Упражнение 1

Кочетков Николай ПМИ 3-1

2/25/2020

Вариант 11 Модели: сглаживающие сплайны. Данные: сгенерированные.

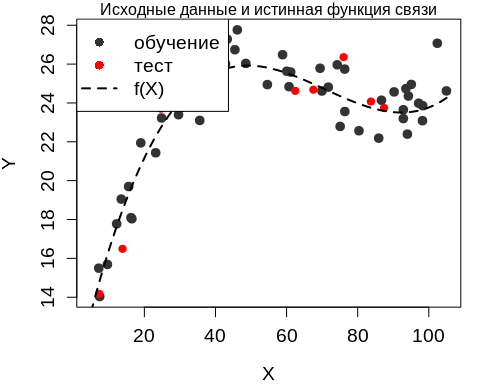
Рассмотрим пример из лекции: как меняется поведение ошибок на тестовой и обучающей выборках при различном числе степеней свободы, если функция зависимости отклика Y от единственного признака X известна. Сгенерируем X и Y: X∼U(5,105) Y=f(X)+ϵ, где f(X)=25 + 0.02 \* x - 0.003 \* (x - 45)^2 + 0.00006 \* (x - 54)^3 ϵ∼N(0,1)

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.

# 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) {25 + 0.02 \* x - 0.003 \* (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  
# 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)  
  
# 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.

# 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)

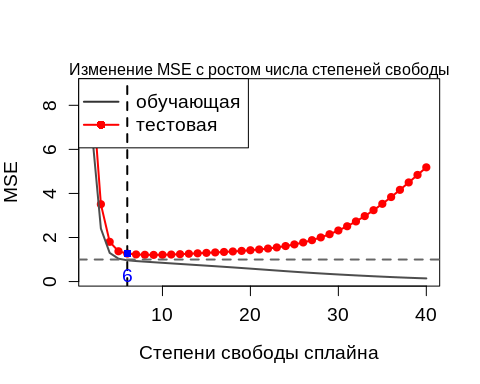
## train test   
## 0.97 1.27

# 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)

## df MSE.train MSE.test  
## 1 2 6.7184957 8.856639  
## 2 3 2.4055302 3.509615  
## 3 4 1.3007478 1.802507  
## 4 5 1.0370028 1.374416  
## 5 6 0.9653311 1.267632  
## 6 7 0.9301312 1.230138

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 <- 6  
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:

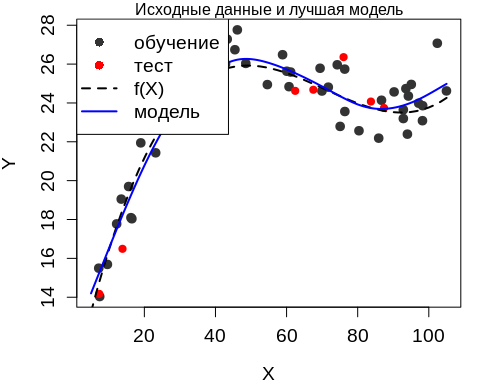
При движении слева направо MSE на обучающей выборке (серая кривая) сокращается, потому что с ростом числа степеней свободы расчёт число узлов, по которым строится сплайн. При этом модельная кривая подгоняется по всё возрастающему количеству точек и становится всё более гибкой. В результате индивидуальные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

При движении слева направо MSE на тестовой выборке (красная кривая) сначала резко сокращается, затем растёт. Нам известна истинная форма связи Y с X, она описывается кубической функцией. Число степеней свободы такой модели равно числу оцениваемых параметров, т.е. 4 (коэффициенты перед X, X2, X3 и константа). Поэтому резкое падение ошибки на тестовой выборке при небольшом числе степеней свободы связано с тем, что модель приближается по гибкости к истинной функции связи. Затем MSE на тестовой выборке довольно долго остаётся стабильной, а затем начинает расти. Этот рост 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 7 и равно 1.27. Визуально по графику мы можем установить, что первое значение MSEТЕСТ, близкое к стабильно низким, соответствует df = 6. Ошибка здесь равна 1, что ненамного отличается от минимума. Именно df = 6 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.

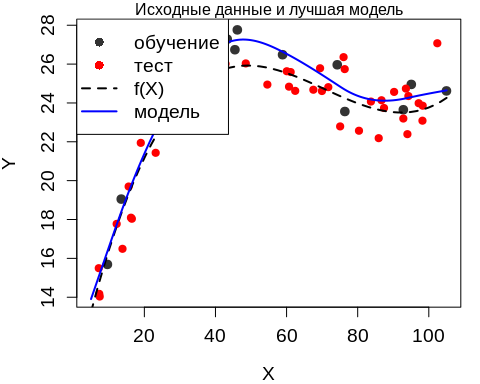
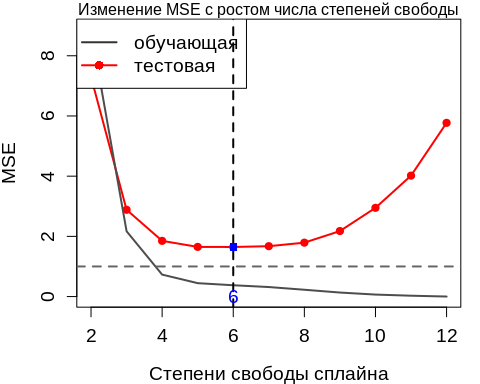
# График 3: Лучшая модель (компромисс между гибкостью и точностью) ############  
  
mod.MSE.test <- smooth.spline(x = x.train, y = y.train, df = df.my.MSE.test)  
  
