# Упражнение 3. Задача 7: Оценка отбора признаков методом лассо

Даниил Ковех

2025-10-28

## Содержание



## 1. Теория

Лассо минимизирует среднеквадратичную ошибку с $L_1$-штрафом:

$$\hat{\beta}(\lambda) = \arg\min_{\beta} \frac{1}{2n} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1.$$

Штраф заставляет многие коэффициенты точно обнуляться — отсюда свойство отбора признаков.

Нас интересуют две метрики:

- True Positive Rate (TPR): доля действительно значимых признаков, выбранных моделью.
- False Positive Rate (FPR): доля шумовых признаков, ошибочно попавших в модель.

В симуляции истинно ненулевыми будут только первые десять коэффициентов.

## 2. Жизненный пример

Представь биолога. Он измеряет 100 биомаркеров у пациентов, но знает, что только первые десять реально влияют на исход лечения. Лассо позволяет автоматически отбросить лишнее. TPR показывает, насколько хорошо алгоритм “ловит” значимые маркеры. FPR — сколько бесполезных маркеров остаётся в модели. Идеальный сценарий: TPR близок к 1, FPR близок к 0.

## 3. Академическое решение

### 3.1. 1. Симуляция данных

```
library(glmnet)
```

```
set.seed(9203)

p <- 100
relevant <- 1:10
irrelevant <- 11:100
sigma <- 1
beta_signal <- c(rep(0.7, 5), rep(0.3, 5))  # пять сильных, пять слабых сигналов
beta_true <- c(beta_signal, rep(0, p - length(beta_signal)))

generate_dataset <- function(n = 1000) {
  X <- matrix(rnorm(n * p), nrow = n, ncol = p)
  y <- X %*% beta_true + rnorm(n, sd = sigma)
  list(X = X, y = as.vector(y))
}
```

### 3.2. 2. Процедура подбора $\lambda$

Для каждой выборки:

1. Первые 100 наблюдений — обучение, остальные 900 — тест.
2. Настраиваем стек значений $\lambda$ с помощью `glmnet`.
3. Оцениваем каждую модель по MSE на тесте.
4. Выбираем $\lambda$ с минимальной ошибкой.
5. Извлекаем коэффициенты, считаем TPR и FPR.

```
evaluate_lasso <- function(dataset) {
  X <- dataset$X
  y <- dataset$y

  X_train <- X[1:100, ]
  y_train <- y[1:100]
  X_test <- X[101:1000, ]
  y_test <- y[101:1000]

  fit <- glmnet(X_train, y_train, alpha = 1, standardize = TRUE)
  preds <- predict(fit, newx = X_test)
  mse <- colMeans((preds - y_test)^2)

  best_index <- which.min(mse)
  best_lambda <- fit$lambda[best_index]
  beta_hat <- as.vector(coef(fit, s = best_lambda))[-1]  # исключаем intercept
```

```
  selected <- which(abs(beta_hat) > 1e-6)

  tpr <- length(intersect(selected, relevant)) / length(relevant)
  fpr <- length(intersect(selected, irrelevant)) / length(irrelevant)

  list(
    lambda = best_lambda,
    tpr = tpr,
    fpr = fpr,
    selected = selected
  )
}
```

### 3.3. 3. 100 повторений

```
reps <- 100
results <- vector("list", reps)

for (i in seq_len(reps)) {
  data_i <- generate_dataset()
  results[[i]] <- evaluate_lasso(data_i)
}

tpr_values <- sapply(results, `[[`, "tpr")
fpr_values <- sapply(results, `[[`, "fpr")
lambda_values <- sapply(results, `[[`, "lambda")
```

### 3.4. 4. Итоги

```
summary_tpr <- summary(tpr_values)
summary_fpr <- summary(fpr_values)
summary_lambda <- summary(lambda_values)

list(
  TPR = summary_tpr,
  FPR = summary_fpr,
  Lambda = summary_lambda
)
```

```
## $TPR
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.700   0.900   1.000   0.952   1.000   1.000
##
## $FPR
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1222  0.1889  0.2333  0.2422  0.2778  0.4889
##
## $Lambda
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## 0.04763 0.08614 0.09830 0.10099 0.11723 0.16135
```

