Задание 1

Салимова Э.И.

23 февраля 2018 г

### Оценка точности модели с непрерывной зависимой переменной

В практических примерах ниже показано:

1. Как делить данные на выборки (обучающую и тестовую);
2. Как считать MSE: среднеквадратическую ошибку модели;
3. Как меняются MSE на тестовой и обучающей выборках с изменением гибкости (числа степеней свободы) модели.  
   *Модели:* сглаживающие сплайны.  
   *Данные:* сгенерированные.

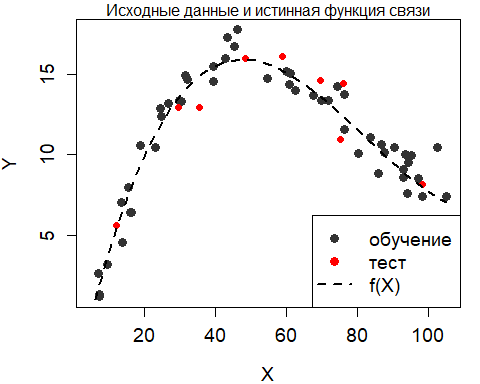
Рассмотрим как меняется поведение ошибок на тестовой и обучающей выборках при различном числе степеней свободы, если функция зависимости отклика **Y** от единственного признака **X** известна. Сгенерируем **X** и **Y**:

**X∼U(5,105)X∼U(5,105)**  
**Y=f(X)+ϵY=f(X)+ϵ**  
где f(x)=15 + 0.02*x - 0.005*  (x - 45)^2 + 0.00006 \* (x-54)^3, ϵ∼N(0,1)ϵ∼N(0,1)

# 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)  
  
# истинная функция взаимосвязи  
y.func <- function(x) {15 + 0.02\*x - 0.005 \* (x - 45)^2 + 0.00006 \* (x-54)^3}  
  
# 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  
  
# Создаём векторы с данными для построения графиков ############################  
  
# наблюдения на обучающей выборке  
x.train <- x[inTrain]  
y.train <- y[inTrain]  
  
# наблюдения на тестовой выборке  
x.test <- x[-inTrain]  
y.test <- y[-inTrain]

Изобразим исходные данные на графике.

# График 1: Исходные данные на график #########################################  
  
# 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)  
  
# 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)  
# заголовок  
mtext('Исходные данные и истинная функция связи', side = 3)  
  
# легенда  
legend('bottomright', 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

# Строим модель №2 из лекции (df = 6) #########################################  
  
# модель 2 (сплайн с df = 6)  
mod <- smooth.spline(x = x.train, y = y.train, df = 6)  
  
# 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 обечающей 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))  
names(MSE) <- c('train', 'test')  
round(MSE, 2)

## train test   
## 0.87 1.65

# 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 обечающей 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 модель  
 tbl[tbl$df == i, c('MSE.train', 'MSE.test')] <- MSE  
}  
  
# первые строки таблицы  
head(tbl)

## df MSE.train MSE.test  
## 1 2 15.1768902 12.671133  
## 2 3 3.5054898 2.041628  
## 3 4 1.3398327 1.150805  
## 4 5 0.9530536 1.452940  
## 5 6 0.8653527 1.645438  
## 6 7 0.8298545 1.718778

Изобразим на графике поведение ошибок при различном количестве степеней свободы.

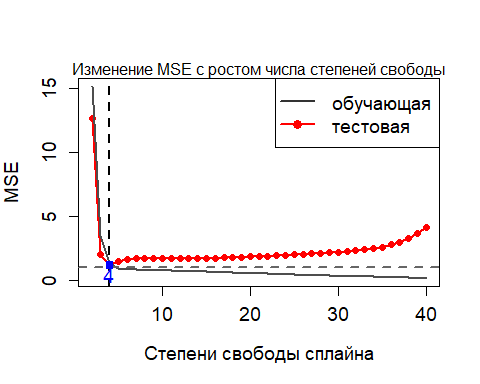
# График 2: Зависимость MSE от гибкости модели ################################  
  
plot(tbl$df, tbl$MSE.test,  
 type = 'l', col = 'red', lwd = 2,  
 xlab = 'Степени свободы сплайна', ylab = 'MSE',  
 ylim = c(min(tbl$MSE.train, tbl$MSE.test),   
 max(tbl$MSE.train, tbl$MSE.test)),  
 cex = 1.2, cex.lab = 1.2, cex.axis = 1.2)  
  
points(tbl$df, tbl$MSE.test,  
 pch = 21, col = 'red', bg = 'red')  
  
lines(tbl$df, tbl$MSE.train,   
 col = grey(0.3), lwd = 2)  
# неустранимая ошибка  
abline(h = res.sd,  
 lty = 2, col = grey(0.4), lwd = 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']  
  
# сообщение в консоль  
message(paste0('Наименьшая MSE на тестовой выборке равна ',   
 round(min.MSE.test, 2),   
 ' и достигается при df = ', df.min.MSE.test, '.'))

