Упражнение 1

Дроздецкая Анна

08 03 2020

## Постновка задачи

**Задача 1.** На данных своего варианта повторить три графика из первой практики, выбрав число степеней свободы как компромисс между точностью (оценкой ошибки на тестовой выборке) и простотой модели (числом степеней свободы). Все рисунки сохранить в графические файлы в формате png.

**Задача 2.** Решить задачу 1, изменив характеристики данных (см. свой вариант). Почему при таком изменении данных MSE меняется именно так? Все рисунки сохранить в графические файлы в формате png.

Вариант 6. Функция для задачи 1:

Характеристики для задачи 2:

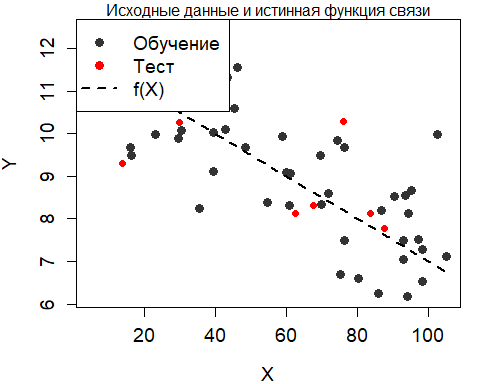
## Задача 1. Построение графиков

Сгенерируем и :  
\*   
\*

# Генерируем данные  
  
my.seed <- 1486372882 # ядро  
n.all <- 60 # наблюдений всего  
train.percent <- 0.85 # доля обучающей выборки  
res.sd <- 1 # стандартное отклонение случайного шума  
x.min <- 5 # границы изменения X: нижняя  
x.max <- 105 # и верхняя  
  
# Фактические значения X  
set.seed(my.seed)  
x <- runif(x.min, x.max, n = n.all)  
  
# Случайный шум  
set.seed(my.seed)  
res <- rnorm(mean = 0, sd = res.sd, n = n.all)  
  
# Отбираем наблюдения в обучающую выборку  
set.seed(my.seed)  
inTrain <- sample(seq\_along(x), size = train.percent\*n.all)  
  
# Истиная функция взаимосвязи  
y.func <- function(x) {12 - 0.05 \* x}  
  
# Для графика истинной взаимосвязи  
x.line <- seq(x.min, x.max, length = n.all)  
y.line <- y.func(x.line)  
  
# Фактические значения y (с шумом)  
y <- y.func(x) + res  
  
# Создаем векторы с данными для построения графиков  
  
# Наблюдения для обучающей выборки  
x.train <- x[inTrain]  
y.train <- y[inTrain]  
  
# Наблюдения для тестовой выборки  
x.test <- x[-inTrain]  
y.test <- y[-inTrain]

**Изобразим исходные данные на графике**

# График 1: Исходные данные на график  
  
# Убираем широкие поля рисунков  
par(mar = c(4, 4, 1, 1))  
  
# Наименьшие/наибольшие значения по осям  
x.lim <- c(x.min, x.max)  
y.lim <- c(min(y), max(y))  
  
# Наблюдения с шумом (обучающая выборка)  
plot(x.train, y.train,  
 col = grey(0.2), bg = grey(0.2), pch = 21,  
 xlab = 'X', ylab = 'Y',  
 xlim = x.lim, ylim = y.lim,  
 cex = 1.2, cex.lab = 1.2, cex.axis = 1.2)  
# Заголовок  
mtext('Исходные данные и истинная функция связи', side = 3)  
# Наблюдения тестовой выборки  
points(x.test, y.test, col = 'red', bg = 'red', pch = 21)  
# Истинная функция  
lines(x.line, y.line, lwd = 2, lty = 2)  
# Легенда  
legend('topleft', legend = c('Обучение', 'Тест', 'f(X)'),  
 pch = c(16, 16, NA),  
 col = c(grey(0.2), 'red', 'black'),  
 lty = c(0, 0, 2), lwd = c(1, 1, 2), cex = 1.2)



В качестве модели используем сплайны со степенями свободы от 2 (прямая) до 40 (количество узлов равно 2/3 наблюдений). Строим модели с различным количеством степеней свободы и в каждом случае считаем среднеквадратическую ошибку модели на обучающей и тестовой выборках.

