Multinomial Logistic Regression

Caden Hewlett

2024-08-25

Multinomial Logistic Regression

Multinomial Logistic Regression (MLR) is an extension of binary logistic regression, where the response Y is one of K potentially ordinal categories, $Y \in \{1, 2, \dots K\} \subseteq \mathbb{N}$ where $K \geq 3$

The multinomial logistic model assumes that data are case-specific; that is, each independent variable has a single value for each case.

The general expression for MLR is given as follows:

$$\mathbb{P}(y_i = k \mid \mathbf{x}_i) = \frac{\exp(\mathbf{x}_i^\top \boldsymbol{\beta}_k)}{1 + \sum_{\ell=1}^{K-1} \exp(\mathbf{x}_i^\top \boldsymbol{\beta}_\ell)}, \text{ where } k \leq K - 1$$

And for the reference category K, the probability is dervied from the Law of Total Probability and given as:

$$\mathbb{P}(y_i = K \mid \mathbf{x}_i) = \frac{1}{1 + \sum_{\ell=1}^{K-1} \exp(\mathbf{x}_i^\top \boldsymbol{\beta}_\ell)}$$

There is no closed-form solution to the system of equations minimizing the regression coefficients with respect to RSS (or other loss functions,) and hence the coefficients $\boldsymbol{\beta_i}$ and intercept β_{i0} are generally found via optimization techniques maximizing the likelihood function, occasionally with constraints.

Likelihood Function: Derivation

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\beta}_1, \dots \boldsymbol{\beta}_K \mid \mathbf{X}) &= \prod_{i=1}^{n} \prod_{k=1}^{K} \Big(\mathbb{P}(y_i = j \mid \mathbf{x}_i)^{\mathbb{1}(y_i = j)} \Big) \\
\ell(\boldsymbol{\beta}_1, \dots \boldsymbol{\beta}_K \mid \mathbf{X}) &= \sum_{i=1}^{n} \sum_{k=1}^{K} \log \Big(\mathbb{P}(y_i = j \mid \mathbf{x}_i)^{\mathbb{1}(y_i = j)} \Big) \\
\ell(\boldsymbol{\beta}_1, \dots \boldsymbol{\beta}_K \mid \mathbf{X}) &= \sum_{i=1}^{n} \sum_{k=1}^{K} \mathbb{1}(y_i = j) \log \Big(\mathbb{P}(y_i = j \mid \mathbf{x}_i) \Big) \\
\ell(\boldsymbol{\beta}_1, \dots \boldsymbol{\beta}_K \mid \mathbf{X}) &= \sum_{i=1}^{n} \Bigg(\mathbb{1}(y_i = K) \log \Big(\frac{1}{1 + \sum_{\ell=1}^{K-1} \exp(\mathbf{x}_i^\top \boldsymbol{\beta}_\ell)} \Big) + \sum_{k=0}^{K-1} \log \Big(\frac{\exp(\mathbf{x}_i^\top \boldsymbol{\beta}_k)}{1 + \sum_{\ell=1}^{K-1} \exp(\mathbf{x}_i^\top \boldsymbol{\beta}_\ell)} \Big) \Bigg) \\
\ell(\boldsymbol{\beta}_1, \dots \boldsymbol{\beta}_K \mid \mathbf{X}) &= \sum_{i=1}^{n} \Bigg(-\log \Big(1 + \sum_{\ell=1}^{K-1} \exp(\mathbf{x}_i^\top \boldsymbol{\beta}_\ell) \Big) + \sum_{k=1}^{K-1} \mathbb{1}(y_i = k) \mathbf{x}_i^\top \boldsymbol{\beta}_k \Bigg) \ \square
\end{aligned}$$

Letting $\mathbf{B} = \begin{bmatrix} \boldsymbol{\beta}_1 & \dots & \boldsymbol{\beta}_K \end{bmatrix}$, the above log-likelihood expression is given in a simplified form as $\ell(\mathbf{B} \mid \mathbf{X})$.

Objective Function: Constraints

In addition, one may wish to impose constraints on the optimization to penalize overfitting. These include Ridge, Lasso and Elastic Net. They all depend on hyperparemeter λ controlling the strength of the penalization, which is tuned via cross-validation.

Lasso Penalty

For $\ell(\mathbf{B} \mid \mathbf{X})$, the Lasso (Least Absolute Shrinkage and Selection Operator) imposes an L1 penalty and hence performs variable selection.

$$F_{\text{lasso}}(\mathbf{B}) = \ell(\mathbf{B} \mid \mathbf{X}) - \lambda \|\mathbf{B}\|_1 = \ell(\mathbf{B} \mid \mathbf{X}) - \lambda \sum_{i=1}^{n} \sum_{k=1}^{K} |\beta_{i,k}|\,, \text{ for } \lambda \in \mathbb{R}^+$$

Ridge Penalty

For $\ell(\mathbf{B} \mid \mathbf{X})$, the Ridge Penalty uses the L2 norm - it is a stronger penalty but does not perform variable selection.

$$F_{\text{ridge}}(\mathbf{B}) = \ell(\mathbf{B} \mid \mathbf{X}) - \lambda \|\mathbf{B}\|_2^2 = \ell(\mathbf{B} \mid \mathbf{X}) - \lambda \sum_{i=1}^{n} \sum_{k=1}^{K} \beta_{i,k}^2\,, \text{ for } \lambda \in \mathbb{R}^+$$