## Наименьшая MSE на тестовой выборке равна 1.15 и достигается при df = 4.

# компромисс между точностью и простотой модели по графику  
df.my.MSE.test <- 4  
my.MSE.test <-   
 tbl[tbl$df == df.my.MSE.test,  
 'MSE.test']  
  
# легенда  
legend('topright', 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)  
  
# сообщение в консоль  
message(paste0('Компромисс между точностью и сложностью модели при df = ',   
 df.my.MSE.test, ', MSE = ', round(my.MSE.test, 2), '.'))

## Компромисс между точностью и сложностью модели при df = 4, MSE = 1.15.

# ставим точку на графике  
abline(v = 4,  
 lty = 2, lwd = 2)  
points(4, 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)  
  
mtext('Изменение MSE с ростом числа степеней свободы', side = 3)

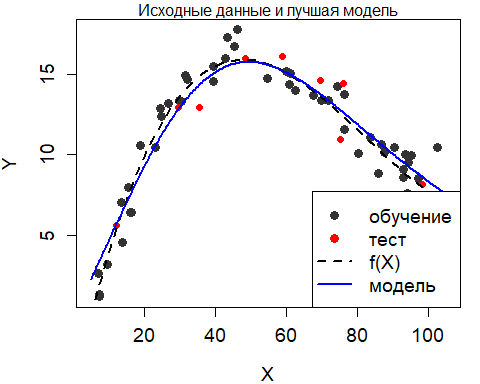


На этом графике:

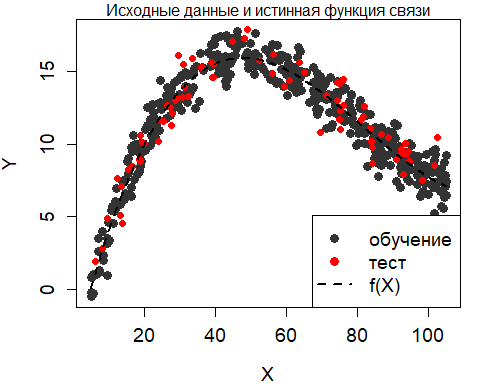
* При движении слева направо MSE на обучающей выборке (серая кривая) сокращается, потому что с ростом числа степеней свободы расчёт число узлов, по которым строится сплайн. При этом модельная кривая подгоняется по всё возрастающему количеству точек и становится всё более гибкой. В результате индивидуальные расстояния от фактических наблюдений за YY до их модельных оценок сокращаются, что приводит к сокращению MSE.
* При движении слева направо MSE на тестовой выборке (красная кривая) плавно растёт. Нам известна истинная форма связи Y с X, она описывается прямой. Число степеней свободы такой модели равно числу оцениваемых параметров, т.е. 4 (коэффициенты перед X, и константа). Рост MSE на тестовой выборке объясняется эффектом переобучения модели: она всё 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 4 и равно 1.15. Что и является компромиссом между простотой и точностью модели.

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.train, 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)  
  
# 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')  
  
# заголовок  
mtext('Исходные данные и лучшая модель', side = 3)  
  
# легенда  
legend('bottomright', 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)



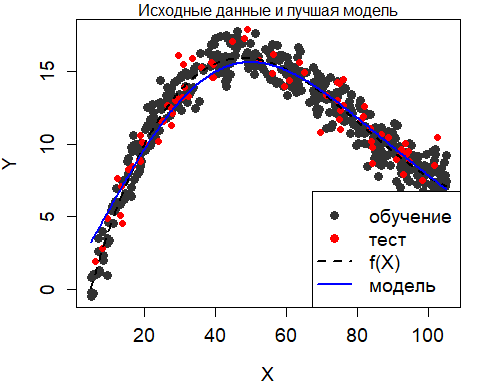
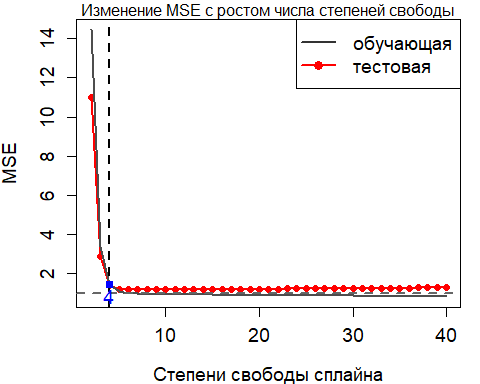
Теперь изменим исходные данные, пусть количество наблюдений станет равным 500. Ниже можно увидеть графики, описывающие изменение моделей.



## df MSE.train MSE.test  
## 1 2 14.4574876 11.017805  
## 2 3 3.3974115 2.884912  
## 3 4 1.4418342 1.407084  
## 4 5 1.0801152 1.180858  
## 5 6 0.9903026 1.162918  
## 6 7 0.9574958 1.171244

## Наименьшая MSE на тестовой выборке равна 1.16 и достигается при df = 6.

## Компромисс между точностью и сложностью модели при df = 4, MSE = 1.41.



Сравнив полученные модели, можно увидеть, что при увеличении числа наблюдений лучшая модель практически не поменяла своего положения относительно истинной функции, значение ошибки MSE на тестовой выборке увеличивается, в данном случае она изменилась с 1.15 до 1.4.