ミクロデータサイエンス
Problemset2

2125178
廣江友哉

2024/6/19

1　囚人と主観的確率

1.1　３つの扉

事前確率として, 扉が正解である事象 X の確率は $P(X) = \frac{1}{3}$ とする. また, 看守がいずれかの扉を選ぶ事象を Y とする. 例えば, 囚人が A の扉を選び看守が不正解の扉として C の扉を開けたとすると, A と B それぞれの扉が正解の扉である事後確率はベイズの定理を用いて以下のように表せる.

$$P(X = A|Y = C) = \frac{P(Y = C|X = A)}{P(Y = C)}P(X = A) \tag{1}$$

$$P(X = B|Y = C) = \frac{P(Y = C|X = B)}{P(Y = C)}P(X = B) \tag{2}$$

ここで, A の扉が正解だったとすると以下のように具体的な確率を求めることができる.

$$P(Y = C) = \frac{1}{2} \quad P(Y = C|X = A) = \frac{1}{2} \tag{3}$$

$$P(Y = C) = \frac{1}{2} \quad P(Y = C|X = B) = 1 \tag{4}$$

上記の式は, 囚人が正解の扉を選んでいる場合看守は残りのどちらの扉も選ぶことができるのでその確率を仮に $\frac{1}{2}$ とし, 囚人が不正解の扉を選んでいる場合に看守は扉を一つしか選択する余地がないので確率が 1 となることを表している. 上記の確率の値を使用して, ベイズの定理の式 1 と式 2 を計算すると,

$$P(X = A|Y = C) = \frac{\frac{1}{2}}{\frac{1}{2}} \times \frac{1}{3} = \frac{1}{3} \tag{5}$$

$$P(X = B|Y = C) = \frac{1}{\frac{1}{2}} \times \frac{1}{3} = \frac{2}{3} \tag{6}$$

したがって, 囚人は A から B へ扉を変更した方が正解の確率が上がるので, 囚人の扉を選び直すという行為は正しいことがわかる.

1.2 3 人の囚人

囚人が釈放される事象 X の確率が均等に $\frac{1}{3}$ とすると, 看守が教える囚人の名前を事象 Y とする. ここで, 便宜上看守が伝えた処刑される囚人の名前を C とすると, 前問のベイズの定理の式と同じ形になる.

$$P(X=A|Y=C) = \frac{P(Y=C|X=A)}{P(Y=C)}P(X=A) \tag{7}$$

$$P(X=B|Y=C) = \frac{P(Y=C|X=B)}{P(Y=C)}P(X=B) \tag{8}$$

具体的な確率を計算すると,

$$P(Y=C) = \frac{1}{2} \quad P(Y=C|X=A) = \frac{1}{2} \tag{9}$$

$$P(Y=C) = \frac{1}{2} \quad P(Y=C|X=B) = 1 \tag{10}$$

これを元にベイズの定理に沿って計算を進めると,

$$P(X=A|Y=C) = \frac{\frac{1}{2}}{\frac{1}{2}} \times \frac{1}{3} = \frac{1}{3} \tag{11}$$

$$P(X=B|Y=C) = \frac{1}{\frac{1}{2}} \times \frac{1}{3} = \frac{2}{3} \tag{12}$$

したがって, 囚人 A が釈放される事後確率は $\frac{1}{3}$ となるため問題文の囚人 A は釈放される確率が高まると喜んでいるのは誤りである.

1.3 比較

問題 A と B のどちらも, 対象となる人物の事前確率と事後確率に変化がないことがわかる.

