Problem doboru zmiennych w modelu liniowym

W poniższej tabeli zebrano dane dotyczące wartości miesięcznej sprzedaży (Volume) w 20 aptekach. Zebrano także informacje dotyczące powierzchni apteki (Floor space), procent całkowitej powierzchni zajmowanej przez dział sprzedaży leków na receptę (PrescRX), liczby miejsc parkingowych, czy apteka jest w centrum handlowym, dochód na osobę okolicznej ludności
Dobrać odpowiedni model wyjaśniający wielkość sprzedaży apteki

	Ott 461					
	1 Volume	2 Floor space	3 Presc RX	4 Parking	5 ShopCntr	6 Income
1	22	4900	9	40	tak	18
2	19	5800	10	50	tak	20
3	24	5000	11	55	tak	17
4	28	4400	12	30	nie	19
5	18	3850	13	42	nie	10
6	21	5300	15	20	tak	22
7	29	4100	20	25	nie	8
8	15	4700	22	60	tak	15
9	12	5600	24	45	tak	16
10	14	4900	27	82	tak	14
11	18	3700	28	56	nie	12
12	19	3800	31	38	nie	8

Kryteria oceny modelu

Kryterium AIC jest oparte na teorii informacji. Przypuśćmy że dane są generowane przez pewien proces (mechanizm) losowy f. Rozważmy dwa modele g_1 i g_2 mające reprezentować proces f. Gdybyśmy znali f, moglibyśmy obliczyć stratę informacji o f reprezentując f przez g_1 lub g_2 obliczając "odległość " Kullbacka-Leiblera

$$D_{KL}(f,g_i)=\int_{x\in X} f(x)\log\frac{f(x)}{g_i(x)}dx \quad i=1,2.$$

(ogólnie $D_{KL}(P,Q_i)=\int_{x\in X}\log\frac{P(dx)}{Q_i(dx)}P(dx),\ P<<Q$).

Z tych dwóch modeli wybieramy ten, który charakteryzuje się mniejszą stratą informacji. Niestety nie znamy modelu f. Jednak jak pokazał Akaike (1974) możemy poprzez jego kryterium AIC szacować, o ile więcej (lub mniej) informacji o f tracimy wybierając g_1 zamiast g_2. Tak więc ze skończonego zbioru modeli $g_i; i=1,...,M$ możemy wybrać ten, który charakteryzuje się najmniejszą stratą informacji o f. Niech $\Lambda=-2\ln L(\hat{\boldsymbol{\beta}},\hat{\sigma}^2)$ oznacza maksimum logarytmu funkcji wiarogodności a Φ liczbę wszystkich parametrów modelu. Wówczas

$$AIC=-2\ln L(\hat{\boldsymbol{\beta}},\hat{\sigma}^2)+2\Phi=n+n\log(2\pi)+n\log(\frac{RSS}{n})+2\Phi$$

W przypadku modelu liniowego z p zmiennymi objaśniającymi $\Phi=\dim\beta+1=p+1=k+2$, bo wymiar wektora $\boldsymbol{\beta}$ wynosi $p=k+1$ i jeszcze jeden parametr σ^2.

Oczywiście nie mamy żadnej gwarancji że najlepszy model z rozważanej klasy jest satysfakcjonujący.

Kryterium AIC ma ponadto charakter asymptotyczny i przy małej liczności zbioru obserwacji niezbędne są pewne korekty kryterium AIC. Zastępowane jest ono prze AICc, które ma różną postać dla różnych modeli. Generalnie AICc nakłada większą karę za dodatkowe parametry niż AIC. Podobne, lecz uzyskane w podejściu bayesowskim jest kryterium

$$BIC = -2\ln L(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2) + \ln(n)\ \Phi$$

W programie R używane są

- *GIC* (*Generalized Information Criterion*)

$$GIC = -2\ln L(\hat{\boldsymbol{\beta}}, \hat{\sigma}^2) + h \cdot \Phi$$

gdzie h to pewien współczynnik, Φ -liczba parametrów w aktualnym modelu . Dwa specjalne przypadki *GIC* to kryterium Akaike (*AIC*) w którym h=2 oraz kryterium Schwartza w którym h=log(n) gdzie n oznacza liczbę obserwacji.