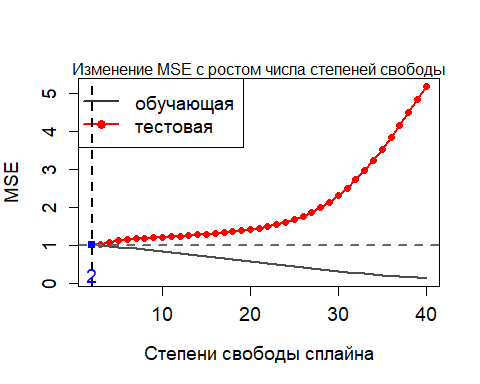
**Модель с df от 2 до 40**

# Теперь строим модели с df от 2 до 40   
  
# Максимальное число степеней свободы для модели сплайна  
max.df <- 40  
  
tbl <- data.frame(df = 2:max.df) # таблица для записи ошибок  
tbl$MSE.train <- 0 # столбец: ошибки на обучающей выборке  
tbl$MSE.test <- 0 # столбец: ошибки на тестовой выборке  
  
# цикл по степеням свободы  
for (i in 2:max.df) {  
 # строим модель  
 mod <- smooth.spline(x = x.train, y = y.train, df = i)  
   
 # модельные значения для расчёта ошибок  
 y.model.train <- predict(mod, data.frame(x = x.train))$y[, 1]  
 y.model.test <- predict(mod, data.frame(x = x.test))$y[, 1]  
   
 # считаем средний квадрат ошибки на обучающей и тестовой выборке  
 MSE <- c(sum((y.train - y.model.train)^2) / length(x.train),  
 sum((y.test - y.model.test)^2) / length(x.test))  
   
 # записываем ошибки в таблицу  
 tbl[tbl$df == i, c('MSE.train', 'MSE.test')] <- MSE  
}  
  
# первые строки таблицы  
head(tbl, 10)

## df MSE.train MSE.test  
## 1 2 1.0183856 1.028921  
## 2 3 1.0036170 1.044839  
## 3 4 0.9785041 1.077521  
## 4 5 0.9562848 1.129732  
## 5 6 0.9387768 1.171181  
## 6 7 0.9211364 1.192722  
## 7 8 0.9004057 1.202439  
## 8 9 0.8766808 1.209150  
## 9 10 0.8512788 1.218002  
## 10 11 0.8254146 1.230699

**Изобразим на графике поведение ошибок при различном количестве степеней свободы**

# График 2: Зависимость MSE от гибкости модели  
  
plot(x = tbl$df, y = tbl$MSE.test,   
 type = 'l', col = 'red', lwd = 2,  
 xlab = 'Степени свободы сплайна', ylab = 'MSE',  
 ylim = c(min(tbl$MSE.train, tbl$MSE.test),   
 max(tbl$MSE.train, tbl$MSE.test)),  
 cex = 1.2, cex.lab = 1.2, cex.axis = 1.2)  
  
# заголовок  
mtext('Изменение MSE с ростом числа степеней свободы', side = 3)  
  
points(x = tbl$df, y = tbl$MSE.test,  
 pch = 21, col = 'red', bg = 'red')  
lines(x = tbl$df, y = tbl$MSE.train, col = grey(0.3), lwd = 2)  
# неустранимая ошибка  
abline(h = res.sd, lty = 2, col = grey(0.4), lwd = 2)  
  
# легенда  
legend('topleft', legend = c('обучающая', 'тестовая'),  
 pch = c(NA, 16),   
 col = c(grey(0.2), 'red'),   
 lty = c(1, 1), lwd = c(2, 2), cex = 1.2)  
  
# степени свободы у наименьшей ошибки на тестовой выборке  
min.MSE.test <- min(tbl$MSE.test)  
df.min.MSE.test <- tbl[tbl$MSE.test == min.MSE.test, 'df']  
  
# компромисс между точностью и простотой модели по графику  
df.my.MSE.test <- 2  
my.MSE.test <- tbl[tbl$df == df.my.MSE.test, 'MSE.test']  
  