2 パレート分布と最尤法

2.1 パレート分布の尤度関数

$$F(x) = \begin{cases} 1 - \left(\frac{\beta}{x}\right)^{\alpha} & \text{if } x \geq \beta, \\ 0 & \text{if } x < \beta \end{cases} \tag{13}$$

密度関数を求めると,

$$f(x) = \begin{cases} \alpha\beta^{\alpha}x^{-\alpha-1} & \text{if } x \geq \beta \\ 0 & \text{if } x < \beta \end{cases} \tag{14}$$

尤度関数は,

$$L(\alpha, \beta) = \prod_{x=\beta}^{n} \alpha \beta^{\alpha} x^{-\alpha-1} \tag{15}$$

対数尤度関数は,

$$\begin{aligned} l(\alpha, \beta) &= \sum_{x=i}^{n} \log \alpha \beta^{\alpha} x^{-\alpha-1} \\ &= \sum_{x=i}^{n} (\log \alpha + \alpha \log \beta - (\alpha+1) \log x) \\ &= n \log \alpha + n\alpha \log \beta - (\alpha+1) \sum_{x=i}^{n} \log x \end{aligned}$$

β は，$x \geq \beta$ より，観測値の最小値が最大値となる．α について偏微分すると，

$$\begin{aligned} \frac{\partial l(\alpha, \beta)}{\partial \alpha} &= \frac{n}{\alpha} + n \log \beta - \sum_{x=i}^{n} \log x = 0 \\ \alpha &= \frac{n}{n \log \beta - \sum_{x=i}^{n} \log x} \end{aligned}$$

2.2 解析解を用いた推定

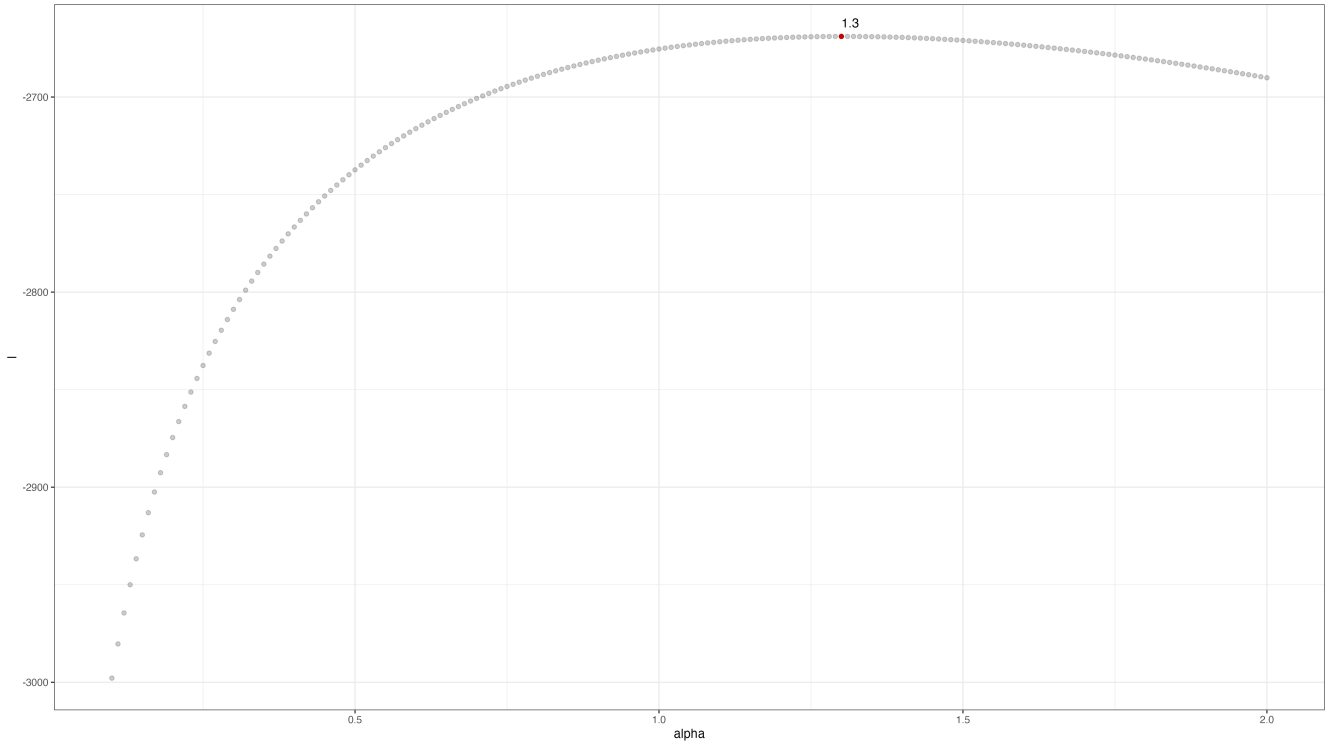
α の値は -1.303965，β の値は 139128 となった．以下にソースコード 1 を添付する．

