Supplementary Material for the Paper: “Marked Cox Models for IBNR Claims Count: Continuous and Discretized Approaches with Dirichlet-Driven Reporting Delays”

Hassan Abdelrahman[1], Andrei L. Badescu[1], Radu Cariu[1] and Sheldon Lin[1]
[1]Department of Statistical Sciences, University of Toronto, ON, Canada.

1 Simulation Study for Parameter Recovery

We conducted three simulation studies to assess the ability of the fitting algorithms for the three models to recover the true model parameters. This section details the simulation setup, assumptions, and results for the three models. The code for the three simulations can be found here: `https://github.com/HassanAbdelrahman/HMM_Simulations`

1.1 Continuous-time Model

Simulation Setup

The simulated data involves a portfolio of 1,000 individuals, each characterized by three distinct covariates:

- `x1`: A numerical variable randomly sampled from integers 1 to 50 for each individual.
- `x2`: A categorical variable with three levels: A, B and C.
- `x3`: A categorical variable with two levels: X and Y.

The simulated dataset spans $T = 100$ discrete time points, representing the observation period for each individual. For the common hidden Markov process, we assume the following parameters:

- **Initial distribution:** $\boldsymbol{\pi} = (1, 0)$, indicating the process always starts in state 1.
- **Number of hidden states:** $g = 2$.
- **Transition matrix:**

$$\boldsymbol{\Gamma} = \begin{bmatrix} 0.70 & 0.30 \\ 0.40 & 0.60 \end{bmatrix}$$

The simulation process involves generating a time series of hidden states $(c_1, c_2, \ldots, c_{100})$. These hidden states, representing the underlying state at each time point, are simulated using the specified initial distribution and transition matrix.

Given the simulated path of hidden states, we proceed to simulate count data $(n_{i,1}, n_{i,2}, \ldots, n_{i,100})$ for each individual i. This count data is a realization of a Poisson process with intensity $\log(\lambda^{c_k}(\boldsymbol{x}_i))$ at each time point k. The Poisson intensities are defined through the following regression functions:

$$\log \lambda(c_t, \boldsymbol{x}) = \begin{cases} 0.2 + 0.001 \cdot \mathtt{x1} + 0.05 \cdot 1\{\mathtt{x2} = B\} + 0.10 \cdot 1\{\mathtt{x2} = C\} - 0.05 \cdot 1\{\mathtt{x3} = Y\}, & \text{if } c_t = 1, \\ 0.4 + 0.002 \cdot \mathtt{x1} + 0.10 \cdot 1\{\mathtt{x2} = B\} + 0.15 \cdot 1\{\mathtt{x2} = C\} - 0.05 \cdot 1\{\mathtt{x3} = Y\}, & \text{if } c_t = 2. \end{cases}$$

Each individual's observed count data is generated as a realization of a Poisson process based on the simulated hidden states and covariates. The exposure term $a_l(\boldsymbol{x})$, representing the probability of reporting, is assumed constant and equal to 1 for all individuals and all time points, indicating that all claims are reported and the exposure is constant across time.

Initialization of Parameters

Before we apply our EM algorithm, we need to initialize the parameters. The parameters were initialized as follows:

- **Initial Distribution ($\boldsymbol{\pi}$):** The initial distribution of the hidden states was set to be uniform across all g states:
$$\pi = \left[\frac{1}{g}, \frac{1}{g}, \ldots, \frac{1}{g}\right].$$
- **Regression Coefficients ($\boldsymbol{\theta}$):** The regression coefficients for $\log(\lambda)$ were initialized based on the observed data:
  1. *Observed Intensity Estimation:* The average observed intensity was computed for each time point as the ratio of total observed counts to total exposure.
  2. *Clustering:* A k-means clustering algorithm was applied to group these average intensities into g clusters, with the cluster centers sorted in ascending order to align with the expected intensity levels for the hidden states.
  3. *Intercept Initialization:* For each hidden state, the intercept was set to the log of the mean intensity within its corresponding cluster. All other regression coefficients were initialized to zero.
- **Transition Matrix ($\boldsymbol{\Gamma}$):** The transition matrix was initialized by fitting a Markov chain to the sequence of cluster assignments derived from the k-means clustering. This matrix represents the probabilities of transitioning between hidden states.

