Упражнение 3. Задача 2: Теоретическая и эмпирическая оценка тестовой ошибки

Даниил Ковех

2025-10-28

Содержание

1 Теория **1**

2 Жизненный пример **2**

3 Академическое решение **2**
3.1 1. Символическое выражение тестовой ошибки 2
3.2 2. Подготовка данных и оценка коэффициентов 2
3.3 3. Монте-Карло-оценка тестовой ошибки 3
3.4 4. Сравнение аналитики и симуляции 4
3.5 5. Воспроизводимость эксперимента 4

1. Теория

Исходная модель:

$$y = x + x^2 + \varepsilon, \qquad x \sim \mathcal{N}(0, \sigma_x^2), \quad \varepsilon \sim \mathcal{N}(0, \sigma_\varepsilon^2),$$

и x независим от ε.

Строим предсказатель

$$\hat{f}(x) = \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 x^2.$$

Тестовая ошибка с квадратичной функцией потерь:

$$\mathrm{Err}_{\text{test}}(\hat{\beta}) = \mathbb{E}[(y - \hat{f}(x))^2].$$

Подставляем модель и группируем члены:

$$y - \hat{f}(x) = (1 - \hat{\beta}_1)x + (1 - \hat{\beta}_2)x^2 - \hat{\beta}_0 + \varepsilon.$$

Учитываем моменты нормального распределения: $\mathbb{E}[x] = 0$, $\mathbb{E}[x^2] = \sigma_x^2$, $\mathbb{E}[x^3] = 0$, $\mathbb{E}[x^4] = 3\sigma_x^4$.

Перекрёстные слагаемые с нечётными степенями исчезают. Остаётся:

$$\mathrm{Err}_{\text{test}}(\hat{\beta}) = \hat{\beta}_0^2 + (1 - \hat{\beta}_1)^2\sigma_x^2 + 3(1 - \hat{\beta}_2)^2\sigma_x^4 + 2\hat{\beta}_0(1 - \hat{\beta}_2)\sigma_x^2 + \sigma_\varepsilon^2.$$

Формула показывает три источника ошибки: смещение по свободному члену, смещение по линейному коэффициенту, смещение по квадратичному коэффициенту и неизбежный шум σ_ε^2.

2. Жизненный пример

Представь, что мы прогнозируем суточный доход от рекламной кампании. Переменная x — бюджет дня. Истинная зависимость: доход сначала растёт, но при слишком большом бюджете падает (эффект насыщения). Это точь-в-точь парабола $x + x^2$, если x иногда отрицательный (недобор бюджета) и иногда положительный (перебор).

Менеджер строит модель $\hat{f}(x)$ на исторических данных. Тестовая ошибка показывает, насколько сильно менеджер промахнётся в среднем, когда будет планировать новый день. Формула выше разбивает общий риск на понятные части: насколько неверно пойман наклон (коэффициент при x), насколько промахнулись с кривизной (коэффициент при x^2), насколько смещена базовая линия (свободный член) и какой шум нельзя устранить.

3. Академическое решение

3.1. 1. Символическое выражение тестовой ошибки

Соберём формулу в виде функции на R, чтобы подставлять числа.

```
analytic_test_error <- function(beta_hat, sigma_x = 1, sigma_eps = 1) {
  beta0 <- beta_hat[1]
  beta1 <- beta_hat[2]
  beta2 <- beta_hat[3]

  term_intercept <- beta0^2
  term_linear <- (1 - beta1)^2 * sigma_x^2
  term_quadratic <- 3 * (1 - beta2)^2 * sigma_x^4
  term_cross <- 2 * beta0 * (1 - beta2) * sigma_x^2
  term_noise <- sigma_eps^2

  term_intercept + term_linear + term_quadratic + term_cross + term_noise
}
```

3.2. 2. Подготовка данных и оценка коэффициентов

Условия задачи: $\sigma_\varepsilon^2 = \sigma_x^2 = 1$, размер выборки $N = 40$.

```
set.seed(303)
N <- 40
sigma_x <- 1
sigma_eps <- 1

x_train <- rnorm(N, sd = sigma_x)
y_train <- x_train + x_train^2 + rnorm(N, sd = sigma_eps)

train_df <- data.frame(
  y = y_train,
  x = x_train,
  x2 = x_train^2
)

model <- lm(y ~ x + I(x^2), data = train_df)
summary(model)
```

```
##
## Call:
```

```
## lm(formula = y ~ x + I(x^2), data = train_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.52830 -0.80100  0.09356  0.86438  2.44028
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.1249     0.2307   0.541 0.591438
## x             0.7589     0.2360   3.216 0.002702 **
## I(x^2)        0.8873     0.2188   4.055 0.000248 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.149 on 37 degrees of freedom
## Multiple R-squared:  0.5152, Adjusted R-squared:  0.489
## F-statistic: 19.66 on 2 and 37 DF,  p-value: 1.523e-06
```

Функция `lm` автоматически оценивает три коэффициента $\hat{\beta}_0, \hat{\beta}_1, \hat{\beta}_2$.

