Logistic regression $P(y|x,w)$

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

$$\begin{cases} x_i = (x_1, x_2, x_3, \cdots x_n)^T \quad (\uparrow \text{feature}) \\ y_i \in \{0, 1\} \end{cases}$$

Probabilistic Model

$$p(y|x,w) = \boxed{Ber(y|\sigma(w^Tx))}$$

Sigmoid Func: $\sigma(x) = \frac{e^x}{e^x+1}$

$$Ber(y|\sigma(w^Tx)) = [\sigma(w^Tx)]^y \cdot [\sigma(w^Tx)]^{1-y}, \; y\in\{0,1\} \Leftrightarrow \begin{cases} P(y=1|x,w) = \sigma(w^Tx) \\ P(y=0|x,w) = 1-\sigma(w^Tx) \end{cases}$$

Since:
$$\log \frac{P(y=1|w,x)}{1-P(y=1|w,x)} = \log \frac{P(y=1|w,x)}{P(y=0|w,x)} = \log \frac{\frac{e^{w^Tx}}{1+e^{w^Tx}}}{\frac{1}{1+e^{w^Tx}}} = w^Tx$$

"Logit" ⇒ Logistic Rregression

Classify "$\hat{y} = \arg\max_y p(y|x,w)$"

⇒ if $P(y=1|w,x) > P(y=0|w,x)$ then $y=1$
else $y=0$ $\Leftrightarrow P(y=1|x,w) > 0.5$

if $w^Tx > 0$ then $P(y=1|w,x) > P(y=0|w,x) \Rightarrow \hat{y}=1$

else $P(y=0|w,x) > P(y=1|w,x) \Rightarrow \hat{y}=0$

∴ A linear Classifier!

How to learn?

1. Principles

Minimize Cross Entropy

⇔ Maximize likelihood

Entropy 熵: $H(x) = \sum_x P(x)\log\frac{1}{P(x)} = -\sum_x P(x)\log P(x)$

"Uncertainty" ↑ ⇒ Entropy ↑.

"K-L" divergence $D_{KL}(P\|Q) = \sum_x P(x)\log\frac{P(x)}{Q(x)}$

(P(x): "True Distribution"; Q(x): A random Distribution)

How much a distribution $Q(x)$ diverge from a "true" probability distribution.

$D_{KL}(P\|Q) \geqslant 0$, equality holds iff $P(x)=Q(x)$

$$D_{KL}(P\|Q) = \sum_x P(x)\log\frac{1}{Q(x)} - \sum_x P(x)\frac{1}{P(x)}$$
$$= H(P,Q) - H(P)$$

$H(P,Q)$: "Cross Entropy"; $H(P)$: Entropy (Will remain unchanged)

Conditional Cross Entropy

$$H(P,Q) = \int P(x,y)\log\frac{1}{Q(y|x)}\,dx\,dy$$

$H(P,Q)$ ↑ $Q(y|x)$ is far from $P(y|x)$
$H(P,Q)$ ↘ $Q(y|x)$ is close to $P(y|x)$

"Supervised Learning"

$$P(x,y) \xrightarrow{\text{Sampling}} D=\{x_i,y_i\}_{i=1}^{N} \xrightarrow{\text{learning}} Q(y|x)$$

($P(x,y)$: "Unknown Distribution")

$$H(P,Q) = \int P(x,y)\log\frac{1}{Q(y|x)}\,dx\,dy$$
$$= -\int P(x,y)\log Q(y|x)\,dx\,dy$$
$$\approx -\frac{1}{N}\sum_{i=1}^{N}\log Q(y_i|x_i) \quad \text{Minimize!}$$

"用经验分布近似真实分布"

Likelihood

$$L(\theta|D) = P(D|\theta)$$

在θ的条件下，形成D的分布的概率

Maximum Likelihood Estimation (MLE)

$$L(\theta^*|D) = \arg\max_\theta L(\theta|D)$$

i.i.d.:

1. independence: $P(D_1, D_2, \cdots, D_m|\theta) = \prod_{i=1}^{m} P(D_i|\theta)$

2. identically distributed:

∴ $P(D_i=H)=\theta$, $P(D_i=T)=1-\theta$ for all i.

$$\Rightarrow L(\theta|D) = P(D|\theta) = P(D_1|\theta)\cdot P(D_2|\theta)\cdots P(D_m|\theta)$$
$$= \theta^{m_h}(1-\theta)^{m_t} \quad \text{Binomial Likelihood}$$

Where $m_h \leftarrow$ # heads
$m_t \leftarrow$ # tails.

⇒ loglikelihood.

$\log L(\theta|D) = m_h\log\theta + m_t\log(1-\theta)$

denoted $l(\theta|D)$

$$\frac{d\,l(\theta|D)}{d\theta} = m_h\cdot\frac{1}{\theta} - m_t\frac{1}{1-\theta}$$

When derivative → 0.

$$m_h\cdot\frac{1}{\theta} = m_t\cdot\frac{1}{1-\theta}.$$

$$\frac{m_h}{m_t} = \frac{\theta}{1-\theta} \Rightarrow \theta^* = \frac{m_h}{m_h+m_t} = \frac{m_h}{m}$$

Cross Entropy for logistic Rregression

$$L(w) = -\frac{1}{N}\sum_{i=1}^{N}\log P(y|x_i,w)$$

Since y is binary.

$$P(y|x,w) = \hat{y}^y\cdot(1-\hat{y})^{1-y}$$

$$\log P(y|x,w) = y\log\hat{y} + (1-y)\log(1-\hat{y})$$
$$= y\log\sigma(w^Tx) + (1-y)\log(1-\sigma(w^Tx))$$

$$L(w) = -\frac{1}{N}\sum_{i=1}^{N}[y_i\log\sigma(w^Tx_i) + (1-y_i)\log(1-\sigma(w^Tx))] \quad \text{Minimize it!}$$

Gradient Descent

$w \leftarrow w + \alpha L'(w)$ ↗ ① initialize w ② repeat, until convergence

gradient: $\nabla L = (\frac{\partial L}{\partial w_1}, \frac{\partial L}{\partial w_2}, \cdots, \frac{\partial L}{\partial w_D})^T$

"Chain Rule"

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

$$\frac{\partial}{\partial w_j}\log P(y|x,w)$$
$$\frac{\partial}{\partial w_j}(y\log\sigma(z) + (1-y)\log(1-\sigma(z)))$$
$$= y\cdot\frac{1}{\sigma(z)}\sigma(z)(1-\sigma(z))\cdot\frac{\partial\sigma(z)}{\partial w} + (1-y)\frac{1}{1-\sigma(z)}\sigma(z)(1-\sigma(z))\cdot\frac{\partial\sigma(z)}{\partial w}$$
$$= (y-\sigma(z))\frac{\partial\sigma(z)}{\partial w}$$
$$= [y-\sigma(z)]\,x_j$$

$\{x_{i,1}, x_{i,2}, x_{i,3}, \cdots x_{i,D}\}$ features ← $x_1, x_2, x_3, x_4, \vdots$ | label $y_1 \to \{0,1\}$, $y_2, y_3, y_4, \vdots$

$$\boxed{\frac{\partial L(w)}{\partial w_j} = -\frac{1}{N}\sum_{i=1}^{N}\frac{\partial}{\partial w_j}\log P(y_i|w,x_i) = -\frac{1}{N}\sum_{i=1}^{N}[y_i - \sigma(w^Tx_i)]\cdot x_{i,j}}$$

☆

$$w_j \leftarrow w_j + \alpha\frac{1}{N}\sum_{i=1}^{N}[y_i - \sigma(w^Tx_i)]\,x_{i,j}$$

(α ↑ learning rate)

"Stochastic Gradient Descent"

$$w_j \leftarrow w_j + \alpha\cdot\frac{1}{|B|}\sum_{i\in B}[y_i - \sigma(w^Tx_i)]\,x_{i,j}$$

"L_2 regularization"

$$L(w) \longleftarrow L(w) + \frac{1}{2}\|w\|_2^2$$

Softmax Rregression

Data: $\{x_i, y_i\}_{i=1}^{N}$ $y_i \in \{1, 2, \cdots, C\}$.

$$p(y=c|x,W) = \boxed{\frac{1}{Z(x,W)}\cdot\exp(w_c^Tx)}$$

partition function: $Z(x,W) = \sum_{c'=1}^{C}\exp(w_{c'}^Tx)$

$W = \{w_1, w_2, \cdots, w_C\}$

Rule:
$$\hat{y} = \arg\max_y P(y=c|x,W)$$

When $C=2$. Let $w = w_2 - w_1$ Then We get the Logistic regression model

$$L(w) = -\frac{1}{N}\sum_{i=1}^{N}\log P(y_i|x_i,w)$$
$$= -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(w_{y_i}^Tx)}{Z(x_i,W)}$$
$$= -\frac{1}{N}\sum_{i=1}^{N}[w_{y_i}^Tx - \boxed{\log\sum_{c'=1}^{C}\exp(w_{c'}^Tx_i)}]$$

LogSumExp

$$LSE(x_1,x_2,x_3) = \log(e^{x_1}+e^{x_2}+e^{x_3})$$
$$\approx Max(x_1,x_2,x_3)$$

$\log\sum_{c'=1}^{C}\exp(w_{c'}^Tx_i)$

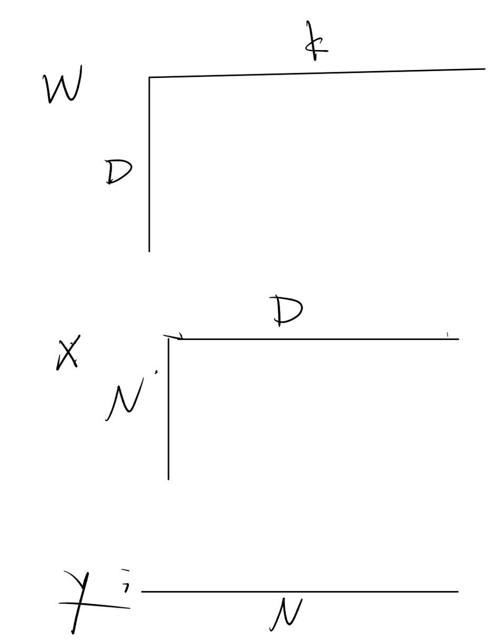
"soft maximum" of the logits of all classes

$$\frac{\partial L(w)}{\partial w_{c,j}} = -\frac{1}{N}\sum_{i=1}^{N}[\frac{\partial}{\partial w_{c,j}}\sum_{c=1}^{C}1(y_i=c)\,w_c^Tx_i - \frac{\partial}{\partial w_{c,j}}\log\sum_{c'=1}^{C}\exp(w_{c'}^Tx_i)]$$

(y_i only belongs to 1 class)

$$= -\frac{1}{N}\sum_{i=1}^{N}[\sum_{c=1}^{C}1(y_i=c)\,x_{i,j} - \frac{1}{\sum\exp(w_{c'}^Tx_i)}\cdot\exp(w_c^Tx_i)\,x_{i,j}]$$
$$= -\frac{1}{N}\sum_{i=1}^{N}[w_{y_i}^Tx_i - \log\sum_{c'=1}^{C}\exp(w_{c'}^Tx_i)]\,x_{i,j}$$

$$w_c \longleftarrow w_c + \alpha\frac{1}{N}\sum_{i=1}^{N}[1(y_i=c) - p(y_i=c|x_i,W)]\,x_i.$$



Performance Metrics

① Confusion Matrix

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of prediction}}$$

$$\frac{TP+TN}{TP+FP+TN+FN}$$

Precision = $\frac{TP}{\boxed{TP+FP}}$ (被判断成对) Recall = $\frac{TP}{\boxed{TP+FN}}$ (真是对的)

$$F_1 = 2\frac{P\cdot R}{P+R}$$

Harmonic Mean.