econometrics

MDPI

Article

Dirichlet Process Log Skew-Normal Mixture with a Missing-at-Random-Covariate in Insurance Claim Analysis

Minkun Kim [1,*], David Lindberg [2], Martin Crane [1] and Marija Bezbradica [1]

[1] ADAPT Centre, School of Computing, Dublin City University, D09 PX21 Dublin, Ireland; martin.crane@adaptcentre.ie (M.C.); marija.bezbradica@adaptcentre.ie (M.B.)
[2] Department of Statistics, University of Florida, Gainesville, FL 32611, USA; dlindberg@ufl.edu
* Correspondence: minkun.kim@adaptcentre.ie; Tel.: +353-089-459-8519

Abstract: In actuarial practice, the modeling of total losses tied to a certain policy is a nontrivial task due to complex distributional features. In the recent literature, the application of the Dirichlet process mixture for insurance loss has been proposed to eliminate the risk of model misspecification biases. However, the effect of covariates as well as missing covariates in the modeling framework is rarely studied. In this article, we propose novel connections among a covariate-dependent Dirichlet process mixture, log-normal convolution, and missing covariate imputation. As a generative approach, our framework models the joint of outcome and covariates, which allows us to impute missing covariates under the assumption of missingness at random. The performance is assessed by applying our model to several insurance datasets of varying size and data missingness from the literature, and the empirical results demonstrate the benefit of our model compared with the existing actuarial models, such as the Tweedie-based generalized linear model, generalized additive model, or multivariate adaptive regression spline.

Keywords: Bayesian nonparametric model; heterogeneity; missing at random; log-normal sum approximation; aggregate insurance claims; clustering; generative model; latent class

Citation: Kim, Minkun, David Lindberg, Martin Crane, and Marija Bezbradica. 2023. Dirichlet Process Log Skew-Normal Mixture with a Missing-at-Random-Covariate in Insurance Claim Analysis. *Econometrics* 11: 24. https://doi.org/10.3390/econometrics11040024

Academic Editor: Marc S. Paolella

Received: 28 May 2023
Revised: 6 October 2023
Accepted: 9 October 2023
Published: 12 October 2023

1. Introduction

In short-term insurance contracts, predicting insurance claim amounts is essential for major actuarial decisions such as pricing or reserving. In particular, the development of a full predictive distribution of aggregate claims is fundamental to understanding potential risks. However, it is often not easy to develop the loss distribution properly due to its complex distributional features, such as high skewness, zero inflation, hump shape, and multi-modality. It is known that such complexity stems from the presence of diverse, interconnected unknown risk classes and uncertainty in loss events. Accordingly, there have been many attempts by actuaries to develop loss models, accommodating multiple risk classes and quantifying the uncertainty. This includes the parametric mixture modeling approaches based on log-normal, Weibull, Burr, Pareto, etc. distributions to capture the various aspects of the loss of data (see Hogg and Klugman 2009). The parametric approaches have been popular because they are conceptually simple, relying on established statistical principles. However, the reality is that we never know how many true risk classes are associated with the loss of data we have. Therefore, it is not surprising that many parametric approaches are often met with model misspecification biases. With respect to this, a Bayesian nonparametric (BNP) approach has been gradually recognized to solve such distributional conundrums in insurance loss analysis. The major difference between the traditional parametric models and the BNP is that the parametric model is built upon a fixed number of risk classes imagined by actuaries, while the BNP does not allow the number of risk classes to be fixed but instead lets the data determine the number of risk classes. In other words, the BNP framework theoretically supports an infinite number of clusters or

parameters until it finishes investigating every corner of the parameter space with the Monte Carlo simulation technique. Aligned with such conceptual appeal, there have been several BNP frameworks studied and applied in actuarial practice recently, such as the Gaussian process, Dirichlet process, and Pitman–Yor process. (e.g., Suwandani and Purwono 2021; Hong and Martin 2018; Shams Esfand Abadi 2022). Focusing on the Dirichlet process prior, Hong and Martin (2018) recently developed the Dirichlet process mixture (DPM) model as a BNP approach that maximizes the fitting flexibility of the loss distribution with the presence of unknown risk classes. In this paper, as an extension of their work, we attempt to go beyond the search for the maximized fitting flexibility, addressing the issues that arise from the presence of covariates, missing data, and aggregate losses (total amount of losses). The implication is that the predictive distribution for the expected aggregate claims developed under Hong and Martin's Dirichlet process framework cannot obviate the chance of model misspecification bias with the incorporation of covariate effects and log-normal convolution. For example, as covariates add new information that differentiates the data points of the outcome variable, a new structure can be introduced into the data space, and this increases the within-cluster heterogeneity (see Neuhaus and McCulloch 2006). That aside, the incorporation of missing covariates may exacerbate the existing heterogeneity. Additionally, given that the outcome variable describes the aggregate losses, rather than individual claim amounts, it is difficult to compute the log-normal convolution as it does not have a closed-form solution. In this regard, our study extends their work by addressing the following research questions:

- **RQ1.** If an additional unobservable heterogeneity is introduced by the inclusion of covariates, then what is the best method to capture the within-cluster heterogeneity in modeling the total losses, comparing several conventional approaches?
- **RQ2.** If an additional estimation bias results from the use of the incomplete covariates under missing-at-random (MAR) conditions, then what is the best way to increase the imputation efficiency, comparing several conventional approaches?
- **RQ3.** If an individual loss is distributed with log-normal densities, then what is the best way to approximate the sum of the log-normal outcome variables, comparing several conventional approaches?

2. Discussion on the Research Questions and Related Work

Let $Y_i,\ i = 1, 2, \ldots, N$ be the independent claim amount (reported by each policyholder for a single policy) random variable, defined on a common probability space $(\Omega, \mathcal{F}, P)$ from a certain loss distribution, such as a log-normal distribution. Let $\boldsymbol{X}$ be a vector of the covariates and $N(t)$ be the total claim count, denoting the number of individual claims for a single policy up to time t (policy period). The aggregate claim $S_h(t)$ for a single policy h given time t can be expressed as a convolution, where $S_h(t) = \sum_{i=1}^{N(t)} Y_i = Y_1 + Y_2 + \ldots + Y_{N(t)}$ (assuming that each policy h is a group policy referring to the insurance coverage provided to a group of individuals under a single policy). At the end of the policy period t, let $\tilde{S}(t)$ be the total aggregate claim amounts from the total policies received by an insurer. Then, $\tilde{S}(t) = \sum_{h=1}^{H} S_h(t) = S_1(t) + S_2(t) + \ldots + S_H(t)$, in which H is the total number of independent policies in the entire portfolio. Note that both convolutions described so far are built upon the assumption that the summands—$Y_i,\ i = 1, 2, \cdots, N(t)$ and $S_h,\ h = 1, 2, \cdots, H$—are mutually independent and identically distributed (to maintain the homogeneity of each loss).

The involvement of covariates and the lack of closed-form solutions for the log-normal sum bring about several challenges that violate the assumptions for an accurate estimation of the total aggregate losses $\tilde{S}(t)$. To begin with, the use of covariates gives rise to an additional within-cluster heterogeneity. Kaas et al. (2008) described a standard aggregate loss modeling principle, denoting that the expected aggregate claims $E[S_h]$ are obtained by the product of the mean claim counts and severities, where $E[S_h] = E[N]E[Y]$. With the inclusion of covariates $\boldsymbol{X}$, however, a new unknown structure or heterogeneity is introduced into the data space of Y_i, and this means that $Y_1|\boldsymbol{X}_1, Y_2|\boldsymbol{X}_2, \cdots, Y_N|\boldsymbol{X}_N$ within

a single policy can still be independent but cannot be identically distributed. Therefore, $E[S_h|\boldsymbol{X}] \neq E[N|\boldsymbol{X}]E[Y|\boldsymbol{X}]$, and the total aggregate losses $\tilde{S}(t)$ becomes difficult to compute with the conventional collective risk modeling approach. In addition, assuming that the severity Y_i follows a log-normal distribution, the computation of $\tilde{S}(t)$ becomes quite difficult, as its convolution S_h is not known to have a closed form (see Beaulieu and Xie 2003). Another challenge is the missing covariates in $S_h|\boldsymbol{X}$. As shown by Ungolo et al. (2020), the missing covariates under the missing-at-random (MAR) assumption lead to biased parameter estimations because the uncertainty in the estimation results of the parameters describing the outcome Y is heavily affected by the quality of the covariates $\boldsymbol{X}$. Again, in this case, $\tilde{S}(t)$ cannot be computed properly.

Compounding all this, we propose the Dirichlet process log skew-normal mixture to model $S_h|\boldsymbol{X}$. We aim to cope with the within-cluster heterogeneity as suggested by Braun et al. (2006); Hong and Martin (2018) while employing the log skew-normal approximation studied by Li (2008) to compute each $S_h|\boldsymbol{X}$ and the sum of log-normal random variables $\sum_{i=1}^{N(t)} Y_i|\boldsymbol{X}$. When it comes to the problem of missing covariates, we exploit the generative capability of the Dirichlet process to capture the latent structure of data, which allows for a rigorous statistical treatment of MAR covariates.

2.1. Can the Dirichlet Process Capture the Heterogeneity and Bias? RQ1 and RQ2

Figure 1 illustrates the unpredictable and heterogeneous nature of the aggregate losses S_h and how this can be addressed by the Dirichlet process. A series of independent, identically distributed S_h developed by Y_i for each policy h can be observed and collected within a certain policy period t. However, the presence of unsettled amounts of losses Y_h^* incurred from unknown policyholders or other unobservable features of the policyholders often increase the heterogeneity of each aggregate loss S_h as well as the total aggregate losses $\tilde{S}_h(t)$. This is because any policyholders in different risk classes can raise claims at any time over a fixed time horizon t, and their unsettled claim amounts (i.e., random Y_h^*) will not be known in advance. In order to understand the aggregate losses S_h properly, one might need to answer questions such as "How much is Y_h^*?" and "By which policy or risk class Y_h^* is incurred?"

