

Statistical computing MATH10093

Coursework B 2019/20

Finn Lindgren

20/3/2020

Summary

Handout Friday 20/3/2020, handin as pdf via Learn by noon Tuesday 14/4/2020. Discussion of the assignment with others is permitted, but handin must be individual solutions. The work will be marked out of 50, and counts for 50% of the total grade.

General notes

- Use RMarkdown (or knitr) used to produce the PDF-handin. See the Learn Announcement about working in the free web based [RStudio.cloud](#) if you do not have a local rstudio installation.
- Ordinary text should be typeset as text, and not as R code chunk comments.
- Use LaTeX math typesetting with $\$ \cdots \$$ for inline formulas and $\$ \$ \cdots \$ \$$ for displayed equations (see lecture 8).
- The code in the [CWB2020code.R](#) file should be included via `source()`, and not included in your report.
- Write readable code.
- *Do not* hide R code with `echo=FALSE`.
- *Do* hide unnecessary R *output*, such as long data listings, with `results='hide'` as RMarkdown code chunk
- Avoid unnecessarily repeating identical code, for example when adding to a previous plot, use the `pl + new_stuff()` technique for `ggplot()`.

Suggested RMarkdown startup code chunk:

```
set.seed(12345L)
source("CWB2020code.R")
suppressPackageStartupMessages(library(tidyverse))
theme_set(theme_bw())
# Read data for part 2 of the assignment:
filament <- read.csv("filamentCWB.csv", stringsAsFactors = FALSE)
# Note: this is a different data file than in CWA.
```

The functions from [CWB2020code.R](#) that you need to call in your own code are either mentioned in the text or part of provided code outlines. Look at the code file for documentation, and to see functions that are used internally and that you do not need to call yourself.

Part 1: Archaeology



“Anno Domini MCCCLXI feria III post Jacobi ante portas Visby in manibus Danorum ceciderunt Gutenses, hic sepulti, orate pro eis!”

“In the year of our Lord 1361, on the third day after St. Jacob, the Goth fell outside the gates of Visby at the hands of the Danish. They are buried here. Pray for them!”

In 1361 the Danish king Valdemar Atterdag conquered Gotland¹ and captured the rich Hanseatic town of Visby. The conquest was followed by a plunder of Visby. Most of the defenders² were killed in the attack and are buried in a field, *Korsbetningen*³, outside of the walls of Visby.

In the 1920s the gravesite was subject to several archaeological excavations. A total of 493 femurs⁴ (256 left, and 237 right) were found. We want to figure out how many persons were likely buried at the gravesite. It must reasonably have been at least 256, but how many more?

Statistical model

To build a simple model for this problem, we assume that the number of left ($y_1 = 256$) and right ($y_2 = 237$) femurs are two independent observations from a $\text{Bin}(N, \phi)$ distribution. Here N is the total number of people buried and ϕ is the probability of finding a femur, left or right, and both N and ϕ are unknown parameters.

The probability function for a single observation $y \sim \text{Bin}(N, \phi)$ is

$$p(y|N, \phi) = \binom{N}{y} \phi^y (1 - \phi)^{N-y}.$$

The function `arch_loglike()` in `CWB2020code.R` evaluates the combined log-likelihood $\log[p(\mathbf{y}|N, \phi)]$ for a collection \mathbf{y} of y -observations. If a `data.frame` with columns `N` and `phi` is provided, the log-likelihood for each row-pair (N, ϕ) is returned.

Questions

1. An archaeological researcher tries to obtain 95% confidence intervals for N and ϕ . They realise that since ϕ must fall in the finite interval $(0, 1)$, it is

¹Strategically located in the middle of the Baltic sea, Gotland had shifting periods of being partly self-governed, and in partial control by the Hanseatic trading alliance, Sweden, Denmark, and the Denmark-Norway-Sweden union, until settling as part of Sweden in 1645. Gotland has an abundance of archaeological treasures, with coins dating back to Viking era trade routes via Russia to the Arab Caliphates.

²Primarily local farmers that could not take shelter inside the city walls.

³Literal translation: *the grazing field that is marked by a cross*, as shown in the picture.

