Упражнение 3. Задача 4: Маргинальное правдоподобие для Бернуллиевой модели

Даниил Ковех

2025-10-28

Содержание

1. Теория

Наблюдаем N независимых исходов $y = (y_1, \dots, y_N)$, где $y_i \in \{0, 1\}$. Вероятность успеха обозначим θ.

- Условное правдоподобие:

$$p(y \mid \theta) = \theta^s (1-\theta)^{N-s}, \quad s = \sum_{i=1}^{N} y_i.$$

- Приоритетное распределение $\theta \sim \text{Beta}(\alpha, \beta)$ с плотностью

$$p(\theta \mid \alpha, \beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1}.$$

Маргинальное правдоподобие (доказательство классической формулы Бета-Биномиального распределения):

$$p(y \mid M) = \int_0^1 \theta^{s+\alpha-1}(1-\theta)^{N-s+\beta-1} \, \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \, \mathrm{d}\theta.$$

Интеграл — это нормировочный множитель $\text{Beta}(s+\alpha, N-s+\beta)$. Поэтому

$$p(y \mid M) = \frac{\mathrm{B}(s+\alpha, N-s+\beta)}{\mathrm{B}(\alpha, \beta)},$$

где $\mathrm{B}(a, b) = \frac{\Gamma(a)\Gamma(b)}{\Gamma(a+b)}$.

Это решение полностью аналитическое и легко вычисляется в любом CAS.

2. Жизненный пример

Допустим, мы анализируем долю пользователей, которые завершили покупку после просмотра новой рекламной креативы. Аудитория небольшая, и мы хотим аккуратно учитывать неопределённость. Бета-распределение — удобный способ априори зажать вероятность успеха в диапазоне $[0, 1]$. Маргинальное правдоподобие показывает, насколько данные поддерживают модель. Если сравниваем несколько гипотез (например, разные креативы), то нормировочный множитель выступает оценкой "качества" модели с учётом неопределённости параметра.

3. Академическое решение

3.1. 1. Формула маргинального правдоподобия

Обозначим s — количество успехов. Тогда

$$p(y \mid M) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} \frac{\Gamma(s+\alpha)\Gamma(N-s+\beta)}{\Gamma(N+\alpha+\beta)}.$$

Это удобно вычислять в логарифмической шкале, используя `lgamma`.

```
marginal_log_likelihood <- function(s, N, alpha = 1, beta = 1) {
  log_beta_prior <- lgamma(alpha) + lgamma(beta) - lgamma(alpha + beta)
  log_beta_posterior <- lgamma(s + alpha) + lgamma(N - s + beta) - lgamma(N + alpha + beta)
  log_beta_posterior - log_beta_prior
}

marginal_likelihood <- function(s, N, alpha = 1, beta = 1) {
  exp(marginal_log_likelihood(s, N, alpha, beta))
}
```

3.2. 2. Вычисление $-2 \log p(y \mid M)$ для разных N

Условия задачи: $\alpha = \beta = 1$ (равномерный приор), вероятность успеха 0.1. Значит $s \in \{1, 10, 100, 1000\}$ при $N \in \{10, 100, 1000, 10000\}$.

```
scenarios <- data.frame(
  N = c(10, 100, 1000, 10000),
  s = c(1, 10, 100, 1000)
)

scenarios$log_marginal <- mapply(
  marginal_log_likelihood,
  s = scenarios$s,
  N = scenarios$N,
  alpha = 1,
  beta = 1
)

scenarios$neg2log_marginal <- -2 * scenarios$log_marginal
scenarios
```

```
##       N    s log_marginal neg2log_marginal
## 1    10    1     -4.70048         9.400961
## 2   100   10    -35.09744        70.194888
## 3  1000  100   -328.82204       657.644084
## 4 10000 1000  -3255.71995      6511.439908
```

3.3. 3. Интерпретация

- При малом N данные не слишком уверенно отличают вероятность 0.1 от других значений, потому что маргинальное правдоподобие относительно “плоское”.
- По мере роста N значение $-2\log p(y \mid M)$ увеличивается почти линейно, что отражает концентрацию апостериорного распределения вокруг истинного значения.
- Такую метрику можно использовать для сравнения моделей с разными априорными предположениями или разным числом параметров.