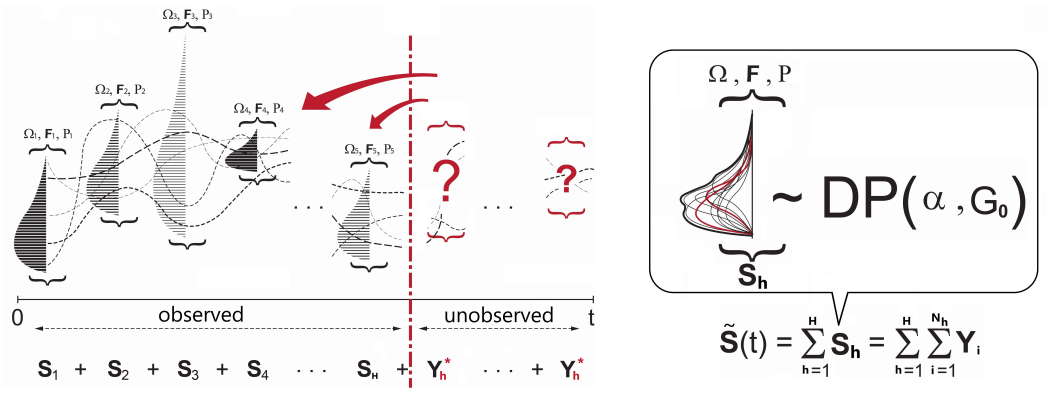

Figure 1. A series of independent and identically distributed aggregate losses S_h for each policy h and the emergence of unsettled losses Y_h^* tied to unknown policyholders that increases the heterogeneity of S_h (**left**). The DPM as a mixture model to accommodate the inherent heterogeneity of S_h (**right**).

With respect to this, Hong and Martin (2018) presented a DPM framework that takes into account such sources of heterogeneity via the extensive simulation of S_h and the investigation of multiple mixture scenarios of S_h. By associating each unobservable loss Y_h^* with every possible risk-clustering scenario built upon an infinite dimensional parametric structure, their DPM optimizes the prediction values for the future amount of S_t. Braun et al. (2006) also carried out a useful study of the DPM in insurance practice to capture unobservable heterogeneity in the loss data, such as intracorrelation between claim amounts Y_i in the different risk classes. In short, no matter how complex the distribu-

tion of the loss is, the DPM is capable of accommodating any distributional properties—multi-modes, skewness, heavy tails, etc.—resulting from unobservable heterogeneity and therefore dramatically minimizes model misspecification biases.

Having said that, however, if considering covariates $\boldsymbol{X}$ to better understand the different risk classes, then one might introduce an additional source of heterogeneity into the scene, which prevents each cluster S_h from being identically distributed. In regard to this, Huang and Meng (2020) pointed out that the covariate effect can be incorporated into the DPM framework by considering the weight of the mixture component to be covariate-dependent rather than constant. This allows the mixture model to keep each risk class homogeneous while identifying the unique loss patterns (relationship between the loss amount and the risk factors) of different types of policyholders. The research on covariate-dependent weights includes a series of stick-breaking method studies on regression, time series, loss distribution fitting problems, etc. in insurance. Sethuraman (1994) presented the general stick-breaking framework to construct a prior distribution over an infinite number of mixture components, ensuring that the mixing weights are probabilities and their distribution is discrete with a probability of one. Griffin and Steel (2006, 2011) studied the order-based stick-breaking method that allows the mixing weights to vary over the covariate effect. They ordered the mixing weights according to a covariate-dependent ranking and associated risk classes for similar covariate values with similar orderings. Rodriguez and Dunson (2011) developed the probit stick-breaking process, aiming to diversify families of prior distributions while preserving computational simplicity. They suggest that the covariate effects can be integrated with the probit transformations of normal random variables to produce mixing weights, which is the replacement for the characteristic beta distribution in the stick formulation. The stick-breaking-based mixing weights can be used in determining the clustering structures in the time series or regression analysis. Bassetti et al. (2014) and Billio et al. (2019) studied applications of the stick-breaking process to hierarchical prior development for the coefficients of autoregressive time series models. Hannah et al. (2011) and Richardson and Hartman (2018) proposed combining the stick-breaking-based prior with the Gaussian density to build a regression model.

In this article, we use a generalized representation of the stick-breaking process developed by Sethuraman (1994) to incorporate the covariate effects into the mixing weight. This is because, with the inclusion of the covariates subject to missingness, the advanced stick-breaking approaches listed above cannot be viable solutions. On the contrary, the DPM with Sethuraman's stick-breaking formulation offers a useful bedrock (such as a multi-purpose joint distribution) for a MAR covariate treatment. As a generative modeling approach, the DPM framework coupled with Sethurman's stick-breaking method models both the outcomes S_h and covariates $\boldsymbol{X}$ jointly to produce cluster memberships. This is used as key knowledge to identify the latent structure of the data and thus estimate the missing information (see Shahbaba and Neal 2009). For example, in the domain of medicine research, Roy et al. (2018) developed a novel imputation strategy for the MAR covariate using the joint model of the BNP framework. A further survey of imputation methods based on the nonparametric Bayesian framework can be found in the work of Si and Reiter (2013) and the references therein.

2.2. Can a Log Skew-Normal Mixture Approximate the Log-Normal Convolution? RQ3

The log-normal distribution has been considered a suitable claim amount Y_i distribution due to its nonnegative support, right-skewed curve, and moderately heavy tail to accommodate some outliers. However, if one generalizes the individual claim amount Y_i by introducing a log-normal distribution, then the convolution computation for S_h fails because the exact closed form for the log-normal sum is unknown.

Furman et al. (2020) presented several existing methods for the log-normal sum approximation that have been studied in the literature. This includes the moment-matching approximation approaches such as minimax approximation, least squares approximation, log-shifted gamma approximation, and log skew-normal approximation. The distance mini-

mization approaches—minimax approximation or least squares approximation—described by Beaulieu and Xie (2003); Zhao and Ding (2007) are conceptually simple, but they require fitting the entire cumulative densities to the sum of the claim amounts, which can be computationally expensive and fail easily when the number of summands Y_i increases. The log-shifted gamma approximation suggested by Lam and Le-Ngoc (2007) has less strict distributional assumptions, but it is not particularly accurate at the lower region of the distribution. In our study, special attention is paid to the possibility of the log skew-normal approximation method for the sake of simplicity. A skew-normal distribution as an extension of a normal distribution has a third parameter to naturally explain skewness apart from the other parameters (for a location and spread). Li (2008) pointed out that one can exploit the third parameter of the skew-normal distribution to capture different skewness levels of each summand. By taking the log of the skew-normal densities, we can approximate S_h, the sum of the log-normal Y_i. Using the log skew-normal as the underlying distribution for S_h in the DPM framework, one can eliminate the need to compute the cumulative density curve, and its closed-form density and the optimal distribution parameters for S_h can be easily obtained by the moment-matching technique. For further details, see Li (2008) and the references contained within.

2.3. Our Contributions and Paper Outline

The contributions of this study are twofold. First, we propose a new method to efficiently model the sum of log-normal outcome variables representing the aggregate insurance losses S_h. Using the log skew-normal model in the BNP framework, we cope with the (1) lack of a closed form for the log-normal convolution and (2) heterogeneity in the log-normal random variable at the same time. Second, we tackle the adverse impact triggered by the inclusion of covariates $\boldsymbol{X}$ into the aggregate loss modeling framework. This encompasses the added heterogeneity across Y_i and the missing information fed by the MAR covariates $\boldsymbol{X}$. To our knowledge, there have been no previous attempts to estimate the log skew-normal mixture within the BNP framework or use the DPM to handle the MAR covariate in insurance loss modeling.

The rest of this paper is structured as follows. In Section 3, we describe the proposed modeling framework for S_h, assuming a log-normal distributed Y_i and the inclusion of both continuous and discrete covariates $\boldsymbol{X}$. This section also presents our novel imputation approach for the MAR covariate within the DPM framework. Section 4 clarifies the final forms of the posterior and predictive densities accordingly. Section 5 presents our empirical results and validates our approach by fitting to two different datasets with different sample sizes drawn from the R package **CASdatasets** and the Wisconsin Local Government Property Insurance Fund (LGPIF). This is followed by a discussion in Section 6.

3. Model: DP Log Skew-Normal Mixture for $S_h|\boldsymbol{X}$

3.1. Background

Consider that there are multiple unknown risk classes (clusters) across the claim Y_i information within each policy, and then the individual aggregate claims S_h for the policy h would have diverse characteristics that cannot be explained by fitting a single log skew-normal distribution. In order to approximate the distribution that captures such diverse characteristics in S_h, we seek to investigate diverse clustering scenarios. To this end, as suggested by Hong and Martin (2018), we exploit the infinite mixture of log skew-normal clusters and their complex dependencies by employing a Dirichlet process. The Dirichlet process produces a distribution over clustering scenarios (with clustering parameters):

$$\begin{aligned} \{\boldsymbol{\theta}_j,\ \boldsymbol{w}_j\} &\sim G \\ G &\sim DP(\alpha,\ G_0) \end{aligned}$$

where G denotes the clustering scenarios, and the important components of G are as follows:

- $\boldsymbol{\theta}_j$: the parameters of the outcome variable defined by cluster j.

- $\boldsymbol{w}_j$: the parameters of the covariates defined by cluster j.