# ставим точку на графике  
abline(v = df.my.MSE.test,   
 lty = 2, lwd = 2)  
points(x = df.my.MSE.test, y = my.MSE.test,   
 pch = 15, col = 'blue')  
mtext(df.my.MSE.test,   
 side = 1, line = -1, at = df.my.MSE.test, col = 'blue', cex = 1.2)



На этом графике:

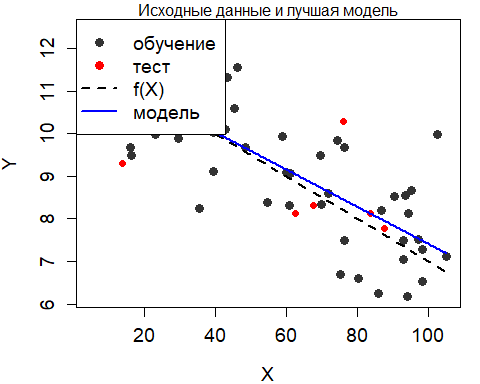
При движении слева направо MSE на обучающей выборке (серая кривая) сокращается, потому что с ростом числа степеней свободы растет число узлов, по которым строится сплайн. При этом модельная кривая подгоняется по всему возрастающему кол-ву точек и становится всё более гибкой. В результате индивидуальные расстояния от фактических наблюдений за Y до их отдельных оценок сокращается, что приводит к сокращению MSE.

При движении слева на право MSE на тестовой выборке (красная кривая) растет. Нам известна истиная форма связи Y с X, она описывается линейной функцией. MSE на тестовой выбоке довольно долго остается стабильной, а затем начинает расти. Этот рост объясняется эффектом переобучения модели: она все лучше описывает обучающую выборку, и при этом постепенно становится неприменимой ни к одному другому набору наблюдений.

Наименьшее значение MSE на тестовой выборке соответствует числу степеней свободы 2 и равно 1.028921. Визуально по графику мы можем установить, что первое значение , близкое к стабильно низким, соответствует df = 2. Именно df = 2 было выбрано в качестве компромисса между точностью (минимальной MSE на тестовой выборке) и простотой модели (чем меньше степеней свободы, тем модель проще).

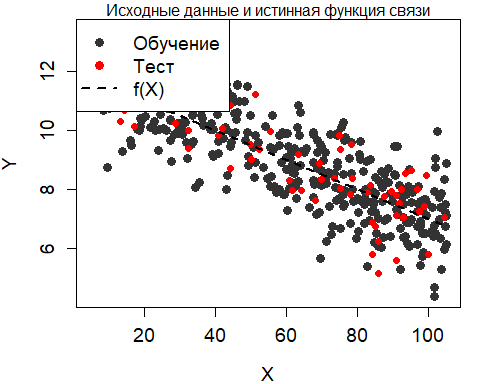
**График с моделью, выбранной в качестве лучшей, показан на рисунке ниже**

# График 3: Лучшая модель (компромисс между гибкостью и точностью)  
  
mod.MSE.test <- smooth.spline(x = x.train, y = y.train, df = df.my.MSE.test)  
  
# для гладких графиков модели  
x.model.plot <- seq(x.min, x.max, length = 250)  
y.model.plot <- predict(mod.MSE.test, data.frame(x = x.model.plot))$y[, 1]  
  
# убираем широкие поля рисунка  
par(mar = c(4, 4, 1, 1))  
  
# наименьшие/наибольшие значения по осям  
x.lim <- c(x.min, x.max)  
y.lim <- c(min(y), max(y))  
  
# наблюдения с шумом (обучающая выборка)  
plot(x.train, y.train,   
 col = grey(0.2), bg = grey(0.2), pch = 21,  
 xlab = 'X', ylab = 'Y',   
 xlim = x.lim, ylim = y.lim,   
 cex = 1.2, cex.lab = 1.2, cex.axis = 1.2)  
  
# заголовок  
mtext('Исходные данные и лучшая модель', side = 3)  
  
# наблюдения тестовой выборки  
points(x.test, y.test,   
 col = 'red', bg = 'red', pch = 21)  
  
# истинная функция  
lines(x.line, y.line,   
 lwd = 2, lty = 2)  
  
# модель  
lines(x.model.plot, y.model.plot,   
 lwd = 2, col = 'blue')  
  
# легенда  
legend('topleft', legend = c('обучение', 'тест', 'f(X)', 'модель'),  
 pch = c(16, 16, NA, NA),   
 col = c(grey(0.2), 'red', 'black', 'blue'),   
 lty = c(0, 0, 2, 1), lwd = c(1, 1, 2, 2), cex = 1.2)



