From claim counts to interarrival times using a small neural framework

Anne van der Scheer

Perunum
ACTUARIEEL ADVIES

Insurance Data Science Conference, London
19 - 20 June 2025

Outline

- From claim counts to interarrival times
- Time-to-event neural framework
- Example Weibull distribution
- Conclusions
- References

From claim counts to interarrival times

Introduction
Claim counting is deeply rooted in actuarial modelling, especially with the common use of Generalised Linear Models (GLMs) and distributions including Poisson and Negative Binomial. The growing use of neural networks often comes with a one-to-one translation of extensions (e.g. zero-inflation) of traditional models.

In the context of assumption-free machine learning, it is natural to consider also the undisclosed data: the exact occurence dates of claims. Then the next step is to model the claim interarrival times instead of the aggregated counts. This leads to the field of time-to-event (or survival) analysis. For our goal it is important to have a generic model.

From claim counts to interarrival times

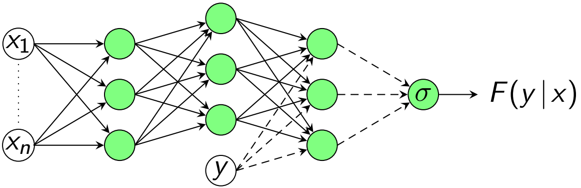
Choosing a neural framework
In recent years, several neural models have been introduced. We mention DeepSurv and DeepHit and refer to the work of Katzman et al. (2018) respectively Lee et al. (2018). DeepHit offers competing events without the restriction of proportional hazards (DeepSurv), but the modelled time is discrete where for actuarial modelling a continuous model is preferred.

Therefore, we introduce an alternative, building on a very simple but effective neural framework for modelling a continuous cumulative distribution function (CDF). We combine this model with a model for categorical distribution to derive at a time-to-event model. The choice for modelling a CDF turns out to be convenient for the implementation of censoring.

Time-to-event neural framework

Component 1: continuous distribution

For a continuous dependent variable we model the cumulative distribution function (CDF), following an architecture of Chilinski (2020):



- the neuron in the last layer has sigmoid activation, other neurons have tanh activation
- dashed arrows indicate non-negative weights to ensure a non-decreasing CDF for all input values

Time-to-event neural framework

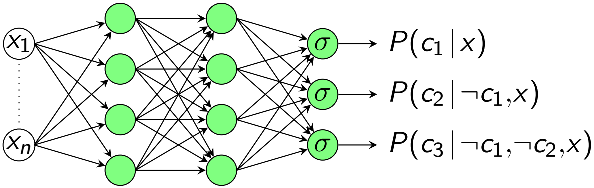
Why modelling the CDF instead of the probability density function (PDF)?

- CDF has easy characteristics: smooth increasing function between 0 and 1 is sufficient
- quantiles can be calculated from the monotone increasing F^{-1} by using binary search
- random sampling is possible by calculating $F^{-1}(\text{rand})$
- calculation of mean, variance, skewness etc. can be calculated from a derived set of random or "systematic" quantiles, using uniformly distributed fixed sets between 0 and 1

Time-to-event neural framework

Component 2: categorical distribution

For a categorical dependent variable, we model the PDF with the following architecture:



- the neuron in the last layer has sigmoid activation, other neurons have tanh activation
- the outcome y in the above example can take four values: c_1, c_2, c_3 and c_4

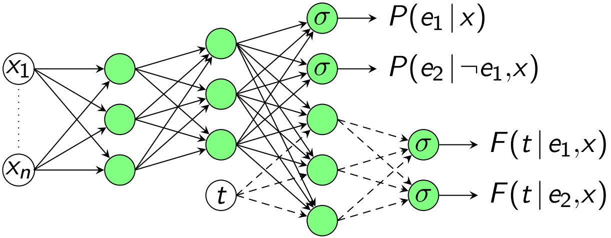
Time-to-event neural framework

- the neurons in the last layer are ordered as a sequential logit and can each separately take any value between 0 and 1
- for two categories the model is equal to the logistic network
- the chosen approach is a natural extension of two categories
- for more than two categories the standard literature uses unconditional probabilities in combination with the softmax function to ensure that the sum of the probabilities equals 1. Our network only needs sigmoid activations.

Time-to-event neural framework

Components 1+2: time-to-event distribution

For the joint distribution of time and events, we combine the earlier models:



Each (winning) event e_i has its own sub-CDF. The joint distribution can be written as

$$f(t, e_i \,|\, x) = P(e_i \,|\, x) \cdot f(t \,|\, e_i, x)$$

Time-to-event neural framework

Notice that the PDF and hazard functions can be directly derived from the network.

Left-, right- and interval censored observations contribute to the likelihood in the following way:

- in case of event e_i in (t_1, t_2) :
  $P(e_i \mid x) \cdot (F(t_2 \mid e_i, x) - F(t_1 \mid e_i, x))$
- no event in $(-\infty, t)$:
  $1 - \sum_i P(e_i \mid x) + \sum_i P(e_i \mid x) \cdot (F(\infty \mid e_i, x) - F(t \mid e_i, x))$

In practice, the function F does not have exact limits 0 and 1. This explains the formulation of the expression in the case of no event.

In case of event e_i at time t we just use $P(e_i \mid x) \cdot f(t \mid e_i, x)$.