- Zmodyfikowany współczynnik R^2_{adj}
- Statystyka *Cp Mallowsa* $E\sum_{i=1}^{n}(\hat{y}_i - E(y_i|X_i)^2/\sigma^2$ szacowana jako

$$C_p(M) = \frac{RSS_p}{S^2} - n + 2p,$$

gdzie RSS_p to suma kwadratów reszt w modelu z p zmiennymi a S^2 to suma kwadratów reszt w modelu ze wszystkimi zmiennymi. Im mniejsza wartość C_p tym model lepszy. Statystykę C_p wyznacza funkcja ols_mallows_cp {olsrr}.

Idea konstrukcji wskaźnika C_p Mallowsa jest następująca. Mając próbę uczącą $(\mathbf{x}_i, y_i)\ i = 1,...,n$ oraz funkcję straty L możemy przyjąć kryterium jakości rozwiązania $\hat{f}$ rozważanego problemu

$$\frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{f}(\mathbf{x}_i)).$$

Takie przybliżenie ryzyka, a zatem prawdziwej mocy predykcyjnej rozwiązania $\hat{f}$ jest oczywiście przybliżeniem optymistycznym. Niech miarą optymizmu będzie wielkość

$$\text{op} = \frac{1}{n}\sum_{i=1}^{n} E_{y_i^*} L(y_i^*, \hat{f}(\mathbf{x}_i)) - \frac{1}{n}\sum_{i=1}^{n} E_{y_i} L(y_i, \hat{f}(\mathbf{x}_i)),$$

gdzie $y_i^* = i = 1,...,n$ oznaczają nowe obserwacje zmiennej odpowiedzi niezależne od y_i i odpowiadające tym samym punktom próby uczącej $\mathbf{x}_i$ traktowanym jako punkty ustalone. Model

$\hat{y}_i = \hat{f}(\mathbf{x}_i)$ jest dopasowany do próby uczącej $(\mathbf{x}_i, y_i)$ $i = 1,...,n$ i dlatego optymizm przyjmuje zwykle wartości dodatnie. Można udowodnić, że przy ogólnych założeniach i kwadratowej funkcji straty (możliwe są też inne postacie funkcji straty)

$$\text{op} = \frac{2}{n}\sum_{i=1}^{n} Cov(\hat{y}_i, y_i).$$

Łatwo zauważyć, że im bardziej model jest dopasowany do próby uczącej tym większy będzie optymizm. Jeżeli prawdziwy model jest postaci $y = f(\mathbf{x}_i) + \varepsilon$ gdzie funkcję f można zapisać jako liniową kombinację p znanych funkcji bazowych oraz ε jest błędem losowym o zerowej wartości oczekiwanej i wariancji σ_ε^2, to $\sum_{i=1}^{n} Cov(\hat{y}_i, y_i) = p\sigma_\varepsilon^2$. Ostatecznie zamiast minimalizować $\frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{f}(\mathbf{x}_i))$ bardziej uzasadniona jest minimalizacja wskaźnika

$$\frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{f}(\mathbf{x}_i)) + \frac{2}{n} p\sigma_\varepsilon^2,$$

znanego jako wskaźnik C_p Mallowsa. W języku R używana jest równoważna postać

$$C_p = \frac{RSS_p}{S^2} - n + 2p,$$

RSS_p= deviance() model z p zmiennymi objaśniającymi
s<-summary_fullmodel$sigma #ocena sigma dla pełnego modelu dostępna dla obiektu klasy summary.lm

