<Logistic Regression & Softmax & Cross-Entropy Loss>

<Logistic Regression>

- **odds** : x라는 데이터가 주어졌을 때, 성공(c_1)확률/실패(c_2)확률
  : $p(c_1|x)=y,\ p(c_2|x)=1-y$
  : $y/(1-y)$
  : 범위 : $0 < odds < \infty$
- **logit** : odds에 log를 씌운 것
  : $\log(odds)=\log(y/1-y)$
  : 범위 : $-\infty <$ logit $< \infty$

- f(x)의 결과값(z)를 logit으로 표현 -> $-\infty \leqq$logit(y)=z$\leqq \infty$

$$\log\left(\frac{y}{1-y}\right)=f(x)$$
$$\frac{y}{1-y}=e^{f(x)}$$
$$\frac{1-y}{y}=e^{-f(x)}$$
$$1/y=1+e^{-f(x)}$$
$$y=\frac{1}{1+e^{-f(x)}}$$

- f(x)의 종속변수(z)를 sigmoid function의 입력값으로 -> $0 \leqq$sigmoid(z)=y$\leqq 1$

-> logit(y)=z, sigmoid(z)=y이기 때문에, logit과 sigmoid는 역함수 관계

∴ f(x)가 선형식일 때, sigmoid(t)로 regression하는 것 : Logistic Regression

<결정경계>

- logistic regression을 통한 class의 결정은 임계값(threshold)을 통해 이루어짐-> $f(x)$의 값에 대한 임계값이 생김 -> 선형분류와 같음(즉 logistic regression은 비선형함수이지만, 비선형 분류를 할 수 있는 모델은 아님)

<Multiple Logistic Regression>

- 클래스가 k개인 경우($C_1, C_2, \dots, C_k$)
- C_k를 기준으로 사용
- C_1에 대한 logit = $\log\frac{p(C_1|x)}{p(C_k|x)}=f_1(x)$
- C_2에 대한 logit = $\log\frac{p(C_2|x)}{p(C_k|x)}=f_2(x)$
- C_i에 대한 logit = $\log\frac{p(C_i|x)}{p(C_k|x)}=f_i(x)$

- C_k에 대한 logit = $0 = \log\frac{p(C_k|x)}{p(C_k|x)} = f_i(x)$

-> exponential 형태로 표현한 후, class (1~k-1)까지의 양변을 더함

$$\sum_{i=1}^{k-1}\frac{p(C_i|x)}{p(C_k|x)} = \sum_{i=1}^{k-1}\exp(z_i)$$

$p(C_k|x) = \frac{1}{1+\sum_{i=1}^{k-1}\exp(z_i)}$, -> $p(C_i|x) = \frac{\exp(z_i)}{1+\sum_{i=1}^{k-1}\exp(z_i)}$ (C_i에 대한 logit과 $p(C_k|x)$ 이용)

이 때, $1 = \exp(z_k)$이기 때문에 1 자리에 이를 대입

-> $p(C_k|x) = \frac{\exp(z_k)}{\sum_{i=1}^{k}\exp(z_i)}$, $p(C_i|x) = \frac{\exp(z_i)}{\sum_{i=1}^{k}\exp(z_i)}$ -> "$softmax(z_i)$"

<Cross Entropy Loss – Multiple Logistic Regression>

○ 각 클래스 예측 확률 : $\vec{y}$ = $[p(C_1|x), p(C_2|x), \ldots, p(C_k|x)]$

○ 각 클래스 실제 확률 : $\vec{t} = [t_1, t_2, \ldots, t_k]$(만약 클래스1이라면 [1,0, ..., 0])

-> Cross Entropy($H(p|q) = -\sum_{i=1}^{n} p_i \log(q_i)$)를 이용하여 예측의 Loss를 구함

$L = -\sum_{i=1}^{k} t_i \log y$ -> "cross entropy loss"

<Cross Entropy Loss – Binary Logistic Regression>

○ 예측 값 : $y = P(C_1|x)$

○ 실제 값 : $t = 1\,\mathrm{or}\,0$

-> Cross Entropy를 이용하기 위해 예측값과 실제값 변환

- 예측값 : $\vec{y} = [y, 1-y]$
- 실제값 : $\vec{t} = [0, 1]$

$L = -\sum_{i=1}^{2} t_i \log y = -[t log(y) + (1-t)\log(1-y)]$ -> "cross entropy loss"

<Log Loss & Cross Entropy Error>

- binary classification일 때, log likelihood를 최대화 하는 것은 CEE를 최소화하는 것과 같음

따라서 CEE는 log loss 혹은 negative log likelihood로 불리기도 함

ex)

- D: binary classification 문제의 어떤 dataset(데이터 개수:n개)
- y : target이 1일 확률

-> $likelihood = p(D|y)$

= $\Pi_{i=1}^{n}(\theta)^{t_i}(1-\theta)^{1-t_i}$

$$\therefore \log - likelihood = \sum_{i=1}^{n}(t_i \log y_i + (1-t_i)\log(1-y_i)) = - CEE$$

해당 식은 binary-classification의 cross-entropy의 음과 같음

- (하나의 x에 대한 log-likelihood = $[tlog(y) + (1-t)\log(1-y)] = - CEE$)