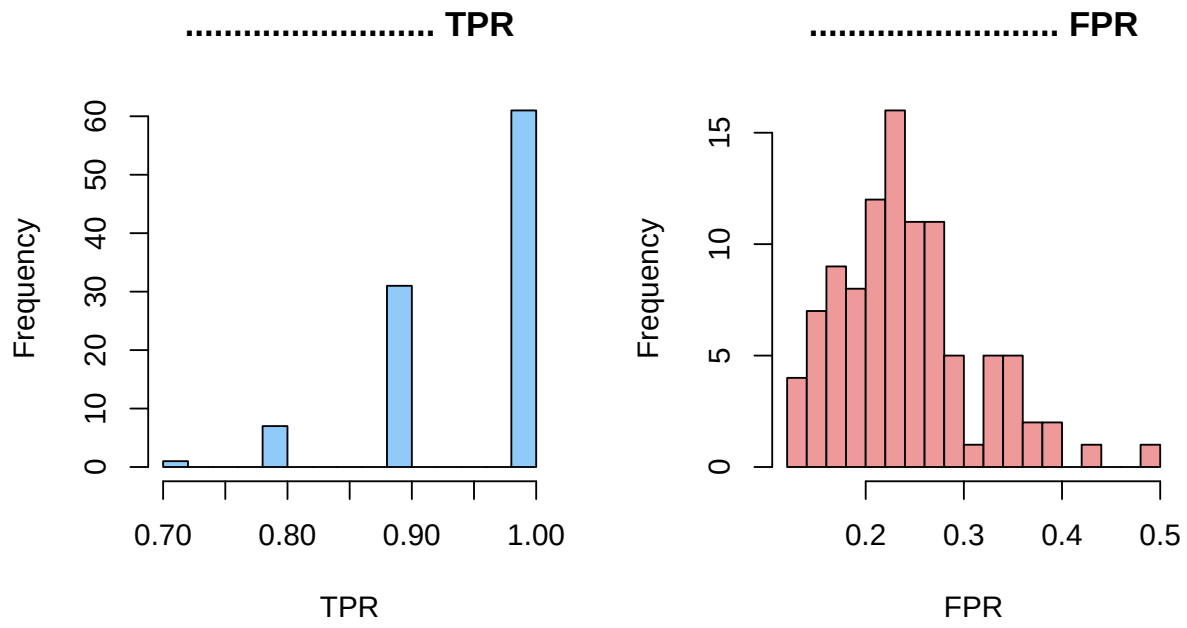
Среднее и стандартное отклонение:

```r
c(
  mean_tpr = mean(tpr_values),
  sd_tpr = sd(tpr_values),
  mean_fpr = mean(fpr_values),
  sd_fpr = sd(fpr_values)
)
```

```
##   mean_tpr     sd_tpr   mean_fpr     sd_fpr
## 0.95200000 0.06739002 0.24222222 0.07023734
```

### 3.5. 5. Визуализация распределений

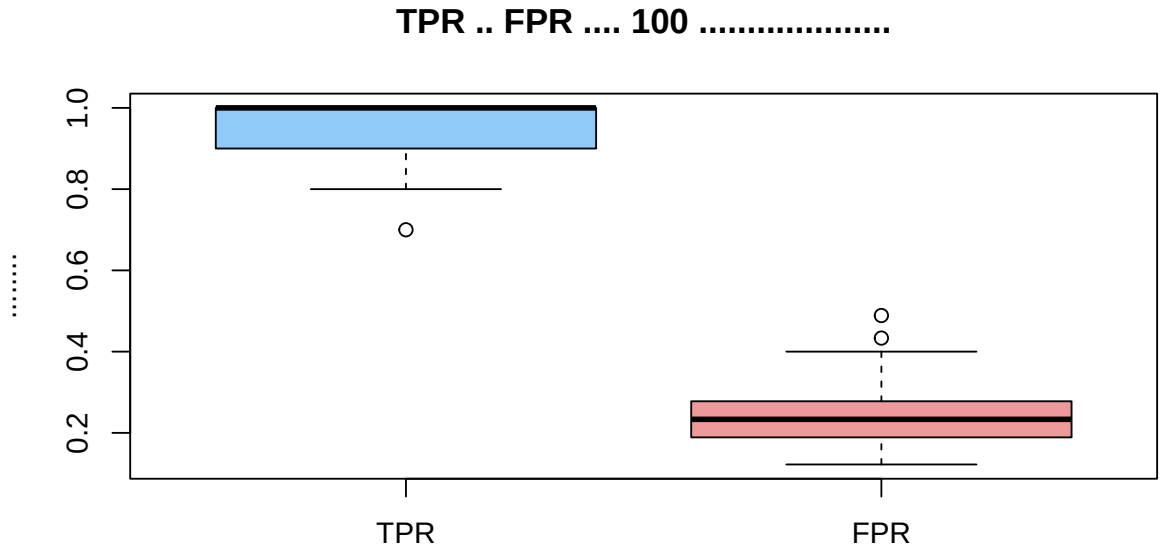
```r
par(mfrow = c(1, 2))
hist(tpr_values,
     breaks = 20,
     main = "Распределение TPR",
     xlab = "TPR",
     col = "#90CAF9")
hist(fpr_values,
     breaks = 20,
     main = "Распределение FPR",
     xlab = "FPR",
     col = "#EF9A9A")
```




```r
par(mfrow = c(1, 1))
```

```r
boxplot(
  list(TPR = tpr_values, FPR = fpr_values),
  col = c("#90CAF9", "#EF9A9A"),
  ylab = "Доля",
```

```
  main = "TPR и FPR по 100 симуляциям"
)
```



