Model Assessment and Selection

By Mehrab Atighi

مقدمه‌ای بر یادگیری آماری

در این درس می‌خواهیم به این دو مسله در مسایل رگرسیون بپردازیم

Model selection

- estimating the performance **of different models in** order to choose the **best one.**

Model assessment

- having chosen a final model, **estimating its prediction error** (generalization error) on **new data**

انواع یادگیری آماری

Model selection

- It is difficult to give a general rule on how to choose the number of observations in each of the three parts, as this depends on the signal to noise ratio in the data and the training sample size. A typical split might be 50% for training, and 25% each for validation and testing

Model assessment

- The methods of this chapter approximate the validation step either analytically (**AIC, BIC, MDL, SRM**) or by efficient sample re-use (crossvalidation and the bootstrap).

داده ها و بخش‌بندی کردن آنها

داده‌های آموزشی: این داده های supervise هستند و ما مدل خود را براساس این داده ها تقسیم‌بندی می‌کنیم.

داده‌های آزمایشی: این داده ها نیز برای تست کردن و بدست آوردن دقت مدل ما یا همان خطای مدل رگرسیونی ما مورد استفاده قرار می‌گیرند.

باید به این نیز توجه داشته باشیم که داده‌های آموزشی و داده‌های آزمایشی ما شامل زوج‌های p متغیره از داده ها و متغیرپاسخ و متغیرهای مستقل ما نیز هستند که بطور کلی داده‌هایمان به این دو تقسیم می‌شوند.

```
sample <- sample(c(TRUE,FALSE),nrow(Data),replace = T,prob = c(0.6,0.4))
train <-Data[sample,]
test <-Data[!sample,]
```

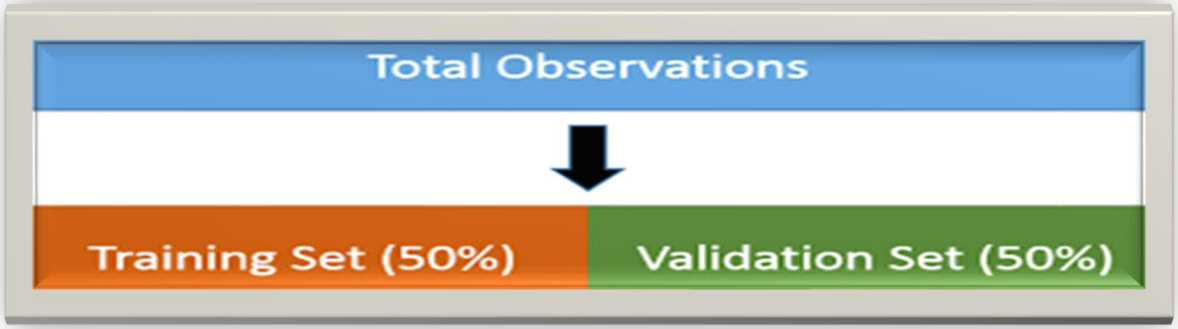
Validation set Approch

ما در این روش زمانی که داده‌هایمان را به داده‌های آموزش و آزمون تقسیم بندی می‌کنیم بصورت ۶۰-۴۰ یا ۷۰-۳۰ این کار را انجام می‌دهیم یعنی اینکه مثلا ۶۰ درصد داده‌های ما آموزش باشند، که به کمک آنها مدل را بسازیم و مدل ساخته شده را روی ۴۰ درصد مابقی که شامل داده‌های تست یا آزمون ما هستند، تست کرده و میزان خطای آنرا براساس معیار موردنظر بررسی بکنیم.

$$sample <- sample(c(TRUE, FALSE), nrow(Data), replace = T, prob = c(0.6,0.4))$$
$$train <- Data[sample,]$$
$$test <- Data[!sample,]$$

برای داشتن یک مدل رگرسیون درجه دو سه و یا بیشتر بصورت روبرو انجام می‌دهیم:

$$lm(y \sim poly(x, deg))$$



Cross validation

- **ارزیابی متقابل:**

چرا روش‌های بازنمونه‌گیری را بکار ببریم؟

گاهی اوقات که ما می‌آییم و بصورت تصادفی و مکرر از داده‌های آموزشی خود مجموعه‌های مختلفی را انتخاب می‌کنیم و هربار اطلاعات جدیدتری به ما ارایه می‌شود. و ما بدنبال دو هدف ارزیابی و انتخاب مدل خود هستیم.

