BIOS:7600 Homework 8

Chuan Lu

May 4, 2019

1. Problem 10.1, Logistic regression: Score and Hessian

   (a) *Proof.* First, consider
   $$\frac{\partial L}{\partial \pi_i} = -\frac{1}{n}\frac{y_i - \pi_i}{\pi_i(1-\pi_i)}, \tag{1}$$
   and
   $$\frac{d\pi_i}{d\eta_i} = -\frac{\partial_{\eta_i} f}{\partial_{\pi_i} f} = \pi_i(1-\pi_i), \quad \frac{d\pi_i}{d\eta_j} = 0, \tag{2}$$
   Then
   $$\frac{\partial L}{\partial \eta_i} = \sum_{j=1}^{n} \frac{\partial L}{\partial \pi_j}\frac{\partial \pi_j}{\partial \eta_i} = -\frac{1}{n}(y_i - \pi_i). \tag{3}$$
   So
   $$-\frac{\partial L}{\partial \eta} = \frac{1}{n}(y - \pi). \tag{4}$$
   □

   (b) *Proof.* First,
   $$\frac{\partial^2 L}{\partial \eta^2} = \frac{\partial}{\partial \eta}(\frac{\partial L}{\partial \eta}) = -\frac{1}{n}\frac{\partial}{\partial \eta}(y - \pi). \tag{5}$$
   Then for $i \neq j$,
   $$\left(\frac{\partial^2 L}{\partial \eta^2}\right)_{ij} = \frac{1}{n}\frac{\partial}{\partial \eta_i}\pi_j = 0, \tag{6}$$
   and
   $$\left(\frac{\partial^2 L}{\partial \eta^2}\right)_{ii} = \frac{1}{n}\frac{\partial}{\partial \eta_i}\pi_i = \frac{1}{n}\pi_i(1-\pi_i). \tag{7}$$
   □

2. Problem 10.2, Quadratic approximation to loss functions.

   Proof. By Taylor expansion,
   $$L(\eta) = L(\tilde{\eta}) - v \cdot (\tilde{\eta} - \eta) + \frac{1}{2}(\eta - \tilde{\eta})^\top A(\eta - \tilde{\eta}) + o(|\tilde{\eta} - \eta|^3). \tag{8}$$
   Let $r = \tilde{\eta} - \eta$, then
   $$L(\eta) - L(\tilde{\eta}) = -v \cdot r + \frac{1}{2}r^\top A r + o(|r|^3) = \frac{1}{2}(r - A^{-1}v)^\top A(r - A^{-1}v) - v^\top A^{-1}v + o(|r|^3). \tag{9}$$
   Then
   $$L(\beta) \approx \frac{1}{2}(z - X\beta)^\top A(z - X\beta). \tag{10}$$
   □

3. Problem 13.1, Group lasso analysis of leukemia data

   (a) With this code below, this model selected 54 genes, while the ordinary lasso selected 26 genes. So group lasso is more liberal than ordinary lasso.

```
library("hdrm")
library("splines")
library("grpreg")
downloadData("Golub1999")
attachData("Golub1999")

group_X = c();
for(i in 1:ncol(X)) {
  group_X = cbind(group_X, ns(X[, i], df = 3));
}

fit = cv.grpreg(group_X, y, rep(1:ncol(X), each = 3), family = "binomial");
lambda = which.min(fit$cve);
beta = fit$fit$beta[, lambda];
nparam = which(beta != 0)
```

(b) As can be seen in this table, for both mean error and standard error, allowing nonlinear effects helps to increse the accuracy.

	group lasso	ordinary lasso
cve	0.2587123	0.3856799
cvse	0.06825018	0.1192066