G, as a single realization of the joint cluster probability vector $\{G(A_1), G(A_2)\ldots\}$ sampled from the DPM model, takes independent partitions $A_1, A_2, \ldots$ of the sample space $\bigcup_{k=1}^{\infty} A_k = A$ of the support of G_0. Through sufficient simulations of G, the Dirichlet process investigates all possible clustering scenarios rather than relying on a single best guess. The overall production of G is controlled with two parameters: the precision α and a base measure G_0. The precision α controls a variance of sampling G in the sense that larger α generates new clusters more often to account for the unknown risk classes. The base measure G_0, as the mean of DP(α, G_0), is a DP prior over the joint space of all parameters for the outcome model, covariate model, and the precision α, as shown in Ghosal (2010).

Note that the original research on DPM by Hong and Martin (2018) mainly focused on the random cluster weights $\boldsymbol{\omega}_j$ that were not tied to the covariates $\boldsymbol{X}$. On the other hand, in our model, the covariate effects are incorporated into the development of the cluster weights $\boldsymbol{\omega}_j$. All calculations for the development of the DPM modeling components in this paper are based on the principles introduced by Ferguson (1973), Antoniak (1974), and Sethuraman (1994).

3.2. Model Formulation with Discrete and Continuous Clusters

If the goal of modeling is to perform prediction and uncertainty quantification with the presence of heterogeneity (resulting from previously unseen risk factors), then the DPM framework exploits the generative process to this end. This process provides all the necessary components to construct the predictive distribution, using the infinite clustering scenarios based on the joint distribution of observed outcomes, covariates, as well as hidden variables. Let the outcome be $S = \{S_1, S_2, \ldots, S_H\}$, denoting the H different aggregate claims (incurred by the H different policies). We assume that the covariate $\boldsymbol{x}_1$ is binary and $\boldsymbol{x}_2$ is Gaussian, and then our baseline DPM model can be expressed as follows:

$$\begin{aligned} S_h|x_{1h},x_{2h},\boldsymbol{\beta}_j,\sigma_j^2,\xi_j,\tilde{\boldsymbol{\beta}}_j & \\ \sim \delta(\boldsymbol{X}_h^T\tilde{\boldsymbol{\beta}}_j)\ \mathbb{1}(S_h=0) + \left[1-\delta(\boldsymbol{X}_h^T\tilde{\boldsymbol{\beta}}_j)\right]\ & LogSN\left(\boldsymbol{X}_h^T\boldsymbol{\beta}_j,\ \sigma_j^2,\ \xi_j\right) \\ x_{1h}|\pi_j \sim\ & Bern\left(\pi_j\right) \\ x_{2h}|\mu_j,\tau_j^2 \sim\ & N\left(\mu_j,\ \tau_j^2\right) \\ \{\boldsymbol{\theta}_j,\ \boldsymbol{w}_j,\boldsymbol{\omega}_j\} \sim\ & G \\ G \sim\ & DP\left(\alpha,\ G_0\right) \end{aligned} \tag{1}$$

where j is the risk class index, $\boldsymbol{X}_h = \{x_{1h}, x_{2h}\}$ for the covariates, $\boldsymbol{\theta}_j = \{\boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j\}$ for parameters describing the outcome, and $\boldsymbol{w}_j = \{\pi_j, \mu_j, \tau_j^2\}$ for parameters explaining the covariates. S_h is modeled as a mixture of a point mass at 0 with positive values distributed with a log skew-normal density to address the complications of zero inflation in the loss data, while $\delta(\boldsymbol{X}^T\tilde{\boldsymbol{\beta}}_j)$ models the probability of the outcome being zero using a multivariate logistic regression. Variable Definitions has a brief description of all parameters used in this study.

When considering a Dirichlet process log skew-normal mixture to house the multiple unknown risk classes in S_h, it is necessary to differentiate the forms of the mixture components, depending on the types of clusters they use: discrete and continuous. While keeping the inference of the cluster parameters data-dominated, the DPM first develops discrete clusters based on the given claim information and then extrapolates certain unobservable clusters of claims by examining the heterogeneity (or hidden risk classes) of each cluster. In this process, the DPM develops new continuous clusters additionally and assesses them with some probabilistic decision-making algorithms, rendering the parameter estimations computationally efficient and asymptotically consistent (see Hong and Martin 2017).

The discrete mixture components (clusters) in the DPM framework have the standard form that is useful in accounting for the observed classes, such as policy information

for aggregate losses S_h (see Diebolt and Robert 1994). In calculating the discrete cluster probabilities, we assume that the nonzero outcome and covariates are distributed with the densities denoted by

$$f_{LSN}(S_h|\boldsymbol{X}_h^T\boldsymbol{\beta}_j,\ \sigma_j^2,\ \xi_j) = \frac{2}{S_h\sigma_j}\phi\Big(\frac{\log S_h - \boldsymbol{X}_h^T\boldsymbol{\beta}_j}{\sigma_j}\Big)\cdot\Phi\Big(\xi_j\cdot\frac{\log S_h - \boldsymbol{X}_h^T\boldsymbol{\beta}_j}{\sigma_j}\Big) \tag{2a}$$

$$f_{Bern}(x_{1h}|\pi_j) = \pi_j^{x_{1h}}(1-\pi_j)^{1-x_{1h}} \tag{2b}$$

$$f_N(x_{2h}|\mu_j,\tau_j^2) = \frac{1}{\sqrt{2\pi\tau_j^2}}\exp\Big\{-\frac{1}{2\tau_j^2}(x_{2h}-\mu_j)^2\Big\} \tag{2c}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are standard normal probability and cumulative density functions for the log skew-normal density, respectively. To model the outcome data $S_h|\boldsymbol{X}_h$ for the policy h, the DPM takes the general form of the mixture

$$f(S_h|\boldsymbol{X}_h,\boldsymbol{\theta}) = \sum_{j=1}^{\infty}\boldsymbol{\omega}_j\Big(\delta(\boldsymbol{X}_h^T\tilde{\boldsymbol{\beta}}_j)\ \mathbb{1}(S_h=0) + \big[1-\delta(\boldsymbol{X}_h^T\tilde{\boldsymbol{\beta}}_j)\big]\ f_{LSN}(S_h|\boldsymbol{X}_h,\boldsymbol{\theta}_j)\Big) \tag{3}$$

where j is the cluster index, $\boldsymbol{\theta}_j = \{\boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j\}$ and $\boldsymbol{w}_j = \{\pi_j, \mu_j, \tau_j^2\}$ are the outcome and covariate parameters to explain the risk clusters, respectively, and $\boldsymbol{\omega}_j$, for the functions of the covariates $\boldsymbol{\omega}_j(\boldsymbol{X}_h|\boldsymbol{w}_j)$, represents the cluster component weights (mixing coefficient) satisfying $\sum_{j=1}^{\infty}\boldsymbol{\omega}_j = 1$. However, when the total number of mixture components $J \le H$ is determined later from the data we have, the new continuous clusters can be introduced by G_0 (with its infinite-dimensional parametric structure) in order to tackle the additional unknown risk classes. This involvement of G_0 can address the within-class heterogeneity in S_h by confronting the current discrete clustering result and investigating the homogeneity more closely. As the new clusters are considered countably infinite, their corresponding forms for the outcome and covariate models to obtain the continuous cluster are given by

$$f_0(S_h|\boldsymbol{X}_h) = \int f(S_h|\boldsymbol{X}_h,\boldsymbol{\theta})\ dG_0(\boldsymbol{\theta}) \tag{4a}$$

$$f_0(x_{1h}) = \int f_{Bern}(x_{1h}|\boldsymbol{w})\ dG_0(\boldsymbol{w}) \tag{4b}$$

$$f_0(x_{2h}) = \int f_N(x_{2h}|\boldsymbol{w})\ dG_0(\boldsymbol{w}) \tag{4c}$$

They are also known as a "parameter-free outcome model" and a "parameter-free covariate model", respectively, for developing the new continuous cluster mixture. Given a collection of outcome-covariate data pairs $D = \{S_h, \boldsymbol{X}_h\}_{h=1}^{H}$, the DPM puts together the current discrete clusters and new continuous clusters to update the mixture form in Equation (3), with help from the Monte Carlo Markov chain method (using sufficiently simulated samples of the major parameters $\boldsymbol{\theta}_j, \boldsymbol{w}_j$). Consequently, the sample G described in Equation (1) becomes $G = f(S_h|\boldsymbol{X}_h, D) = \sum_{j=1}^{\infty}\boldsymbol{\omega}_j\cdot\delta_{z_j}$, where δ_{z_j} denotes discrete clusters and the continuous cluster as a point mass distribution at the random locations sampled from G_0. Aligned with such flexible cluster development, the form of the predictive distribution can be molded based on the knowledge extracted from G and the finite number of clusters J as follows:

$$f(S_h|\boldsymbol{X}_h,\boldsymbol{\theta},\boldsymbol{w},\alpha) = \frac{\boldsymbol{\omega}_{J+1}^*}{\boldsymbol{\omega}_{J+1}^* + \sum_{j=1}^{J}\boldsymbol{\omega}_j^*}\cdot f_0(S_h|\boldsymbol{X}_h) + \frac{\sum_{j=1}^{J}\boldsymbol{\omega}_j^*\cdot f(S_h|\boldsymbol{X}_h,\boldsymbol{\theta}_j)}{\boldsymbol{\omega}_{J+1}^* + \sum_{j=1}^{J}\boldsymbol{\omega}_j^*} \tag{5}$$

The finalized cluster weights in Equation (5) are secured through computing the two submodels below for the discrete and continuous cluster weights, respectively, which reflect the properties of the clusters and relevant covariates:

$$\boldsymbol{\omega}^*_{J+1} = \frac{\alpha}{\alpha + H} \cdot f_0(x_{1h}, x_{2h}) \tag{6a}$$

$$\boldsymbol{\omega}^*_{j} = \frac{n_j}{\alpha + H} \cdot f(x_{1h}, x_{2h} | \boldsymbol{w}_j = (\pi_j, \mu_j, \tau^2_j)) \tag{6b}$$

where α is the precision parameter to control the acceptance chances of the new clusters, n_j is the number of observations in cluster j, $f_0(\boldsymbol{X}_h)$ is the parameter-free covariate model in Equations (4b) and (4c) to support the new continuous cluster, and $f(\boldsymbol{X}_h|\boldsymbol{w}_j)$ is the covariate model to support the current discrete clusters. Note that Equation (5) is derived from the joint distribution of $\{S_h, X_h\}$ conditioned on the posterior samples. The mixture components—$\omega^*_j \cdot f(S_h|\boldsymbol{X}_h, \boldsymbol{\theta}_j)$ and $\omega^*_{J+1} \cdot f_0(S_h|\boldsymbol{X}_h)$—as a product comprise the joint distribution $\{S_h, X_h\}$. The mixing weights $\boldsymbol{\omega}_j$ are obtained by the covariate models of $\boldsymbol{x}_1, \boldsymbol{x}_2$ that explain $\boldsymbol{\omega}^*_j$ and $\boldsymbol{\omega}^*_{J+1}$. This is based on a Polya Urn distribution suggested by Blackwell and MacQueen (1973), which is aligned with the result from the generalized stick-breaking representation of the DPM presented by Sethuraman (1994).

