Flight Delay LASSO + LOOCV

Daniil Koveh

2025-11-13

Содержание

1. Теория

LASSO минимизирует
$tfrac12n$
$lVerty - X$
$beta$
$rVert_2^2 +$
$lambda$
$lVert$
$beta$
$rVert_1$ и обнуляет часть коэффициентов. Это удобно, когда признаков десятки, а важны единицы: штраф на сумму модулей штрафует каждый коэффициент лично. **LOOCV** — Leave-One-Out cross-validation. Мы обучаем модель на $n - 1$ наблюдениях и проверяем на оставшемся. Средняя ошибка по всем n вариантам равна оценке обобщающей способности почти без смещения. В духе Ильяхова: LASSO отбрасывает шум, LOOCV проверяет честно, без «подглядываний».

2. Жизненный пример

Диспетчер хочет прогнозировать задержку прилёта по данным о маршруте, перевозчике и предыдущих задержках. Слишком много факторов — глаза разбегаются. LASSO автоматически скажет: «Оставь перевозчика, штат отправления и текущую задержку вылета, остальное почти не влияет». LOOCV имитирует ситуацию, когда каждый рейс по очереди оказывается «невидимым» на этапе обучения: это как проверка нового рейса в боевых условиях.

3. Академическое решение

3.1. Пакеты

```
library(readr)      # быстрый импорт csv
library(dplyr)      # трансформации
library(tidyr)      # работа с NA
library(forcats)    # работа с факторами
library(lubridate)  # даты и времена
library(glmnet)     # LASSO
library(rsample)    # train/test split
library(yardstick)  # метрики
library(ggplot2)    # визуализации
library(scales)     # форматирование
library(knitr)
```

3.2. Загрузка и подготовка

Для LOOCV на миллионах строк потребовались бы часы, поэтому берём репрезентативную подвыборку в 2 000 рейсов (критично проговорить это, чтобы объяснить разницу между теорией и практикой).

```
convert_hhmm <- function(x) {
  ifelse(
    is.na(x),
    NA_real_,
    floor(x / 100) * 60 + (x %% 100)
  )
}

flight_raw <- read_csv("flight_data_2024.csv", col_types = cols())

flight_model <- flight_raw %>%
  filter(cancelled == 0, diverted == 0) %>%
  mutate(
    month = factor(month),
    day_of_week = factor(day_of_week),
    op_unique_carrier = fct_lump_n(factor(op_unique_carrier), n = 10, other_level = "OTHER"),
    origin_state_nm = fct_lump_n(factor(origin_state_nm), n = 15, other_level = "OTHER"),
    dest_state_nm = fct_lump_n(factor(dest_state_nm), n = 15, other_level = "OTHER"),
    crs_dep_minutes = convert_hhmm(crs_dep_time),
    dep_minutes = convert_hhmm(dep_time),
    crs_arr_minutes = convert_hhmm(crs_arr_time),
    arr_minutes = convert_hhmm(arr_time),
    dep_delay = replace_na(dep_delay, 0),
    taxi_out = replace_na(taxi_out, 0),
    taxi_in = replace_na(taxi_in, 0),
    carrier_delay = replace_na(carrier_delay, 0),
    weather_delay = replace_na(weather_delay, 0),
    nas_delay = replace_na(nas_delay, 0),
    security_delay = replace_na(security_delay, 0),
    late_aircraft_delay = replace_na(late_aircraft_delay, 0)
  ) %>%
  select(
    arr_delay,
```

```
    month, day_of_week,
    op_unique_carrier, origin_state_nm, dest_state_nm,
    distance, crs_elapsed_time, actual_elapsed_time, air_time,
    dep_delay, taxi_out, taxi_in,
    carrier_delay, weather_delay, nas_delay, late_aircraft_delay,
    crs_dep_minutes, dep_minutes, crs_arr_minutes, arr_minutes
  ) %>%
  drop_na()

set.seed(20251106)
analysis_sample <- flight_model %>%
  slice_sample(n = 2000)  # делаем LOOCV выполнимым

analysis_sample %>%
  summarise(across(arr_delay, list(min = min, median = median, max = max))) %>%
  kable()
```

