Subgroup Analysis Of Logistic Regression

Report

Yuelin Li

SWUFE

Contents

Introduction

Most of the existing research on multivariate response regression assumes that the collected predictors are sufficient to explain the responses.

However, due to cost constraints or ethical issues, oftentimes there exist unmeasured hidden variables that are associated with the responses as well.

Model

Logistic Regression

Model: $\log \frac{P(Y_i=1|x_i)}{1-p(Y_i=1|x_i)} = x_i^T\beta + \mu_i$

- $Y \in \mathbb{R}^n$:response vector
- $X \in \mathbb{R}^{n\times m}$:observable features
- $\mu \in \mathbb{R}^n$:group-specific intercept

The likelihood function is $\prod_{i=1}^{N} [\frac{exp(x_i^T\beta+\mu_i)}{1+exp(x_i^T\beta+\mu_i)}]^{y_i} [\frac{1}{1+exp(x_i^T\beta+\mu_i)}]^{(1-y_i)}$

The log-likelihood function is

$$
\begin{aligned}
L(\beta, \mu, y, x) = & \sum_{i=1}^{N} [y_i \log \frac{exp(x_i^T \beta + \mu_i)}{1 + exp(x_i^T \beta + \mu_i)} \\
& + (1 - y_i) \log \frac{1}{1 + exp(x_i^T \beta + \mu_i)}] \\
= & \sum_{i=1}^{N} [y_i \log(exp(x_i^T \beta + \mu_i)) + \log \frac{1}{1 + exp(x_i^T \beta + \mu_i)}] \\
= & \sum_{i=1}^{N} [y_i (x_i^T \beta + \mu_i) - \log(1 + exp(x_i^T \beta + \mu_i))] \\
= & \sum_{i=1}^{N} l(\beta, \mu, y, x)
\end{aligned}
$$

Induce penalty item:

$$LASSO \quad p_\kappa(t,\tau) = \tau|t| \tag{1}$$

$$SCAD \quad p_\kappa(t,\tau) = \tau \int_0^t \min(1, \frac{(\kappa - \frac{x}{\tau})_+}{\kappa - 1}) dx, \kappa > 2 \tag{2}$$

$$MCP \quad p_\kappa(t,\tau) = \tau \int_0^t (1 - \frac{x}{\kappa\tau}), \kappa > 2 \tag{3}$$

Specifically, express objective function as a constrained optimization problem

$$\min \quad \mathbf{Q}_n(\beta,\mu,\tau) = -\frac{1}{n} l(\beta,\mu,y_i,x_i) + \sum_{1\le i\le j\le n} p_\kappa(|\delta_{ij}|,\tau)$$

$$s.t. \quad \mu_i - \mu j - \delta_{ij}$$

Take the constraints into account by augmenting the objective function with a weighted sum of the constraintfunctions. We define the Lagrangian L : $Y \in \mathbb{R}^n, X \in \mathbb{R}^{n \times m}$,

$$L(\beta, \mu, \delta, \lambda; Y, X) = -\frac{1}{n} \sum_{i=1}^{n} l(\beta, \mu, y_i, x_i) + \sum_{1 \leq i \leq j \leq n} p_\kappa(|\delta_{ij}|, \tau)$$
$$+ \sum_{1 \leq i \leq j \leq n} \lambda_{ij}(\mu_i - \mu_j - \delta_{ij}) + \frac{\rho}{2} \sum_{1 \leq i \leq j \leq n} (\mu_i - \mu_j - \delta_{ij})^2$$

Goal:

$$(\hat{\beta}, \hat{\mu}, \hat{\delta}.\hat{\lambda}) = argminL(\beta, \mu, \delta, \lambda)$$

Summary of ADMM

Three-step procedure(ADMM):

- Find initial parameter$\mu^{(0)}$and$\beta^{(0)}$from MLEs by letting $\mu_i = \mu$for all i. Set the initial values of the splitting variables $\delta_{ij}$and dual variables λ_{ij} as $\delta_{ij}^{(0)} = \mu_i^{(0)} - \mu_j^{(0)}$ and $\lambda_{ij}^{(0)} = 0$
- At step s,$s \geq 1$, compute$(\beta^{(s)}, \mu^{(s)}, \delta^{(s)}, \lambda^{(s)})$ iteratively by dual ascent;
- Compute residual$r^s = \Delta\mu^s - \delta^s$,Terminate the algorithm if the stopping rule $\|r^s\| <$ is satisfied

Update of β、μ:

$$L(\beta, \mu, \delta^{(s)}, \lambda^{(s)})$$
$$= -\sum_{i=1}^{n} l(\beta, \mu, y_i, x_i) + \frac{\rho}{2} \sum_{1 \le i \le j \le n} (b_i - b_j - \delta_{ij}^{(s)} + \rho^{-1} \lambda_{ij}^{(s)}) + C$$

At the s^{th} iteration, for a given $(\beta^{(s)}, \mu^{(s)}, \delta^{(s)}, \lambda^{(s)})$,
for the $(s+1)^{th}$ iteration, β and μ can be efficiently updated by minimizing it with numerical optimization algorithms such as Newton–Raphson or Quasi-Newton algorithms.

