Risk Modelling in Insurance, Final project

Academic Year 2020-2021

KU Leuven and University of Ljubljana

Instructions for the final project

You should provide answers in English. Motivate all your answers extensively, you get points for the methods you use and for your clear explanation of them, not just for the final answer.

Success!

Deliverables for the final project

Please hand in on or before February 2, 2021 (end of day) via Canvas:

1. A report (format: html or pdf file) in which you answer all the *Assignment Questions* stated below and which contains all the figures you are asked to create. We encourage you to use R markdown! We value and will grade your writing, the structure and the lay-out of your report. You should carefully think about the structure and the lay-out or template used for your report, starting from the documentation available on Canvas.
2. An R script or R markdown file that you have used for all your calculations. Your code should be well-organized and easy to read: if an answer in your report is correct but we cannot follow your calculation then you will receive no points for that calculation.

Please submit one report per team. You can work in teams of 2-3 students.

Questions

In the **first** part of the project you will fit various parametric models to loss data with claim frequencies and you will compare the model fits. The file `NonFleetCo507.txt` contains information on 159 947 insurance contracts.

- The column `Clm_Count` shows how many claims were filed on the contract. This is the loss frequency data that we will use for model fitting.
- The column `TLength` shows the period of exposure during which the contract was active and the `Clm_Count` was filed.
- The other columns show characteristics of the policyholder and the insured vehicle. We will **not** use those in this assignment.

1. We start by reading the data into R. Describe the structure of the data set, and the empirical distribution of `Clm_Count`. What is the empirical claim frequency and variance? How do you calculate these when taking exposure information `TLength` into account?

   We will fit several count distributions to the observed claim count data: the Poisson, the Negative Binomial, the Zero Inflated Poisson and the Hurdle Poisson.

2. We start by fitting the Poisson distribution to the loss frequency data. Compute the log-likelihood function for this model, taking exposure into account, and maximize it numerically to find the MLE of the (annual) mean parameter. Compute (numerically) the corresponding standard error. Also compute the Akaike Information Criterion (AIC) for this model, for later comparison with other models.

   Note: we will define a function which constructs the negative log-likelihood function. The log-likelihood should be seen as a function of the unknown parameter (vector), which depends on the observed claim counts and exposures. We define the negative log-likelihood since Newton-type algorithms exist in R for minimization problems. Minimizing the negative log-likelihood is equivalent to maximizing the likelihood. In order for the optimization problem to be unconstrained, we set $\lambda = \exp(\beta)$ and optimize for β. This ensures that the (annual) mean λ of the Poisson distribution will be positive for any real value of β. We use the non-linear minimization function `nlm` to carry out the minimization. This routine requires a starting value, which is here simply set to 1. The function returns a list which contains the parameter value for which minus the log-likelihood is minimized as well as the hessian at the estimated minimum. In general, the covariance matrix of the maximum likelihood estimators can be estimated by the inverse of this hessian of minus the log-likelihood at the estimated minimum. By taking the square root of its diagonal elements, we obtain the estimated standard error of each parameter estimate.

3. You will now verify your solution for question 2 (where you computed the log-likelihood function yourself and then did the maximization numerically) with the `glm` function in R. `glm` fits Generalized Linear Models (GLMs). We put focus on the Poisson regression model, as one specific example of a GLM. In such a model the response N, the number of claims reported, is Poisson distributed as follows

$$\begin{aligned} N &\sim \text{POI}(\mu) \\ g(\mu) &= \log(\text{exposure}) + x'\beta. \end{aligned}$$

   In this set-up $g(.)$ is the so-called link function, transforming the mean μ. The log (exposure) is an offset term; it is part of the linear predictor but the corresponding regression parameter β is fixed at 1. Using a logarithmic link function in the above model specification, we obtain the following model

$$\begin{aligned} N &\sim \text{POI}(\mu) \\ \mu &= \exp(\log(\text{exposure}) + x'\beta) \\ \mu &= \text{exposure} \cdot \exp(x'\beta). \end{aligned}$$

   In R you will fit such a model via the `glm` function by completing the following instructions:

```
fm_pois <- glm(... ~ ..., family=poisson(link="log"), offset=...)
summary(fm_pois)
```

Fill in the dots so that `Clm_Count` is modelled with a Poisson distribution, taking exposure to risk into account. Compare your result with the result from question 2.

4. Repeat question 2 and 3 for Negative Binomial, Zero Inflated Poisson and Hurdle Poisson. That is, compute the log-likelihood function and find the MLEs of the parameters for each of these models. Also compute the AIC for each model. Then verify your calculations with the results of functions directly available in R. Explain how you do this in R.

   Note: use the Negative Binomial fit `glm.nb` from the `MASS` package, and the `zeroinfl` and `hurdle` functions from the `pscl` package. Install and load those packages for this exercise. Fill in the dots and explain what is going on:

```
fm_nb <- glm.nb(... ~ ... + offset(...), link=log)
summary(fm_nb)

fZIP <- zeroinfl(... ~ ..., offset=..., dist= "...")
summary(fZIP)

fhurdle <- hurdle(... ~ ..., offset=..., dist= "...",
                              zero.dist=c("binomial"))
summary(fhurdle)
```

5. Compare the AIC values for the parametric models considered above. Which model gives the best fit according to AIC?

6. We will now compare the frequency models by comparing the expected number of zeros with the actually observed number of zero claims. What do you conclude? Which model is your overall recommendation for this data set on frequency data?