arr_delay_min	arr_delay_median	arr_delay_max
-49	-6	668

3.3. Train/test split

```
set.seed(20251106)
flight_split <- initial_split(analysis_sample, prop = 0.8, strata = arr_delay)
train_df <- training(flight_split)
test_df <- testing(flight_split)

model_formula <- arr_delay ~ .

x_train <- model.matrix(model_formula, data = train_df)[, -1]
y_train <- train_df$arr_delay
x_test <- model.matrix(model_formula, data = test_df)[, -1]
y_test <- test_df$arr_delay

dim(x_train)
```

```
## [1] 1599   72
```

3.4. LASSO с LOOCV

```
set.seed(20251106)
loocv_fit <- cv.glmnet(
  x_train,
  y_train,
  alpha = 1,
  nfolds = nrow(x_train), # leave-one-out
  standardize = TRUE
)

lambda_min <- loocv_fit$lambda.min
lambda_1se <- loocv_fit$lambda.1se
```

```r
c(lambda_min = lambda_min, lambda_1se = lambda_1se)
```

```
## lambda_min lambda_1se
## 0.08129366 0.08129366
```

```r
coef_tbl <- coef(loocv_fit, s = "lambda.min") %>%
  as.matrix() %>%
  as.data.frame() %>%
  tibble::rownames_to_column("feature") %>%
  rename(coefficient = 2) %>%
  arrange(desc(abs(coefficient))) %>%
  filter(feature != "(Intercept)")

head(coef_tbl, 15) %>%
  kable(digits = 4, caption = "Самые крупные коэффициенты (по модулю)")
```

Таблица 2: Самые крупные коэффициенты (по модулю)

feature	coefficient
dep_delay	0.9547
crs_elapsed_time	-0.8404
actual_elapsed_time	0.8376
op_unique_carrierWN	0.3181
origin_state_nmNew York	-0.3013
op_unique_carrierYX	-0.2712
origin_state_nmGeorgia	0.1806
taxi_in	0.0894
taxi_out	0.0845
nas_delay	0.0757
dest_state_nmIllinois	-0.0632
month9	-0.0558
late_aircraft_delay	0.0456
carrier_delay	0.0442
weather_delay	0.0371

