Statistical computing MATH10093
Coursework B 2018/19

Finn Lindgren

20/3/2019

Summary

Handout Wednesday 20/3/2019, handin as pdf via Learn by midnight, end of Wednesday 10/4/2019. 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 credits

A total of 10 marks is awarded for general acquired skills:

1. RMarkdown or knitr used to produce the handin [2 marks]
2. Mathematical formula typesetting with RMarkdown [4 marks]
3. Code readability, useful code comments [4 marks]

Include all your code required to generate the results, but suppress unused R output with `results='hide'` as RMarkdown code chunk option when appropriate (e.g. do not include long data listings). The file `CWB2019code.R` must be included with a call to `source()`, and none of the functions in that file should be included in the report. The following four questions are worth 10 marks each, with subtask marks as indicated.

Question 1: (Archaeology, part 1)

"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 S:t 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 Gotland[1] and captured the rich Hanseatic town of Visby. The conquest was followed by a plunder of Visby. Most of the defenders[2] 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 femurs[4] (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) and right (y_2) 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

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

Here we used that $k! = \Gamma(k+1)$.

Since we need ϕ to be a probability in $(0,1)$, change variables to $\theta = \log(\phi) - \log(1-\phi)$. This function and its inverse $\phi = \frac{1}{1+\exp(-\theta)}$ are implemented in the supplementary code as `logit()` and `ilogit()`, respectively. The combined *negated log-likelihood* $l(N,\theta) = -\log p(y_1, y_2|N,\phi)$ for the data set $\{y_1, y_2\}$ is then given by

$$\begin{aligned} l(N,\theta) = &\log\Gamma(y_1+1) + \log\Gamma(y_2+1) \\ &+ \log\Gamma(N-y_1+1) + \log\Gamma(N-y_2+1) - 2\log\Gamma(N+1) \\ &+ 2N\log(1+\mathrm{e}^\theta) - (y_1+y_2)\theta \end{aligned}$$

We will treat N as a continuous parameter.

[1] Strategically 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.

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

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

[4] thigh bone

Tasks for Question 1

1. Write a function `negloglike(param, Y)` with inputs

   `param`: vector with 2 values, N and θ.

   `Y`: vector with 2 values, y_1 and y_2.

   The output should be a scalar, equal to $l(N, \theta)$ when $N \geq \max(y_1, y_2)$, and equal to $+\infty$ (`+Inf`) when $N < \max(y_1, y_2)$.
   Use the `lgamma()` function to evaluate $\log \Gamma(x)$ in the log-likelihood expression, and remember that most R functions can operate element by element on vectors; your function should use at most three calls to `lgamma`, and no indexing into `Y`. [4 marks]

2. Use `negloglike` and `optim` to compute the maximum likelihood estimate of N and θ. Also transform the estimate $\widehat{\theta}$ to an estimate of ϕ.
   Hint: The optimisation must be started for some value $N > \max(y_1, y_2)$. [3 marks]

3. Obtain the Hessian $\boldsymbol{H}$ of $l(N, \theta)$ at the mode $(\widehat{N}, \widehat{\theta})$ via `optimHess(...)`, and compute a 95% confidence interval for N based on Normal approximation with joint covariance matrix $\boldsymbol{H}^{-1}$. Does the interval respect the lower bound $\max(y_1, y_2)$? [3 marks]

Question 2: (Archaeology, part 2)

1. Derive expressions for the derivatives $\frac{\partial l}{\partial \theta}$ and $\frac{\partial l}{\partial N}$. Then derive expressions for the second order derivatives $\frac{\partial^2 l}{\partial \theta^2}$, $\frac{\partial^2 l}{\partial \theta \partial N}$, and $\frac{\partial^2 l}{\partial N^2}$.
   Hints:
   The derivative of $\log \Gamma(x)$ is the *digamma* function, $\Psi(x) = \frac{\mathrm{d} \log \Gamma(x)}{\mathrm{d}x}$.
   The derivative of $\Psi(x)$ is the *trigamma* function, $\Psi'(x) = \frac{\mathrm{d}\Psi(x)}{\mathrm{d}x}$. [4 marks]