## Задача 2

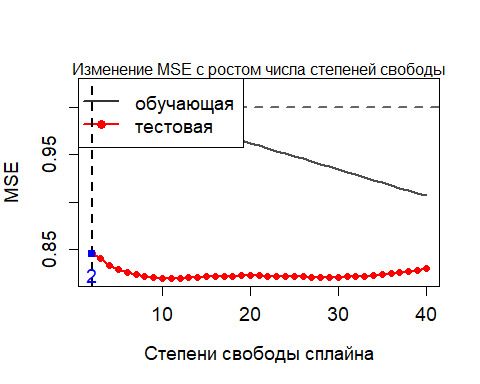
**Изобразим исходные данные на графике**



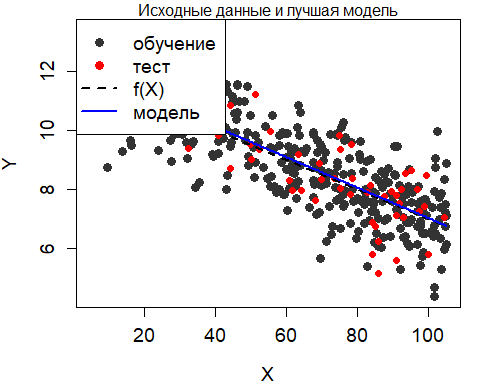
**Модель с df от 2 до 40**

## df MSE.train MSE.test  
## 1 2 1.0220729 0.8458160  
## 2 3 1.0198036 0.8408279  
## 3 4 1.0167390 0.8339040  
## 4 5 1.0142818 0.8293501  
## 5 6 1.0113693 0.8262744  
## 6 7 1.0077859 0.8236847  
## 7 8 1.0038797 0.8216438  
## 8 9 0.9999296 0.8203999  
## 9 10 0.9959589 0.8199228  
## 10 11 0.9920084 0.8199810

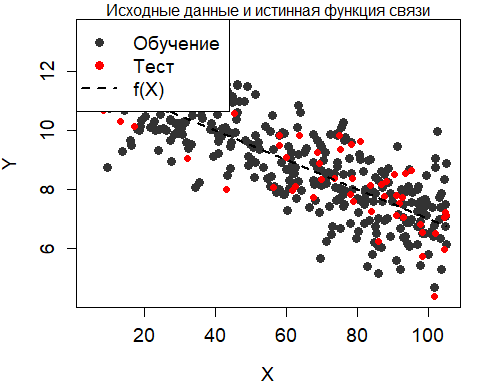
**Изобразим на графике поведение ошибок при различном количестве степеней свободы**



**График с моделью, выбранной в качестве лучшей, показан на рисунке ниже**



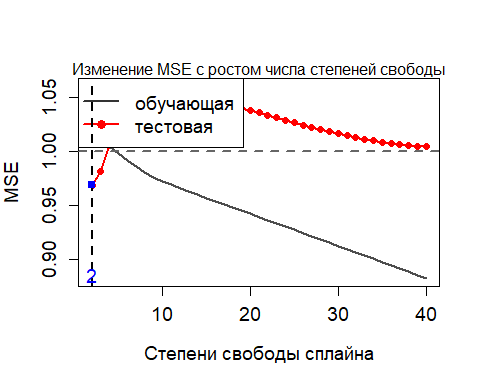
**Изобразим исходные данные на графике**