ソースコード 1: 解析解を用いた推定

```
1  # (b)
2  sample_size <- 200
3  beta <- min(df$population)
4  alpha_hat <- length(df$population) /
5    (length(df$population) * log(beta) - sum(log(df$population)))
```

2.3 グリッド探索を用いた推定

次のグラフのように，尤度が最大になる時の α は 1.3 となり，概ね解析会を用いた推定値と一致する．



並び替えた完成系の *simulate_mle* 関数は以下のようになった.

ソースコード 2: 解析解を用いた推定

```
# (c)
simulate_mle <- function(df, beta, sample_size) {
  a <- "((alpha + 1)*sum(log(df$population))))"
  b <- "sample_size*alpha*log(beta)"
  c <- "(sample_size*log(alpha)"
  d <- "-"
  e <- "+"

  # 以下のコードを正しく並べ替えること
  df_likelihood <- dplyr::tibble(
    alpha = seq(0.1, 2, by = 0.01),
    l = eval(parse(text = paste0(c, e, b, d, a)))
  ) |>
    dplyr::mutate(
      alpha_hat = dplyr::if_else(max(l) == l, 1, 0)
    )


  ggplot() +
    geom_point(data = df_likelihood |> dplyr::filter(alpha_hat == 1), mapping = aes(x =
        alpha, y = l), color = "red") +
    geom_point(data = df_likelihood, mapping = aes(x = alpha, y = l), alpha = 0.2) +
    geom_text(data = df_likelihood |> dplyr::filter(alpha_hat == 1), mapping = aes(x = alpha
        , y = l, label = alpha), hjust = 0, vjust = -1) +
    theme_bw()
}
```

2.4 勾配法を用いた推定

以下に示すように，初期値をどこから始めても $\alpha = 1.304$ となる．計算時間が最も短いのは，初期値が 2 と 200 の時で，0.009 秒だった．ここから求める α 日かいい値から始めた方が勾配法では計算量が減ることがわかる．

ソースコード 3: 勾配法を用いた推定

```
  0.009 sec elapsed
  [1] "alpha: 1.304"
  [1] "-log_likelihood: 2668.926"
  [1] "counts: 25" "counts: 6"

  0.009 sec elapsed
  [1] "alpha: 1.304"
  [1] "-log_likelihood: 2668.926"
  [1] "counts: 37" "counts: 13"

  0.046 sec elapsed
  [1] "alpha: 1.304"
  [1] "-log_likelihood: 2668.926"
  [1] "counts: 74" "counts: 20"
```

ソースコード 4: コード

```
#  (d)
log_likelihood <- function(alpha_1, df, beta, sample_size) {
  a <- "((alpha_1[1] + 1)*sum(log(df$population))))"
  b <- "sample_size*alpha_1[1]*log(beta)"
  c <- "(sample_size*log(alpha_1[1])"
  d <- "-"
  e <- "+"

  likelihood_fn <- eval(parse(text = paste0("-", c, e, b, d, a)))
}

run_gradient_method <- function(alpha_0, df, beta, sample_size) {
  tictoc::tic()
  list_par <- optim(
    c(alpha_0),
    log_likelihood,
    df = df,
    beta = beta,
    sample_size = sample_size,
    method = "BFGS"
  )
  tictoc::toc()

  print(paste0("alpha: ", round(list_par$par, digits = 3)))
  print(paste0("-log_likelihood: ", round(list_par$value, digits = 3)))
  print(paste0("counts: ", list_par$counts))
}

# (d)
run_gradient_method(alpha_0 = 2, df = df, beta = beta, sample_size = sample_size)
run_gradient_method(alpha_0 = 200, df = df, beta = beta, sample_size = sample_size)
run_gradient_method(alpha_0 = 20000, df = df, beta = beta, sample_size = sample_size)
```

2.5 パレート分布の仮定の検証 (Adv.)