# для гладких графиков модели  
x.model.plot <- seq(x.min, x.max, length = 250)  
y.model.plot <- predict(mod.MSE.test, data.frame(x = 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)



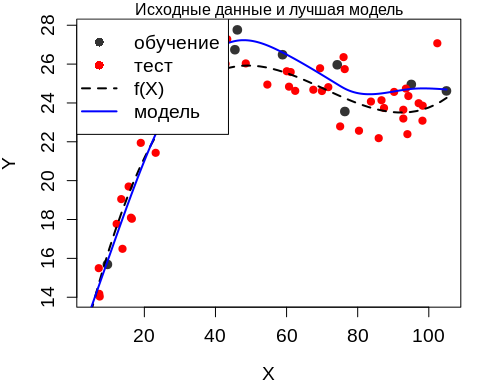
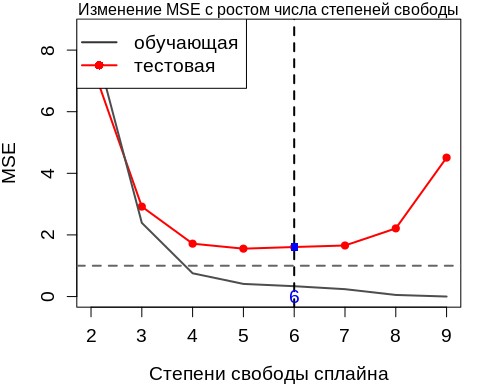
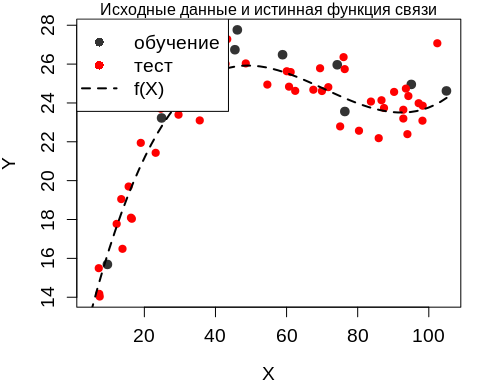
Задача 2 Задача 2. Решить задачу 1, изменив характеристики данных (11 вариант). Почему при таком изменении данных MSE меняется именно так? Все рисунки сохранить в графические файлы в формате png. train.percent=0.2, train.percent=0.15, train.percent=0.1 Требуется проанализировать результаты построения моделей при изменении доли обучаемой выборки (в данном случае при ее уменьшении) Создадим функцию, которая выполняет все действия, требуемые задачей 1, но сделаем так, чтобы функция принимала аргумент train.percent “доля обучаемой выборки”. Далее нет необходимости демонстрировать “чанки” с кодом, так как будут использоваться вышеприведенные действия. train.percent=0.2, train.percent=0.15, train.percent=0.1

Графики для train.percent=0.2, train.percent=0.15, train.percent=0.1 соотвественно

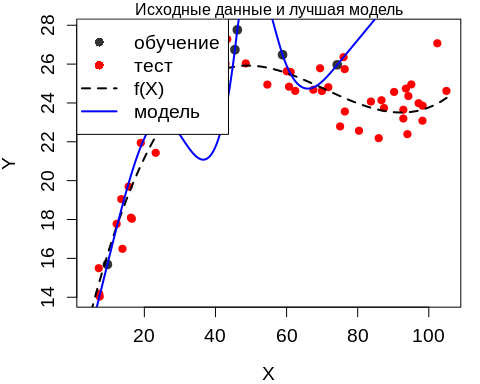
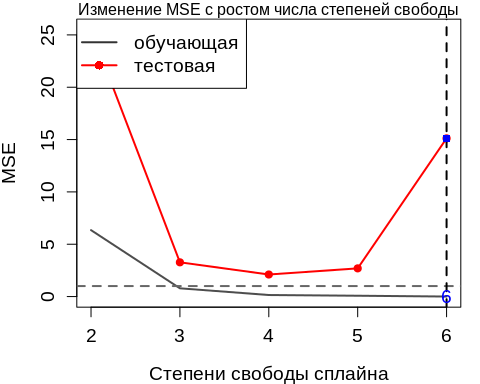
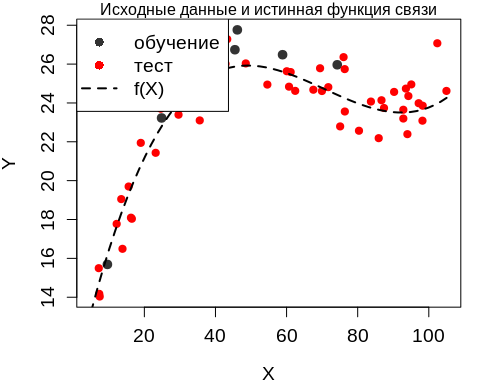
changeTrainPercent(0.2)



changeTrainPercent(0.15)



changeTrainPercent(0.1)

 С уменьшением размера обучающей выборки падает качество модели: повышается MSE и график модели все меньше становится похожим на исходный f(x).