Artificial Intelligence/Machine Learning/Deep Learning:
'Bridging the Skills Gap'

Optional: Normal Equation

There is an analytical/closed form solution to a linear regression problem with Mean Squared Error as cost function → no gradient descent needed!

So why don't we use this all the time?

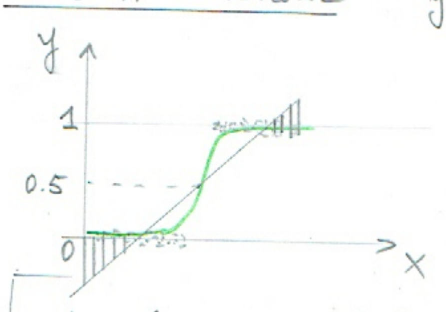
- Because it is heavy taxation on the memory → requires d^2 memory → avoid when dimension d of feature vector is large
- X^TX requires lots of memory if dimension of X is large
- Inverting X^TX is also complex
-

logistic Regression → classifier < BINARY / Multi-class

Problem statement

y_i is a categorical variable

→ $y=0$ 'no cancer'
$y=1$ 'cancer'



↳ linear Regression $y_p = w_0 + w_1x_1 + w_2x_2 + \dots$

values < 0
> 1 → NOT GOOD!

values between 0 and 1 → NOT Good!

Definition: Odds Ratio : $\frac{\text{Successes}}{\text{Failures}}$

6 coin tosses ↓ 4 Hs, 2 Ts

→ $\frac{4}{2}$ → 2 to 1

↓

BERNOULLI $B \sim p$ ← Probability of success vs Probability of failure → $\frac{4/6}{2/6}$ → $\frac{4}{2}$ → 2

$\xrightarrow{ln}$ $\underbrace{\ln\left(\frac{p}{1-p}\right)}_{\text{log odds}} = \underbrace{w_0 + w_1x_1 + \dots + w_dx_d}_{W^Tx}$ → linear function of features x_i

$$\rightarrow \frac{p}{1-p} = e^{+W^Tx}$$

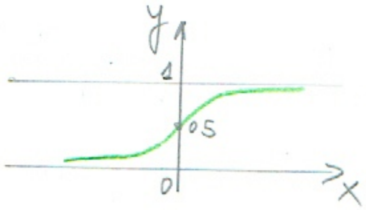
$$\rightarrow p = \left(e^{+W^Tx}\right)(1-p)$$

$$= e^{+W^Tx} - pe^{+W^Tx}$$

$$\rightarrow e^{+W^Tx} = p\left(1 + e^{+W^Tx}\right)$$

$$\rightarrow p = \frac{e^{+W^Tx}}{1+e^{+W^Tx}} \cdot \frac{e^{-W^Tx}}{e^{-W^Tx}} \rightarrow \boxed{p = \frac{1}{1+e^{-W^Tx}}}$$

probability of success



$P(y|x) > 0.5 \rightarrow y=1$ → CANCER

$P(y|x) \leqslant 0.5 \rightarrow y=0$ → NO CANCER

Naive Bayes $P(y|x) \sim \underbrace{P(x|y) \cdot P(y)}$

you get $P(y|x)$ by estimating $P(x|y)$ and $P(y)$

$P(x|y)$ → likelihoods

$P(y)$ → PRIORS

logistics Regression $\boxed{P(y|x) = \dfrac{1}{1+e^{-W^T x}}} = \sigma(W^T x_i)$ ← Sigmoid

Gradient $D = \{(x_1, y_1), \ldots, (x_d, y_d)\}$ m samples

$y_i \sim$ BERNOULLI

MLE: $\underset{W}{\text{Argmax}}\ P(D|W)$ → Find Ws that maximizes the likelihood of seeing the DATA D

$$P(D|W) = \prod_{i=1}^{m} P(y^{(i)}|x^{(i)}, w)$$

SET $\alpha^{(i)} = \sigma(W^T x^{(i)})$

$$= \prod_{i=1}^{m} \alpha^{(i)\, y^{(i)}} (1-\alpha^{(i)})^{1-y^{(i)}} \quad y_i \in \{0,1\}$$

↑ pmf BERNOULLI

→ We will not be able to solve for W because non-linearity of α

→ Newton's method $\begin{cases}\nabla \\ \mathcal{H}\end{cases}$ → 2nd order optimization ③

- no α
- Very fast convergence

$$\boxed{w_1 = w_0 - \frac{c'(w_0)}{c''(w_0)}}$$

$\rightarrow \mathcal{H}^{-1}$ → can be challenge to calculate

$\xrightarrow{\text{newton}}$ if $d < 1000$

$$\mathcal{L}(w) \overset{\text{SET}}{=} \underbrace{-\log p(D|w)}_{-\text{log-likelihood}} \rightarrow \text{WE WANT to minimize!}$$

$$= -\sum_{i=1}^{m} y_i \log \alpha_i + (1-y_i)\log(1-\alpha_i)$$

INTERMEZZO: $\frac{\partial}{\partial w_j} \log \alpha_i = + \frac{x_j e^{-w^T x}}{1+e^{-w^T x}} = \boxed{x_j(1-\alpha)}$

$$\log \alpha = \log \sigma(w^T x) = \log \frac{1}{1+e^{-w^T x}} = 0 - \log(1+e^{-w^T x})$$

$$\log(1-\alpha) = -w^T x - \log(1+e^{-w^T x})$$

$$\frac{\partial}{\partial w_j} \log(1-\alpha) = -x_j + x_j(1-\alpha) = \boxed{-\alpha x_j}$$

$$\rightarrow \mathcal{L}(w) = -\sum_{i=1}^{m} y_i \log \alpha_i + (1-y_i)\log(1-\alpha_i)$$

$$\frac{\partial \mathcal{L}(w)}{\partial w_j} = -\sum_{i=1}^{m} y_i x_{ij}(1-\alpha_i) - (1-y_i)x_{ij}\alpha_i$$

$$= -\sum_{i=1}^{m} y_i x_{ij} - \cancel{y_i x_{ij}\alpha_i} - x_{ij}\alpha_i + \cancel{y_i x_{ij}\alpha_i}$$

$$= \sum_{i=1}^{m} (\alpha_i - y_i)\, x_{ij}$$

(predict - real y_i)

DOT product of column j of X WITH $(\alpha - y_i)$

DOT

1st ROW $(x_1)^T$

design MATRIX

$$X = \begin{pmatrix} x_{11} & x_{i1} & x_{1d} \\ \vdots & \vdots & \vdots \\ x_{m1} & x_{im} & x_{md} \end{pmatrix}$$

d - features $\rightarrow x_{i1}, \dots, x_{id}$ $\downarrow$ i-th sample!

m - samples

$$\rightarrow (\alpha - y)^T X \rightarrow \left((\alpha - y)^T X\right)^T = \underbrace{\boxed{X^T(\alpha - y)}}_{\nabla_W \mathcal{L}}$$

<u>next</u>: Hessian

$$\mathcal{H}_W \rightarrow \frac{\partial^2}{\partial w_j \partial w_k} \mathcal{L}_W$$

jk entry of $\mathcal{H}$

$$\mathcal{H}_W \Rightarrow \sum_{i=1}^{m} x_{ij} \frac{\partial}{\partial w_k} \alpha_i \quad (1)$$

$$\partial \log \alpha = \frac{\partial \alpha}{\alpha}$$

$$\partial \alpha = \alpha\, \partial \log \alpha = \alpha\, x_j (1-\alpha)$$

$$\frac{\partial}{\partial w_k} \alpha_i = x_{ik}\, \alpha_i (1-\alpha_i)$$

$\downarrow$ x_i

$$(1) = \sum_{i=1}^{m} \overbrace{x_{ij} x_{ik}}^{\rightarrow \text{ROW}^{(i)}} \alpha_i (1-\alpha_i)$$

$$= z_j^T B z_k$$

WITH $z_j = (x_{1j}, \dots, x_{mj})$, $z_k = (x_{1k}, \dots, x_{mk})$

DIAGONAL

WITH $B \begin{pmatrix} \alpha_1(1-\alpha_1) & 0 \\ 0 & \ddots \; \alpha_m(1-\alpha_m) \end{pmatrix}$

$x_j = (x_{j1}, \dots, x_{jd})^T$ → ROW (i)

$z_j = x_{ij}, \dots, x_{mj}$ → COLUMN (j)

to make it a ROW

$X^T B X$

→ $\boxed{\nabla^2_W \mathcal{L} = X^T B X}$

positive semi Definite!

$\mathcal{L}$ is convex!

$\alpha_i = \sigma(w^T x_i)$

↳ always > 0 AND < 1

$0 < \alpha_i (1-\alpha_i) < 1$

Newton: → iterative Reweighted least squares!

↳ Fast!

$w_{t+1} = w_t - H^{-1} \nabla$ (see page 3)

→ $\boxed{w_{t+1} = w_t - \underbrace{(X^T B A)^{-1}}_{\text{ASSUME is invertible}} . X^T (\alpha - Y)}$

$= w_t = (X^T B X)^{-1} X^T B \underbrace{(A w_t - B^{-1}(\alpha - Y))}_{z_t}$

$= w_t = (X^T B X)^{-1} X^T B z_t$