Two inferential problems

Consider the linear model

$$y = \mathcal{N}(1_n\beta_0 + X\beta, \sigma^2 I_n)$$

where

- $\underset{n\times 1}{y} = (y_1,\ldots,y_n)'$ is the response on n observations
- $\underset{n\times p}{X}$ is the design matrix containing the measurements on p variables
- $\underset{p\times 1}{\beta} = (\beta_1,\ldots,\beta_p)'$ is the vector of coefficients of interest
- β_0 and σ^2 are nuisance parameters
- $\underset{n\times 1}{1_n} = (1,1,\ldots,1)'$ is a vector of ones of length n and $\underset{n\times n}{I_n}$ is the identity matrix

Multiple testing

The first inferential problem concerns multiple testing, in particular what happens when we perform simultaneously a large number of tests.

In the linear model described above, we can for example consider testing

$$H_j : \beta_j = 0 \quad \text{vs} \quad \bar{H}_j : \beta_j \neq 0, \qquad j = 1,\ldots,p$$

If we reject the null hypothesis H_j, then we can say that the jth variable is 'important' in explaing the response.

Consider the following scenario:

- $n = 100$, $p = 25$
- X with generic element $x_{ij} \sim U(0,1)$, $\beta_0 = 2$
- $\beta_j = 1$ if $j \in \{1,2,\ldots,5\}$ and $\beta_j = 0$ for $j \in \{6,\ldots,25\}$

thus only the first 5 variables are important.

Now we generate the data:

```r
rm(list=ls())
set.seed(123)
n = 100
p = 25
# design matrix
X = matrix(runif(n*p), ncol=p)
colnames(X) = paste0("X",1:p)
# betas
beta = c(rep(1,5),rep(0,p-5))
# response
y = 2 + X %*% beta + rnorm(n)
# data
yX = data.frame(y,X)
# linear model
fit <- lm(y~., yX)
summary(fit)
```

```
##
## Call:
## lm(formula = y ~ ., data = yX)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.23153 -0.45874 -0.03247  0.59968  2.86874
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.17284    0.85273   3.721 0.000384 ***
## X1           1.13970    0.36029   3.163 0.002263 **
## X2           0.73219    0.37462   1.954 0.054424 .
## X3           0.83722    0.35605   2.351 0.021369 *
## X4           1.35891    0.35763   3.800 0.000295 ***
## X5           0.85090    0.38758   2.195 0.031268 *
## X6          -0.93589    0.37564  -2.491 0.014962 *
## X7           0.34423    0.39484   0.872 0.386125
## X8          -0.17207    0.34938  -0.492 0.623836
## X9          -0.47943    0.37020  -1.295 0.199324
## X10         -0.01770    0.36849  -0.048 0.961820
## X11          0.84622    0.40053   2.113 0.037997 *
## X12          0.28015    0.37484   0.747 0.457186
## X13         -0.02829    0.35072  -0.081 0.935928
## X14          0.03171    0.40718   0.078 0.938134
## X15         -0.57929    0.35571  -1.629 0.107661
## X16          0.11734    0.41671   0.282 0.779047
## X17         -0.01173    0.38927  -0.030 0.976033
## X18         -0.56350    0.35575  -1.584 0.117462
## X19         -0.74730    0.38972  -1.918 0.059034 .
## X20          0.82704    0.39103   2.115 0.037793 *
## X21         -0.46234    0.34215  -1.351 0.180715
## X22         -0.56520    0.37710  -1.499 0.138183
## X23          0.11756    0.35679   0.329 0.742722
## X24         -0.49497    0.36884  -1.342 0.183715
## X25          0.51115    0.34910   1.464 0.147368
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.925 on 74 degrees of freedom
## Multiple R-squared:  0.5001, Adjusted R-squared:  0.3312
## F-statistic: 2.961 on 25 and 74 DF,  p-value: 0.0001618
```

If we reject all the null hypotheses which have p-values less than 5%, we commit 3 type I errors:

```
# type I errors
names(which(summary(fit)$coef[-c(1:6),4] < 0.05))
```

```
## [1] "X6"  "X11" "X20"
```

We would like to avoid to conclude that X6, X11 and X20 are important variables.

Inference after model selection

This is our second inferential problem. Inference after model selection was typically done ignoring the model selection process.

Consider the following *high-dimensional* scenario:

- $n = 25$, $p = 100$
- X with generic element $x_{ij} \sim U(0,1)$, $\beta_0 = 2$
- $\beta_j = 1$ if $j \in \{1, 2, \ldots, 5\}$ and $\beta_j = 0$ for $j \in \{6, \ldots, 25\}$

thus only the first 5 variables are important.