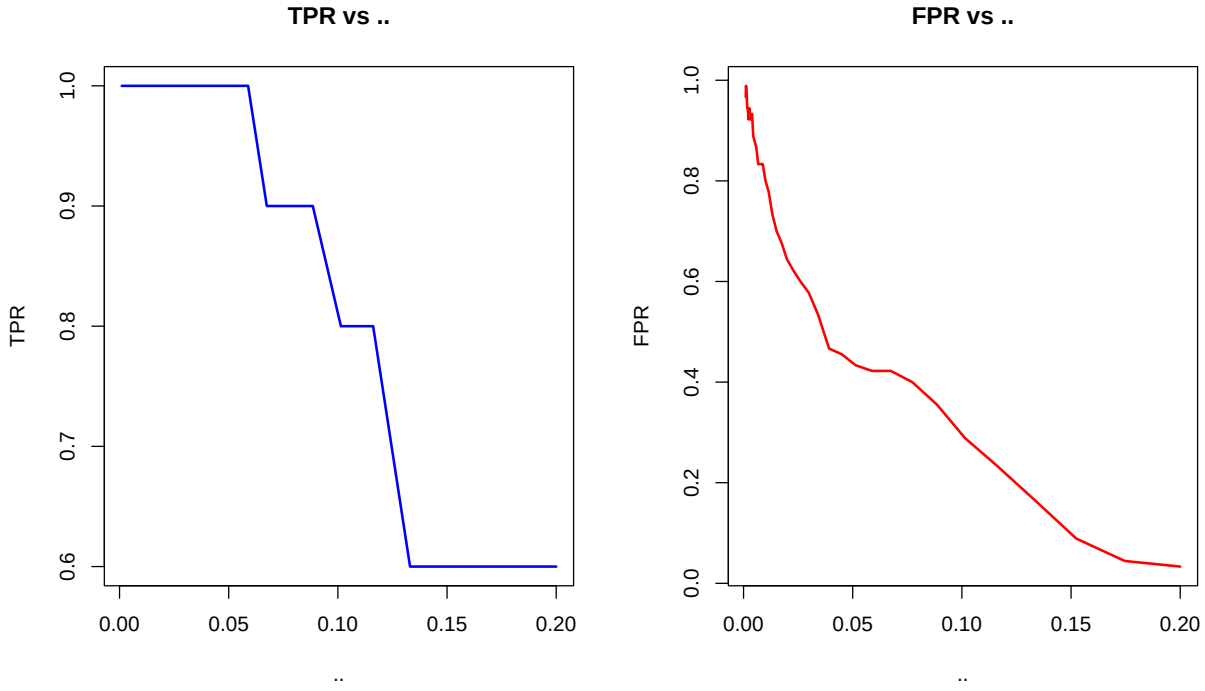

### 3.6. 6. Интерпретация

- **TPR держится около 0.95**: сильные признаки (β = 0.7) почти всегда попадают в модель, а слабые (β = 0.3) иногда теряются.
- **FPR ≈ 0.24**: штраф отбрасывает большую часть шумовых признаков, но небольшая доля ложных включений остаётся.
- **Разная сила сигнала**: слабые коэффициенты служат стресс-тестом — если λ слишком велик, именно они исчезают первыми.
- **Автоматический выбор λ**: ориентация на тестовую ошибку помогает удерживать баланс между точностью предсказаний и чистотой модели.