Results

We fitted the model using $g = 2, 3, 4, 5$ hidden states and selected the best model based on the Bayesian Information Criterion (BIC) (see **?**, Chapter 6 for details). The following table presents the BIC values for models with different numbers of hidden states:

Number of Hidden States (g)	**BIC**
2	122,871.4
3	122,897.0
4	122,939.8
5	122,991.5

Table 1: BIC values for the Continuous-time model with varying g

The BIC suggested that the best model was with $g = 2$, which aligns with the simulation setup. For $g = 2$, the estimated parameters were:

- **Initial distribution:**

$$\boldsymbol{\pi} = (1.00, 0.00).$$

- **Transition matrix:**

$$\boldsymbol{\Gamma} = \begin{bmatrix} 0.73 & 0.27 \\ 0.41 & 0.59 \end{bmatrix}.$$

- **Regression coefficients:**

	Intercept	x_1	$1\{x_2 = B\}$	$1\{x_2 = C\}$	$1\{x_3 = Y\}$
State 1	0.21	0.0010	0.035	0.093	-0.048
State 2	0.40	0.0023	0.090	0.14	-0.050

The values for the initial distribution, transition matrix, and regression coefficients closely match the true parameters used in the simulation, confirming that the fitting algorithm was able to recover the model parameters successfully. Note that with a larger portfolio or more time points, the estimates become more accurate.

1.2 Multinomial Model

Simulation Setup

We use the generated data from the previous simulation, assuming the same time points, number of individuals, and parameters for the Poisson process and the underlying HMM.

The reporting delay is now incorporated by assuming a maximum reporting delay of 4 periods. For each occurrence at time t, the event can be reported at time $t + d$, where $d \in \{0, 1, 2, 3, 4\}$. The reporting delay distribution depends on the covariate x_2 as follows:

- For individuals with $x_2 = A$, the probabilities of reporting in each delay category are uniform:

$$\boldsymbol{p}_A = (0.2, 0.2, 0.2, 0.2, 0.2)$$

- For individuals with $x_2 = C$, the probabilities decrease with the delay:

$$\boldsymbol{p}_C = \left(\frac{5}{15}, \frac{4}{15}, \frac{3}{15}, \frac{2}{15}, \frac{1}{15}\right) = (0.333, 0.267, 0.200, 0.133, 0.067)$$

- For individuals with $x_2 = B$, the probabilities are the average of those for $x_2 = A$ and $x_2 = C$:

$$\boldsymbol{p}_B = (0.267, 0.233, 0.200, 0.167, 0.133)$$

For each individual i, given the total count at time t (same counts obtained from the first simulation), we simulate delayed counts for each delay category $d \in \{0, 1, 2, 3, 4\}$ from a multinomial distribution, using the corresponding probabilities $\boldsymbol{p}_A$, $\boldsymbol{p}_B$, and $\boldsymbol{p}_C$ based on the value of x_2.

The regression function corresponding to the targeted $\boldsymbol{p}_A$, $\boldsymbol{p}_B$, and $\boldsymbol{p}_C$ is given by:

$$\log(q(d; \boldsymbol{x}_i)) = \log\left(\frac{p(d; \boldsymbol{x}_i)}{\sum_{j=0}^{d} p(j; \boldsymbol{x}_i)}\right) = \text{Intercept} + \delta_1 \cdot 1\{x_2 = B\} + \delta_2 \cdot 1\{x_2 = C\}, \quad d = 1, 2, 3, 4,$$

with coefficients:

Intercept	δ_1	δ_2	Delay
-0.69314718	-0.06899287	-0.11778304	$d = 1$
-1.0986123	-0.1541507	-0.2876821	$d = 2$
-1.3862944	-0.2623643	-0.5596158	$d = 3$
-1.6094379	-0.4054651	-1.0986123	$d = 4$

Initialization of Parameters

The initialization of π, $\boldsymbol{\Gamma}$, and $\boldsymbol{\theta}_j'$s in this simulation follows a similar methodology to the original simulation. However, the initialization for $\boldsymbol{\Gamma}$, and $\boldsymbol{\theta}_j'$s are based on a reduced dataset that includes data only up to $t \leq 96$, ensuring that all information about the events and delays is fully observed.