3.3. Modeling $S_h|X_h$ with a Complete Case Covariate

The joint posterior update for the outcome and covariate parameters—$\boldsymbol{\theta}_j, \boldsymbol{w}_j$—in Equations (5) and (6) can be made through the DPM Gibbs sampler given in Algorithm A2 in Appendix B. In a nutshell, the DPM Gibbs sampler obtains draws from the analytically intractable posterior, alternating between two stages to ensure convergence: (1) updating the cluster membership for each observation and (2) updating the parameters given the cluster partitioning. By looping through this algorithm many times (e.g., M = 100,000 iterations), each iteration might give a slightly different selection of the new clusters based on the Polya Urn scheme (see Gershman and Blei 2012), but the log-likelihood calculated at the end of each iteration can help keep track of the convergence of the selections. A detailed description of these two stages in Algorithm A2 is given below.

Stage 1. Cluster membership update:

Step I. Let the cluster-index $j = 1, 2, \cdots, J$ for the observation h be s_h. First, the cluster membership j is initialized by some clustering methods such as hierarchical or k-means clustering. This provides an initial clustering of the data $(S_h, \boldsymbol{X}_h)$ as well as the initial number of clusters.

Step II. Next, with the parameters sampled from the DPM prior G_0 described in Section 4.1 and the conditional probability term $p(s_h|s_{-h})$ on lines 6 and 9 in Algorithm A2 for the observation assignment, the ultimate probabilities of the selected observation h being in the current discrete clusters and the proposed continuous cluster are computed, respectively. (The use of such a nonparametric prior to the development of a new continuous cluster allows the shape of the cluster to be driven by the data). Note that the term $p(s_h|s_{-h})$ is known as the *Chinese Restaurant process* (see Blei and Frazier 2011) probability given by

$$p(s_h|s_{-h}) = \begin{cases} c \cdot \dfrac{n_j^{-h}}{\alpha + H - 1}, & \text{for } h \text{ entering into the existing cluster: } s_h = j. \\ c \cdot \dfrac{\alpha}{\alpha + H - 1}, & \text{for } h \text{ entering into the new cluster: } s_h = J + 1. \end{cases} \tag{7}$$

where c is a scaling constant to ensure that the probabilities add up to one and s_{-h} is the collection of cluster indices $(s_1, s_2, \cdots, s_{h-1}, s_{h+1}, \cdots, s_H)$ assigned to every observation without the cluster index s_h of the obser-

vation h. A larger α results in a higher chance of developing the new continuous cluster and adding to the collection of the existing discrete clusters. Since the number of clusters is not fixed, and the sequence of cluster assignment to observation cannot be ordered, one might be concerned about the sampling variance or convergence problem in the Gibbs sampler. In this regard, we expect that Equation (7) can carry out stable simulations with the Gibbs sampler. Neal (2000) pointed out that from the example of Escobar's algorithm, the sequence in which the observation h arrives in the cluster s_h is exchangeable under this conditional probability distribution described in Equation (7). This means that the ultimate joint distribution to update the cluster memberships from lines 4 to 10 in Algorithm A2 does not depend on the order of the sequence in which the observations arrive.

Step III. Lastly, the new cluster membership is determined and updated by the Polya Urn scheme using a multinomial distribution based on the resulting cluster probabilities. This is briefly illustrated in Figure 2. Please note how the development of the cluster weighting components $\omega_j^*, \omega_{J+1}^*$ in Equations (6a) and (6b) is made in Figure 2.

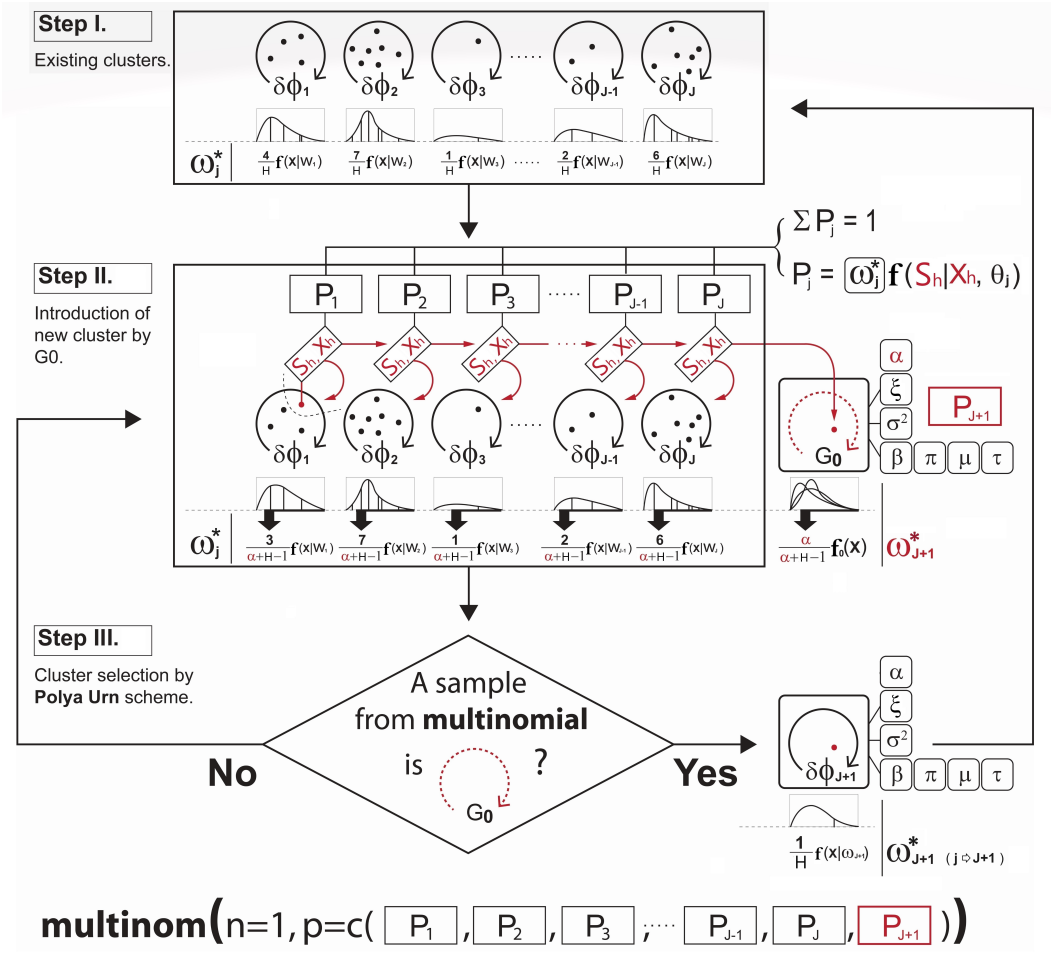

Figure 2. A schematic of the cluster membership update process in Stage 1. In Step I, the algorithm initializes the cluster memberships and parameters including ω_j. In Step II, the cluster probabilities $\boldsymbol{P}$ of the selected observation h are computed. In Step III, the new cluster membership is determined by the Polya Urn scheme, and the new clustering weight component ω_{J+1}^* is created.

Stage 2. **Parameter update:**
Once all observations have been assigned to particular clusters $j = 1, 2, \cdots, J$ at a given iteration in the Gibbs sampling, the parameters of our interest—α and $\boldsymbol{\theta}_j, \boldsymbol{w}_j$—for each cluster are updated, given the new cluster membership. This is accomplished using the posterior densities denoted by $p(\alpha|J)$, $p(\boldsymbol{\theta}|S_h, \boldsymbol{X}_h)$, and $p(\boldsymbol{w}|\boldsymbol{X}_h)$, in which $S_h, \boldsymbol{X}_h$ represents all observations in cluster j. When it comes to the forms of the prior and posterior densities from lines 17 to 23 in Algorithm A2 that are used to simulate the parameters $\{\alpha^*, \boldsymbol{\theta}_j^*, \boldsymbol{w}_j^*\}$, we detail them in Appendix A.

The DPM model described here can be characterized by the investigation of the infinite number of clustering scenarios coupled with covariates. The simulated outcome model $f(S_h|\boldsymbol{X}_h, D) = \sum_{j=1}^{\infty} \boldsymbol{\omega}_j \cdot \delta_{z_j}$ and its predictive model in Equation (5) show that although the DPM framework allows infinite-dimensional clustering, the dimension of the sampling output G is adaptive, as it is a mixture with at most finite components determined by the data themselves (its dimension cannot be greater than the total sample size H). This gives the model flexibility, and throughout such modeling flexibility, the clustering scenarios G accommodate all distributional properties of the given claims as well as the additional unknown claims. In this process, the DPM captures the within-class heterogeneity across the observations, and thus the resulting clusters can be kept as homogeneous as possible. As a result, the unobserved claim problem mentioned in Figure 1 can be addressed, which leads to a better prediction of the future value of S_h.

