Logistic Regression

Seungjin Choi

Department of Computer Science and Engineering
Pohang University of Science and Technology
77 Cheongam-ro, Nam-gu, Pohang 37673, Korea
seungjin@postech.ac.kr

Binary Classification

- Consider a binary classification problem, i.e., $\mathcal{C} = \{\mathcal{C}_1, \mathcal{C}_2\}$.
- Given a particular vector $\boldsymbol{x}$, we wish to assign it to one of the two classes. To this end, we use Bayes' rule and calculate the relevant posterior probability:

$$
\begin{aligned}
P(\mathcal{C}_1|\boldsymbol{x}) &= \frac{P(\boldsymbol{x}|\mathcal{C}_1)P(\mathcal{C}_1)}{P(\boldsymbol{x})} \\
&= \frac{P(\boldsymbol{x}|\mathcal{C}_1)P(\mathcal{C}_1)}{P(\boldsymbol{x}|\mathcal{C}_1)P(\mathcal{C}_1) + P(\boldsymbol{x}|\mathcal{C}_2)P(\mathcal{C}_2)} \\
&= \frac{1}{1 + \frac{P(\boldsymbol{x}|\mathcal{C}_2)}{P(\boldsymbol{x}|\mathcal{C}_1)}\frac{P(\mathcal{C}_2)}{P(\mathcal{C}_1)}} \\
&= \frac{1}{1 + \exp\left\{-\log\left[\frac{P(\boldsymbol{x}|\mathcal{C}_1)}{P(\boldsymbol{x}|\mathcal{C}_2)}\right] - \log\left[\frac{P(\mathcal{C}_1)}{P(\mathcal{C}_2)}\right]\right\}}
\end{aligned}
$$

- It can be written in the form of the logistic function:

$$P(\mathcal{C}_1|\boldsymbol{x}) = \frac{1}{1+e^{-\xi}},$$

where

$$\xi = \underbrace{\log\left[\frac{P(\boldsymbol{x}|\mathcal{C}_1)}{P(\boldsymbol{x}|\mathcal{C}_2)}\right]}_{\text{likelihood ratio}} + \underbrace{\log\left[\frac{P(\mathcal{C}_1)}{P(\mathcal{C}_2)}\right]}_{\text{prior ratio}}.$$

- We must choose a particular form for the class-conditional density function $P(\boldsymbol{x}|\mathcal{C}_i)$.

Multivariate Gaussian Model

Assume that the class-conditional densities are multivariate Gaussian with identical covariance matrix $\boldsymbol{\Sigma}$:

$$P(\boldsymbol{x}|\mathcal{C}_i) = \frac{1}{(2\pi)^{\frac{m}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\left\{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_i)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_i)\right\}.$$

After a simple calculation, we have

$$P(\mathcal{C}_1|\boldsymbol{x}) = \frac{1}{1+e^{-(\boldsymbol{w}^\top \boldsymbol{x}+b)}},$$

where

$$\begin{aligned} \boldsymbol{w} &= \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2), \\ b &= \frac{1}{2}(\boldsymbol{\mu}_2+\boldsymbol{\mu}_1)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{\mu}_2-\boldsymbol{\mu}_1) + \log\left[\frac{P(\mathcal{C}_1)}{P(\mathcal{C}_2)}\right]. \end{aligned}$$

Logistic Regression

- Predict a binary output $y_t \in \{0, 1\}$ from an input $\boldsymbol{x}_t$.
- The logistic regression model has the form

