LOOCV+BIC

Zheng Yuan

2019/10/7

Bayesian Information Criterion

The BIC (Bayesian information criterion) was developed by Gideon E. Schwarz and published in a 1978 paper,

$$Choose\ the\ model\ for\ which\ logM_j(X_1, X_2, ..., X_n) - \frac{1}{2}k_j logn\ is\ the\ largest$$

Later, the BIC is formally defined as

$$BIC = -2log(\hat{L}) + klogn$$

- $\hat{L}$= the maximized value of the likelihood function of the model M, i.e., $\hat{L} = p(X|\hat{\theta}, M)$, where θ are the parameter values that maximize the likelihood function

- X: the observed data

- n: the sample size

- K: the number of parameters estimated by model M

BIC in Gaussian Linear Models

Under the assumption that the model errors or disturbances are independent and identically distributed according to a normal distribution and that the boundary condition that the derivative of the log-likelihood with respect to the true variance is zero, the maximum log-likelihood can be expressed as a function of $\hat{\sigma}^2$, or equivalently, as a function of RSS. Note that for each Gaussian linear model $M : Y = X\beta + \epsilon$,

$$\begin{aligned} logL(\hat{\beta}, \hat{\sigma^2}) &= log\Big\{(2\pi\hat{\sigma^2})^{-\frac{n}{2}} exp\Big\{-\frac{1}{2\hat{\sigma^2}}||Y - X\hat{\beta}||^2\Big\}\Big\} \\ &= -\frac{n}{2}log(2\pi\hat{\sigma^2}) - \frac{1}{2\hat{\sigma^2}}||Y - X\hat{\beta}||^2 \\ &= -\frac{n}{2}log(2\pi) - \frac{n}{2}log\frac{RSS}{n} - \frac{n}{2} \\ &= C - \frac{n}{2}log\frac{RSS}{n}. \end{aligned}$$

Therefore, BIC becomes

$$BIC = nlog\frac{RSS}{n} + klogn = nlog(\hat{\sigma}^2) + klogn$$

up to an additive constant, which depends only on n and not on the model. The one with the lowest BIC is preferred when selecting from several models.

Leave One Out Cross Validation

In the Bayesian Information Criterion, we use

$$\hat{\sigma}^2 = \frac{RSS}{n}$$

and it represents the **in-sample** estimation of prediction error, therefore a natural thought would be to replace it with an **out-sample** estimation of prediction error, i.e.,the estimation of $\Gamma_{\alpha,n}$ for model M_α, as the same notation as in (Shao, 1993).

Cross-validation (CV) is a class of model selection methods widely used in statistical learning practice. CV does not require the candidate models to be parametric, and it works as long as the data are permutable and one can assess the predictive performance based on some measure.

A specific type of CV is the Leave One Out CV (LOOCV) method. For brevity, given n observations, we leave each one observation out in turn and attempt to predict that data point by using the $n-1$ remaining observations and record the average prediction loss over n rounds. Interestingly, the LOOCV was shown to be asymptotically equivalent to AIC under some regularity conditions.

First, let us start from the CV(1) estimate of $\Gamma_{\alpha,n}$, the true prediction error. In the specific case where the models are Gaussian linear models, CV(1) can be evaluated through projection matrix by a trick in linear algebra.

The CV(1) estimate of $\Gamma_{\alpha,n}$ in Gaussian linear models :

$$\begin{aligned}\hat{\Gamma}^{CV}_{\alpha,n} &= \frac{1}{n}\sum_{i}^{n}[(1-h_{i\alpha})^{-1}(y_i - x_{i\alpha}^T\hat{\beta}_\alpha)]^2 \\ &= \frac{1}{n}\sum_{i=1}^{n}\frac{(Y_i - \hat{Y_{i,\alpha}})^2}{(1-h_{i\alpha})^2}\end{aligned}$$

where $h_{i\alpha}$ is the ith diagonal entry in the projection matrix for M_α, $h_{i\alpha} = [X_\alpha(X_\alpha^T X_\alpha)^{-1}X_\alpha^T]_{ii}$

By comparison with $\frac{RSS}{n}$, the CV(1) estimate can be interpreted as adding a scaling term $\frac{1}{(1-h_{i\alpha})^2}$ to each squared error term $(Y_i - \hat{Y_{i,\alpha}})^2$ in the RSS.