⁴thigh bone

reasonable to reparameterise to $\theta = \log(\phi) - \log(1 - \phi)$, the so-called *logit*-transformation of ϕ . They then attempt to apply Maximum Likelihood theory to N and θ , where the estimates are assumed to be approximately Gaussian, by running the following code:

```
like_est <- arch_likelihood_estimation(c(256, 237))
```

The code automatically transforms the confidence interval limits for θ back to the ϕ scale, as $\phi = 1/(1 + e^\theta)$, and they obtain the intervals

$$\begin{aligned} \text{CI}_N &= (-51, 827), \\ \text{CI}_\phi &= (0.073, 0.975). \end{aligned}$$

After also introducing a log-transformation for N in the internal calculations,⁵ the computed confidence interval for N changes to (125, 1199), with only minor changes to the confidence interval for ϕ .

Explain, in words, what is problematic with these frequentistic confidence intervals. What assumptions were violated?

2. You should now do a Bayesian analysis of the problem (see Lecture 8), to make an improvement over the researchers failed frequentistic attempt.

Let N have a Geometric(ξ), $\xi > 0$, prior distribution, and let ϕ have a Beta(a, b), $a, b > 0$, prior distribution:

$$\begin{aligned} p_N(n) &= \text{P}(N = n) = \xi (1 - \xi)^n, \quad n = 0, 1, 2, 3, \dots, \\ p_\phi(\phi) &= \frac{\phi^{a-1} (1 - \phi)^{b-1}}{B(a, b)}, \quad \phi \in [0, 1]. \end{aligned}$$

Before the excavation took place, the archaeologist believed that around 1000 individuals were buried, and that they would find around half on the femurs. To encode that belief in the Bayesian analysis, set $\xi = 1/(1 + 1000)$, which corresponds to an expected total count of 1000, and $a = b = 2$, which makes ϕ more likely to be close to 1/2 than to 0 or 1.

The posterior probability function (PF) for N given $\mathbf{y} = (y_1, y_2)$ is given by

$$p_{(N|Y)}(n) = \text{P}(N = n | \mathbf{Y} = \mathbf{y}) = \frac{\text{P}(N = n, \mathbf{Y} = \mathbf{y})}{\text{P}(\mathbf{Y} = \mathbf{y})}$$

The numerator can be obtained by integrating the joint distribution for N , ϕ , and \mathbf{Y} ,

$$\text{P}(N = n, \mathbf{Y} = \mathbf{y}) = \int_0^1 p_N(n) p_\phi(\phi) \text{P}(\mathbf{Y} = \mathbf{y} | N = n, \phi) d\phi.$$

⁵This model reparameterisation would only require minor changes to `arch_param_to_theta()` and `arch_param_from_theta()`, but this assignment doesn't involve doing that.

The denominator can be obtained by summing $P(N = n, \mathbf{Y} = \mathbf{y})$ for all n ,

$$P(\mathbf{Y} = \mathbf{y}) = \sum_{n=\max(y_1, y_2)}^{\infty} P(N = n, \mathbf{Y} = \mathbf{y}).$$

Estimate the posterior probability function (PF) $p_{N|y}(n) = P(N = n|y_1, y_2)$ by first performing Monte Carlo integration⁶ over ϕ to obtain $P(N = n, \mathbf{Y} = \mathbf{y})$ for $n = \max(y_1, y_2), \dots, 3000$. The remaining terms, for $n > 3000$ are negligible. Use 1000 Monte Carlo samples of $\phi \sim p_{\phi}(\phi)$ in the integration.

You can use the following code template for the calculations, replacing each `#####` with suitable code

```
y <- c(256, 237)
N_xi <- 1/(1 + 1000)
phi_ab <- list(a = 2, b = 2)
N <- max(y):3000
n_mc <- 1000
# Use the prior for phi for sampling:
phi <- #####
P_NY <- numeric(length(N))
for (loop_phi in phi) {
  # Compute and accumulate the integrand:
  P_NY <- P_NY + #####
}
df <- data.frame(N = N,
                  P_N_posterior = #####)
```

3. Draw a figure with the prior and posterior probability functions for N . Pay attention to the fact that the prior and posterior distributions for N have different supports.
4. Compute a Bayesian 95% credible interval for N , with equal upper and lower tail probabilities, by finding the appropriate quantiles via the posterior cumulative distribution function for N ,

$$P(N \leq n|y_1, y_2) = \sum_{k=0}^n P(N = k|y_1, y_2).$$

Compare the result to the frequentistic confidence intervals the archaeologist computed.

