End-term project: Advanced Statistical Computing 2020

Instructions

In this project you will use nothing but code to solve a challenging modeling problem in insurance. The tasks below walk you through the modeling process. Do not hand in a point-by-point reply, however. Your submission should have the form of a coherent report.

Form

The following would be a suitable structure:

1. *Introduction*
2. *Methodolgy*
3. *Simulation study*
4. *Results*

All figures should have explanatory captions. Be concise in your writing, but provide enough details such that a fellow student seeing this for the first time would understand what's going on. In particular,

- clearly state each algorithm that you use (and provide a reference to other packages in case you use them),
- clearly specify the method/parameters you use to obtain your results (like bootstrap type, number of replications, etc.).

Submission

To hand in your submission, upload the following documents on Brightspace:

- a pdf file of your written report,
- the code you used to complete the assignment (preferably as .Rmd).

All results and figures in the written report must be reproducible from the code.

Tips for R markdown

You can specify options of a code chunk by starting it with `{r, option1 = v1, options = v2, ...}`. Here are a few useful ones:

- `fig.cap="A caption"`: adds captions to a figure.
- `include=FALSE` hide code and output from document.
- `echo=FALSE` hides only the code, but not the output.

See here for more options.

The problem

Context

ANV is an insurance company focusing on corporate clients. It offers different types of insurance policies (called business lines). In the last year two of the business lines were simultaneously affected by a huge claim related to a single client. The two business lines are:

- Professional liability insurance (PLI)
- Workers' compensation (WC)

The problematic client holds policies from both business lines and messed up big time.

To protect themselves from such risks in the future, ANV approaches a *reinsurance* company. Such companies offer insurance policies for other insurance companies. ANV has a rather specific idea how this policy should look like: For some threshold $t = 100, 110, \ldots, 200$

- If $PLI + WC \leq t$, ANV pays the claim themselves.
- If $PLI + WC > t$, the reinsurance company pays the claim.

Depending on the threshold t, the reinsurance company asks the following price for the insurance: $P(t) = 40\,000 \exp(-t/7)$ (in million euros). ANV is only willing to buy the policy if its expected value

$$V(t) = E[(PLI + WC)1(PLI + WC > t)]$$

exceeds the price $P(t)$. To make a decision, ANV want to use statistical modeling to approximate $V(t)$.

The data

The file `insurance.csv` contains data from clients that occurred losses in both PLI and WC business lines. There are three columns:

- `ID`: client ID,
- `PLI`: loss incurred for PLI (in million euros),
- `WC`: loss incurred for WC (in million euros).

The model

To simplify notation, define $X_1 = \text{CLI}$ and $X_2 = \text{WC}$. To approximate $V(t)$, we need a model for the joint distribution F_{X_1,X_2} of both business lines. It is particularly important to adequately reflect the dependence between lines.

Copula models are very popular in such situations. In a copula model, the joint density f_{X_1,X_2} is decomposed into

$$f_{X_1,X_2}(x_1, x_2) = f_{X_1}(x_1) f_{X_2}(x_2) c\big(F_{X_1}(x_1), F_{X_2}(x_2)\big),$$

where $c(u_1, u_2)$ is called *copula density* and induces the dependence between the marginals f_{X_1} and f_{X_2}. The function c is the joint density of the probability integral transforms $U_1 = F_{X_1}(X_1)$ and $U_2 = F_{X_2}(X_2)$.

Preliminary experiments suggested the following parameteric models:

- $f_{X_1}(\cdot; \mu_1, \sigma_1) \sim \text{Lognormal}(\mu_1, \sigma_1)$, $\mu_1 \in \mathbb{R}, \sigma_1 > 0$.
- $f_{X_2}(\cdot; \mu_2, \sigma_2) \sim \text{Lognormal}(\mu_2, \sigma_2)$, $\mu_2 \in \mathbb{R}, \sigma_2 > 0$.
- $c(\cdot; \theta) \sim \text{Joe}(\theta), \theta \geq 1$.

