STA 314: Statistical Methods for Machine Learning I

Lecture 8 - Multi-class Logistic Regression

Xin Bing

Department of Statistical Sciences
University of Toronto

Review

In the last lecture, we have learned the logistic regression for binary classification with $Y \in \{0, 1\}$.

- Estimating the Bayes rule at any observation $\mathbf{x} \in \mathcal{X}$ is equivalent to estimate the conditional probability $\mathbb{P}(Y = 1 \mid X = \mathbf{x})$.
- Logistic regression parametrizes the conditional probability by

$$\mathbb{P}(Y = 1 \mid X = \mathbf{x}) = \frac{e^{\beta_0 + \mathbf{x}^\top \boldsymbol{\beta}}}{1 + e^{\beta_0 + \mathbf{x}^\top \boldsymbol{\beta}}}.$$

- We estimate the coefficients by using MLE which can be solved by (stochastic) gradient descent.

Extension to multi-class classification

When $Y \in \{0, 1, \ldots, K\}$ for $K \geq 2$, we need to estimate

$$p_k(\mathbf{x}) := \mathbb{P}(Y = k \mid X = \mathbf{x}), \qquad \forall 1 \leq k \leq K.$$

We assume

$$p_0(\mathbf{x}) = \frac{1}{1 + \sum_{k=1}^{K} e^{\beta_0^{(k)} + x^\top \boldsymbol{\beta}^{(k)}}},$$

$$p_1(\mathbf{x}) = \frac{e^{\beta_0^{(1)} + x^\top \boldsymbol{\beta}^{(1)}}}{1 + \sum_{k=1}^{K} e^{\beta_0^{(k)} + \mathbf{x}^\top \boldsymbol{\beta}^{(k)}}}.$$

$$\vdots$$

$$p_K(\mathbf{x}) = \frac{e^{\beta_0^{(K)} + x^\top \boldsymbol{\beta}^{(K)}}}{1 + \sum_{k=1}^{K} e^{\beta_0^{(k)} + \mathbf{x}^\top \boldsymbol{\beta}^{(k)}}}$$

Choice of the baseline (which is $Y = 0$) is arbitrary.

Classification

Equivalently,

$$
\begin{aligned}
\log\left(\frac{p_1(\mathbf{x})}{p_0(\mathbf{x})}\right) &= \beta_0^{(1)} + \beta_1^{(1)} x_1 + \cdots + \beta_p^{(1)} x_p \\
\log\left(\frac{p_2(\mathbf{x})}{p_0(\mathbf{x})}\right) &= \beta_0^{(2)} + \beta_1^{(2)} x_1 + \cdots + \beta_p^{(2)} x_p \\
&\vdots \\
\log\left(\frac{p_K(\mathbf{x})}{p_0(\mathbf{x})}\right) &= \beta_0^{(K)} + \beta_1^{(K)} x_1 + \cdots + \beta_p^{(K)} x_p
\end{aligned}
$$

So classification can be done immediately once $\boldsymbol{\beta}^{(k)}$'s are estimated,

How to estimate coefficients?

A naive approach: separate binary logistic regressions

$$\log\left(\frac{p_k(\mathbf{x})}{p_0(\mathbf{x})}\right) = \beta_0^{(k)} + \beta_1^{(k)} x_1 + \cdots + \beta_p^{(k)} x_p$$

Split the data into $\{\mathcal{D}^{train}{}_{(1)}, \ldots, \mathcal{D}^{train}{}_{(K)}\}$ with $\mathcal{D}^{train}{}_{(k)}$ containing all data with $y \in \{0, k\}$.

1. For each $1 \le k \le K$, use $\mathcal{D}^{train}{}_{(k)}$ to perform binary logistic regression to estimate $\beta^{(k)}$ and estimate
$$\frac{p_k(\mathbf{x})}{p_0(\mathbf{x})}$$
2. Assign class label by comparing
$$1, \frac{p_1(\mathbf{x})}{p_0(\mathbf{x})}, \frac{p_2(\mathbf{x})}{p_0(\mathbf{x})} \ldots, \frac{p_K(\mathbf{x})}{p_0(\mathbf{x})}$$

Why naive?

- Estimation of $\boldsymbol{\beta}^{(k)}$
  - only uses $\mathcal{D}^{train}{}_{(k)}$, data points in class $\{0, k\}$
  - ignore all data points in other classes

- The event $\{y_i = k\}$ is **dependent** on all other $\{y_i = k'\}$ for $k' \neq k$. Intuitively, this dependence helps to estimate $\boldsymbol{\beta}^{(k)}$ by pooling data from all classes.