3.4. 4. Проверка численной устойчивости

Для очень больших N нужно работать в логарифмах. В R функция `lgamma` решает проблему переполнения.

```r
large_case <- marginal_log_likelihood(s = 1000, N = 10000, alpha = 1, beta = 1)
large_case
```

```
## [1] -3255.72
```

Значение отрицательное, что отражает, что плотность вероятности меньше единицы. Перемножать факторы напрямую нельзя — продукты гамма-функций были бы огромны. Логарифмическая форма сохраняет точность.

3.5. 5. График зависимости

```r
plot(scenarios$N, scenarios$neg2log_marginal,
     type = "b",
     xlab = "Размер выборки N",
     ylab = expression(-2 * log(p(y %|% M))),
     main = "-2 логарифм маргинального правдоподобия (равномерный приор)")
```

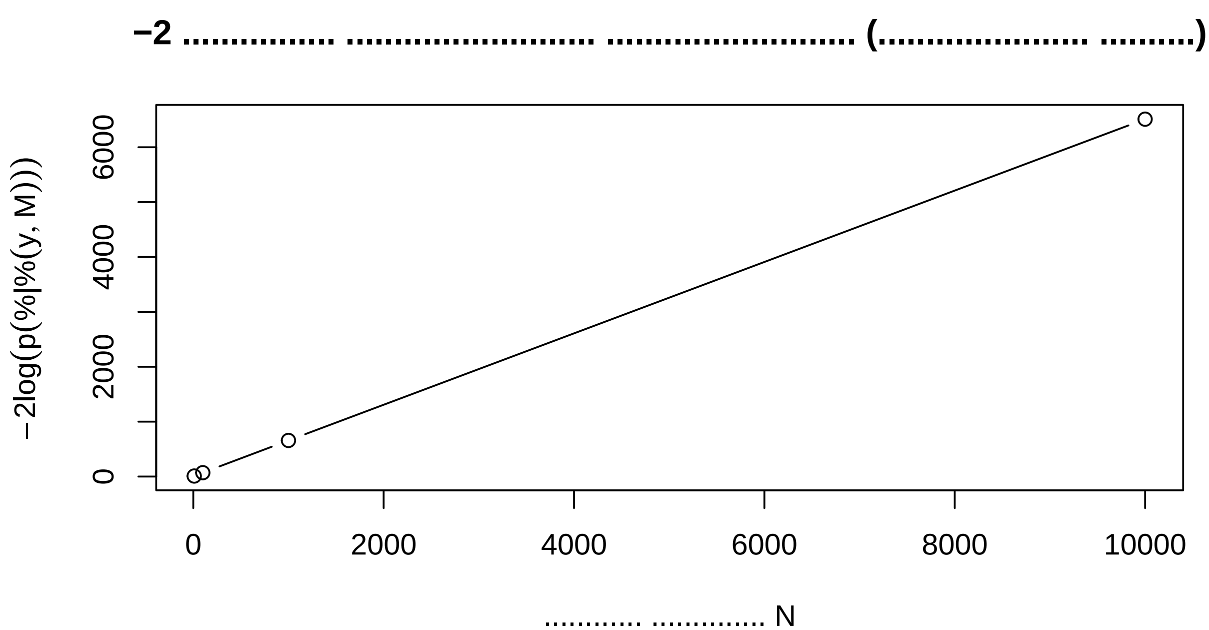

График подчёркивает, что показатель растёт вместе с размером выборки. Дополнительные данные дают больше информации, и нормировочный множитель становится всё меньше, поскольку априорное распределение “сжимается” в районе наблюдаемых данных.