For the regression coefficients $\boldsymbol{\delta}$ associated with the delay probabilities $q(d)$, the initialization is as follows:

- **Intercept Initialization:** The intercept for each delay d is set to the logarithm of the average q_d, calculated as the ratio of the total counts observed at delay d to the cumulative total counts across all delays $j \leq d$.

- **Other Coefficients:** All other regression coefficients in $\boldsymbol{\delta}$ are initialized to zero, reflecting an uninformative prior before model fitting.

This initialization corresponds to assuming same delay probabilities across covariate groups ($\boldsymbol{p}_A = \boldsymbol{p}_B = \boldsymbol{p}_C$) with the delay probabilities initialized as $(0.267, 0.235, 0.198, 0.167, 0.133)$. These values align with the observed proportions in the reduced dataset.

Results

We fitted the model to the censored reported data using $g = 2, 3, 4, 5$ hidden states and selected the best model based on the BIC. The following table presents the BIC values for models with different numbers of hidden states:

Number of Hidden States (g)	**BIC**
2	122,088.0
3	122,122.8
4	122,158.4
5	122,210.6

Table 2: BIC values for the Multinomial model with varying g

Similar to the first simulation, the BIC suggested that the best model was with $g = 2$, which aligns with the simulation setup. For $g = 2$, the estimated parameters were:

- **Initial distribution:**

$$\boldsymbol{\pi} = (1.00, 0.00).$$

- **Transition matrix:**

$$\boldsymbol{\Gamma} = \begin{bmatrix} 0.73 & 0.27 \\ 0.41 & 0.59 \end{bmatrix}.$$

- **Regression coefficients:**

	Intercept	x_1	$1\{x_2 = B\}$	$1\{x_2 = C\}$	$1\{x_3 = Y\}$
State 1	0.21	0.00097	0.035	0.092	-0.046
State 2	0.40	0.0023	0.089	0.14	-0.050

- **Delay coefficients:**

Intercept	δ_1	δ_2	Delay
-0.70	-0.058	-0.086	$d = 1$
-1.11	-0.15	-0.29	$d = 2$
-1.36	-0.28	-0.59	$d = 3$
-1.61	-0.40	-1.11	$d = 4$

These coefficients correspond to $\boldsymbol{p}_A = (0.202, 0.197, 0.197, 0.205, 0.199)$, $\boldsymbol{p}_B = (0.268, 0.234, 0.199, 0.166, 0.133)$, and $\boldsymbol{p}_C = (0.330, 0.274, 0.199, 0.132, 0.065$, which closely match the true probability vectors.

The results confirm that the fitting algorithm successfully recovered the model parameters. As in the continuous-time model, larger datasets or longer time horizons improve estimation accuracy.

1.3 Dirichlet-Multinomial Model

Simulation Setup

We retain the same portfolio characteristics, Poisson counts, and underlying assumptions as in Simulation 1.1. The occurrence counts at each time point are generated identically. Additionally, we assume the same maximum reporting delay of 4 as in Simulation 1.2. However, in this simulation, the reporting delay probabilities are modeled using a Dirichlet distribution.

The reporting delay probabilities, $\boldsymbol{p}(\boldsymbol{x}_i) = (p(0; \boldsymbol{x}_i), p(1; \boldsymbol{x}_i), \ldots, p(4; \boldsymbol{x}_i))$, follow a Dirichlet distribution, with the expected delay probabilities depending on `x2` as follows:

$$
\begin{aligned}
\mathbb{E}(\boldsymbol{p}_A) &= (0.2, 0.2, 0.2, 0.2, 0.2), \\
\mathbb{E}(\boldsymbol{p}_B) &= (0.267, 0.233, 0.200, 0.167, 0.133), \\
\mathbb{E}(\boldsymbol{p}_C) &= (0.333, 0.267, 0.200, 0.133, 0.067),
\end{aligned}
$$

which align with the targeted delay probabilities from Simulation 1.2. For each individual i, the delay probabilities $\boldsymbol{p}_i$ are sampled from the corresponding Dirichlet distribution. Once $\boldsymbol{p}_i$ is sampled, the delayed counts for each delay category d are simulated using a multinomial distribution with the sampled $\boldsymbol{p}_i$, as in Simulation 1.2.

