Statistical computing MATH10093
Coursework A 2019/20

Finn Lindgren

4/2/2020

Summary

Handout Tuesday 4/2/2020, handin as pdf via Learn by noon Tuesday 25/2/2020. 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
- Ordinary text should not be typeset as R comments
- The code in the `CWA2020code.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 unneccessarily repeating identical code, for example when adding to a previous plot, use the `pl + new_stuff()` technique for `ggplot()`

Problem description

The aim is to build and assess statistical models of material use in a 3D printer. The printer uses rolls of *filament* that gets heated and squeezed through a moving nozzle, gradually building objects. The objects are first designed in a CAD program (Computer Aided Design), that also estimates how much material will be required to print the object. The data file `filament.csv` contains the data needed for model estimation and assessment, one 3D-printed object per row. The columns are

- `CAD_Length`: The CAD-estimated required filament length (metres)
- `CAD_Weight`: The CAD-estimated required filament weight (gram)
- `Actual_Weight`: The measured actual object weight (gram)
- `Material`: The material colour
- `.copy`: Index for duplicate rolls; mostly 1, but can be 2 if more than one filament roll of a given material colour is present in the data set
- `Date`: The date the object was printed.

- `Class`: Either `"obs"` or `"test"`, classifying each observation according to whether it should be treated as an observation available for model estimation, or one only available for model assessment.

Use `read.csv()` to load the data into an R object called `filament`, with additional argument `stringsAsFactors = FALSE`. Suggested setup code chunk:

```
source("CWA2020code.R")
suppressPackageStartupMessages(library(tidyverse))
theme_set(theme_bw())
filament <- read.csv("filament.csv", stringsAsFactors = FALSE)
```

Tasks

1. Plot the CAD and Actual weight data, with matching colours for each material.
   Hint: use `scale_colour_manual` to specify the colours.

2. Unlike lab 3, the `model_Z` function in `CWA2020code.R` now takes more arguments, allowing more flexibility in how to define the models. Define a function `model_estimate` that takes parameters `formulas`, `data`, `response`, where `formulas` and `data` have the same interpretation as the input arguments to `model_Z`, and `response` is a character/string variable naming the response (measured outcome) variable column of the data.

   The function should estimate the model by numerical optimisation, and return a list with three named elements: `theta`, `formulas`, and `Sigma_theta`, containing the information needed to call the `model_predict` function.

   In the following tasks, each model estimation result should be stored in variables with different names so they can be accessed by later tasks if needed.

3. Estimate a model for `Actual_Weight` with an intercept and covariate `CAD_Weight` for the model expectations, but only an intercept for the model (log-)variances. Only use the `Class == "obs"` observations for model estimation.

4. Plot the test data together with the prediction intervals for the estimated model from task 3.

5. Now estimate a model that uses an intercept and covariate `CAD_Weight` formula for both the model expectations and (log-)variances.

6. Plot the test data together with the prediction intervals for the two estimated models from tasks 3 and 5.

7. Compute the Squared Error, Dawid-Sebastiani, and Interval scores ($\alpha = 0.1$) for the two estimated models. Consider the collections of pairwise score differences of each score type. Do the scores agree about which model seems better?

Hint: Write a function `model_scores` with suitable input and output parameters to avoid unneccessary code repeating.

8. The printer user wonders if the CAD system is equally (or bad) at predicting the weight of all the different materials. Estimate a model with material dependent `CAD_Weight` coefficients for the expectation part of the model, and compare the prediction scores with those from task 7.
   Hint: Find out what the *interaction* syntax `A:B` does in a model formula, when `A` is a factor and `B` is a continuous variable.

9. Often, we want to assess predictions of $z = 0$ or 1 *event indicator* variables, expressed as the probability $p_F = \mathrm{P}_F(z = 1) = \mathrm{P}_F(\text{The event occurred})$. The *Brier score* for such prediction can be written as the same as a Squared Error score, since $p_F = \mathrm{E}_F(z)$:

   $$S_{\text{Brier}}(F, z) = (z - p_F)^2$$

   Under the assumption that the predictive distributions are well approximated by Gaussian distributions, compute and plot the probabilities for the event that "more than 10% extra weight is needed compared with `CAD_Weight`" for the models from tasks 3, 5, and 8 as a function of `CAD_Weight`. Compute and compare the Brier scores for the test data for the events.

10. Simulate N observations y from a Cauchy distribution (see `rcauchy` and related functions) with `location=2` and `scale=5`, for $N = 5$, 10, 20, 40, etc. Estimate the model parameters using numerical maximum likelihood estimation, and compute the average Brier scores for event indicators $z = \mathbb{I}(y \leq 0)$ based on prediction probabilities from the estimated models, and compare with the average Brier scores based on prediction from the true model. Do the parameter estimates and score differences stabilise? Should we expect a similar comparison for Squared Error (with respect to `mu=location`) or Dawid-Sebastiani (with respect to `mu=location` and `sigma=scale`) scores to stabilise? Why/why not?