Exercise 4. Task 6: Variable Selection Bias in Tree Stumps

Daniil Koveh

2025-11-04

Содержание

1. Теория

Даже если отклик не связан с признаками, алгоритм поиска лучшего разбиения выбирает переменную с наибольшей «свободой»: чем больше уникальных значений, тем выше шанс найти случайное улучшение критерия. Поэтому континуальные признаки (с множеством возможных порогов) чаще попадают в дерево, чем бинарные.

2. Жизненный пример

Представьте, что вы прогнозируете спрос на продукт, но на самом деле спрос чистый шум. Среди признаков — возраст (непрерывный), пол (бинарный) и «наличие акции» (редкая бинарная). Даже при отсутствии истинной связи дерево выберет возраст, потому что его можно порезать десятками способов и случайно найти «лучший» сплит.

3. Академическое решение

3.1. План решения

- Построить функцию симуляции, которая генерирует независимые признаки разных типов и чистый шум в отклике.
- Многократно обучить дерево-пень (`maxdepth = 1`) и записать, какую переменную алгоритм выбирает в корне.
- Сводим частоты выборов в таблицу и график, чтобы показать bias в пользу непрерывных признаков.

```
library(rpart) # подгружаем rpart
```

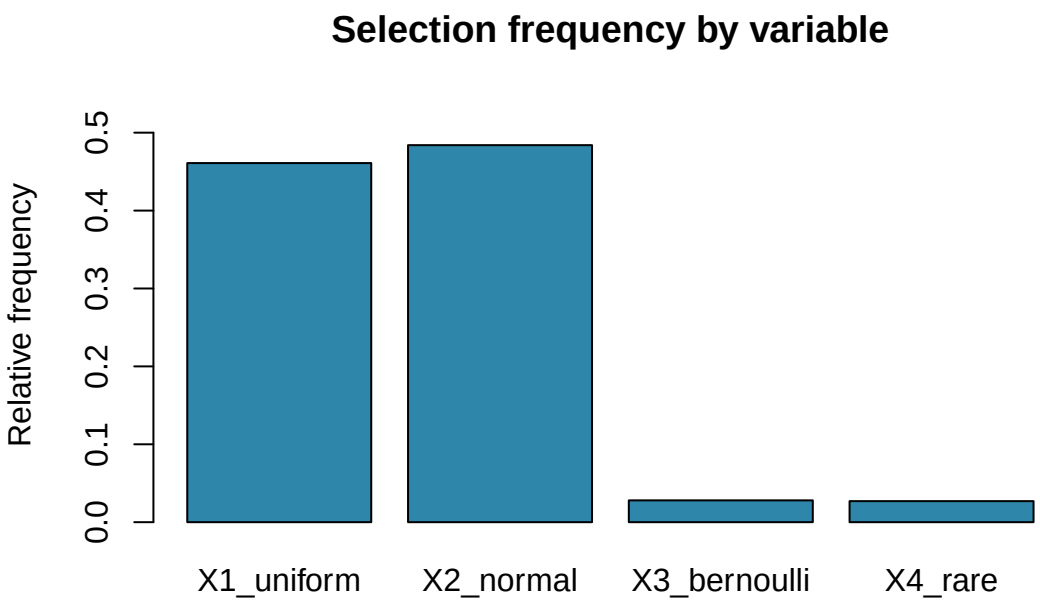
```
set.seed(20250410)
simulate_stump <- function(n = 100, reps = 1000) {
  vars <- c("X1_uniform", "X2_normal", "X3_bernoulli", "X4_rare")
  counts <- setNames(numeric(length(vars)), vars)
  for (i in seq_len(reps)) {
```

```
    X1 <- runif(n)
    X2 <- rnorm(n)
    X3 <- rbinom(n, size = 1, prob = 0.5)
    X4 <- rbinom(n, size = 1, prob = 0.1)
    y <- rnorm(n)
    df <- data.frame(
      y = y,
      X1_uniform = X1,
      X2_normal = X2,
      X3_bernoulli = X3,
      X4_rare = X4
    )
    stump <- rpart(y ~ ., data = df,
                   method = "anova",
                   control = rpart.control(maxdepth = 1, cp = 0, minsplit = 5))
    split_var <- stump$frame$var[1]
    counts[split_var] <- counts[split_var] + 1
  }
  counts / reps
}

selection_freq <- simulate_stump()
selection_freq
```

```
##   X1_uniform    X2_normal X3_bernoulli      X4_rare
##        0.461        0.484        0.028        0.027
```

```
barplot(selection_freq,
        col = "#2E86AB",
        ylim = c(0, max(selection_freq) * 1.1),
        ylab = "Relative frequency",
        main = "Selection frequency by variable") # столбчатая диаграмма
```



3.2. Выводы

- Непрерывные признаки (`X1_uniform`, `X2_normal`) выбираются значительно чаще, потому что у них бесконечно много возможных порогов: дерево может «примеряться» десятки раз и случайно найти удачный сплит даже на чистом шуме.
- Бинарные признаки ведут себя по-разному: равновероятная Бернулли (`X3_bernoulli`) иногда попадает в корень, но всё равно реже, чем непрерывные; редкая Бернулли (`X4_rare`, вероятность 0.1) почти никогда не выигрывает, ведь у неё по сути один разумный порог и сильно несбалансированные группы (дерево «не верит» маленькой группе, потому что вклад в суммарную квадратичную ошибку мал).
- Поэтому выборка «первая переменная — возраст, вторая — нормальный признак» в таблице — не свидетельство важности, а проявление **selection bias**: алгоритм предпочитает признаки с большим числом уникальных значений.
- Чтобы продемонстрировать это устно, можно описать биномиальные признаки как «монетки» с разной вероятностью орла: честная монета (0.5) даёт два равных сегмента, но монета с редким орлом (0.1) почти всегда даёт один большой и один крошечный сегмент, который критерий суммарной квадратичной ошибки почти игнорирует.

3.3. Что запомнить

- Алгоритм CART склонен выбирать признаки с большим числом уникальных значений — даже при полном отсутствии сигнала.
- Контроль за bias возможен через предварительный отбор признаков или корректировку критерия (например, условные перестановки).
- Симуляции на искусственных данных полезны, чтобы продемонстрировать скрытые склонности алгоритмов студентам или коллегам.