   Note: the expected number of zeros can be written as $E[\sum_{i=1}^{n} I_i]$ where indicator I_i equals 1 if observation i is zero and 0 otherwise. Work out this expected value to see that the expected number of zeros in each of the fitted models can be obtained by summing the probability of a zero over all the observations.

   - **Poisson**: applying `fitted` to the Poisson `glm` object returns the fitted mean for each observation which we combine with `dpois` at zero.
   - **NB**: applying `fitted` to the `negbin` object returns the fitted mean for each observation which we combine with `dnbinom` at zero and the common size / dispersion parameter from the NB fit.

- **ZIP**: applying `fitted` to the `zeroinfl` object only returns the fitted means for each observation, so we use `predict` instead with `type = "prob"`. This returns a matrix containing the probabilities for each observation (in the rows) to equal 0, 1, 2 and so on. Selecting the first column gives the probabilities of a zero for each observation. Note that using argument `type = "zero"` is not the same: these contain only predicted probabilities for the zero component (and not the count component).
- **Hurdle**: similar to ZIP.

In the **second** part of the project, you will fit various parametric models to censored/truncated loss data and compare results. The file `SeverityCensoring.txt` contains information about 9062 claims paid by an insurance company over some observation period.

- The column `claimAmount` shows how much the insurance company paid on each claim. This is the loss data that we will use for model fitting.
- The column `deductible` shows that there is a fixed deductible of 100 EUR for each policy, which means that all the observed claim amounts are truncated from the left at 100.
- The column `rc` shows whether right-censoring is present or not: `NA` indicates that the claim is fully settled by the end of the observation period and therefore the observed claim amount is the full (uncensored) loss associated with this claim. On the other hand, a number in the `rc` column indicates that the claim is not yet fully settled, so the observed claim amount is right-censored.

1. We start by fitting an exponential distribution to the loss data. Compute the negative log-likelihood function for this model, taking truncation and censoring into account, and minimize it numerically (using `nlm` or `optimize`) to find the MLE of the rate parameter. Also compute the Akaike Information Criterion (AIC) for this model, for later comparison with other models.

2. Repeat exercise 1 for lognormal, inverse Gaussian and Burr distributions. That is, compute the negative log-likelihood function under truncation and censoring, and minimize it numerically (using `nlm` or `optim`) to find the MLEs of the parameters in each model. Also compute the AIC for each model.

   Note: For best results in numerical optimization, try to pick "reasonable" initial values for parameters whenever you can, e.g. using method of moments. If a parameter θ is constrained to be positive, it is better to enter it as $\exp(\beta)$ in the log-likelihood function and optimize over unconstrained β. Also: the pdf and cdf of the inverse Gaussian and Burr distributions are provided in the `actuar` package, which should be installed and loaded for this exercise.

3. Next, we want to fit an Erlang mixture distribution with 5 components to the loss data. Recall that this distribution has a pdf of the form

$$f(x;\alpha_1,\ldots,\alpha_5,r_1,\ldots,r_5,\theta)=\sum_{j=1}^{5}\alpha_j\frac{x^{r_j-1}e^{-x/\theta}}{\theta^{r_j}(r_j-1)!},\quad x>0.$$

We will make use of the EM (Expectation-Maximization) algorithm to estimate parameters. Thankfully, the R code for implementing the EM algorithm for Erlang mixtures was developed in Verbelen et al. (2015) and is available for our use. Download the file EM_MixedErlang.R into your working directory and run:

```
source("EM_MixedErlang")
loss <- ... ; nrc <- ...
fit.ME <- ME_fit(loss, nrc, trunclower=100, M=5, s=3)
```

Fill in the dots so that loss is the vector of 9062 losses as provided in the claimAmount column, and nrc is the same as loss, but with censored loss amounts replaced by NA's. This will take a few minutes to run, and will produce some warning messages that you can ignore.

Inspect the object fit.ME, identify the MLEs for the five weights $\alpha_1, \ldots, \alpha_5$, the five shape parameters $r_1, \ldots, r_5$, and the scale parameter θ. Also identify the AIC for this model.

4. Plot the Kaplan-Meier estimate of the survival function for the loss data, by installing the package survival and then running

```
deds <- ... ; loss <- ... ; full <- ...
fit <- survfit(Surv(deds, loss, full) ~ 1)
plot(fit, mark.time=F, conf.int=F)
```

Fill in the dots so that deds is the vector of the 9062 deductibles (all equal to 100) as provided in the deductible column, loss is the vector of the 9062 losses as provided in the claimAmount column, and full is a logical vector of length 9062 that has a TRUE for non-censored (full) losses and FALSE for censored losses.

Note: For more information regarding survival analysis in R using the survival package, you can check out this online tutorial.

5. Add the plots of the best-fitting (i) exponential, (ii) lognormal, (iii) inverse Gaussian, (iv) Burr and (v) Erlang mixture survival functions to the Kaplan-Meier plot. Recall that we have a left-truncation at 100, so you should plot the curve $\frac{1-F(x)}{1-F(100)}$ for each of the five models, with F denoting the cdf with the best-fitting parameters. Which of the five parametric models seems closest to the Kaplan-Meier estimate?

   Note: The mixed Erlang cdf can be computed as ME_cdf(x, theta, shape, alpha).

6. Compare the AIC values for the five parametric models considered above. Which model gives the best fit according to AIC? Is this consistent with your answer to exercise 5?

Roel Verbelen, Lan Gong, Katrien Antonio, Andrei Badescu, and X. Sheldon Lin. Fitting mixtures of Erlangs to censored and truncated data using the EM algorithm. *ASTIN Bulletin*, 45(3):729–758, 2015.