Initialization of Parameters

The initialization of π, $\boldsymbol{\Gamma}$, and $\boldsymbol{\theta}_j$'s in this simulation is exactly the same as the initialization in Simulation 1.2. We choose weak priors for the Dirichlet-distributed delays, which correspond to the mean probabilities $\mathbb{E}(\boldsymbol{p}_A) = \mathbb{E}(\boldsymbol{p}_B) = \mathbb{E}(\boldsymbol{p}_C) = (0.266, 0.233, 0.200, 0.168, 0.133)$, similar to the initialization in Simulation 1.2. These values align with the observed proportions of delayed counts in the reduced dataset.

Results

We fitted the model to the censored reported data using $g = 2, 3, 4, 5$ hidden states and selected the best model based on the BIC. The following table presents the BIC values for models with different numbers of hidden states:

The BIC suggested that the best model was with $g = 2$, which aligns with the simulation setup. For $g = 2$, the estimated parameters were:

- **Initial distribution:**

$$\boldsymbol{\pi} = (1.00, 0.00).$$

Number of Hidden States (g)	BIC
2	122,174.4
3	122,194.3
4	122,233.3
5	122,308.3

Table 3: BIC values for the Dirichlet-Multinomial model with varying g

- **Transition matrix:**

$$\mathbf{\Gamma} = \begin{bmatrix} 0.73 & 0.27 \\ 0.41 & 0.59 \end{bmatrix}.$$

- **Regression coefficients:**

	Intercept	x_1	$1\{x_2 = B\}$	$1\{x_2 = C\}$	$1\{x_3 = Y\}$
State 1	0.21	0.00097	0.035	0.092	−0.048
State 2	0.40	0.0023	0.092	0.14	−0.050

- **Expected Delay Probabilities:**

$$\begin{aligned} \mathbb{E}(\boldsymbol{p}_A) &= (0.203, 0.202, 0.201, 0.195, 0.198), \\ \mathbb{E}(\boldsymbol{p}_B) &= (0.258, 0.232, 0.199, 0.173, 0.139), \\ \mathbb{E}(\boldsymbol{p}_C) &= (0.318, 0.257, 0.195, 0.140, 0.090). \end{aligned}$$

These values closely match the true values. It is important to note that the estimation of the mean delay probabilities is highly dependent on the prior; our simulation is based on a weak prior.

1.4 Comparison of Runtime Per Iteration

We compared the runtime per iteration for $g = 2$ across the three models to provide additional insights into their computational performance in our simulation studies. The results are based on runs conducted on a laptop with an Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 2.00 GHz processor. The average runtimes were as follows:

- **Continuous-Time Model:** 0.491 seconds per iteration, with a total runtime of 0.982 seconds.
- **Multinomial Model:** 5.621 seconds per iteration, with a total runtime of 16.863 seconds.
- **Dirichlet-Multinomial Model:** 35.682 seconds per iteration, with a total runtime of 2 minutes and 23 seconds.

While the runtime per iteration for the Continuous-Time Model is considerably smaller, it is important to note that this model does not explicitly incorporate the reporting delay component within the EM algorithm. Instead, the reporting delay is addressed in a preprocessing step. Before fitting the model, one must first fit a reporting delay model and compute the integral:

$$\int_{d_{l-1}}^{d_l} P_{U|\boldsymbol{x}_i}(\tau - t)\,dt, \quad \forall l \text{ and } i.$$

This additional preprocessing step can add substantial computational overhead. In contrast, the Multinomial and Dirichlet-Multinomial models address the reporting delay component directly within the EM framework, leading to a more integrated and streamlined estimation process at the cost of increased runtime per iteration.