3.4. Modeling $S_h|X_h$ with the MAR Covariate

The DPM model for complete case data $\{S_h, \boldsymbol{X}_h\}$ was discussed in Section 3.3. In this Section, we present our novel imputation strategy for the MAR covariate in the DPM framework in which the missing values are explained by the observed data and the cluster membership. We focus on the missingness in the binary type covariate. With the model definition in Equation (1), suppose the binary covariate $\boldsymbol{x}_1$ has missingness within it. To handle this MAR covariate, we suggest the following modifications (additional steps) to add to the DPM Gibbs sampler given in Algorithm A2:

(a) **Adding an imputation step in the parameter update stage:**
It is true that the missing covariate impacts on the parameter—$\boldsymbol{\theta}, \boldsymbol{w}$—update. For the parameters for the covariates $\boldsymbol{w}_j = \{\pi_j, \mu_j, \tau_j^2\}$, only the observations h without the missing covariate are used for updating. If the cluster does not have any observations with complete data for that covariate, then a draw from the prior distribution for $\{\pi_j, \mu_j, \tau_j^2\}$ would be used to update it. For the parameters for the outcome $\boldsymbol{\theta}_j = \{\boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j\}$, however, we must first impute values for the missing covariates x_{1h} for all observations h within the cluster j. Since we already defined a full joint model—$f(S_h|\boldsymbol{X}_h, \boldsymbol{\theta}_j) \cdot f(\boldsymbol{X}_h|\boldsymbol{w}_j)$—in Section 3.2, we can obtain draws for the MAR covariate x_{1h} from the imputation model, such as

$$f_{Bern}(x_{1h}|S_h, x_{2h}, \boldsymbol{\theta}_j, \boldsymbol{w}_j) \propto f(S_h|\boldsymbol{X}_h, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j) \cdot f_{Bern}(x_{1h}|\pi_j) \tag{8}$$

at each iteration in the Gibbs sampling. Each imputation model is proportional to the joint distribution as a product of the outcome model and the covariate model that has missing data. The imputation process is illustrated in depth in Figure 3. Once all missing covariate values have been imputed, then the parameters of each cluster $\boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j$ are recalculated and sampled from the posterior of $\boldsymbol{\theta}_j$. After this cycle is complete in the Gibbs sampling, the imputed data are discarded, and the same imputation steps are repeated for every iteration.

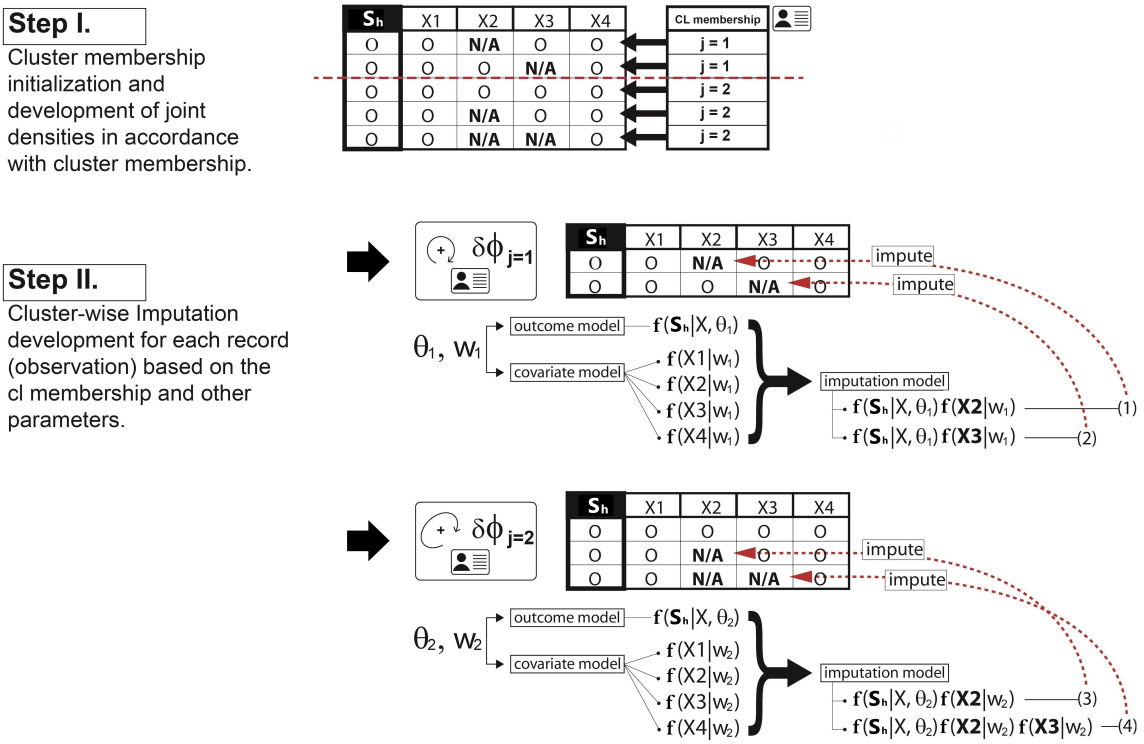

Figure 3. An example of the MAR imputation for the parameter update stage in the DPM Gibbs sampler for Step I and Step II. The imputations are made cluster membership-wise.

(b) **Adding a reclustering step in the cluster membership update stage:**
To calculate each cluster probability after the parameter updates, the algorithm redefines the two main components: (1) the covariate model and (2) the outcome model. For the covariate model $f(\boldsymbol{X}_h|\boldsymbol{w}_j)$, we set this equal to the density functions of only those covariates with complete data for observation h. Assuming that $\boldsymbol{X}_h = \{x_{1h}, x_{2h}\}$, and the covariate $\boldsymbol{x}_1$ is missing for observation h, then we drop x_{1h} and only use x_{2h} in the covariate model:

$$f(\boldsymbol{X}_h|\boldsymbol{w}_j) = f_N(x_{2h}|\boldsymbol{w}_{2j}) \tag{9}$$

This is the refined covariate model for the cluster j with the observation h, where the data in $\boldsymbol{x}_1$ are not available. For the outcome model $f(S_h|\boldsymbol{X}_h, \boldsymbol{\theta}_j)$, the algorithm simply takes the imputation model in Equation (8) for the observation h and integrates it out of the covariates with missingness x_{1h}. This reduces the degrees of variance introduced by the imputations. In other words, as the covariate $\boldsymbol{x}_1$ is missing for observation h, this missing covariate can be removed from the $\boldsymbol{X}_h$ term that it is being conditioned on. Therefore, the refined outcome model is

$$f(S_h|x_{2h}, \boldsymbol{\theta}_j) \propto \int f(S_h|\boldsymbol{X}_h, \boldsymbol{\theta}_j) \cdot f_{Bern}(x_{1h}|\boldsymbol{w}_{1j}) dx_{1h} \tag{10}$$

The same process is performed for each observation with missing data and each combination of missing covariates. Hence, using Equations (9) and (10), the cluster probabilities and the predictive distribution can be obtained as illustrated in Step III in Figure 4.

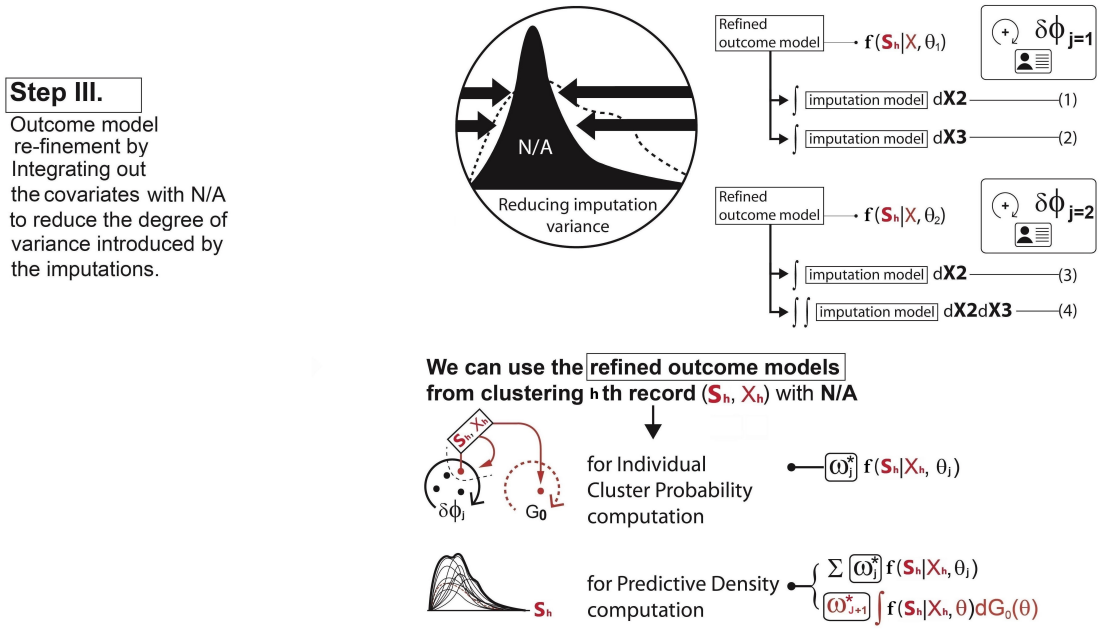

Figure 4. An example of the refined outcome model development for the cluster membership update stage in the DPM Gibbs sampler: Step III. Using these models, each cluster probability and the predictive density can be calculated.