$$y_t = \sigma(\boldsymbol{w}^\top \boldsymbol{x}_t) + \epsilon_t,$$

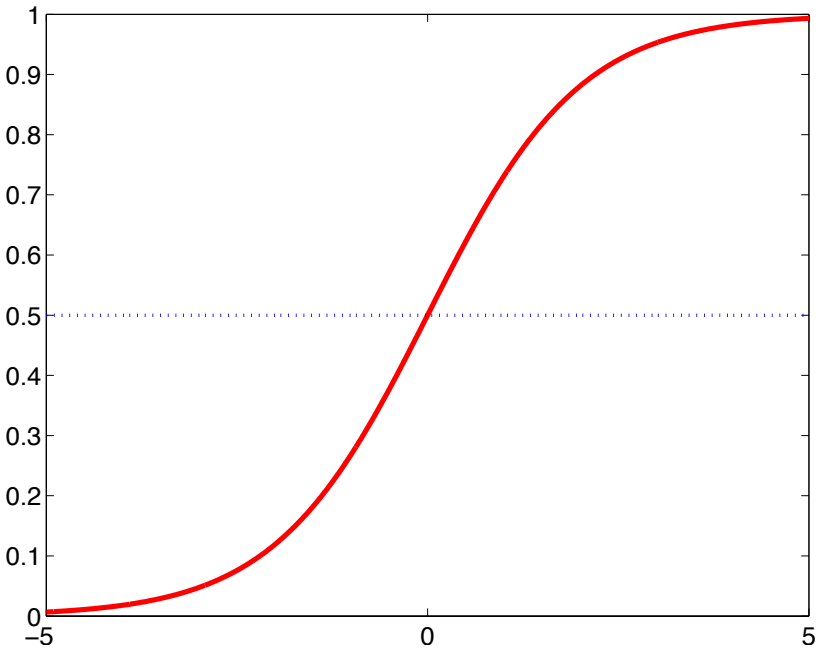
where

$$\sigma(\xi) = \frac{1}{1 + e^{-\xi}} = \frac{e^{\xi}}{1 + e^{\xi}}.$$

- Model input-output by a conditional Bernoulli distribution

$$\boxed{P(y_t = 1 | \boldsymbol{x}_t) = \sigma(\boldsymbol{w}^\top \boldsymbol{x}_t).}$$

Properties of Logistic Function



- $\sigma(\xi) \to 0$ as $\xi \to -\infty$.
- $\sigma(\xi) \to 1$ as $\xi \to \infty$.
- $\sigma(-\xi) = 1 - \sigma(\xi)$.
- $\frac{d}{d\xi}\left[\sigma(\xi)\right] = \sigma(\xi)\sigma(-\xi)$.

Logistic Regression: MLE

- Given $\{(\boldsymbol{x}_t, y_t)\}_{t=1}^N$, the likelihood is given by

$$\begin{aligned} p(\boldsymbol{y}|\boldsymbol{X}, \boldsymbol{w}) &= \prod_{t=1}^{N} p(y_t = 1|\boldsymbol{x}_t)^{y_t} \left(1 - p(y_t = 1|\boldsymbol{x}_t)\right)^{1-y_t} \\ &= \prod_{t=1}^{N} \sigma(\boldsymbol{w}^\top \boldsymbol{x}_t)^{y_t} \left(1 - \sigma(\boldsymbol{w}^\top \boldsymbol{x}_t)\right)^{1-y_t}. \end{aligned}$$

- Then log-likelihood function is given by

$$\mathcal{L} = \sum_{t=1}^{N} \log p(y_t|\boldsymbol{x}_t) = \sum_{t=1}^{N} \left\{ y_t \log \sigma_t + (1 - y_t) \log(1 - \sigma_t) \right\},$$

  where $\sigma_t = \sigma(\boldsymbol{w}^\top \boldsymbol{x}_t)$.

- This is a nonlinear function of $\boldsymbol{w}$ whose maximum cannot be computed in a closed form.
- Iterative re-weighted least squares (IRLS) is a popular algorithm, derived from Newton's method.

Mathematical Preliminaries

- Gradient
- Hessian matrix
- Gradient descent/ascent
- Newton's method

Gradient

- Let us consider a real scalar function of a real vector $\boldsymbol{x} \in \mathbb{R}^m$, $f(\boldsymbol{x})$,

$$f(\boldsymbol{x}) : \mathbb{R}^n \longrightarrow \mathbb{R}.$$

- The gradient of $f(\boldsymbol{x})$ is defined by

$$\begin{aligned} \nabla f(\boldsymbol{x}) &= \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}} \\ &= \left[\frac{\partial f}{\partial x_1} \cdots \frac{\partial f}{\partial x_m} \right]^\top . \end{aligned}$$

