DAV/DGVFM
Jahrestagung
2022

Micro-level Prediction of Outstanding Claim Counts using Neural Networks

Alexander Rosenstock
ARAG SE

DAV
DEUTSCHE
AKTUARVEREINIGUNG e.V.

DGVFM
DEUTSCHE GESELLSCHAFT
FÜR VERSICHERUNGS- UND
FINANZMATHEMATIK e.V.

DAV/DGVFM-Jahrestagung, 27.–29. April 2022

Problem Statement

Given a non-life insurance portfolio, what are the losses I've already incurred for each individual policy?

Goals

- Obtain individual-level IBNR and RBNS reserves to inform by-case decision making.
- Give more accurate reserve estimates for better risk management.
- Quickly detect and adapt to changes in development process.
- Minimize manual calibration effort to save time.

Solution: A micro-level model

- This neccessitates a micro-level approach to modelling insurance claims.
- The RBNS reserve case has been widely studied already.
- We focus on a micro-level claim count model for now.

Our Approach

Several components go hand-in-hand to deliver a complete micro-level claim count model

1. A formal process model for occurrence and reporting of claims.
2. A new flexible distribution family applicable to reporting delays.
3. A neural-network regression approach to fit the flexible distribution model to randomly truncated data.

New approach outperformed Chain Ladder by 50% on real data!

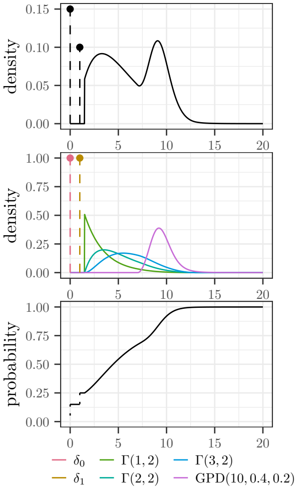
A Claim Process Model

For each policy indexed by i, having features $x^{(i)} \in \mathfrak{X}$ – among which the coverage period $C^{(i)} = [t^{(i)}_{\text{start}}, t^{(i)}_{\text{end}}]$, we have a *position-dependent marked poisson process* with intensity $\lambda(x^{(i)}, t)$

$$\xi^{(i)} = \sum_{j=1}^{N^{(i)}} \delta_{(T_j^{(i)}, Y_j^{(i)}, D_j^{(i)})}$$

1. $N^{(i)} \sim \mathrm{Poi}\Big(\int_{C^{(i)}} \lambda(x^{(i)}, t)\,\mathrm{d}t\Big)$: number of incurred claims
2. $T_j^{(i)} \in C^{(i)}$: occurrence dates of these claims
3. $Y_j^{(i)} \in \mathfrak{Y}$: features of these claims (e.g. type)
4. $D_j^{(i)} \in [0, \infty)$: reporting delay, the time between $T_j^{(i)}$ and the filing of the claim with the insurer.

BDEGP: A suitable semi-parametric family $\mathcal{F}$



$\mathrm{BDEGP}(n, m, \kappa, \varepsilon)$

- n *Dirac* components at $0, \ldots, n-1$.
- m translated *Erlang* components with common scale.
- $\mathrm{GPD}_{\mu=\kappa}$ *generalized Pareto* tail.
- continuous density, by blending the GPD tail with the erlang mixture body on $(\kappa - \varepsilon, \kappa + \varepsilon)$.

A Reporting Delay Model

Find a function $g \in \mathcal{G}$ and a distribution family $\mathcal{F}$ on $[0, \infty)$ with parameter space Θ such that the reporting delay distribution of all claims is described by

$$P_{D|X,T,Y} = F_{g(x,t,y)}; \qquad g : \mathfrak{X} \times \bigcup_i C^{(i)} \times \mathfrak{Y} \to \Theta$$

$\mathcal{G}$ will be a family of neural networks and $\mathfrak{X}, \mathfrak{Y}$ contain all necessary policy-level and claim-level information.

Complete Method

1. Fit a BDEGP-Family to the complete reporting delay distribution.
2. Train a neural network to predict the reporting delay distribution parameters for each individual reported claim.
3. Compute the probability that the claim was reported by time τ in the first place.
4. Use

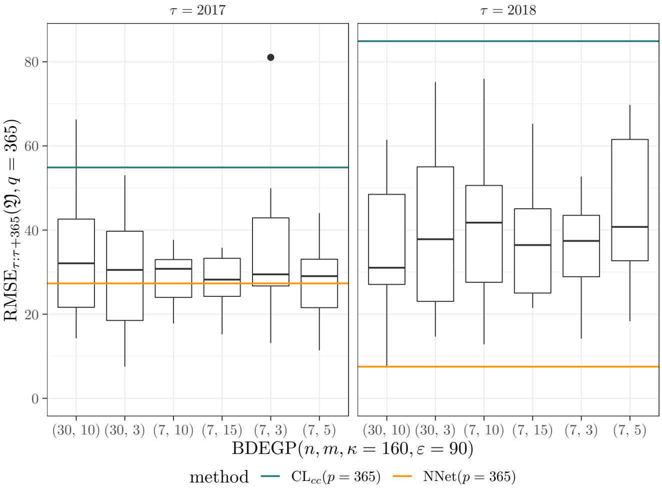
$$\hat{N} := \frac{1}{P(T + D_j^{(i)} \leq \tau | X = x^{(i)}, T \in I(t_j^{(i)}), Y = y_j^{(i)})}$$

as a predictor for the total number of claims[1].

[1] $I(t)$ is a suitably small region around the accident time t.

Proven on real data

New approach outperformed Chain Ladder by 50% on real data!



Outlook

The proposed predictor can be improved even more by completing the process model, i.e. by fitting a distribution to $Y_j^{(i)} | X^{(i)}, T_j^{(i)}$ and estimating the claim frequency $\lambda(x^{(i)}, t)$ to:

$$\begin{aligned}\hat{N}^* &:= \text{reported claims} + \int_{C^{(i)}} \lambda(x^{(i)}, t)\,\mathrm{d}t \cdot P(T + D_j^{(i)} > \tau | X = x^{(i)}) \\ &= \text{reported claims} + \text{expected IBNR count}\end{aligned}$$

Implemented as an R package

We implemented an **R** package that

- Can fit distributions to randomly truncated data.
- Supports the new BDEGP family (among many others).
- Integrates with **TensorFlow**.

https://github.com/AshesITR/reservr

```
R> remotes::install_github("AshesITR/reservr")
```

Preprint available on SSRN: https://dx.doi.org/10.2139/ssrn.3949754