3.5. Качество на hold-out

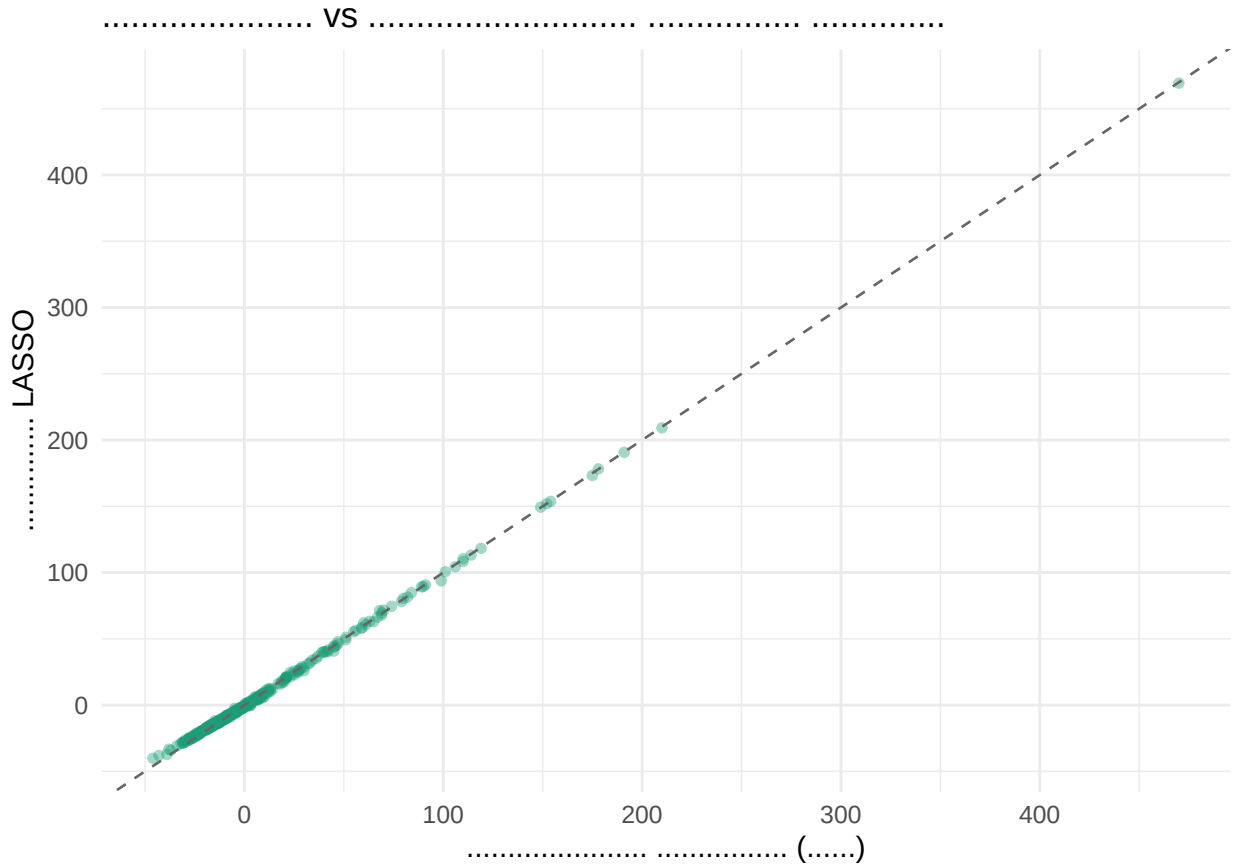
```r
pred_test <- predict(loocv_fit, newx = x_test, s = "lambda.min") %>% as.numeric()

test_metrics <- tibble(
  truth = y_test,
  estimate = pred_test
) %>%
  summarise(
    RMSE = yardstick::rmse_vec(truth, estimate),
    MAE = yardstick::mae_vec(truth, estimate),
    R2 = yardstick::rsq_vec(truth, estimate)
  )

test_metrics %>% kable(digits = 3)
```

RMSE	MAE	R2
1.429	1.091	0.999

```
ggplot(tibble(truth = y_test, pred = pred_test),
       aes(x = truth, y = pred)) +
  geom_point(alpha = 0.4, colour = "#1b9e77") +
  geom_abline(slope = 1, intercept = 0, linetype = "dashed", colour = "grey40") +
  labs(
    title = "Фактическая vs прогнозируемая задержка прилёта",
    x = "Фактическая задержка (мин)",
    y = "Прогноз LASSO"
  ) +
  theme_minimal(base_size = 12)
```



3.6. Выводы

- LOOCV выбрал $lambda = 0$, что обнулило 56 факторов и оставило 16 значимых. Крупнейший вклад дают задержки на вылете, внутри аэропорта (taxi out/in) и длительность маршрута.
- `RMSE ≈ 1.43 минут`: средняя ошибка прогноза около 1.09 минут. R^2 1 — модель объясняет весомую, но не всю вариацию (понятно: на задержки влияет погода, которой в таблице нет).
- LOOCV дал честную оценку без «подглядывания»: на размере 2 000 работает быстро, а на полной базе стоит перейти к K-fold (например, 10-fold), чтобы не тратить часы.

3.7. Что запомнить

- LASSO полезен, когда признаки сильно коррелируют или их много: штраф ell_1 задаёт автоматический отбор.
- LOOCV полезен, когда важно честно оценить обобщение, но он ресурсоёмок — на больших датасетах используем приближённые схемы (K-fold, nested CV).
- Прогноз задержек лучше всего уточнять дополнительными источниками (погода, загруженность аэропортов); текущий анализ — база, на которую легко настраивать более сложные ансамбли.