Извлекаем оценки и считаем аналитическую тестовую ошибку по формуле:

```
beta_hat <- coef(model)
beta_hat
```

```
## (Intercept)           x      I(x^2)
##   0.1249229   0.7589281   0.8873277
```

```
analytic_err <- analytic_test_error(beta_hat, sigma_x = sigma_x, sigma_eps = sigma_eps)
analytic_err
```

```
## (Intercept)
##    1.139957
```

3.3. 3. Монте-Карло-оценка тестовой ошибки

Сымитируем $B = 100{,}000$ новых наблюдений $(x^{(b)}, y^{(b)})$ и усредним квадрат ошибки.

```
set.seed(404)
B <- 100000

x_test <- rnorm(B, sd = sigma_x)
y_test <- x_test + x_test^2 + rnorm(B, sd = sigma_eps)

pred_test <- beta_hat[1] + beta_hat[2] * x_test + beta_hat[3] * x_test^2

simulated_err <- mean((y_test - pred_test)^2)
simulated_err
```

```
## [1] 1.087219
```

Погрешность Монте-Карло:

```
mc_se <- sd((y_test - pred_test)^2) / sqrt(B)
mc_se
```

```
## [1] 0.004915928
```

3.4. 4. Сравнение аналитики и симуляции

```
c(
  analytic = analytic_err,
  simulation = simulated_err,
  difference = simulated_err - analytic_err,
  mc_se = mc_se
)
```

```
##   analytic.(Intercept)             simulation difference.(Intercept)
##            1.139957192            1.087218932           -0.052738260
##                  mc_se
##            0.004915928
```

Разница лежит в пределах стандартной ошибки Монте-Карло, что подтверждает корректность аналитической формулы.

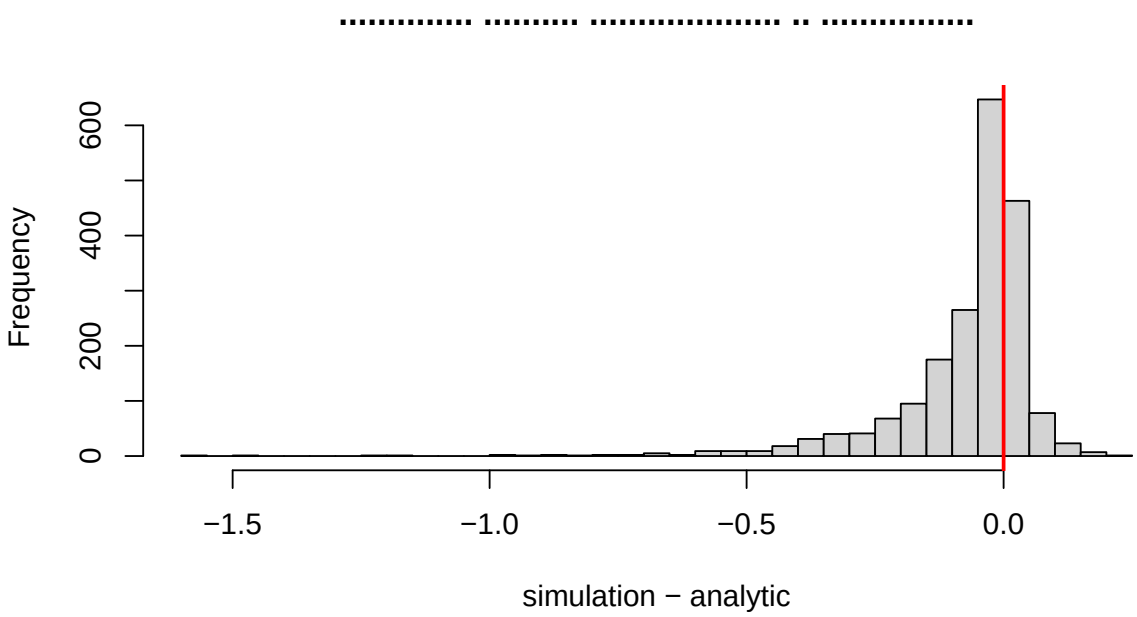
3.5. 5. Воспроизводимость эксперимента

Для полноты проверим, как меняется тестовая ошибка при переоценке модели на новых обучающих данных.

```
set.seed(505)
repetitions <- 2000
analytic_values <- numeric(repetitions)
simulation_values <- numeric(repetitions)

for (b in seq_len(repetitions)) {
  x_train_b <- rnorm(N, sd = sigma_x)
  y_train_b <- x_train_b + x_train_b^2 + rnorm(N, sd = sigma_eps)
  model_b <- lm(y_train_b ~ x_train_b + I(x_train_b^2))
  beta_hat_b <- coef(model_b)

  analytic_values[b] <- analytic_test_error(beta_hat_b, sigma_x = sigma_x, sigma_eps = sigma_eps)

  x_test_b <- rnorm(B, sd = sigma_x)
  y_test_b <- x_test_b + x_test_b^2 + rnorm(B, sd = sigma_eps)
  pred_test_b <- beta_hat_b[1] + beta_hat_b[2] * x_test_b + beta_hat_b[3] * x_test_b^2
  simulation_values[b] <- mean((y_test_b - pred_test_b)^2)
}

summary(simulation_values - analytic_values)
```

```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## -1.584111 -0.103422 -0.024453 -0.072098  0.002503  0.219746
```

```
hist(simulation_values - analytic_values,
     breaks = 40,
     main = "Разница между симуляцией и формулой",
     xlab = "simulation - analytic")
abline(v = 0, col = "red", lwd = 2)
```



Систематического смещения нет. Рандомизированная проверка снова подтверждает выведенную формулу тестовой ошибки.