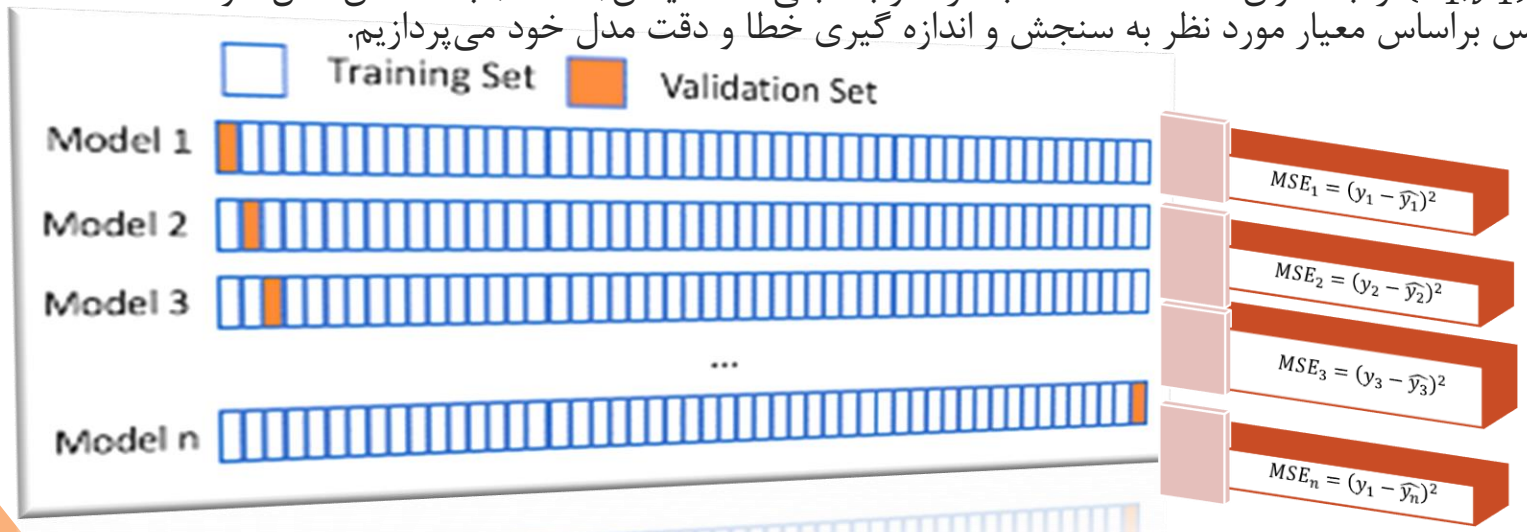
- Leave one out cross validation
- K fold cross validation
- Cross validation for classification problem miss rate

Leave one out cross validation

فرض کنید که تعداد داده‌های ما خیلی زیاد نیست ما از این روش استفاده می‌کنیم با فرض داشتن n داده داریم:

(x_1, y_1) (x_2, y_2) ... (x_n, y_n)

در این روش ما (x_1,y_1) را به عنوان داده تست انتخاب کرده و با مابقی داده‌هایمان(n-1تا) به ساختن مدل خود می‌پردازیم و سپس براساس معیار مورد نظر به سنجش و اندازه گیری خطا و دقت مدل خود می‌پردازیم.



حال این nتا معیار نیز می‌توانند به ازای هر درجه مختلفی از مدل رگرسیون مثلا درجه یک دو سه و... nتا داده باشند و داریم:

$$Cv_n = \frac{1}{n}\sum_{i=1}^{n} MSE_i$$

Leave one out cross validation

- #R codes:

$$glm.fit <- glm(y \sim x, data = \cdots, \ldots)$$

$$loocv.err <- cv.glm(data, glm.fit)$$

$$str(loocv.err)$$

$$loocv.err\$\, delta[n]$$

همان mse_n ما می‌باشد:

در این روش MSE ما بسیار کم می‌شود ولی از نظر زمانی هرچه تعداد داده‌های ما بیشتر شود، زمان بر می‌شود و محاسبات زیادتر و سنگین‌تری خواهد داشت.

K Fold Cross Validation

ارزیابی متقابل kتایی:

دراین روش تعداد داده‌های ما باید ضریبی از k باشد، فرض کنیم nتا مشاهده داریم و آنها را به k گروه تقسیم می‌کنیم.

درمرحله اول گروه اول را به‌عنوان داده‌های تست و مابقی گروه‌ها به‌عنوان داده‌های آموزش استفاده می‌شوند.

درمرحله دوم گروه دوم به‌عنوان داده‌های تست و مابقی گروه‌ها به‌عنوان داده‌های آموزش استفاده می‌شوند و به همین ترتیب، درنهایت گروهk ام به‌عنوان داده‌های تست و مابقی گروه‌ها به‌عنوان داده‌های آموزش استفاده می‌شوند.

وبطور معمول نیز k=10 یا k=5 قرار می‌دهند.

K Fold Cross Validation

$\#R\ codes\ for\ K\ Fold\ Cross\ Validation$:

درکدنویسی همان تابع $cv.glm$ را استفاده می‌کنیم با این تفاوت که دراینجا باید میزان kرا ذکر کنیم:

$kv.fit <- lm(y \sim x, data = \cdots)$

$cv.glm(data, kv.fit)\$delta[n] \quad = MSE_n$

$kfcv - err <- function(x)\{$

$fit <- lm(y \sim x, data = \cdots)$
$cv.glm(data\ , fit, k = 1\ or\ 10\ or\ 5\ )\$delta[1]$

$\}$

$library("purr")$
$1{:}\,5\% > \%\ map - db; (kfcv - err)$ از درجه یک تا درجه ۵ را به ما می‌دهد.

Cross Validation for Classification Problem Miss Rate

فرق این روش در اینجا به این صورت هست که معیار ارزشیابی ما در اینجا دیگر MSE نخواهد بود و از نسبت بد طبقه بندی شده استفاده می‌کنیم.

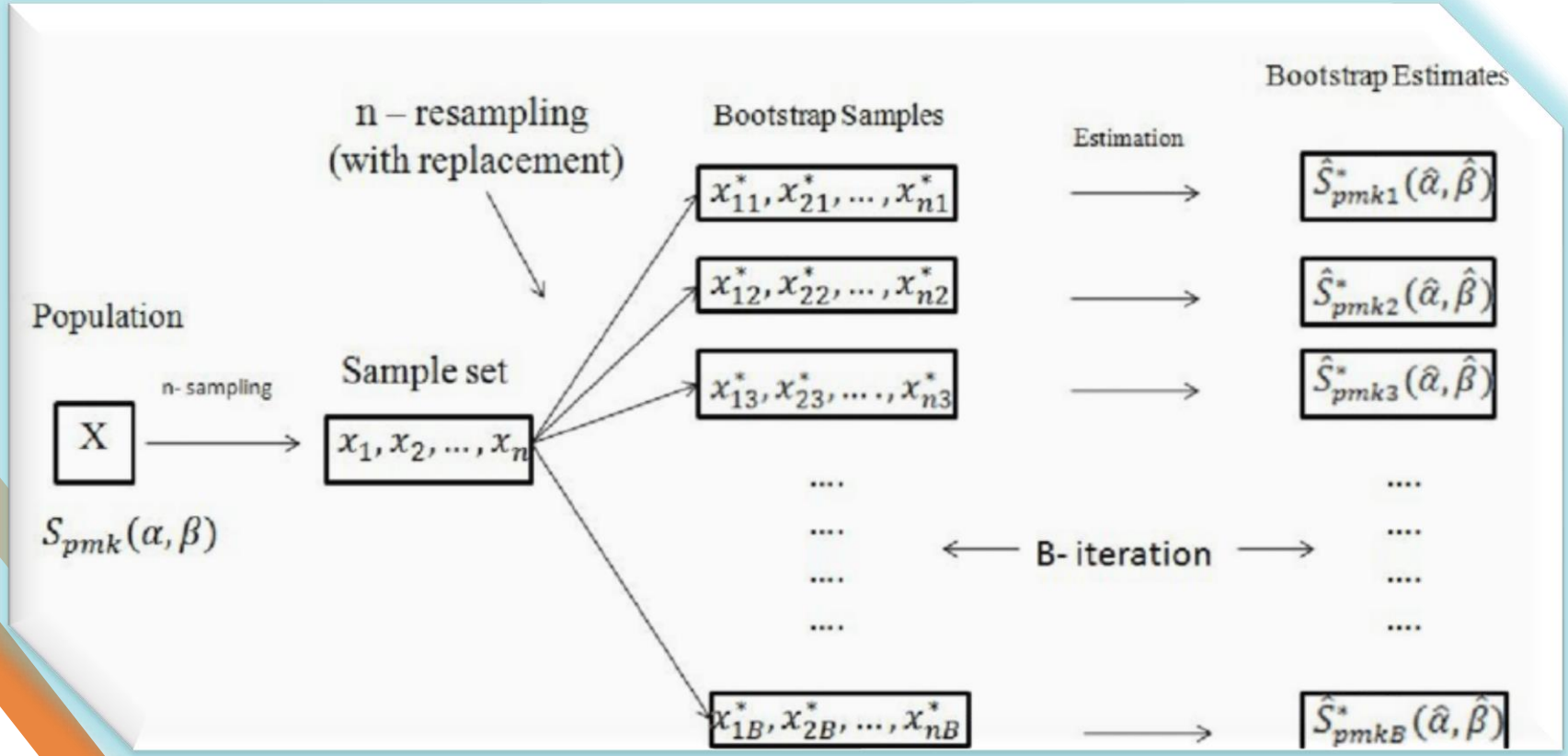
$$\#R\ Codes, Cross\ Validation\ for\ Classification\ Problem\ Miss\ Rate$$
$$glm.fit <- glm(y \sim x\ , family = \text{"binomial"}, data = \cdots)$$
$$Cost <- function(r, pi = 0)\{mean(abs(r - pi) > 0.5)\}$$
$$cv.glm(data, glm.fit, cost, k = 1\ or\ k = 10\ or\ k = 5)\$delta[1]$$

مقدار خروجی بالا هرچه به ۰.۵ نزدیک تر باشد نشانه دقیق‌تر بودن پیشگویی مدل ما می‌باشد.

Boot strapping

خودگردان سازی:

یک روشی‌است که در آن عدم قطعیت یک برآوردگر را مورد بررسی قرار می‌دهیم. ما دراینجا از این روش برای بررسی ضرایب رگرسیونی خود استفاده می‌کنیم و زمانی که نتوانیم خطای استاندارد شده را برآورد بکنیم از این روش استفاده می‌کنیم.



Training and Test Errors

خطای آموزش:

خطای آموزش ما میزان خطایی هست که ما به کمک داده های آموزش خود می‌آییم و طبق معیارمورد نظر میزان دقت و خطای مدلی که بر اساس همین داده‌ها تشکیل شده است با اندازه‌گیری می‌کنیم.

خطای آزمون:

خطای آزمون ما میزان خطایی هست که ما به کمک داده‌های آزمون و یا تست خود روی مدلی که به کمک داده‌های آموزش تشکیل شده است تست کرده و میزان خطارا اندازه‌گیری می‌کنیم.

نکته: هدف ما کم بودن خطای آزمون هست و گاها ممکن است خطای آموزش خیلی کم باشد ولی خطای آزمون بسیار زیاد.

Boot Stapping

#R Codes for Boot strapping Method

ایده کار در نرم افزار Rبرای این روش بدین صورت است که

1) ساختن یک تابعی که آماره موردنظر(ضرایب رگرسیونی) را بسازد.

2) ۲)تابع $boot$ را از پکیج $boot$ استخراج بکنیم تا بتوانیم از آن استفاده بکنیم.

```
library("boot")
Statistic <-function(data, index){
glm.fit
<-glm(y~poly(x, deg), data = ⋯, subset = index , family
= ⋯, ...)
coef(glm.fit)
}
statistic(data , index)
boot(data , statistic, 1000)
```

Criterions for Step3

برای انتخاب کردن معیارهای ارزیابی ۲ روش کلی وجود دارد.

1. بصورت غیرمستقیم خطای آزمون را محاسبه بکنیم و برآورد بکنیم.
2. بصورت مستقیم به بررسی خطای آزمون بپردازیم.

