# Multi-class Logistic Regression

- Assume that there are $s$ different classes.
  Data set: $\{(x_{n,1}, \ldots, x_{n,k}; t_n)\}$, $t_n = 1, 2, \ldots, s$, $n = 1, \ldots, N$
- How to generalize logistic regression?

- $t_n = 1 \Leftrightarrow [1, 0, \ldots, 0]$, $\quad t_n = 2 \Leftrightarrow [0, 1, 0 \ldots, 0]$, ...,
  $t_n = s \Leftrightarrow [0, \ldots, 0, 1]$
- Obtain an $N \times s$ matrix $\mathbf{t} = [t_{n,m}]$ such that

$$t_{n,m} = \begin{cases} 1 & \text{if } t_n = m, \\ 0 & \text{if } t_n \neq m. \end{cases}$$

- Define $\boldsymbol{\sigma} : \mathbb{R}^s \to (0,1)^s$ by

$$\boldsymbol{\sigma}(\boldsymbol{a}) = \left( \frac{e^{a_1}}{\sum_{i=1}^{s} e^{a_i}}, \ldots, \frac{e^{a_s}}{\sum_{i=1}^{s} e^{a_i}} \right),$$

  where $\boldsymbol{a} = (a_1, a_2, \ldots, a_s)$.

  The function $\boldsymbol{\sigma}$ is called the softmax function.

- When $s = 2$, we have

$$\boldsymbol{\sigma}(\boldsymbol{a}) = \left( \frac{e^{a_1}}{e^{a_1} + e^{a_2}}, \frac{e^{a_2}}{e^{a_1} + e^{a_2}} \right) = \left( \frac{1}{1 + e^{a_2 - a_1}}, \frac{e^{a_2 - a_1}}{1 + e^{a_2 - a_1}} \right).$$

- The softmax function is a generalization of the sigmoid function.

- Define $\boldsymbol{y} = [y_1, \ldots, y_s] = \boldsymbol{\sigma}(\boldsymbol{a})$.
- For $m, j = 1, \ldots, s$,

$$\boxed{\frac{\partial y_m}{\partial a_j} = y_m(\delta_{j,m} - y_j)},$$

where $\delta_{j,m}$ is the Kronecker's delta, i.e.

$$\delta_{j,m} = \begin{cases} 1 & \text{if } j = m, \\ 0 & \text{otherwise.} \end{cases}$$

- Consider a $(k+1) \times s$ matrix $\boldsymbol{w} = [w_{p,q}]$.
  Define $\boldsymbol{y} = \boldsymbol{\sigma}(X\boldsymbol{w}) = [y_{n,m}]$,
  where $X$ is as before and $\boldsymbol{\sigma}$ is applied to the rows of $X\boldsymbol{w}$.
  Each row of $\boldsymbol{y}$ consists of probabilities for classes 1 through $m$.
- The likelihood function is given by

$$p(\mathbf{t}|\boldsymbol{w}) = \prod_{n=1}^{N} \prod_{m=1}^{s} y_{n,m}^{t_{n,m}}.$$

- The cross-entropy is

$$E(\boldsymbol{w}) = -\sum_{n=1}^{N} \sum_{m=1}^{s} t_{n,m} \ln y_{n,m}.$$

- One can check

$$\nabla E(\boldsymbol{w}) = \left[\frac{\partial E}{\partial w_{p,q}}\right] = \left[\sum_{n=1}^{N}(y_{n,q} - t_{n,q})x_{n,p}\right] = X^\top(\boldsymbol{y} - \mathbf{t}).$$

(Use
$\sum_{m=1}^{s} t_{n,m}(\delta_{q,m} - y_{n,q}) = \sum_{m=1}^{s}(t_{n,m}\delta_{q,m} - t_{n,m}y_{n,q}) = t_{n,q} - y_{n,q}$.)

- Gradient Descent

$$\boxed{\boldsymbol{w}_{i+1} = \boldsymbol{w}_i - \eta X^\top(\boldsymbol{y} - \mathbf{t})}.$$

- Let $\boldsymbol{w}_i \to \boldsymbol{w}_*$ as $i \to \infty$.

  Given $\boldsymbol{x} = [x_1, \ldots, x_k, 1]$, the coordinates of the vector

$$\boldsymbol{y} = \sigma(\boldsymbol{x}\boldsymbol{w}_*)$$

  represent the probabilities for the classes.

- The (multi-class) logistic regression is the simplest **neural network**.