Generate the data:

```
rm(list=ls())
set.seed(123)
n = 25
p = 100
# design matrix
X = matrix(runif(n*p), ncol=p)
colnames(X) = paste0("X",1:p)
# betas
beta = c(rep(2,5),rep(0,p-5))
# response
y = 2 + X %*% beta + rnorm(n)
# data
yX = data.frame(y,X)
```

Perform the forward selection algorithm and select the model with 5 variables:

```
fml.full <- as.formula(paste("y ~ ", paste(colnames(X), collapse= "+")))
fit.null <- lm(y~1,yX)
fit.fwd <- step(fit.null, scope=fml.full, direction="forward", steps=5, trace=0)
# selected variables
S.fwd <- attr(fit.fwd$coefficients, "names")[-1]
S.fwd
```

```
## [1] "X15" "X64" "X59" "X7"  "X58"
```

None of the selected variables is important. But if we fit the linear model with the selected variables, something is going wrong with the inference on the selected model

```
# inference after model selection
fml.fwd <- as.formula(paste("y ~ ", paste(S.fwd, collapse= "+")))
summary(lm(fml.fwd,yX))
```

```
##
## Call:
## lm(formula = fml.fwd, data = yX)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68756 -0.17049  0.01592  0.22707  0.78753
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  10.4441     0.4331  24.114 1.04e-15 ***
```

```
## X15          -3.1594     0.3570  -8.851 3.62e-08 ***
## X64          -2.7947     0.4090  -6.833 1.60e-06 ***
## X59           2.9191     0.3958   7.375 5.49e-07 ***
## X7           -2.4660     0.5180  -4.760 0.000136 ***
## X58          -1.2984     0.4088  -3.176 0.004972 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4693 on 19 degrees of freedom
## Multiple R-squared:  0.8791, Adjusted R-squared:  0.8472
## F-statistic: 27.62 on 5 and 19 DF,  p-value: 4.285e-08
```

Alternatively, we can use the LASSO algorithm to select the 5 variables, but the same problem happens:

```r
library(glmnet)
fit.lasso <- glmnet(X,y, alpha=1, dfmax=5)
lambda = fit.lasso$lambda[which.max(fit.lasso$df>=5)]
# selected variables
S.lasso = colnames(X)[which(coef(fit.lasso, s=lambda)[-1]!=0)]
S.lasso
```

```
## [1] "X3"  "X15" "X33" "X64" "X70"
```

```r
# inference after model selection
fml.lasso <- as.formula(paste("y ~ ", paste(S.lasso, collapse= "+")))
summary(lm(fml.lasso,yX))
```

```
##
## Call:
## lm(formula = fml.lasso, data = yX)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.55850 -0.29624 -0.07957  0.30725  1.17850
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.6419     0.6582  14.648 8.35e-12 ***
## X3            1.0459     0.5517   1.896 0.073314 .
## X15          -2.3450     0.5476  -4.283 0.000402 ***
## X33          -1.0242     0.5759  -1.778 0.091345 .
## X64          -1.7100     0.6386  -2.678 0.014880 *
## X70          -0.6220     0.4743  -1.311 0.205405
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7163 on 19 degrees of freedom
## Multiple R-squared:  0.7183, Adjusted R-squared:  0.6442
## F-statistic: 9.689 on 5 and 19 DF,  p-value: 0.0001011
```

However, the problem with inference on the selected model seems to disappear if we select the 5 variables at random:

```r
# selected variables
S.random <- sample(colnames(X),5)
S.random
```

```
## [1] "X60" "X16" "X51" "X41" "X80"
```

```r
# inference after model selection
fml.random <- as.formula(paste("y ~ ", paste(S.random, collapse= "+")))
fit.random <- lm(fml.random,yX)
summary(fit.random)
```

```
##
## Call:
## lm(formula = fml.random, data = yX)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.7964 -0.6210  0.1008  0.6575  1.8504
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   7.6726     0.9870   7.774 2.56e-07 ***
## X60           0.3572     0.8614   0.415    0.683
## X16           0.1689     0.8633   0.196    0.847
## X51          -0.2650     0.8154  -0.325    0.749
## X41           0.6259     1.2793   0.489    0.630
## X80          -1.7154     1.0823  -1.585    0.129
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.257 on 19 degrees of freedom
## Multiple R-squared:  0.1329, Adjusted R-squared:  -0.09527
## F-statistic: 0.5825 on 5 and 19 DF,  p-value: 0.7131
```

```r
# confidence intervals
confint(fit.random)[-1,]
```

```
##         2.5 %    97.5 %
## X60 -1.445672 2.1600729
## X16 -1.638041 1.9757922
## X51 -1.971634 1.4415790
## X41 -2.051700 3.3035283
## X80 -3.980704 0.5498592
```

References

- Efron and Hastie (2016) Computer-Age Statistical Inference: Algorithms, Evidence, and Data Science, Cambridge University Press