Evaluate Models

با توجه به C_p که در اسلاید قبلی درمورد آن بحث شد، می‌خواهیم معیارهای دیگر را معرفی بکنیم:

$$AIC = \frac{1}{\widehat{\sigma^2}} C_p$$

این معیار نیز، هرچه مقدار کمتری داشته باشد نشانه بهتر بودن مدل ما خواهد بود.

$$BIC = \frac{1}{n}\left(Rss + \log(\mathrm{n})\mathrm{d}\widehat{\sigma^2}\right)$$

این معیار نیز، هرچه مقدار کمتری داشته باشد نشانه بهتر بودن مدل ما خواهد بود، و اگر $\log(n) > 2$ یا $n > 7$ باشد، ضریب $d\widehat{\sigma^2}$ نسبت به ضریب C_p بیشتر خواهد بود.

$$R^2 = 1 - \left(\frac{Rss}{SST}\right) \quad , Adj.R^2 = 1 - \left(\frac{MSE}{MST}\right)$$

این معیار نیز، هرچه مقدار بیشتری داشته باشد نشانه بهتر بودن مدل ما خواهد بود.

$$\#R\ codes\ For\ AIC, BIC, C_p, Adjusted\ R\ squred:$$
$$result < -summary(best - subset)$$
$$Cp = result\$adjr2$$
$$BIC = result\$bic$$
$$which.\max(adj - R2)\,; which.\min(Cp)\,; which.\min(BIC)$$ اندیس را به ما می‌دهد.

روش مستقیم

در این روش ما بصورت مستقیم به بررسی خطای آزمون می‌پردازیم: در این روش ما باید خطای ارزیابی متقابل خود را برای مدل‌ها حساب کرده و بهترین را انتخاب بکنیم. ما دراینجا برای مدل‌های مختلف با تعداد متغییرهای پیشگو مختلف یک خطای استاندارد(one standard error) داریم. و برای هریک از این مدل‌های p متغییره، می‌آییم و مدلی را که کمترین میانگین انحراف استاندارد را دارد انتخاب می‌کنیم و آنرا با درنمودار خود + و − sd رسم می‌کنیم و بعد هریک از مدل‌هایی که داخل این بازه قرار داشتند، ساده ترین آنها بهترین مدل ما خواهد بود.

$$X = \begin{matrix} obs1 \\ \vdots \\ obsn \end{matrix} \begin{bmatrix} 1 x_1 & \dots & x_p \\ 1 \vdots & \dots & \vdots \\ 1 \vdots & \dots & \vdots \end{bmatrix}_{n*p+1}$$

ما بعد از انتخاب بهترین مدل p متغیره خود، ضرایب آنرا استخراج می‌کنیم که آنها p+1 تا خواهند بود و بعد بصورت زیر مقادیر را پیش‌بینی خواهیم کرد.

$$\hat{y} = \beta X$$

روش مستقیم

```
#R codes for Cross Validaion set Approches Method:
Validation - erros <- -vector("double",lenght=19)
for(i in 1:19){coef - x <- -coef(best - subset, id = i)
pred - x <- -test - m[,names(coef - x)]% * %coef - x
validation - errors[i] = mean((test$y - pred - x)^2)}
Plot(Validation - errors, type = "b")
```

روش مستقیم

```
#R codes for K Fold Cross Validation
predict.regsubset = function(object, newdata, id, ...){
    form <- as.formula(object$call[[2]])
    mat <- model.matrix(form, newdata)
    coefi <- coef(object, id = id)
    xvars <- names(coefi)
    mat[, xvars] %*% coefi
    }
    K <- 10
    Folds <- sample(1:K , nrow(data), replace = TRUE)
    Cv.errors <- matrix (NA, k, 19, dimnames = list(NULL, paste(1:19)))
    for(j in 1:k){
    nest - subset <- regsubset(y~x, data[folds == j, ], nvmax = 19)
    for(i in 1:19){
    pred - x = predict.regsubset(best - subset, data[folds == j, ], id = i)
    Cv - errors[i, j] <- mean(data$y[folds == j] - pred - x)^2)
    }}
```

End

Thanks For Youre Attention

Join us ...