarrow$ want $h(x_i)$ = low ; $1-h(x_i)$ high

log likelihood "LL"

$LL(w) = \log \left[\prod_{i=1}^{N} P(y_i|x_i) \right]$

product of prob of correct prediction

log of all datap in training

datapoints indep. of each other.

want MAX LL(w)

$$= \log\left(\prod_{i=1}^{N} h(x_i)^{y_i}\left(1-h(x_i)\right)^{1-y_i}\right)$$

$$= \sum_{i=1}^{N} \log\left(h(x_i)^{y_i} \cdot \left(1-h(x_i)\right)^{1-y_i}\right)$$

$$= \sum_{i=1}^{N} \left(\log\left(h(x_i)^{y_i}\right) + \log\left(1-h(x_i)\right)^{1-y_i}\right)$$

$$= \sum_{i=1}^{N} \left[y_i \log\left(h(x_i)\right) + (1-y_i)\log\left(1-h(x_i)\right)\right]$$

all train data

$$\log(a \cdot b) = \log(a) + \log(b) + \log(c)$$

$$\log(a^n) = n \log(a)$$

- gradient $\frac{\partial LL}{\partial w} = ?$
- GD update (iterative) $w_{new} = w + \lambda \frac{\partial LL}{\partial w}(w)$ ("ascent")

$$LL = \sum_{i=1}^{n} [y_i \log(h(x_i)) + (1-y_i)\log(1-h(x_i))] = \sum_{i=1}^{n} [y_i \log(g(x_i w)) + (1-y_i)\log(1-g(x_i w))]$$

$\frac{\partial \log}{\partial z} = \frac{1}{z}$ $\qquad$ $\frac{\partial \log}{\partial z} = \frac{1}{z}$

one datapoint x_i:

$$\frac{\partial LL}{\partial w_d} = \left[y_i \frac{1}{g(x_i w)} - (1-y_i) \frac{1}{1-g(x_i w)} \right] \frac{\partial}{\partial w_d} g(x_i w)$$

rule of compound differentials

$$= \left[y_i \frac{1}{g(x_i w)} - (1-y_i) \frac{1}{1-g(x_i w)} \right] \cdot \frac{\partial g}{\partial (x_i w)} \cdot \frac{\partial x_i w}{\partial w_d} \quad x_i^d$$

$$= \left[y_i \frac{1}{g(x_i w)} - (1-y_i) \frac{1}{1-g(x_i w)} \right] \cdot \left[g(x_i w)(1-g(x_i w)) \right] \quad x_i^d$$

$$= [y(1-g(x_i w)) - (1-y_i) \cdot g(x_i w)] \quad x_i^d$$

$$= [y - y_i g(x_i w) - g(x_i w) + y_i g(x_i w)] \quad x_i^d$$

$$= [y - g(x_i w)] x_i^d = [y - h(x_i)] x_i^d$$

GD update for datap x_i: $w^d_{new} = w^d + \lambda(y_i - h(x_i))x_i^d$

GD update for all datap: $w^d_{new} = w^d + \lambda \sum_{i=1}^{N} (y_i - h(x_i))x_i^d$

GD update all comp(d) all datap(i): $W_{new} = w + \lambda [y - h(x)] x$

Regularization Logistic Regression

$L_2 \cdot \sum_{d=1}^{D} (w^d)^2$

$$OBJ = \text{LogLikelihood}(w) + L_2 \|w\|^2 \longrightarrow \text{easy}$$

$$\text{LogLikelihood}(w) = \sum_{i=1}^{N} h(x_i)^{y_i} (1-h(x_i))^{(-y_i)}$$

$$OBJ = \text{LogLikelihood} + L_1 |w|$$

$L_1 \cdot \sum_{d=1}^{D} |w^d| \rightarrow$ not easy.

ML training:

- $\frac{\partial OBJ}{\partial w} = \nabla_w$ (gradient)
- GD update $w_{new} = w + \lambda \cdot \nabla w$

Multiple classes (multinomial) $Y \in \{1, 2, 3, \dots K\}$ K classes

- train a linear score (like Regression) for each class

$$f_k(x) = \langle x \cdot w_k \rangle = \sum_{d=1}^{D} x^d \cdot w_k^d$$

w_k = vector of coef for class k

- binary logistic → k classes Softmax

$$\text{prob}(Y = y_k) = \frac{e^{f_k(x)}}{\text{unnormalized}} = \frac{\exp(f_k(x))}{\sum_{t=1}^{k} \exp(f_t(x))}$$

↓

distribution over possibilities

$Y = y_1 \, y_2 \cdots y_k$

Newton Update instead of GD

unidim:

$$w_{new} = w - \frac{\nabla LL(w)}{\nabla^2 LL(w)}$$

1st diff

2nd diff.

$LL = obj(w)$

∇LL = gradient $\frac{\partial LL}{\partial w}$

$\nabla^2 LL$ = second grad $\frac{\partial LL}{\partial w \partial w}$ (twice differentiated)

Intuition: $f_{(w)} = LL(w)$. use Taylor series expansion for $f(w)$ at point w_n = current w at iteration n

$$f(w) = f(w_n) + \frac{\partial f(w_n)}{\partial w}\frac{(w-w_n)}{1!} + \frac{\partial f(w_n)}{\partial w \partial w}\frac{(w-w_n)^2}{2!} + \frac{\partial f(w_n)}{\partial w \partial w \partial w}\frac{(w-w_n)^3}{3!}$$

$$\approx f(w_n) + \frac{\partial f(w_n)}{\partial w}(w-w_n) + \frac{1}{2}\frac{\partial f(w_n)}{\partial w \partial w}(w-w_n)^2 \quad (\text{two terms})$$

solve for w

assume $f(w) \approx f(w_n)$

$$\frac{\partial f(w_n)}{\partial} + \frac{1}{2}\cdot 2\frac{\partial f}{\partial^2 w}(w-w_n) \overset{want}{=} 0$$

$\partial = 0$

$$-\partial f = \partial^2 f\,(w-w_n)$$

$$w = w_n - \frac{\partial f}{\partial^2 f}(w_n)$$

multidim $w = (w^1 w^2 \dots w^D)$:

first differential is $\underset{\text{vector}}{\nabla f} = \frac{\partial f}{\partial w} = \left(\frac{\partial f}{\partial w^1}, \frac{\partial f}{\partial w^2} \dots \frac{\partial f}{\partial w^d}\right)$

2nd differential is Hessian Matrix partial second derv.

$$D^u \begin{bmatrix} \frac{\partial f}{\partial w^1 \partial w^1} & \cdots & \frac{\partial f}{\partial w^1 w^D} \\ \vdots & & \vdots \\ \frac{\partial f}{\partial w^u \partial w^1} & \cdots & \frac{\partial f}{\partial w^u \partial w^v} \\ \vdots & & \vdots \\ \frac{\partial f}{\partial w^d w^1} & \cdots & \frac{\partial f}{\partial w^d w^D} \end{bmatrix}^V$$

GD update: $w_{new} = w - \lambda H^{-1} \nabla f(w)$

learn rate

intuition: $\frac{\partial f}{\partial^2 f}$ 1st der / 2nd dif

Newton: Hessian for LL=OBJ of Logistic Regression

$H = X^T \underline{DIAG} \cdot X$ - DIAG: