$$P(y=c|x;\theta) \propto \underbrace{P_\theta(x|y;\theta)}_{N} P_\theta(y)$$

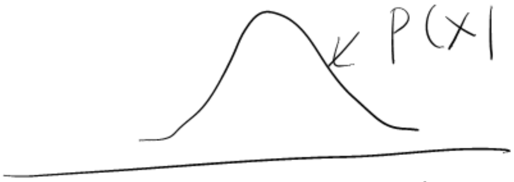
$$\propto P_\theta(x,y;\theta)$$ ← generative classifier

$P_\theta(x,y)$

$P(x; y=c)$

side $= N(x|\mu,\sigma^2)$

$P(x|y=cat)$



For class $c = cat$

x_m

$P(Y=c \mid X; \theta)$

In discriminative classifier we model this as a function $X; \theta$

we will write a diret function of X giving us the probability

For Binary case logistic Regression [classification but via regression]

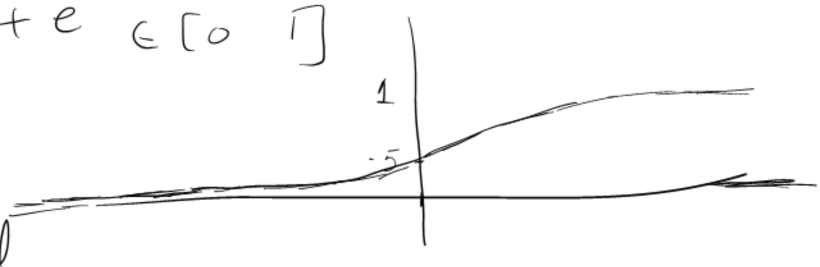
$$P(Y=1 \mid X; W) = Ber(Y_1 \mid \sigma(W^T X)) = \sigma(W^T X)$$

$X \in \mathbb{R}^D$

Side what is σ (sigmoid function) $= \frac{1}{1+e^{-W^T X}}$

$\sigma(x) = \frac{1}{1+e^{-x}} \in [0 \ 1]$ here $x \in \mathbb{R}$

Sigmoid takes a real number and map to $[0 \ 1)$ interval



$\sigma'(x) = \sigma(1-\sigma(x))$

Popular will see that it is easy to extend For multi-class

- using Kernel trick can model non-linear $\phi(x)$

what is decision surface

if $p(y=1 \mid x, w) = p(y=0 \mid x, w)$

$$\frac{1}{1+e^{-W^T X}} = 1 - \frac{1}{1+e^{-W^T X}}$$

$$\frac{e^{W^T X}}{\cancel{e^{W^T X}+1}} = \frac{e^{-W^T X}}{\underbrace{1+e^{-W^T X}}} = \frac{1}{\cancel{e^{W^T X}+1}}$$

$$e^{W^T X} = 1$$

take $\log_e$ $\quad W^T X = 0$ or $W_1 X_1 + W_2 X_2 + \quad W_d X_d = 0$

note $x = \begin{bmatrix} x_1 \\ x_2 \\ x_d \\ 1 \end{bmatrix}$

Hence decision boundary is a line in high dim or is hyperplane

Model Fitting (estimate w)

$$D = \{ \{x_i, y_i\} \}_{i=1}^{N}$$

conditional log likelihood

I.I.D

$$\log P(D) = \log \prod_{i=1}^{N} p(y_i | x_i; w)$$

$$= \log \prod_{i}^{N} \sigma(w^T x_i)^{\mathbb{I}(y_i=1)} (1 - \sigma(w^T x_i))^{\mathbb{I}(y_i=0)}$$

$\mathrm{Ber}(x, \theta)$
$= \theta$ if $x = 1$
$= 1-\theta$ if $x = 0$

side

$\mathrm{Ber}(x, \theta) = \theta^{x} (1-\theta)^{1-x}$

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

$$NLL(D, w) = - \sum_{i=1}^{N} \left(y_i \log \sigma(w^T x_i) + (1-y_i) \log (1 - \sigma(w^T x_i)) \right)$$

$$\mathrm{Ped}(x_i) = \begin{bmatrix} \sigma(w^T x_i) \\ 1 - \sigma(w^T x_i) \end{bmatrix}$$ also see $\mathrm{oneHot}(y_i)$ $\begin{bmatrix} y_i \\ 1-y_i \end{bmatrix}$

$\Sigma = 1$

$$= - \sum_{i=1}^{N} \mathrm{cross}(\mathrm{Ped}(x_i), \mathrm{oneHot}(y_i))$$

$a = \begin{bmatrix} a_1 \\ \vdots \\ a_k \end{bmatrix}$ $b = \begin{bmatrix} b_1 \\ \vdots \\ b_k \end{bmatrix}$

$\mathrm{cross}(a, b) = -\sum_{i=1}^{k} a_i \log b_i$

$-\sum_{i=1}^{k} a_i \log a_i$ entropy

can't write MLE solution in closed form?

Solution optimazation algorithms (iterative)

$$\theta_{k+1} = \theta_k - \eta \left(\frac{\partial NLL(\theta)}{\partial \theta} \right)_{\text{at } k}$$

η step size Learning rate

derivative (direction of steep descent)

effect of this is

$$NLL(\theta_{k+1}) \leq NLL(\theta_k)$$

$$\theta_{k+1} = \theta_k - \eta \left(\frac{\partial NLL(\theta)}{\partial \theta}\right)_k + \mu(\theta_k - \theta_{k-1})$$

Momentum update $\qquad 0 \le \mu \le 1$

$D = \{x_i, y_i\}_{i=1}^{N}$

$T = \{x_i, y_i\}_{i=1}^{N/2} \quad V = \{x_i, y_i\}_{i=N/2}^{N}$

8.3.6

$$f(w) = \underbrace{NLL(w)}_{+ve} + \lambda \underbrace{||w||^2}_{+ve}$$

hyper parameter

regularization (weight decay)

can LASSO

or $f(w) = NLL(w) + \lambda |w|$