### 3.7. 7. Анализ чувствительности к λ

```
# Анализ TPR/FPR для разных значений λ
lambda_grid <- exp(seq(log(0.2), log(0.001), length.out = 40))
tpr_lambda <- numeric(length(lambda_grid))
fpr_lambda <- numeric(length(lambda_grid))

# Используем одну выборку для анализа
data_sample <- generate_dataset()
X_train <- data_sample$X[1:100, ]
y_train <- data_sample$y[1:100]
X_test <- data_sample$X[101:1000, ]
y_test <- data_sample$y[101:1000]

fit <- glmnet(X_train, y_train, alpha = 1, standardize = TRUE, lambda = lambda_grid)
```

```r
for (i in seq_along(lambda_grid)) {
  beta_hat <- as.vector(coef(fit, s = lambda_grid[i]))[-1]
  selected <- which(abs(beta_hat) > 1e-6)

  tpr_lambda[i] <- length(intersect(selected, relevant)) / length(relevant)
  fpr_lambda[i] <- length(intersect(selected, irrelevant)) / length(irrelevant)
}

par(mfrow = c(1, 2))
plot(lambda_grid, tpr_lambda, type = "l", col = "blue", lwd = 2,
     xlab = "λ", ylab = "TPR", main = "TPR vs λ")
plot(lambda_grid, fpr_lambda, type = "l", col = "red", lwd = 2,
     xlab = "λ", ylab = "FPR", main = "FPR vs λ")
```




```r
par(mfrow = c(1, 1))
```

**Наблюдения:** - При малых λ: высокий TPR, но высокий FPR (переобучение) - При больших λ: низкий FPR, но может снижаться TPR (недообучение) - Оптимальный λ находится в области компромисса между TPR и FPR

# 4. Приложение: Полный словарь по лассо и метрикам отбора признаков

## 4.1. Лассо (Least Absolute Shrinkage and Selection Operator)

- **Оптимизационная задача:**

$$\hat{\beta}(\lambda) = \arg\min_{\beta} \left\{ \frac{1}{2n} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\}.$$

Вторая часть $\lambda\|\beta\|_1$ — L1-штраф. Он заставляет коэффициенты обнуляться, поскольку производная L1 имеет “клин” в нуле.

- **Интерпретация $\lambda$.** Малое $\lambda$ почти не штрафует, модель похожа на OLS. Большое $\lambda$ подавляет коэффициенты, превращая модель в простую (вплоть до нуля).
- **Геометрическая интуиция.** Лассо ограничивает коэффициенты $\beta$ ромбом (L1-шар), который имеет вершины на осях. Пересечение этого ромба с эллиптическими уровнями RSS часто происходит в вершинах, поэтому многие коэффициенты становятся точным нулём.
- **Сравнение с ridge.** Ridge использует L2-штраф, который приводит к сжатию, но не обнуляет коэффициенты. Лассо делает реальные отборы признаков.

## 4.2. Метрики TPR и FPR

- **True Positive Rate (TPR):**

$$\text{TPR} = \frac{\#\{\text{релевантные признаки, выбранные моделью}\}}{\#\{\text{всех релевантных признаков}\}}.$$

Значение 1 означает, что модель нашла все важные признаки.

- **False Positive Rate (FPR):**

$$\text{FPR} = \frac{\#\{\text{нерелевантные признаки, ошибочно выбранные моделью}\}}{\#\{\text{всех нерелевантных признаков}\}}.$$

Низкое значение показывает, что модель почти не добавляет лишних признаков.

- **Компромисс.** Лассо регулирует баланс между TPR и FPR одним параметром $\lambda$. Низкое $\lambda$ даёт высокий TPR и высокий FPR (много признаков), высокое $\lambda$ — наоборот.

## 4.3. Данные симуляции

- **Размеры:** $p = 100$ признаков, только 10 из них релевантны. $n = 1000$ наблюдений, из которых 100 для обучения, 900 для теста. Такое соотношение демонстрирует задачу “много признаков, мало обучающих данных”.
- **Сигнал:** пять признаков имеют коэффициенты $0.7$, ещё пять — $0.3$; шум $\varepsilon \sim N(0, 1)$. Так мы различаем сильные и слабые сигналы.
- **Генерация:** `generate_dataset` формирует матрицу $X$ с независимыми стандартно-нормальными признаками и соответствующий вектор $y$.

## 4.4. Процедура выбора λ

- **`glmnet`.** Реализует решение лассо с помощью алгоритма координатного спуска. Генерирует сетку значений $\lambda$.
- **Тестовая выборка.** Для каждой $\lambda$ считаем предсказания на тесте и выбираем $\lambda$ с минимальным MSE. Это имитация выбора по отложенным данным.
- **Альтернатива: `cv.glmnet`.** Встроенная кросс-валидация. В упражнении используем явный тест для контроля. Можно адаптировать код под `cv.glmnet`, тогда оптимальный $\lambda$ выбирается по 10-fold CV.