(c) **Re-updating the parameters:**
The cluster probability computation is followed by the parameter reestimation for each cluster, which is illustrated via the diagram in Figure 5. This is the same idea as what we have discussed about the parameter $(\boldsymbol{\theta}, \boldsymbol{w})$ update in Section 3.3.

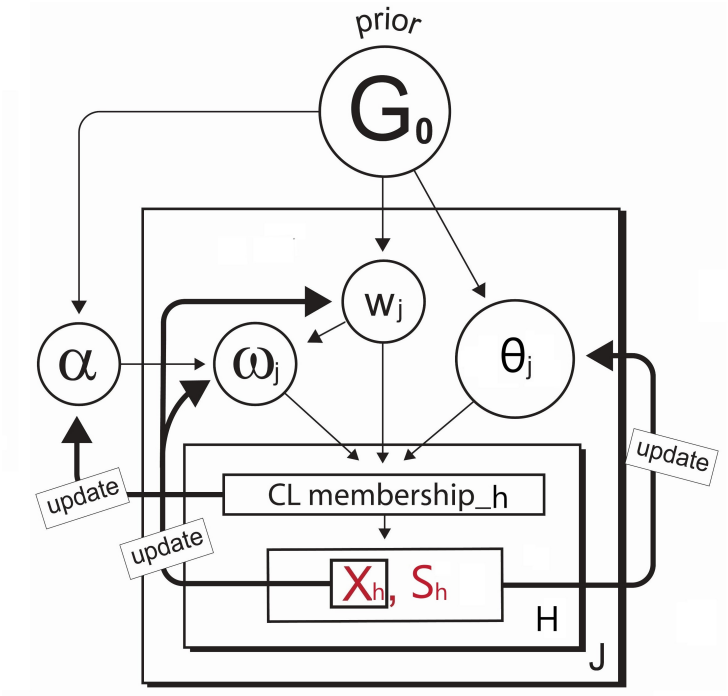

Figure 5. Parameter reestimation after the reclustering with imputation in the Gibbs sampler. This diagram articulates flows of the parameter updates using the acyclic graphical representation. The process cycles until achieving convergence.

3.5. Gibbs Sampler Modification in Detail for the MAR Covariate

Now, we set out some modifications for the DPM Gibbs sampler in Algorithm A2 to address the MAR covariate $\boldsymbol{x}_1$. We aim to provide the details of the DPM implementation integrated with the MAR imputation strategy discussed in Section 3.4. The Gibbs sampler will alternate between imputing missing data and drawing parameters until it reaches convergence. We elaborate below on the modifications that fit into Algorithm A2 to update the clustering scenarios and the posterior cluster parameters properly:

(a) In line 6, with the presence of a missing covariate x_{1h}, the modification of the cluster probability for the observation $(S_h, \cancel{x_{1h}}, x_{2h})$ that belongs to the discrete cluster j can be made as follows:

$$P(s_h = j) = p(s_h|s_{-h}) \cdot f(x_{2h}|\mu_j, \tau_j^2) \cdot f(S_h|x_{2h}, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j)$$

where $f(x_{2h}|\mu_j, \tau_j^2)$ is from Equation (12) and $f(S_h|x_{2h}, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j)$ is from Equation (14).

(b) In line 9, with the presence of a missing covariate x_{1h}, the modification of the cluster probability for the observation $(S_h, \cancel{x_{1h}}, x_{2h})$ that belongs to the continuous cluster $J+1$ can be made as follows:

$$P(s_h = J+1) = p(s_h|s_{-h}) \cdot f_0(x_{2h}) \cdot f_0(S_h|x_{2h})$$

where $f_0(x_{2h})$ is from Equation (13) and $f_0(S_h|x_{2h})$ is from Equation (15).

(c) In line 22, with the presence of a missing covariate x_{1h}, the imputation should be made before simulating the parameter $\boldsymbol{\theta}_j^*$ as follows:

$$\begin{cases} \begin{cases} \text{First, sample } x_{1h} \sim f(S_h|\boldsymbol{X}_h, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j) \cdot f_{Bern}(x_{1h}|\pi_j) \\ \text{Then sample } \boldsymbol{\theta}_j^* \text{from the posterior: } p(\boldsymbol{\theta}|S_h, \boldsymbol{X}_h) \end{cases} & \text{if } x_{1h} \text{ is missing.} \\ \text{Sample } \boldsymbol{\theta}_j^* \text{from the posterior: } p(\boldsymbol{\theta}|S_h, \boldsymbol{X}_h) & \text{otherwise} \end{cases}$$

The imputation model formulation above was discussed in Section 3.4.

Again, these modifications allow us to draw the missing covariate values from the conditional posterior density at each iteration using the Metropolis–Hastings algorithm with a random walk.

4. Bayesian Inference for $S_h|X_h$ with the MAR Covariate

In this section, we examine the parameter models and data models in depth to update the parameters of the DPM model given in Algorithm A2 under the assumption that the binary covariate x_1 is subject to missingness. The efficient simulation for the model parameters $\boldsymbol{\theta} : \{\boldsymbol{\beta}, \sigma^2, \xi\, \tilde{\boldsymbol{\beta}}\}$, $\boldsymbol{w} : \{\pi, \mu, \tau^2\}$, and α requires proper parameterization in the parameter models: the prior parameter model and posterior parameter model. The accurate estimations of cluster probabilities relies on the legitimate development of data models—the outcome model and covariate model—and the model parameter simulation results that govern the data model behaviors.

4.1. Parameter Models and the MAR Covariate

Our study is based on a three-level hierarchical structure. The first level regards data models such as the log skew-normal outcome model and the Bernoulli and Gaussian covariate models, the second level involves parameter models such as $p(\boldsymbol{\theta}|S_h, \boldsymbol{X}_h)$, $p(\boldsymbol{w}|\boldsymbol{X}_h)$ to explain the data, and the third level is developed from the generalized regression to explain the parameters or the related hyperparameters, such as $a_0, b_0, \nu_0, c_0, d_0, \mu_0, \tau_0^2, e_0, \gamma_0, g_0$ and h_0, to set a probabilistic distribution on the parameter vectors $\boldsymbol{\theta} = \{\boldsymbol{\beta}, \sigma^2, \xi, \tilde{\boldsymbol{\beta}}\}$,

$\boldsymbol{w} = \{\pi, \mu, \tau^2\}$. See Variable Definitions for further information on the variables. Given the model definition in Equation (1), we consider a set of conjugate parameter models due to its computational advantages (see Cairns et al. 2011). For $S_h \sim \delta(\boldsymbol{X}_h^T \tilde{\boldsymbol{\beta}}_j)\ \mathbb{1}(S_h = 0) + \left[1 - \delta(\boldsymbol{X}_h^T \tilde{\boldsymbol{\beta}}_j)\right]\ LogSN(\boldsymbol{X}_h^T \boldsymbol{\beta}_j, \sigma_j^2, \xi_j)$, $\boldsymbol{x}_1 \sim Bern(\pi_j)$, and $\boldsymbol{x}_2 \sim N(\mu_j,\ \tau_j^2)$, the prior models come in as

$$
\begin{aligned}
&p_0(\sigma_j^2|a_0,\ b_0) : \boldsymbol{InvGa}(a_0,\ b_0), && p_0(\boldsymbol{\beta}_j|\boldsymbol{\beta}_0,\ \Sigma_0) : \boldsymbol{MVN}(\boldsymbol{\beta}_0,\ \sigma_j^2\Sigma_0), && p_0(\xi_j|\nu_0) :\ \boldsymbol{T}(\nu_0)\\
&p_0(\tilde{\boldsymbol{\beta}}_j|\tilde{\boldsymbol{\beta}}_0,\ \tilde{\Sigma}_0) : \boldsymbol{MVN}(\tilde{\boldsymbol{\beta}}_0,\ \tilde{\Sigma}_0), && p_0(\pi_j|c_0,\ d_0) : \boldsymbol{\beta}(c_0,\ d_0), && p_0(\mu_j|\mu_0,\ \tau_0^2) :\ \boldsymbol{N}(\mu_0,\ \tau_j^2),\\
&p_0(\tau_j^2|e_0,\ \gamma_0) : \boldsymbol{InvGa}(e_0,\ \gamma_0), && p_0(\alpha|g_0,\ h_0) : \boldsymbol{Ga}(g_0,\ h_0)
\end{aligned}
$$

and their corresponding kernels chosen in this study are listed in Appendix A.1. Accordingly, the Dirichlet process prior (probability measure) G_0 in our case can be defined as $G_0 = \boldsymbol{MVN}(\boldsymbol{\beta}_0,\ \Sigma_0) \times \boldsymbol{InvGa}(a_0, b_0) \times \boldsymbol{T}(\nu_0) \times \boldsymbol{MVN}(\tilde{\boldsymbol{\beta}}_0,\ \tilde{\Sigma}_0) \times \boldsymbol{\beta}(c_0,\ d_0) \times \boldsymbol{N}(\mu_0,\ \tau_j^2) \times \boldsymbol{InvGa}(e_0,\ \gamma_0) \times \boldsymbol{Ga}(g_0,\ h_0)$. With a feed of the observed data inputs (S_h, x_{1h}, x_{2h}), the prior models for each cluster j described above will be updated into the following posterior models analytically apart from $\boldsymbol{\theta}_j = \{\boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j\}$:

$$
\begin{aligned}
&p(\pi_j|c_0,\ d_0,\ S,\ \boldsymbol{x}_1) : \boldsymbol{\beta}(c_{new},\ d_{new})\\
&p(\mu_j|\mu_0,\ \tau_0^2,\ S,\ \boldsymbol{x}_2) :\ \boldsymbol{N}(\mu_{new},\ \tau_{new}^2), \quad p(\tau_j^2|e_0,\ \gamma_0,\ S,\ \boldsymbol{x}_2) :\ \boldsymbol{InvGa}(e_{new},\ \gamma_{new}) \qquad (11)\\
&p(\alpha|g_0,\ h_0,\ h,\ J,\ \eta,\ \pi_\eta) :\ \pi_\eta \boldsymbol{Ga}(g_0 + J,\ h_0 - log(\eta)) + (1 - \pi_\eta)\boldsymbol{Ga}(g_0 + J - 1,\ h_0 - log(\eta))
\end{aligned}
$$

and their corresponding parameterizations are elaborated upon in Appendix A.2. Note that the value of the precision parameter α relies on the total cluster number J and thus does not vary by the cluster membership j, and its derivation of the posterior parameterization is not subject to the Bayesian conjugacy. Hence, we instead adapt the form of the posterior density for the α suggested by Escobar and West (1995), and its derivation is shown in Appendix C.1. As for $\boldsymbol{\theta}_j = \{\boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j\}$, there are no conjugate priors available for the log skew-normal likelihood, but their posterior samples can be secured by the conventional Metropolis–Hastings algorithm described in Algorithm A1 in Appendix A.