2. Write a function `myhessian(param, Y)` that takes the same input parameters as `negloglike()`, and constructs the 2×2 Hessian matrix for $l(N, \theta)$. Compare the resulting Hessian for $(\widehat{N}, \widehat{\theta})$ with the Hessian obtained from `optimHess()`. How large are the relative differences for each element?
   Hint: The function calls `digamma(x)` and `psigamma(x, 1)` can be used to evaluate $\Psi(x)$ and $\Psi'(x)$. [2 marks]

3. For $\frac{\partial^2 l}{\partial^2 N}$, compare the absolute difference between the two Hessian evaluations with the bound given in Lecture 5 for 2nd order difference approximations, assuming `optimHess` used $h = 0.0001$.
   Hints:
   Approximate bounds L_0 and L_1 can be obtained by evaluating $l(N, \theta)$ and its derivative $\frac{\partial l}{\partial N}$ at the mode.
   For L_4, use

```
L4 <- abs(sum(psigamma(N - Y + 1, 3)) - 2 * psigamma(N + 1, 3))
```

   which is the absolute value of the fourth derivative. [2 marks]

4. Use the `microbenchmark` function in the `microbenchmark` package to compare the computational costs of the numerical Hessians from `optimHess()` and from `myhessian()`. Comment on the result. [2 marks]

Question 3: (Archaeology, part 3)

1. Write a function `arch_boot(param,J)` to produce J parametric bootstrap samples of parameter estimates, for the model in Question 1. The output should be a matrix of size $J \times 2$. Since the Binomial model requires an integer value for N, use $\lfloor \widehat{N} \rfloor$ instead of $\widehat{N}$. Start the maximum likelihood optimisations at $N = 2\max(y_1^{(j)}, y_2^{(j)})$, $\theta = 0$. [4 marks]

2. Use the output from a call to `arch_boot` to estimate the bias and standard deviations of the estimators for N and θ. Use $J \geq 10000$. Comment on the bias and std.dev. for $\widehat{N}$. [3 marks]

3. The bootstrap distribution for N is very skewed. Construct bootstrap confidence intervals for N and ϕ, under the assumption that the Bootstrap Principle holds on the $\log(N)$ and $\theta = \text{logit}(\phi)$ scales. Compare the interval for N with the one from Question 1.3.
   Hint: Construct the bootstrap intervals on one scale, and then transform the bounds to the target scale. [3 marks]

Question 4: Scottish temperature cross validation score bias and variability

The task is to estimate the bias and variance in the Cross-validation procedures from Lab 6, by using Bootstrap resampling.

Look at the file `CWB2019code.R` and focus on the three functions `read_TMIN_data`, `data_list_resample`, and `cwb4_scores`. The first function reads the training and test data from CWA, and should only be called once. The second function does resampling with replacement from the combined data set, and uses that to produce a new pair of training and test data sets. Note: Semi-parametric residual resampling would have been an alternative, but we will not investigate that here.

Denote the cross-validation score estimate from the training data by $\widehat{S}_0^{\text{train(K)}}$, and the test data score by $\widehat{S}_0^{\text{test}}$, and define $S^{\text{test}} = \mathrm{E}[\widehat{S}_0^{\text{test}}]$.

1. Explain the purpose of each of the four *Steps* in `cwb4_scores`. [4 marks]

2. Use the three functions to generate $J \geq 20$ bootstrap samples of the scores, for $K = 10$. The samples should be stored in two `data.frame` objects, initialised with

```
S_boot_train <- data.frame(SE = numeric(J),
                           DS = numeric(J),
                           Brier = numeric(J))
S_boot_test <- data.frame(SE = numeric(J),
                          DS = numeric(J),
                          Brier = numeric(J))
```

   For the results, show only the output from `head(S_boot_train)` and `head(S_boot_test)` in the report. [2 marks]

3. Compute the sample means and standard deviations for each of the three score types, separately for the training and test scores.
   Hint: See `cwb4_scores` for inspiration for how to do the computations. [1 marks]

4. Use the appropriate score differences to estimate the bias $\mathrm{E}[\widehat{S}_0^{\text{train(K)}} - S^{\text{test}}]$, and the standard deviation of $\widehat{S}_0^{\text{train(K)}}$. Discuss the results. When applied to the original data, the cross-validation scores exceed the test scores. Does this agree with the results here? Do the biases or the variability dominate the score estimates? [3 marks]