Hessian Matrix

If $f(\boldsymbol{x})$ belongs to the class $\mathcal{C}^2$, the Hessian matrix $\boldsymbol{H}$ is defined as the symmetric matrix with the (i,j)-element $\frac{\partial^2 f(\boldsymbol{x})}{\partial x_i \partial x_j}$,

$$
\begin{aligned}
\boldsymbol{H} &= \nabla^2 f(\boldsymbol{x}) \\
&= \left[\frac{\partial^2 f(\boldsymbol{x})}{\partial x_i \partial x_j}\right] \\
&= \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_m} \\ \vdots & & & \vdots \\ \frac{\partial^2 f}{\partial x_m \partial x_1} & \frac{\partial^2 f}{\partial x_m \partial x_2} & \cdots & \frac{\partial^2 f}{\partial x_m^2} \end{bmatrix}. \\
&= \frac{\partial}{\partial \boldsymbol{x}} \left[\frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{x}}\right]^\top \\
&= \frac{\partial}{\partial \boldsymbol{x}} \left[\nabla f(\boldsymbol{x})\right]^\top
\end{aligned}
$$

Gradient Descent/Ascent

- The gradient descent/ascent learning is a simple (first-order) iterative method for minimization/maximization.
- Gradient descent: iterative minimization

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \left(\frac{\partial \mathcal{J}}{\partial \boldsymbol{w}} \right).$$

- Gradient ascent: iterative maximization

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t + \eta \left(\frac{\partial \mathcal{J}}{\partial \boldsymbol{w}} \right).$$

- Learning rate: $\eta > 0$

Newton's Method

The basic idea of Newton's method is to optimize the quadratic approximation of the objective function $\mathcal{J}(\boldsymbol{w})$ around the current point $\boldsymbol{w}_t$.

The second-order Taylor series expansion of $\mathcal{J}(\boldsymbol{w})$ at the current point $\boldsymbol{w}_t$ gives

$$\mathcal{J}(\boldsymbol{w}_{t+1}) \approx \mathcal{J}(\boldsymbol{w}_t) + \nabla^\top \mathcal{J}(\boldsymbol{w}_t)\Delta\boldsymbol{w}_t + \frac{1}{2}\left[\Delta\boldsymbol{w}_t\right]^\top \nabla^2\mathcal{J}(\boldsymbol{w}_t)\left[\Delta\boldsymbol{w}_t\right],$$

where $\nabla^2\mathcal{J}(\boldsymbol{w}_t)$ is the Hessian of $\mathcal{J}(\boldsymbol{w})$ and $\Delta\boldsymbol{w}_t = \boldsymbol{w}_{t+1} - \boldsymbol{w}_t$.

Define

$$\Delta\mathcal{J}(\boldsymbol{w}_t) = \mathcal{J}(\boldsymbol{w}_{t+1}) - \mathcal{J}(\boldsymbol{w}_t).$$

Then, we have

$$\Delta \mathcal{J}(\boldsymbol{w}_t) = \nabla^\top \mathcal{J}(\boldsymbol{w}_t) \Delta \boldsymbol{w}_t + \frac{1}{2} \left[\Delta \boldsymbol{w}_t\right]^\top \nabla^2 \mathcal{J}(\boldsymbol{w}_t) \left[\Delta \boldsymbol{w}_t\right].$$

Differentiate this w.r.t. $\Delta \boldsymbol{w}_t$ and set it equal 0, which leads to

$$\nabla \mathcal{J}(\boldsymbol{w}_t) + \nabla^2 \mathcal{J}(\boldsymbol{w}_t) \Delta \boldsymbol{w}_t = 0,$$

Thus we have

$$\Delta \boldsymbol{w}_t = - \left[\nabla^2 \mathcal{J}(\boldsymbol{w}_t)\right]^{-1} \nabla \mathcal{J}(\boldsymbol{w}_t).$$

Hence the Newton's method has the form

$$\boxed{\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \left[\nabla^2 \mathcal{J}(\boldsymbol{w}_t)\right]^{-1} \nabla \mathcal{J}(\boldsymbol{w}_t).}$$

Remark: The Hessian $\nabla^2 \mathcal{J}(\boldsymbol{w}_t)$ should be positive definite for all t.