Considering that $\boldsymbol{x}_1$ has missing data, although the parameterizations of the posterior densities for the covariate parameter model of $\boldsymbol{w}$ and the precision α listed in Equation (11) are not affected, any outcome data of S_h with missingness should be dropped. Therefore, n_j and $\boldsymbol{x}_1$ are defined with the only observations in cluster j that are not missing. This imputation example is provided in Appendix C.2. For the outcome parameter model of $\boldsymbol{\theta}_j$, the missing covariate $\boldsymbol{x}_1$ must be imputed before its posterior computation shown in Algorithm A2. Once the parameters are updated with the imputation, the data models can be constructed as described in Equations (9) and (10).

4.2. Data Models and the MAR Covariate

Data models are the main components for the cluster probability computations depicted in Figure 2. As with the development of parameter models, the covariate data model of $\boldsymbol{X}$ ignores the observations with missingness, while the outcome data model of S_h requires completing the covariates beforehand. However, the formulation of their densities can be more complex due to the marginalization process with respect to the missing covariate. In addition, as discussed in Section 3.2, the data model development is bound by the types of clusters, such as discrete clusters $f(S_h|\boldsymbol{X}_h, \boldsymbol{\theta}_j)$, $f(\boldsymbol{X}_h|\boldsymbol{w}_j)$ and continuous clusters $f_0(S_h|\boldsymbol{X}_h)$, $f_0(\boldsymbol{X}_h)$:

(a) **Covariate model for the discrete cluster** $f(\boldsymbol{X}_h|\boldsymbol{w}_j)$
Focusing on the scenario where $\boldsymbol{x}_1$ is binary, $\boldsymbol{x}_2$ is Gaussian, and the only covariate with missingness is x_{1h}, we simply drop the covariate x_{1h} to develop the covariate model for the discrete cluster. For instance, when computing the covariate probability term for the hth observation in cluster j, the covariate model $f(x_{1h}, x_{2h}|\pi_j, \mu_j, \tau_j^2)$

simply becomes $f(x_{2h}|\mu_j, \tau_j^2)$ due to the missingness of x_{1h}. As we have $\boldsymbol{x}_2$, which is assumed to be normally distributed as defined in Equation (1), its probability term is

$$f(x_{2h}|\mu_j, \tau_j^2) = \frac{1}{\sqrt{2\pi\tau_j^2}} \exp\{\frac{-(x_{2h}-\mu_j)^2}{2\tau_j^2}\} \tag{12}$$

instead of

$$f(x_{1h}, x_{2h}|\pi_j, \mu_j, \tau_j^2) = \pi_j^{x_{1h}}(1-\pi_j)^{1-x_{1h}} \cdot \frac{1}{\sqrt{2\pi\tau_j^2}} \exp\{\frac{-(x_{2h}-\mu_j)^2}{2\tau_j^2}\}$$

(b) **Covariate model for the continuous cluster** $f_0(\boldsymbol{X}_h)$
If the binary covariate x_{1h} is missing, then by the same logic, we drop the covariate x_{1h} for the continuous cluster. However, using Equation (4), the covariate model for the continuous cluster integrates out the relevant parameters simulated from the Dirichlet process prior G_0 as follows:

$$\begin{aligned} f_0(x_{2h}) &= \int f(x_{2h}|\mu, \tau^2)\, dG_0(\mu, \tau^2) = \int f(x_{2h}|\mu, \tau^2) \cdot p(\mu|\tau^2) \cdot p(\tau^2)\, d\mu\, d\tau^2 \\ &= \frac{\gamma_0^{e_0}\Gamma(e_0+1/2)}{2\sqrt{\pi}\Gamma(e_0)}\Big(\gamma_0 + \frac{(x_{2h}-\mu_0)^2}{4}\Big)^{-(e_0+1/2)} \end{aligned} \tag{13}$$

instead of

$$\begin{aligned} f_0(x_{1h}, x_{2h}) &= \int f(x_{1h}, x_{2h}|\pi, \mu, \tau^2) \cdot p(\pi) \cdot p(\mu|\tau^2) \cdot p(\tau^2)\, d\pi\, d\mu\, d\tau^2 \\ &= \frac{\boldsymbol{B}(x_{1h}+c_0,\ 1-x_{1h}+d_0)}{\boldsymbol{B}(c_0, d_0)} \cdot \frac{\gamma_0^{e_0}\Gamma(e_0+1/2)}{2\sqrt{\pi}\Gamma(e_0)}\Big(\gamma_0 + \frac{(x_{2h}-\mu_0)^2}{4}\Big)^{-(e_0+1/2)} \end{aligned}$$

The derivation of the distributions above is provided in Appendix C.3.

(c) **Outcome model for the discrete cluster** $f(S_h|\boldsymbol{X}_h, \boldsymbol{\theta}_j)$
In developing the outcome model, as with the parameter model case discussed in Section 4.1 and Appendix C.2, it should be ensured that the covariate is complete beforehand. With all missing data in x_{1h} imputed, the outcome model for the discrete cluster is obtained by marginalizing the joint $f(S_h, x_{1h}|x_{2h}, \boldsymbol{\theta}_j, \pi_j)$ out the MAR covariate x_{1h}, which is a log skew-normal mixture expressed as follows:

$$\begin{aligned} f(S_h|x_{2h}, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j) &= \sum_{x_{1h}=0}^{1} f(S_h|x_{1h}, x_{2h}, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j) \cdot f(x_{1h}|\pi_j) \\ &= f(S_h,\ x_{1h}=1|x_{2h}, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j, \pi_j) + f(S_h,\ x_{1h}=0|x_{2h}, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j, \pi_j) \\ &= \delta(\boldsymbol{X}_h^T\tilde{\boldsymbol{\beta}}_j)\ \mathbb{1}(S_h=0) + \big[1-\delta(\boldsymbol{X}_h^T\tilde{\boldsymbol{\beta}}_j)\big] \cdot \frac{2}{\sigma_j S_h} \\ &\cdot \phi\left(\frac{\log S_h - (\beta_{j0}+\beta_{j1}+\beta_{j2}x_{2h})}{\sigma_j}\right) \cdot \Phi\left(\xi_j\frac{\log S_h - (\beta_{j0}+\beta_{j1}+\beta_{j2}x_{2h})}{\sigma_j}\right)\pi_j \\ &+\ \delta(\boldsymbol{X}_h^T\tilde{\boldsymbol{\beta}}_j)\ \mathbb{1}(S_h=0) + \big[1-\delta(\boldsymbol{X}_h^T\tilde{\boldsymbol{\beta}}_j)\big] \cdot \frac{2}{\sigma_j S_h} \\ &\cdot \phi\left(\frac{\log S_h - (\beta_{j0}+\beta_{j2}x_{2h})}{\sigma_j}\right) \cdot \Phi\left(\xi_j\frac{\log S_h - (\beta_{j0}+\beta_{j2}x_{2h})}{\sigma_j}\right) \cdot (1-\pi_j) \end{aligned} \tag{14}$$

instead of

$$f(S_h|x_{1h}, x_{2h}, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j)$$
$$= \delta(\boldsymbol{X}_h^T \tilde{\boldsymbol{\beta}}_j)\ \mathbb{1}(S_h = 0) + \left[1 - \delta(\boldsymbol{X}_h^T \tilde{\boldsymbol{\beta}}_j)\right] \cdot \frac{2}{\sigma_j S_h}$$
$$\cdot \phi\left(\frac{\log S_h - (\beta_{j0} + \beta_{j1} x_{1h} + \beta_{j2} x_{2h})}{\sigma_j}\right) \cdot \Phi\left(\xi_j \frac{\log S_h - (\beta_{j0} + \beta_{j1} x_{1h} + \beta_{j2} x_{2h})}{\sigma_j}\right)$$

(d) **Outcome model for the continuous cluster** $f_0(S_h|\boldsymbol{X}_h)$
Once a missing covariate $\boldsymbol{x}_1$ is fully imputed, and the outcome model is marginalized out and conditioned to the MAR covariate x_{1h}, the outcome model $f_0(S_h|x_{2h})$ for the continuous cluster can also be computed by integrating out the relevant parameters using Equation (4):

$$f_0(S_h|x_{2h}) = \int f(S_h|x_{2h}, \boldsymbol{\beta}, \sigma^2, \xi, \tilde{\boldsymbol{\beta}}) \cdot p(\boldsymbol{\beta}) \cdot p(\sigma^2) \cdot p(\xi) \cdot p(\tilde{\boldsymbol{\beta}})\ d\boldsymbol{\beta}\ d\sigma^2\ d\xi\ d\tilde{\boldsymbol{\beta}} \quad (15)$$

However, it can be too complicated to compute its form analytically. Instead, we can integrate the joint model out of the parameters using Monte Carlo integration. For example, we can perform the following steps for each $h = 1, \cdots, H$:

(i) Sample $\boldsymbol{\beta}, \sigma^2, \xi, \tilde{\boldsymbol{\beta}}$ from the DP prior densities G_0 specified previously;
(ii) Plug these samples into $f(S_h|x_{2h}, \boldsymbol{\beta}, \sigma^2, \xi, \tilde{\boldsymbol{\beta}}) \cdot p(\boldsymbol{\beta}) \cdot p(\sigma^2) \cdot p(\xi) \cdot p(\tilde{\boldsymbol{\beta}})$;
(iii) Repeat the above steps many times, recording each output;
(iv) Divide the sum of all output values by the number of Monte Carlo samples, which will be the approximate integral.