Wybór najlepszego modelu $Y = \beta_0 + \beta_1 X_1 + \cdots + \beta_k X_k + \varepsilon$

Algorytm A (Best model)- przeszukujemy wszystkie 2^k modeli

1. Niech M_0 oznacza model zerowy, który nie zawiera predyktorów. Ten model po prostu przewiduje średnią próbki dla każdej obserwacji.
2. Dla i=1,…,k
   a) Dopasować wszystkie $\binom{k}{i}$ modeli z i predyktorami
   b) Wybierz najlepszy spośród tych $\binom{k}{i}$ modeli i nazwij go M_i. Za najlepszy uważany jest tu model z najmniejszym RSS lub równoważnie największym R^2.
3. Wybierz jeden najlepszy model spośród M_0, . . . ,M_k przy użyciu zweryfikowanego krzyżowo błędu przewidywania, C_p (AIC), BIC lub skorygowanego R^2.

Krok 2 algorytmu A redukuje problem wyboru najlepszego modelu spośród 2^k do wyboru jednego z k+1 modeli.
W kroku 3 wybieramy najlepszy model spośród k+1 modeli stosując jedno z kryteriów CV, AIC, BIC, C_p i R^2_{adj} .

Algorytm B -Regresja krokowa forward

1. Niech M_0 oznacza model zerowy, który nie zawiera predyktorów.
2. Dla i=0,…,k-1
   a) Rozważyć wszystkie k-i modeli które dodają jeden predyktor do M_i
   b) Wybierz najlepszy spośród tych k-i modeli i nazwij go M_{i+1}. Za najlepszy uważany jest tu model z najmniejszym RSS lub równoważnie największym R^2.
3. Wybierz jeden najlepszy model spośród M_0, . . . ,M_k przy użyciu zweryfikowanego krzyżowo błędu przewidywania, Cp , AIC, BIC lub skorygowanego R^2.