Example Weibull distribution

In a synthetic example, we show the application of the time-to-event framework on unit time-interval (0,1) with only one (claim) event:

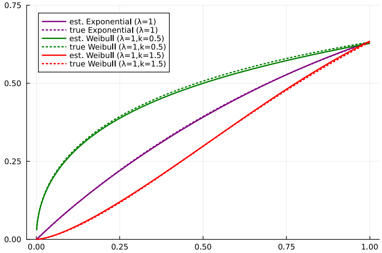
- the population exists of 3 categories with each 25,000 policies having different distributions of interarrival times
- category 1: $t \overset{iid}{\sim} \text{Exp}(\lambda = 1)$
- category 2: $t \overset{iid}{\sim} \text{Weibull}(\lambda = 1, k = 0.5)$
- category 3: $t \overset{iid}{\sim} \text{Weibull}(\lambda = 1, k = 1.5)$
- extra rule for category 2 and 3: at every event, the scale parameter λ is multiplied by 1.05
- for each policy: as long as total time < 1, add a sampled observation to the data, otherwise, add a final right-censored observation to the data

Example Weibull distribution

We utilise the neural framework with the following setup:

- 5 layers with a total of only 18 neurons
- covariates: hot-valued categories and number of past claims
- map unit time interval to $(-\infty, \infty)$ with $t \to \log(\frac{t}{1-t})$

CDF first arrival time

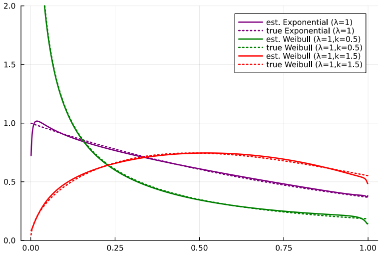


The network reproduces the different distributions accurately.

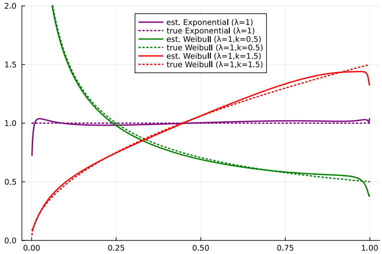
Example Weibull distribution

The PDF and hazard functions for the first arrival are also accurately reproduced for all categories:

PDF first arrival time



Hazard function first arrival time



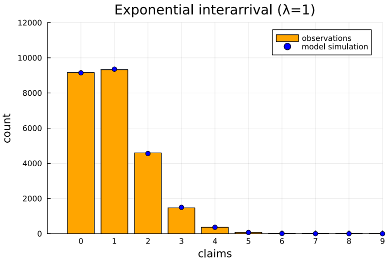
There are "glitches" in the estimated functions near t=0 and t=1, due to the mapping of time on $(-\infty, \infty)$.

Example Weibull distribution

We return to claim counts and verify if the network can reproduce the original data by simulating the population 100 times and take the average counts as outcome.

The claims distribution of category 1 is Poisson distributed, because of the exponential interarrival time.

Claims distribution category 1

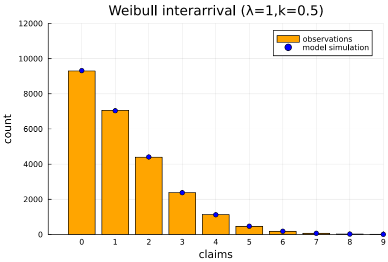


Example Weibull distribution

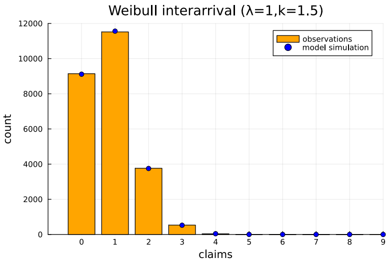
The claims distributions of category 2 and 3 are different from category 1:

- category 2 is overdispersed
- category 3 is underdispersed

Claims distribution category 2



Claims distribution category 3



Example Weibull distribution

Concluding remarks

- the modelled interarrival times are accurately reproduced for all categories
- the claim counts are also accurately reproduced (by performing simulation)
- modelling interarrival times can easily outperform traditional counting GLMs

Implementation

This project has been built in plain Julia code and can be found on https://github.com/perunum/claim-interarrival-times

Conclusions

Modelling claim interarrival times instead of (just) claim counts can offer improved accuracy. The proposed time-to-event neural framework provides a small and assumption-free solution:

- including multiple competing risks
- general left-, right- and interval censoring
- modelling sub-CDFs for every event, PDF and hazard functions can be computed directly

The framework can contribute to a more insightful modelling of claim-generating processes. By providing a more granular connection to policy terms and conditions, it has the potential to enhance pricing. Moreover, the framework is broadly applicable for time-to-event analysis.

References

Chilinski, P., & Silva, R. (2020, August). Neural likelihoods via cumulative distribution functions. In *Conference on Uncertainty in Artificial Intelligence* (pp. 420-429). PMLR.

Katzman, J. L., Shaham, U., Cloninger, A., Bates, J., Jiang, T., & Kluger, Y. (2018). DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC medical research methodology, 18,* 1-12.

Lee, C., Zame, W., Yoon, J., & Van Der Schaar, M. (2018, April). Deephit: A deep learning approach to survival analysis with competing risks. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 32, No. 1).