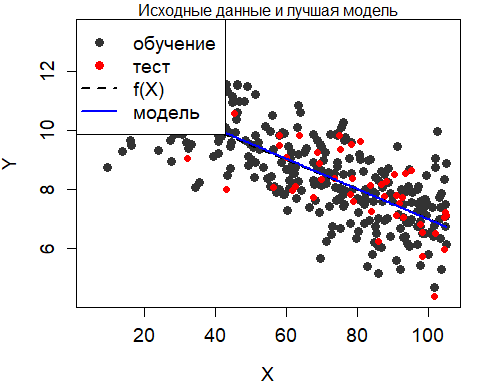
**Модель с df от 2 до 40**

## df MSE.train MSE.test  
## 1 2 1.0199849 0.9691154  
## 2 3 1.0133922 0.9817637  
## 3 4 1.0048340 1.0061059  
## 4 5 0.9979644 1.0299046  
## 5 6 0.9918393 1.0457397  
## 6 7 0.9860321 1.0545619  
## 7 8 0.9809111 1.0588007  
## 8 9 0.9765866 1.0602036  
## 9 10 0.9728387 1.0598592  
## 10 11 0.9694410 1.0584651

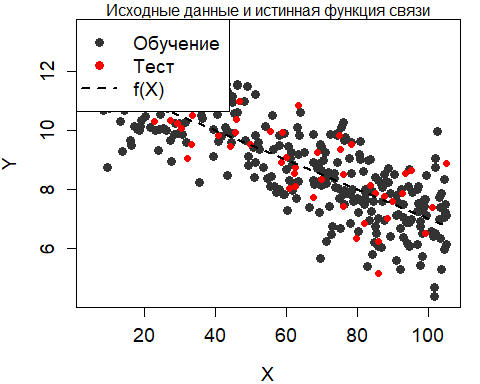
**Изобразим на графике поведение ошибок при различном количестве степеней свободы**



**График с моделью, выбранной в качестве лучшей, показан на рисунке ниже**



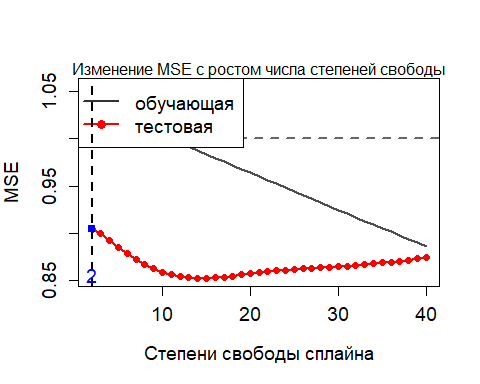
**Изобразим исходные данные на графике**



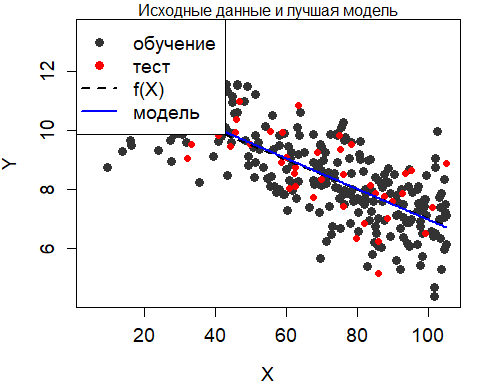
**Модель с df от 2 до 40**

## df MSE.train MSE.test  
## 1 2 1.0550351 0.9044280  
## 2 3 1.0499777 0.9001127  
## 3 4 1.0427360 0.8921804  
## 4 5 1.0367544 0.8854160  
## 5 6 1.0300725 0.8787361  
## 6 7 1.0229562 0.8723881  
## 7 8 1.0163979 0.8670044  
## 8 9 1.0106866 0.8626803  
## 9 10 1.0055786 0.8592251  
## 10 11 1.0008537 0.8565409  
## 11 12 0.9963391 0.8545714  
## 12 13 0.9919674 0.8533027  
## 13 14 0.9877228 0.8527061  
## 14 15 0.9835902 0.8527100  
## 15 16 0.9795641 0.8532037

**Изобразим на графике поведение ошибок при различном количестве степеней свободы**



**График с моделью, выбранной в качестве лучшей, показан на рисунке ниже**



По прежнему оптимальной можно считать модель с 2-мя степенями свободы.

|  |  |  |
| --- | --- | --- |
| n |  |  |
| 450 | 1.0220729 | 0.8458160 |
| 400 | 1.0199849 | 0.9691154 |
| 350 | 1.0550351 | 0.9044280 |

При увеличении объема выборки среднеквадратическая ошибка на обучаемом множестве уменьшается, а на тестовом увеличивается. Это может говорить, что при увеличении объема выборки существует эффект переобучения модели.