Logistic Regression: Gradient Ascent

The gradient ascent learning has the form

$$\boldsymbol{w}^{\text{new}} = \boldsymbol{w}^{\text{old}} + \eta \left(\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} \right).$$

Calculate the gradient

$$\frac{\partial \mathcal{L}}{\partial \boldsymbol{w}} = \sum_t \left\{ y_t - \sigma(\boldsymbol{w}^\top \boldsymbol{x}_t) \right\} \boldsymbol{x}_t.$$

Hence, the gradient ascent update rule for $\boldsymbol{w}$ is

$$\boldsymbol{w}^{\text{new}} = \boldsymbol{w}^{\text{old}} + \eta \sum_t \left\{ y_t - \sigma\left(\left(\boldsymbol{w}^{\text{old}} \right)^\top \boldsymbol{x}_t \right) \right\} \boldsymbol{x}_t.$$

Logistic Regression: IRLS

- Newton's update has the form

$$\Delta \boldsymbol{w} = \underbrace{\left[\sum_{t=1}^{N} \sigma_t \left(1-\sigma_t\right) \boldsymbol{x}_t \boldsymbol{x}_t^\top\right]^{-1}}_{\textit{inverse of Hessian, } [\nabla^2 \mathcal{L}]^{-1}} \underbrace{\left[\sum_{t=1}^{N} \left(y_t - \sigma_t\right) \boldsymbol{x}_t\right]}_{\textit{gradient, } \nabla \mathcal{L}},$$

- Newton's update reduces to iterative re-weighted least squares (IRLS):

$$\Delta \boldsymbol{w} = \left(\boldsymbol{X}^\top \boldsymbol{V}_r \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{V}_r \boldsymbol{Y}^*,$$

where

$$\boldsymbol{V}_r = \begin{bmatrix} \sigma_1(1-\sigma_1) & & 0 \\ \vdots & \ddots & \vdots \\ 0 & & \sigma_N(1-\sigma_N) \end{bmatrix},$$

$$\boldsymbol{Y}^* = \begin{bmatrix} \frac{y_1-\sigma_1}{\sigma_1(1-\sigma_1)} & & 0 \\ \vdots & \ddots & \vdots \\ 0 & & \frac{y_N-\sigma_N}{\sigma_N(1-\sigma_N)} \end{bmatrix}.$$

Logistic Regression: Generalized Linear Model

- Bernoulli distribution for a univariate binary random variable $y \in \{0, 1\}$ with mean p,

$$P(y|p) = p^y(1-p)^{1-y}.$$

  Logit transform (log-odds), $\theta = \log\left(\frac{p}{1-p}\right)$, leads to $[0, 1] \mapsto \mathbb{R}$.

- The logit transform gives $p = \frac{1}{1+e^{-\theta}} = \sigma(\theta)$.
- Thus, we have

$$\begin{aligned} P(y|\theta) &= \sigma(\theta)^y \left(1 - \sigma(\theta)\right)^{1-y} \\ &= \sigma(\theta)^y \sigma(-\theta)^{1-y} \end{aligned}$$

Multiclass Logistic Regression (Multinomial Logistic Regression)

- Model input-output by a softmax transformation of linear functions of inputs:

$$p(y_t = k|\boldsymbol{x}_t) = \frac{\exp\{\boldsymbol{w}_k^\top \boldsymbol{x}_t\}}{\sum_{j=1}^K \exp\{\boldsymbol{w}_j^\top \boldsymbol{x}_t\}}.$$

- Given $\boldsymbol{Y} \in \mathbb{R}^{K\times N}$ (each column $\boldsymbol{y}_t \in \mathbb{R}^K$ follows the 1-of-K coding) and $\boldsymbol{X} \in \mathbb{R}^{D\times N}$, the likelihood is given by

$$p(\boldsymbol{Y}|\boldsymbol{X}, \boldsymbol{w}_1, \ldots, \boldsymbol{w}_K) = \prod_{t=1}^N \prod_{k=1}^K p(y_t = k|\boldsymbol{x}_t)^{Y_{kt}}.$$

- The log-likelihood is given by

$$\mathcal{L} = \sum_{t=1}^N \sum_{k=1}^K Y_{kt} \log\left[p(y_t = k|\boldsymbol{x}_t)\right].$$

- One can apply Newton's update to derive IRLS, as in logistic regression.