- What should we use instead?

MLE for multi-class logistic regression

For $(y_1, \mathbf{x}_1), \ldots, (y_n, \mathbf{x}_n)$, the log-likelihood of $(\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(K)})$ with no intercepts is **proportional to**

$$
\begin{aligned}
&\sum_{i=1}^{n} \log\left(\prod_{k=0}^{K} p_k(\mathbf{x}_i)^{1\{y_i=k\}}\right) \\
&= \sum_{i=1}^{n}\sum_{k=0}^{K} 1\{y_i = k\} \log\left(p_k(\mathbf{x}_i)\right) \\
&= \sum_{i=1}^{n}\left[1\{y_i = 0\}\log\left(p_0(\mathbf{x}_i)\right) + \sum_{k=1}^{K} 1\{y_i = k\}\log\left(p_k(\mathbf{x}_i)\right)\right] \\
&= \sum_{i=1}^{n}\left[\sum_{k=1}^{K} 1\{y_i = k\}\ \mathbf{x}_i^\top \boldsymbol{\beta}^{(k)} - \sum_{k=0}^{K} 1\{y_i = k\}\log\left(1 + \sum_{k=1}^{K} e^{\mathbf{x}_i^\top \boldsymbol{\beta}^{(k)}}\right)\right] \\
&= \sum_{i=1}^{n}\left[\sum_{k=1}^{K} 1\{y_i = k\}\ \mathbf{x}_i^\top \boldsymbol{\beta}^{(k)} - \log\left(1 + \sum_{k=1}^{K} e^{\mathbf{x}_i^\top \boldsymbol{\beta}^{(k)}}\right)\right]
\end{aligned}
$$

Gradient of $\ell(\beta^{(k)})$

For any $1 \leq k \leq K$,

$$\frac{\partial \ell(\boldsymbol{\beta}^{(1)}, \ldots, \boldsymbol{\beta}^{(K)})}{\partial \boldsymbol{\beta}^{(k)}} = \sum_{i=1}^{n} \left[1\{y_i = k\}\ \mathbf{x}_i - \frac{\mathbf{x}_i e^{\mathbf{x}_i^\top \boldsymbol{\beta}^{(k)}}}{1 + \sum_{k=1}^{K} e^{\mathbf{x}_i^\top \boldsymbol{\beta}^{(k)}}} \right]$$

$$= \sum_{i=1}^{n} \left[1\{y_i = k\} - \frac{e^{\mathbf{x}_i^\top \boldsymbol{\beta}^{(k)}}}{1 + \sum_{k=1}^{K} e^{\mathbf{x}_i^\top \boldsymbol{\beta}^{(k)}}} \right] \mathbf{x}_i$$

c.f. the binary case

$$\frac{\partial \ell(\boldsymbol{\beta})}{\partial \boldsymbol{\beta}} = \sum_{i=1}^{n} \left[1\{y_i = 1\} - \frac{e^{\mathbf{x}_i^\top \boldsymbol{\beta}}}{1 + e^{\mathbf{x}_i^\top \boldsymbol{\beta}}} \right] \mathbf{x}_i$$

$$= \sum_{i=1}^{n} \left[y_i - \frac{e^{\mathbf{x}_i^\top \boldsymbol{\beta}}}{1 + e^{\mathbf{x}_i^\top \boldsymbol{\beta}}} \right] \mathbf{x}_i.$$

Gradient descent

Therefore, for $1 \le k \le K$, we update

$$\hat{\beta}^{(k)}_{(t+1)} = \hat{\beta}^{(k)}_{(t)} + \alpha \sum_{i=1}^{n} \left[1\{y_i = k\} - \frac{e^{\mathbf{x}_i^\top \hat{\beta}^{(k)}_{(t)}}}{1 + \sum_{k=1}^{K} e^{\mathbf{x}_i^\top \hat{\beta}^{(k)}_{(t)}}} \right] \mathbf{x}_i.$$

Remark:

- the gradient update uses data points from **all classes**!
- better estimation than the naive approach

An alternative to Logistic Regression

- When the classes are well-separated, the parameter estimates for the logistic regression model are surprisingly unstable[1].
  - Discriminant analysis does not suffer from this problem.

- When n is small and we know more about the data, such as the distribution of $X \mid Y = k$
  - Discriminant analysis has better performance than the logistic regression model.

- Logistic Regression sometimes does not handle multi-class classification well
  - Discriminant analysis is more suitable for **multi-class** classification problems.

[1] A paper on this.