## 4.5. Функции и объекты

- **`glmnet(x_train, y_train, alpha = 1)`.** Строит путь решений для лассо (alpha=1). Стандартно стандартизирует признаки. Можно подавать аргумент `standardize = FALSE`, если уже масштабировали данные.
- **`predict(fit, newx, s = lambda)`.** Возвращает предсказания для конкретного $ . Еслиs‘ не указан, по умолчанию используется целый путь.
- **`coef(fit, s = lambda)`.** Возвращает коэффициенты модели при заданном $ ‘. Вектор включает интерсепт (первый элемент) и $p$ коэффициентов.

- **`abs(beta_hat) > 1e-6`.** Числовой критерий для отличия от нуля. Из-за численных погрешностей коэффициенты редко становятся *точно* нулём, поэтому сравниваем с маленьким порогом.
- **`intersect`.** Находит общие элементы вектора выбранных индексов и множества релевантных (или нерелевантных) признаков.
- **`par(mfrow = c(1, 2))`.** Строит два графика рядом: зависимость TPR и FPR от $\lambda$.

## 4.6. Терминология

- **Регуляризация.** Добавление штрафа к функции потерь для ограничения сложности модели. Лассо — L1 регуляризация.
- **Координатный спуск.** Итеративный метод оптимизации, который последовательно обновляет каждый коэффициент, фиксируя остальные.
- **Сжатие (shrinkage).** Уменьшение абсолютных значений коэффициентов за счёт штрафа. В лассо одновременно выполняет отбор.
- **Гиперпараметр.** Параметр, который не оцениваем напрямую из данных, а настраиваем (например, $\lambda$).
- **Sparsity (разреженность).** Свойство модели иметь много нулевых коэффициентов. Лассо стремится к разреженным решениям.

## 4.7. Практические наблюдения

- **Высокий TPR + низкий FPR** — цель. В нашем сценарии средний TPR составляет ≈0.95, а FPR — ≈0.24: сильные сигналы находятся, но некоторая доля шумовых признаков прорывается.
- **Вариативность.** Даже при фиксированной процедуре TPR и FPR колеблются между симуляциями. Это влияние случайного шума и конкретных выборок. Гистограммы показывают распределение значений.
- **Выбор порога.** Порог $1e-6$ в критерии “коэффициент нулевой” можно адаптировать. Если признаки плохо масштабированы, стоит использовать адаптивный порог (например, $1e-4$).
- **Время вычисления.** Решение 100 симуляций с `glmnet` быстрое благодаря векторизации и использованию эффективных алгоритмов на C/Fortran.

## 4.8. Расширение эксперимента

- **Использование `cv.glmnet`.** Позволяет выбирать $\lambda$ по кросс-валидации, а не по тесту. Возвращает `lambda.min` и `lambda.1se`, аналогично ridge/лассо в задаче 8.
- **Чувствительность к коррелированным признакам.** Если признаки коррелированы, лассо может выбирать один из них и отбрасывать остальные, даже если все релевантны. Для таких сценариев используют elastic net (смешение L1 и L2 штрафов).
- **Переменная интенсивность сигнала.** Если коэффициенты релевантных признаков имеют разную величину, TPR может снижаться: признаки со слабым сигналом первыми “отваливаются” при росте $\lambda$.
- **Шумовые признаки.** Можно вводить признаки с распределением сдвинуто / разной дисперсией, чтобы проверить устойчивость лассо.

## 4.9. Связь с ROC-кривыми

- **TPR и FPR** по изменению $\lambda$ наполняют ROC-кривую (хотя здесь мы смотрим на один участок). По сути, $\lambda$ управляет точкой на ROC: низкое $\lambda$ — точка с высокими TPR и FPR, высокое — со сниженным TPR и FPR.
- **Precision/Recall.** Можно аналогично считать точность и полноту. Для отбора признаков особенно полезен F1-score.

## 4.10. Вопросы для окончательного понимания

1. Почему лассо обнуляет коэффициенты, а ridge — нет?

2. Как выбрать оптимальный $\lambda$, если нет тестовой выборки?
3. Что произойдёт с TPR/FPR, если увеличить шум в данных (увеличить $\sigma$)?
4. Как изменить код, чтобы использовать `cv.glmnet` вместо тестового отбора?
5. Что случится, если релевантные признаки коррелированы между собой?
6. Как вычислить стандартные ошибки коэффициентов в лассо?
7. Почему важно масштабировать признаки перед применением лассо?
8. В чём отличие между лассо и forward selection?
9. Как elastic net сочетает свойства лассо и ridge?

Разобрав все эти вопросы и термины, можно детально рассказывать о работе лассо, интерпретировать коэффициенты и объяснять стратегию выбора $\lambda$.