Algorytm C- Regresja krokowa backward

1. Niech M_k oznacza model pełny, który zawiera wszystkie predyktory.
2. Dla i=k,k-1,…, 1
   a) Rozważyć wszystkie k modeli które zawierają wszystkie oprócz jednego predyktora M_i dla wszystkich i-1 predyktorów
   b) Wybierz najlepszy spośród tych i modeli i nazwij go M_{i-1}. Za najlepszy uważany jest tu model z najmniejszym RSS lub równoważnie największym R^2.
3. Wybierz jeden najlepszy model spośród M_0, . . . ,Mk_p przy użyciu zweryfikowanego krzyżowo błędu przewidywania, Cp , AIC, BIC lub skorygowanego R^2.

```
library(openxlsx)
#wybór optymalnego modelu
dane2 <- read.xlsx("C:/Users/User/Documents/Dydaktyka/Modele liniowe/Ćwiczenia/Ott461.xlsx",colNames =
TRUE)
dane3<-dane2 #przygotowanie danych
colnames(dane3)<-c("Volume","Floor_space","Presc_RX","Parking","ShopCntr","Income")
dane3$ShopCntr <- factor(dane3$ShopCntr)
levels(dane3$ShopCntr) <- c("no","yes")
attach(dane3)
zm=c("Floor_space","Presc_RX","Parking","ShopCntr","Income")
#indeksy wszystkich podzbiorów indeksowanego zbioru length(zm)
library(e1071)
help("bincombinations")#zwraca macierz  2^p wektorów o długości p
bincombinations(2)#przykład 1
bincombinations(3) #przykład 2
bincombinations(3)[-1,] #pomijamy pierwszy wiersz odpowiadający modelowi pustemu
wsp = (bincombinations(length(zm))==1 )[-1,]
params=matrix(0,nrow(wsp),5)#przygotowujemy macierz na wskaźniki jakości poszczególnych modeli
library(olsrr)# do Cp
fullmodel<-lm(Volume ~ Floor_space+Presc_RX+Parking+ShopCntr+Income,data=dane3)#potrzebne do Cp

for (i in 1:nrow(wsp)) {
  form = as.formula(paste("Volume~",paste(zm[wsp[i,]],collapse="+")))
  model=lm(form,data=dane3)
  params[i,1]=AIC(model,k=log(nrow(dane3)))
```

```
  params[i,2]=model$rank
  params[i,3]=summary(model)$adj.r.squared
  params[i,4]=AIC(model)
  params[i,5]=ols_mallows_cp(model, fullmodel)
}


# model optymalny w sensie BIC
as.formula(paste("Volume~",paste(zm[wsp[which.min(params[,1]),]],collapse="+")))

# model optymalny w sensie R^2_adj
as.formula(paste("Volume~",paste(zm[wsp[which.max(params[,3]),]],collapse="+")))

# model optymalny w sensie  AIC
as.formula(paste("Volume~",paste(zm[wsp[which.min(params[,4]),]],collapse="+")))

# model optymalny w sensie  Cp
as.formula(paste("Volume~",paste(zm[wsp[which.min(params[,5]),]],collapse="+")))


AIC(lm(Volume ~ Floor_space + Presc_RX))
#wyznaczenia na piechotę wskaźnika AIC
-2*logLik(lm(Volume ~ Floor_space + Presc_RX))+8 #4 parametry:wyraz wolny,Floor_space,Presc_RX, sigma^2
extractAIC(lm(Volume ~ Floor_space + Presc_RX))#  help(extractAIC)
# zwraca  p=dim(coeff) i AIC=nlog(RSS/n)+2p  (pomija stałą n+nln(2*pi)) i nie wlicza sigma^2 jako parametru
#wyznaczenia na piechotę wskaźnika Cp
deviance(lm(Volume ~ Floor_space + Presc_RX))
summary_fullmodel<-summary(fullmodel)
s<-summary_fullmodel$sigma #ocena sigma dla pełnego modelu dostępna dla obiektu klasy summary.lm
deviance(lm(Volume ~ Floor_space + Presc_RX))/s^2+6-20 #6=2*(2+1)
deviance(lm(Volume ~ Floor_space + Presc_RX))/s^2+2*lm(Volume ~ Floor_space + Presc_RX)$rank-20
#6=2*(2+1)
```

- Wybór modelu w oparciu o $RMSE_{LOOCV}$
  leave-one-out cross validated RMSE

plik Dp.386.R

tworzymy dane do modelu $y_i = 3 + x_i + 4x_i^2 + \varepsilon_i$

```
make_poly_data = function(sample_size = 11) {
x = seq(0, 10)
y = 3 + x + 4 * x ^ 2 + rnorm(n = sample_size, mean = 0, sd = 20)
data.frame(x, y)
}
set.seed(1234)
poly_data = make_poly_data()


fit_quad = lm(y ~ poly(x, degree = 2), data = poly_data) # dopasowujemy wielomian st. 2
fit_big  = lm(y ~ poly(x, degree = 8), data = poly_data) # dopasowujemy wielomian st. 8
plot(y ~ x, data = poly_data, ylim = c(-100, 400), cex = 2, pch = 20)
xplot = seq(0, 10, by = 0.1)
lines(xplot, predict(fit_quad, newdata = data.frame(x = xplot)),col = "dodgerblue", lwd = 2, lty = 1)
lines(xplot, predict(fit_big, newdata = data.frame(x = xplot)),col = "darkorange", lwd = 2, lty = 2)
```

Wyznaczamy RMSE dla obu modeli $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_i^2} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} \hat{\varepsilon}_i^2}$

```
sqrt(mean(resid(fit_quad) ^ 2))
sqrt(mean(resid(fit_big) ^ 2))
```

