MX_Exact_Logistic

Yang Meng

9/26/2019

Abstract

The article uses Model_X knockoff filter to select variables in logistic regression, with small amplitude. When the amplitude is as small as $\sqrt{\frac{logp}{n}}$, the filter is conservative.

```
library(knockoff)
library(glmnet)
library(MASS)
```

1. Weak Amplitude

$logit(y) = X\beta + \epsilon$, $X \sim N(0, \Sigma)$

$p = 300$, $n = 200$, $k = 15$(non-null variables)

Amplitude $= \sqrt{\frac{logp}{n}}$

$FDP = 0.5$

```
#Simulating data X and logistic y
set.seed(0)
p=300; n=200; k=15
rho=0.4; amplitude=sqrt(2*log(p)/n)
Sigma = toeplitz(rho^(0:(p-1)))
X = matrix(rnorm(n*p),n) %*% chol(Sigma)
nonzero = sample(p, k)
beta = amplitude * (1:p %in% nonzero)
y = X %*% beta + rnorm(n)
y = as.numeric(exp(y) / (1+exp(y)) > .5)
#Generate X.tilt
set.seed(0)
s <- create.solve_sdp(Sigma)
mu = X - X %*% solve(Sigma) %*% diag(s,p)
V = 2 * diag(s,p) - diag(s,p) %*% solve(Sigma) %*% diag(s,p)
X.t = matrix(NA, nrow=n, ncol=p)
for(i in 1:n) X.t[i,] = mvrnorm(n=1, mu[i,], V)
#Variable Selection
set.seed(0)
XX = cbind(X, X.t)
cv.lasso <- cv.glmnet(XX, y, alpha = 1, family = "binomial")
fit = glmnet(XX, y, family = "binomial", alpha = 1, lambda = cv.lasso$lambda.min)
b = as.vector(coef(fit))[-1]
w = abs(b[1:p]) - abs(b[(p+1):(2*p)])
threshold = function(w, q=.5){
  for (t in sort(abs(w[w!=0]))){
    if (((1 + sum(w<=-t)) / sum(w>=t)) <= q) return(t)
```

```
  }}
myresult = which(w>=threshold(w))
myresult
```

```
## [1]  24  55  66 118 154 195 213
```

```
sort(nonzero)
```

```
##  [1]  24  50  66  70  83  85  86 118 131 146 154 213 244 261 275
```

```
#fdp
fdp = function(selected) sum(beta[selected]==0) / max(1, length(selected))
fdp(myresult)
```

```
## [1] 0.2857143
```