One drawback of selecting model with CV(1) is that CV(1) is too conservative in selecting Gaussian linear models, in the sense that it may select a model of excessive size, unless the optimal model is the one with size p. (Shao 1993)

LOOBIC

By combining the LOOCV and BIC together, a new information criterion, LOOBIC is proposed and is defined as follows:

$$LOOBIC = n * log(\hat{\Gamma}^{CV}_{\_,n}) + klogn$$

LOOBIC automatically derives a model selection procedure based on itself. Specifically, the one with the lowest LOOBIC is preferred when selecting from several models.

The next step is to implement new model selection procedure in R. Therefore, “cvbic” function is generated in R to evaluate “LOOBIC” criterion value for a certain Gaussian linear model, and model selection is based on this criterion value.

```
loocv = function(fit) {

  h = lm.influence(fit)$h
  mean((residuals(fit)/(1-h))^2)

}

cvbic = function(fit) {

  dim(fit$model)[1]*loocv(fit)+(fit$rank)*log(dim(fit$model)[1])

}
```

Traditional BIC criterion value

```
mse <- function(object) {

  mean(residuals(object)^2)
}

bic = function(fit) {
  dim(fit$model)[1]*mse(fit)+(fit$rank)*log(dim(fit$model)[1])
}
```

Simulation

(1) LOOBIC VS LOOCV VS BIC

Dataset Generator

```
def <- defData(varname = "x0", dist = "nonrandom", formula = 1)%>%

defData(,varname = "x1", dist="uniform",formula = "10;20")%>%

defData(,varname = "x2", dist="uniform",formula = "0;3")%>%

defData(,varname = "x3", dist="uniform",formula = "0;5")%>%

defData(,varname = "x4", dist="uniform",formula = "5;10")%>%

defData(, varname = "y1", formula = "2*x0+4*x3", variance = 1)%>%

defData(, varname = "y2", formula = "2*x0+4*x3+8*x4", variance = 1)%>%

defData(, varname = "y3", formula = "2*x0+9*x1+4*x3+8*x4", variance = 1)%>%

defData(, varname = "y4", formula = "2*x0+9*x1+6*x2+4*x3+8*x4", variance = 1)
```

One variable: $y = \beta_0 + \beta_3 x_3$, $\beta_1 = \beta_2 = \beta_4 = 0$

```
md1<-c()
md2<-c()
md3<-c()
for(i in 1:1000){
```

```r
dt <- genData(500, def)

fit1 <- lm(y1 ~ x3, data = dt)
fit2 <- lm(y1 ~ x2+x3, data = dt)
fit3 <- lm(y1 ~ x1+x3, data = dt)
fit4 <- lm(y1 ~ x4+x3, data = dt)
fit5 <- lm(y1 ~ x1+x3+x4, data = dt)
fit6 <- lm(y1 ~ x2+x3+x4, data = dt)
fit7 <- lm(y1 ~ x1+x2+x3, data = dt)
fit8 <- lm(y1 ~ x1+x2+x3+x4, data = dt)


md1<-c(md1,which.min(c(cvbic(fit1),cvbic(fit2),cvbic(fit3),cvbic(fit4),cvbic(fit5),cvbic(fit6),cvbic(fi
md2<-c(md2,which.min(c(loocv(fit1),loocv(fit2),loocv(fit3),loocv(fit4),loocv(fit5),loocv(fit6),loocv(fi
md3<-c(md3,which.min(c(bic(fit1),bic(fit2),bic(fit3),bic(fit4),bic(fit5),bic(fit6),bic(fit7),bic(fit8))
}

table(md1)/1000## empirical probability for each model
```

```
## md1
##     1     2     3     4
## 0.991 0.003 0.004 0.002
```

```r
table(md2)/1000
```

```
## md2
##     1     2     3     4     5     6     7     8
## 0.619 0.105 0.102 0.110 0.019 0.018 0.023 0.004
```

```r
table(md3)/1000
```

```
## md3
##     1     2     3     4
## 0.969 0.008 0.012 0.011
```