Widać, że model big ma mniejszy RMSE
Definiujemy nowe kryterium

leave-one-out cross validated $RMSE_{\text{LOOCV}} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} e_{[i]}^2} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{e_i}{1-h_i}\right)^2}$,

gdzie $e_{[i]}$ oznacza błąd predykcji obserwacji y_i w modelu z usuniętym i-tym wierszem macierzy planu eksperymentu

```
#funkcja wyznaczająca RMSE_LOOCV
calc_loocv_rmse = function(model) {
  sqrt(mean((resid(model) / (1 - hatvalues(model))) ^ 2))
}
#wyznaczamy RMSE_LOOCV dla obu modeli
calc_loocv_rmse(fit_quad)
calc_loocv_rmse(fit_big)
```

Widać, że model big ma teraz znacznie większy $RMSE_{LOOCV}$ niż model quad

Pokażemy graficznie jak zachowują się oba modele przy usunięciu trzeciej obserwacji

```
fit_quad_removed = lm(y ~ poly(x, degree = 2), data = poly_data[-3, ])
fit_big_removed  = lm(y ~ poly(x, degree = 8), data = poly_data[-3, ])
plot(y ~ x, data = poly_data, ylim = c(-100, 400), cex = 2, pch = 20)
xplot = seq(0, 10, by = 0.1)
lines(xplot, predict(fit_quad_removed, newdata = data.frame(x = xplot)),
      col = "dodgerblue", lwd = 2, lty = 1)
lines(xplot, predict(fit_big_removed, newdata = data.frame(x = xplot)),
      col = "darkorange", lwd = 2, lty = 2)
```

Regresja krokowa

```
library(faraway) dp
help(seatpos) #opis zbioru danych
```

HtShoes –wzrost w butach w cm
Ht– wzrost bez butów w cm
Seated – wysokość na siedząco w cm
Arm – długość przedramienia w cm
Thigh –długość uda w cm
Leg – długość podudzia w cm
hipcenter –pozioma odległość środka bioder od ustalonego miejsca w samochodzie w mm

```
hipcenter_mod = lm(hipcenter ~ ., data = seatpos) #model addytywny ze wszystkimi zmiennymi
coef(hipcenter_mod)
extractAIC(hipcenter_mod) # zwraca  p i AIC=nlog(RSS/n)+2p

#regresja krokowa backward z kryterium AIC
hipcenter_mod_back_aic = step(hipcenter_mod, direction = "backward")

#regresja krokowa backward z kryterium BIC
n = length(resid(hipcenter_mod))
hipcenter_mod_back_bic = step(hipcenter_mod, direction = "backward", k = log(n))


#regresja krokowa forward z kryterium AIC
hipcenter_mod_start = lm(hipcenter ~ 1, data = seatpos)
```

```
hipcenter_mod_forw_aic = step(
  hipcenter_mod_start,
  scope = hipcenter ~ Age + Weight + HtShoes + Ht + Seated + Arm + Thigh + Leg,
  direction = "forward")

#regresja krokowa forward z kryterium BIC
hipcenter_mod_forw_bic = step(
  hipcenter_mod_start,
  scope = hipcenter ~ Age + Weight + HtShoes + Ht + Seated + Arm + Thigh + Leg,
  direction = "forward", k = log(n))

#regresja krokowa both z kryterium BIC
hipcenter_mod_both_bic = step(
  hipcenter_mod_start,
  scope = hipcenter ~ Age + Weight + HtShoes + Ht + Seated + Arm + Thigh + Leg,
  direction = "both", k = log(n))

#regresja krokowa backward dla danych aptecznych
model_b=lm(Volume~.,data=dane3)
extractAIC(model_b)
model_b_aic = step(model_b, direction = "backward")


#regresja krokowa forward dla danych aptecznych
model_start=lm(Volume ~ 1,data=dane3)
model_fwd_aic=step(
  model_start,
  scope=Volume ~ Floor_space + Presc_RX + Parking + ShopCntr + Income,
  direction = "forward")
```

Funkcja regsubsets() z pakietu leaps