Tasks

1. Give a high-level introduction to the scientific problem in your own words. Illustrate the dependence in the data with a graph.
2. Give a precise mathematical specification of the model (similar to the above).
3. Explain how maximum likelihood estimation can be used to estimate the parameters of $f_{X_j}, j = 1, 2$. When implementing the method, try to come up with sensible starting parameters (depending on the input data) to speed up optimization.

4. If the probability integral transforms $U_j = F_{X_j}(X_j), j = 1, 2$ were observed, we could also estimate the copula parameter θ by MLE. However, we might use *pseudo-observations* $\hat{U}_j = F_{\hat{\mu}_j, \hat{\sigma}_j}(X_j), j = 1, 2$ and use those for estimation. Write a function that fits the parameter of a Joe copula model. Explain why this might work.

5. Now write a function that estimates all model parameters as follows:

   i. Compute estimates $(\hat{\mu}_1, \hat{\sigma}_1)$ by maximum likelihood.
   ii. Compute estimates $(\hat{\mu}_2, \hat{\sigma}_2)$ by maximum likelihood.
   iii. Set $\hat{U}_j = F_{\hat{\mu}_j, \hat{\sigma}_j}(X_j)$, $j = 1, 2$ and estimate the parameters of a Joe copula model for $(\hat{U}_1, \hat{U}_2)$.

   To compute the density of the Joe copula, use

```
library(copula)
# Joe(theta)-density evaluated at u; u is a (n x 2) matrix
dCopula(u, joeCopula(theta))
```

6. Write a function that simulates from the joint model for (X_1, X_2) for a given set of parameters $(\mu_1, \sigma_1, \mu_2, \sigma_2, \theta)$. To simulate `n` samples from a Joe copula with parameter θ, you can use `rCopula(n, joeCopula(theta))`.

7. Generate simulated data from an estimated model (from step 4). Compare the simulated data with the observed data. If your implementation are correct, the two data sets should look similar. Illustrate how the properties of the data change when you increase/decrease a parameter.

8. We want to better understand the inner workings of our method for parameter estimation. To do so, we conduct a simulation study. Fix $\mu_1 = 1, \sigma_1 = 2, \mu_2 = 3, \sigma_1 = 0.5, \theta = 2$. For $r = 1, \ldots, 100$:

   i. Simulate n observations $(X_{i,1}, X_{i,2})$, $i = 1, \ldots, n$ from the joint model.
   ii. Fit the model parameters while tracking the time it takes (for example using `system.time()`).

   Then compute the RMSE for each parameter and method. *(This will run for a while, so better test your code with less replications first.)*

   Repeat the whole procedure for $n = 200, 500, 1000$ and plot the RMSE and average computing time as functions of n. If your implementation is correct, the RMSE should be decreasing in n. Which parameter values seem easier/harder to estimate? Can you explain why? How does the computation time scale with respect to the sample size?

9. Now fit all model parameters to the observed data. Explain how to compute the expected payout of the reinsurance $V(t) = E[1(X_1 + X_2 > t)(X_1 + X_2)]$, $t \in \mathbb{R}$, using Monte Carlo simulation. Make a graph of $V(t)$ for $t = 100, 110, \ldots, 200$ based on 10^5 Monte Carlo samples.

10. You will see that the the $V(t)$ values are quite noisy because we're dealing with highly improbable events. Compute the values again but this time using importance sampling. (*Hint: generate data from the same model but change the parameters such that the events of interest become more likely.*)
    Plot $V(t)$ again for $t = 100, 110, \ldots, 200$ and compare it to the price $P(t)$ asked by the reinsurance company. At which value of t should the company buy the reinsurance policy?

11. Because our model parameters are estimated, $V(t)$ is only an estimated quantity (even when ignoring the MC approximation error). Use a bootstrap method to compute 80% confidence intervals for $V(t)$. Explain your implementation and why it does/doesn't account for a) estimation error, b) MC approximation error. Add the confidence intervals to the plot in 10. Does that change your recommendation in 10?