Two variables: $y = \beta_0 + \beta_3 x_3 + \beta_4 x_4$, $\beta_1 = \beta_2 = 0$

```r
md1<-c()
md2<-c()
md3<-c()
for(i in 1:1000){

dt <- genData(500, def)

fit1 <- lm(y2 ~ x3+x4, data = dt)
fit2 <- lm(y2 ~ x2+x4, data = dt)
fit3 <- lm(y2 ~ x1+x4, data = dt)
fit4 <- lm(y2 ~ x1+x3+x4, data = dt)
fit5 <- lm(y2 ~ x2+x3+x4, data = dt)
fit6 <- lm(y2 ~ x1+x2+x3+x4, data = dt)


md1<-c(md1,which.min(c(cvbic(fit1),cvbic(fit2),cvbic(fit3),cvbic(fit4),cvbic(fit5),cvbic(fit6))))
md2<-c(md2,which.min(c(loocv(fit1),loocv(fit2),loocv(fit3),loocv(fit4),loocv(fit5),loocv(fit6))))
md3<-c(md3,which.min(c(bic(fit1),bic(fit2),bic(fit3),bic(fit4),bic(fit5),bic(fit6))))
}
```

```r
table(md1)/1000## empirical probability for each model
```

```
## md1
##     1     4     5 
## 0.997 0.001 0.002
```

```r
table(md2)/1000
```

```
## md2
##     1     4     5     6 
## 0.732 0.134 0.115 0.019
```

```r
table(md3)/1000
```

```
## md3
##     1     4     5 
## 0.984 0.009 0.007
```

Three variables: $y = \beta_0 + \beta_1 x_1 + \beta_3 x_3 + \beta_4 x_4$, $\beta_2 = 0$

```r
md1<-c()
md2<-c()
md3<-c()
for(i in 1:1000){

dt <- genData(500, def)

fit1 <- lm(y3 ~ x3+x4, data = dt)
fit2 <- lm(y3 ~ x1+x3+x4, data = dt)
fit3 <- lm(y3 ~ x2+x3+x4, data = dt)
fit4 <- lm(y3 ~ x1+x2+x3+x4, data = dt)


md1<-c(md1,which.min(c(cvbic(fit1),cvbic(fit2),cvbic(fit3),cvbic(fit4))))
md2<-c(md2,which.min(c(loocv(fit1),loocv(fit2),loocv(fit3),loocv(fit4))))
md3<-c(md3,which.min(c(cvbic(fit1),cvbic(fit2),cvbic(fit3),cvbic(fit4))))
}

table(md1)/1000## empirical probability for each model
```

```
## md1
##     2     4 
## 0.996 0.004
```

```r
table(md2)/1000
```

```
## md2
##     2     4 
## 0.832 0.168
```

```r
table(md3)/1000
```

```
## md3
##     2     4 
## 0.996 0.004
```

Full variables: $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + \beta_4 x_4$

```r
md1<-c()
md2<-c()
md3<-c()
for(i in 1:1000){

dt <- genData(500, def)

fit1 <- lm(y4 ~ x1+x2+x3, data = dt)
fit2 <- lm(y4 ~ x1+x3+x4, data = dt)
fit3 <- lm(y4 ~ x2+x3+x4, data = dt)
fit4 <- lm(y4 ~ x1+x2+x3+x4, data = dt)


md1<-c(md1,which.min(c(cvbic(fit1),cvbic(fit2),cvbic(fit3),cvbic(fit4))))
md2<-c(md2,which.min(c(loocv(fit1),loocv(fit2),loocv(fit3),loocv(fit4))))
md3<-c(md3,which.min(c(bic(fit1),bic(fit2),bic(fit3),bic(fit4))))
}

table(md1)/1000## empirical probability for each model
```

```
## md1
## 4
## 1
```

```r
table(md2)/1000
```

```
## md2
## 4
## 1
```

```r
table(md3)/1000
```

```
## md3
## 4
## 1
```