Update of δ:

the update of at the (s + 1)th iteration is given by minimizing

$L(\beta^{(s+1)}, \mu^{(s+1)}, \delta, \lambda^{(s)})$

$= \sum_{1\le i\le j\le n} p_\kappa(|\delta_{ij}|, \tau) + \sum_{1\le i\le j\le n} \lambda_{ij}^s(b_i^{(s+1)} - b_j^{(s+1)} - \delta_{ij})$

$+\frac{\rho}{2}(b_i^{(s+1)} - b_j^{(s+1)} - \delta_{ij})^2 + C$

$=$

$\sum_{1\le i\le j\le n}[\frac{\rho}{2}(\mu_i^{(s+1)} - \mu_j^{(s+1)} + \rho^{-1}\lambda_{ij}^{(s)} - \delta_{ij})^2] + \sum_{1\le i\le j\le n} p_\kappa(|\delta_{ij}|, \tau)$

$= \sum_{1\le i\le j\le n}[\frac{\rho}{2}(u_{ij} - \delta_{ij})^2 + p_\kappa(|\delta_{ij}|, \tau)] + C,$

where $u_{ij} = \mu_i^{(s+1)} - \mu_j^{(s+1)} + \rho^{-1}\lambda_{ij}^s$

This function is separable in δ_{ij} and the minimizer of each summand with respective to δ_{ij} collectively minimizes the entire function. Surprisingly, the minimizer of each summand,

i.e. $\min \quad \frac{\rho}{2}(u_{ij} - \delta_{ij})^2 + p_\kappa(|\delta_{ij}|, \tau)$

Update of δ:

Subgradients

- We say a vector $g \in \mathbb{R}^n$ is a subgradient of $f : \mathbb{R}^n \to \mathbb{R}$ at $x \in domf$ if for all $z \in domf$, $f(z)$ $f(x) + g^T(z - x)$. If f is convex and differentiable, then its gradient at x is a subgradient. But a subgradient can exist even when f is not differentiable at x.
- A function f is called subdifferentiable at x if there exists at least one subgradient at x. The set of subgradients of f at the point x is called the subdifferential of f at x, and is denoted $\partial f(x)$. A function f is called subdifferentiable if it is subdifferentiable at all $x \in domf$.

closed-form expression:

Subgradients

Example: Suppose $f : \mathbb{R} \to \mathbb{R}$, $f(x) = |x|$

- For $x \neq 0$, unique subgradient $g = \mathit{sign}(x)$
- For $x = 0$, g is any element of $[-1, 1]$

Subgradient optimality condition point x^* is the optimal solution of $f(x)$ (convex or not) if and only if subdifferential contains 0, i.e.

$$f(x^*) = \min f(x) \Longleftrightarrow 0 \in \partial f(x^*)$$

Update of δ:

Define the soft-thresholding operator as $ST(t, \tau) = sgn(t)(|t| - \tau)_+$.

- SCAD penalty

$$\delta_{ij} = \begin{cases} ST(u_{ij}, \tau/\rho), & |u_{ij}| \leq \tau(1+\rho^{-1}) \\ \dfrac{ST(u_{ij}, \kappa\tau/((\kappa-1)\rho))}{1-((\kappa-1)\rho)^{-1}}, & \tau(1+\rho^{-1}) \leq |u_{ij}| \leq \kappa\tau \\ u_{ij}, & |u_{ij}| > \kappa\tau \end{cases} \tag{4}$$

- MCP penalty

$$\delta_{ij} = \begin{cases} \dfrac{ST(u_{ij}, \tau/\rho)}{1-(\kappa\tau)^{-1}}, & |u_{ij}| \leq \kappa\tau \\ u_{ij}, & |u_{ij}| > \kappa\tau \end{cases} \tag{5}$$

Update of λ:

$$\lambda_{ij}^{(s+1)} = \lambda_{ij} + \rho(\mu_i^{(s)} - \mu_j^{(s)} - \delta_{ij}^{(s+1)})$$

Dual Ascent.

For any λ_{ij},We define the Lagrange dual function g:

$$g(\lambda) = \inf_{\delta} L(\beta, \mu, \delta, \lambda)$$

i.e.$g(\lambda) = L(\delta^*(\lambda), \lambda), \delta^* = argmin_\delta L(\delta, \lambda)$

$$\frac{dg}{d\lambda} = \frac{\partial L}{\partial \delta^*}\frac{\partial \delta^*}{\partial \lambda} + \frac{\partial L}{\partial \lambda} = \frac{\partial L}{\partial \lambda} = \mu_i - \mu j - \delta_{ij}$$

$$\lambda^{k+1} = \lambda^k + \alpha \nabla g,$$

where $\alpha = \rho, \nabla g = \mu_i - \mu j - \delta_{ij}$ □

Simulation

We simulate data from

$$P(Y=1|x) = \frac{exp(\beta^T x+\mu)}{1+exp(\beta^T x+\mu)}$$
$$P(Y=0|x) = \frac{exp(\beta^T x+\mu)}{1+exp(\beta^T x+\mu)} \quad (6)$$

,where the predictors $x_i = (x_{i1}, x_{i2}, x_{i3}, x_{i4}, x_{i5})^T$ are simulated from multivariate normal distribution with mean 0 and diagonal variance matrix with elements 1 and μ_i are generated from distribution $P(\mu_i = \alpha) = P(\mu_i = -\alpha) = 0.5$.The coefficient vector was set to be $\beta = (-0.3, 1, -1, 2, 0.5)^T$. We compare the performance of our our estimators using the two concave penalty functions, namely SCAD and MCP.