⁶Remark: An alternative, Monte Carlo free approach for this particular model would be to show that the conditional posterior distribution for $(\phi|N = n, \mathbf{Y} = \mathbf{y})$ is $\text{Beta}(a + y_1 + y_2, b + 2n - y_1 - y_2)$. Then the identity $p(n, \mathbf{y}) = \frac{p(\phi, n, \mathbf{y})}{p(\phi|n, \mathbf{y})}$ would provide a direct expression for the joint probability function for N and \mathbf{Y} , eliminating the need for numerical integration.

Part 2: LOOCV and randomisation tests

Recall the 3D printer filament problem from Coursework A. Of the models investigated there, the overall best model had a log-linear model for the variance, that we'll now call model A:

$$y_i \sim N[\theta_1 + \theta_2 x_i, e^{\theta_3 + \theta_4 x_i}],$$

where x_i are the `CAD.Weight` values of `filamentCWB.csv`, and y_i are the observed `Actual.Weight` values. See the suggested setup code chunk for how to load the data file. The data columns are

- `Index`: An observation index, $1, 2, 3, \dots, n$
- `Date`: The date the object was printed.
- `Material`: The material colour
- `CAD.Weight`: The CAD-estimated required filament weight (gram)
- `Actual.Weight`: The measured actual object weight (gram)

The `Date` and `Material` variables are not needed for this assignment.

We are now interested in a slightly different model, that we'll call model B:

$$y_i \sim N[\theta_1 + \theta_2 x_i, e^{\theta_3} + e^{\theta_4} x_i^2].$$

Code for estimating the parameters for each of the two models, and computing predictions for new data, is available in `CWB2020code.R` as the function `estimate_and_predict()`.

1. Show that Model B is mathematically equivalent to an error model that can be written $y_i = \alpha_0 + \gamma_i x_i + e_i$, where γ_i and e_i are random variables with independent Gaussian distributions, $N(\mu_\gamma, \sigma_\gamma^2)$ and $N(0, \sigma_e^2)$. What are the parameters for the γ_i and e_i distributions that yield Model B?

2. Implement leave-one-out cross validation scoring and compare the prediction scores for the two models, for Squared Error (SE), Dawid-Sebastiani (DS), and the Interval score (Interval, use the default 90% intervals).

First, fill in suitable code instead of `#####`:

```
pred <- list(A = estimate_and_predict_output_template(nrow(filament)),
            B = estimate_and_predict_output_template(nrow(filament)))
for (ind in filament$Index) {
  for (model in c("A", "B")) {
    # Leave out one observation, estimate the model, predict the left-out obs:
    pred[[model]][ind,] <- #####
  }
}
```

The scores $S_i^A = S(F_i^A, y_i)$ and $S_i^B = S(F_i^B, y_i)$ can then be computed and stored as follows:

```
scores <- list(A = calc_scores(pred[["A"]], filament[["Actual_Weight"]]),
              B = calc_scores(pred[["B"]], filament[["Actual_Weight"]]))
```

- Let $\overline{S^{A-B}}$ denote the average of the pairwise score differences $S_i^A - S_i^B$, and let G denote the prediction of the true data generating model. For each type of score, estimate the p-value, P_T , for a suitable randomisation test of

$$H_0 : E_G(\overline{S^{A-B}}) = 0,$$

$$H_1 : E_G(\overline{S^{A-B}}) > 0.$$

Use the code below, replacing `#####` with suitable⁷ code. Explain, in words, what assumptions the test is based on.

```
# Compute the test statistics
stat <- colMeans(scores$A - scores$B)
J <- 10000
rand_stat <- data.frame(SE = numeric(J),
                       DS = numeric(J),
                       Interval = numeric(J))
# Compute the randomised test statistics
for (loop in seq_len(J)) {
  rand_stat[loop, ] <- colMeans(#####)
}
# Estimate the P-values for the tests
p_value <- colMeans(rand_stat >= rep(stat, each = J))
```

- For each of the three tests, $Z^{(j)} = \mathbb{I}(T^{(j)} \geq T) \sim \text{Bin}(1, P_T)$, independent across $j = 1, \dots, J$. By the central limit theorem, the estimate \hat{P}_T has, approximately, a $N(P_T, \sigma_T^2)$ distribution. Show that $\sigma_T^2 = P_T(1 - P_T)/J$, and compute a standard deviation estimate $\hat{\sigma}_T$ for each test, by replacing P_T by \hat{P}_T . What is the relative Monte Carlo error (std.dev. divided by the estimate) for each of the three p-value estimates? Also compute an approximate 95% confidence interval for each P_T .

⁷Consult lecture 6 for material on randomisation tests.