5. Empirical Study

5.1. Data

The performance of our DPM framework is assessed based on two insurance datasets. They highlight data difficulties such as unobservable heterogeneity in an outcome variable and MAR covariates. For simplicity, in each dataset, we only consider two covariates—one binary and one continuous—to explain its loss information (outcome variable). In this study, all computations on these two datasets are performed in the same data format:

$$\begin{array}{lc} & Year_1 \quad Year_2 \quad \cdots, \quad Year_y \\ \text{Policy (a):} & \{(S_a,\ \boldsymbol{X}_a),\ (S_a,\ \boldsymbol{X}_a),\ \cdots, (S_a,\ \boldsymbol{X}_a)\} \\ \text{Policy (b):} & \{(S_b,\ \boldsymbol{X}_b),\ (S_b,\ \boldsymbol{X}_b),\ \cdots, (S_b,\ \boldsymbol{X}_b)\} \\ & \vdots \\ \text{Policy (H):} & \{(S_H, \boldsymbol{X}_H),\ (S_H, \boldsymbol{X}_H),\ \cdots, (S_H, \boldsymbol{X}_H)\} \end{array}$$

The first dataset is **PnCdemand**, which is about the international property and liability insurance demand of 22 countries over 7 years from 1987 to 1993. Secondly, we use a dataset drawn from the Wisconsin Local Government Property Insurance Fund (LGPIF) with information about the insurance coverage for government building units in Wisconsin for the years from 2006 to 2010. The first one—**PnCdemand**—can be obtained from the R package **CASdatasets**. The dataset is relatively small as it has $H = 240$ cases with an outcome variable *GenLiab*, the individual loss amount under the policies of general insurance for each case. As for the covariates, we consider one indicator variable of the statutory law system (*LegalSyst*: one or zero) and one continuous variable that measures a risk aversion rate (*RiskAversion*) for each area. For additional background on this dataset, see the work of Browne et al. (2000). In the LGPIF dataset, the insurance coverage samples for the government properties from $H = 5660$ policies are provided. The outcome variable is the sum of all types of losses (*total losses*) for each policy. Only the covariates—*LnCoverage*

and *Fire5*—are considered in our study. *Fire5* is a binary covariate that indicates fire protection levels, while *LnCoverage* is a continuous covariate that informs a total coverage amount in a logarithmic scale. For further details, see the work of Quan and Valdez (2018).

Histograms of the losses of the two datasets are exhibited in Figure 6. Due to the significant skewness, the loss data were log-transformed to attain Gaussianity. As shown in the histograms, each distribution displayed different characteristics in regard to skewness, modality, excess of zeros, etc. Note that the zero-inflated outcome variable in the LGPIF data (Figure 6(b1,b2)) required a two-part modeling technique that distinguished the probabilities of the outcome being zero and positive.

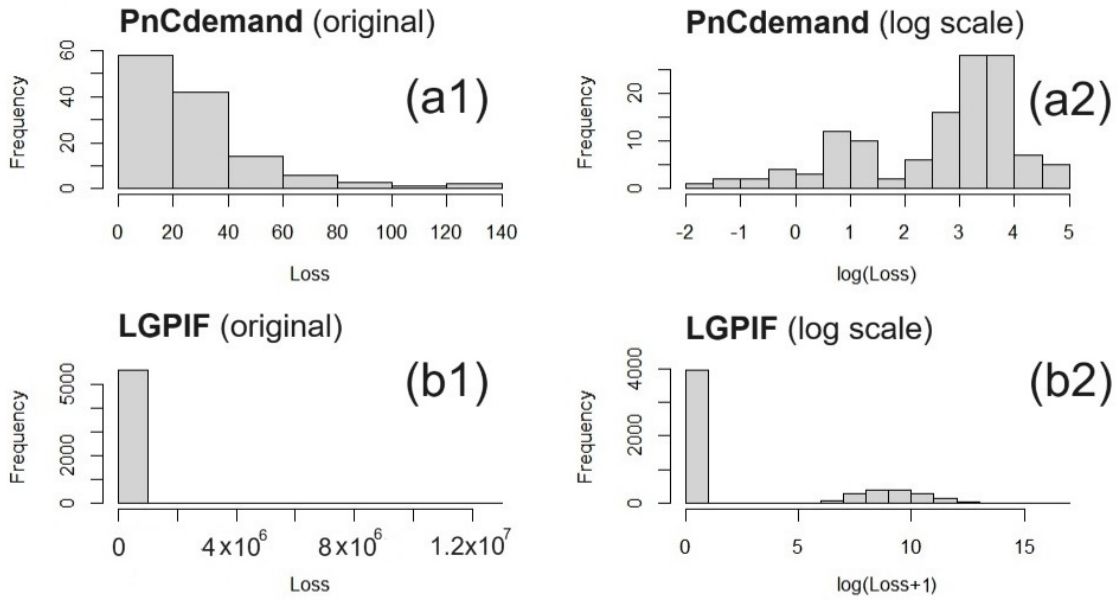

Figure 6. Histograms of the outcomes and log-transformed outcomes for the two datasets: (**a1**,**a2**) **PnCdemand** and (**b1**,**b2**) LGPIF.

5.2. Three Competitor Models and Evaluation

Our DPM framework is compared to other commonly used actuarial models in practice. We employ three predictive models as benchmarks, namely a generalized linear mixture model (GLM), multivariate adaptive regression spline (MARS), and generalized additive model (GAM). In each dataset, we assume different distributions for the outcome variables, and thus the three benchmark models are built upon the different outcome data models. For example, the **PnCdemand** dataset (a1,a2) that appeared in Figure 6 had a high frequency of small losses without zero values, and hence it was safe to use a gamma mixture to explain the outcome data. As for the LGPIF data in Figure 6(b1,b2), we considered the outcome data model based on a Tweedie distribution to accommodate the zero-inflated loss data. The benchmark models were implemented in R with the **mgcv**, **splines**, and **mice** packages.

All four models were trained, and investigations were performed in terms of model fit, prediction accuracy, and the conditional tail expectation (CTE) of the predictive distribution. Note that the goodness of fit value for the DPM is not available. Teh (2010) argued that the goodness of fit evaluation for the DPM is unnecessary, as underfitting is mitigated by the unbounded complexity of the DPM while overfitting is alleviated by the approximation of the posterior densities over each parameter in the DPM. Gelman and Hill (2007) pointed out that the *posterior predictive check*, which compares the simulated data under the fitted DPM to the observed data, can be useful for studying model adequacy, but its usage cannot be for model comparison. Therefore, the goodness of fit was only compared between the rival models.

For the evaluation of prediction performance, the sum of square prediction error (SSPE) $\sum_{h=1}^{H}(g(\boldsymbol{X}_h) - S_h)^2$ and the sum of square absolute error (SAPE) $\sum_{h=1}^{H}|g(\boldsymbol{X}_h) - S_h|$ were used in order to measure the differences between the predicted value $g(\boldsymbol{X}_h)$ and actual value S_h. The SSPE penalizes large deviations much more than the SAPE. We preferred the SAPE over the SSPE because our data were heavily skewed, which could result in outliers occurring more often. In the distribution fitting problem, each data point had equal importance, and we did not need to penalize larger error values that could arise from the outliers.

5.3. Result with International General Insurance Liability Data

For this dataset, a training set of a response and covariate pair $(Y, \boldsymbol{X})$ with $n = 160$ records and a test set of a response and covariate pair $(Y', \boldsymbol{X}')$ with $m = 80$ records were constructed. We implemented the following DPM:

$$\begin{aligned} Y_h|x_{1h}, x_{2h}, \boldsymbol{\beta}_j, \sigma_j^2 &\sim \ LogN\left(\boldsymbol{X}_h^T\boldsymbol{\beta}_j,\ \sigma_j^2\right) \\ x_{1h}|\pi_j &\sim Bern\left(\pi_j\right) \\ x_{2h}|\mu_j, \tau_j^2 &\sim N\left(\mu_j,\ \tau_j^2\right) \\ \{\boldsymbol{\theta}_j,\ \boldsymbol{w}_j\} &\sim G \\ G &\sim DP\left(\alpha,\ G_0\right) \end{aligned}$$

A log-normal likelihood was chosen to accommodate the individual loss Y_h:*GenLiab* for a policy h. The covariate x_2, *RiskAversion*, was subject to missingness and found to depend on Y_h (a MAR case). This was addressed by the internalized imputation process as discussed in Figure 3. The posterior parameters of $\boldsymbol{\theta}_j, \boldsymbol{w}_j$ were estimated with our DPM Gibbs sampler presented in Algorithm A2. The algorithm ran 10,000 iterations until convergence, and the resulting scenarios of the clustering mixture are shown in Figure 7. The plot reveals the overlays of predictive densities on the log scale from the last 100 iterations that were tied to convergence. Figure 8 lists the classical data imputation result using the multivariate imputation chained equation (MICE) and the predictive densities produced from our rival models: GLM, GAM, and MARS.

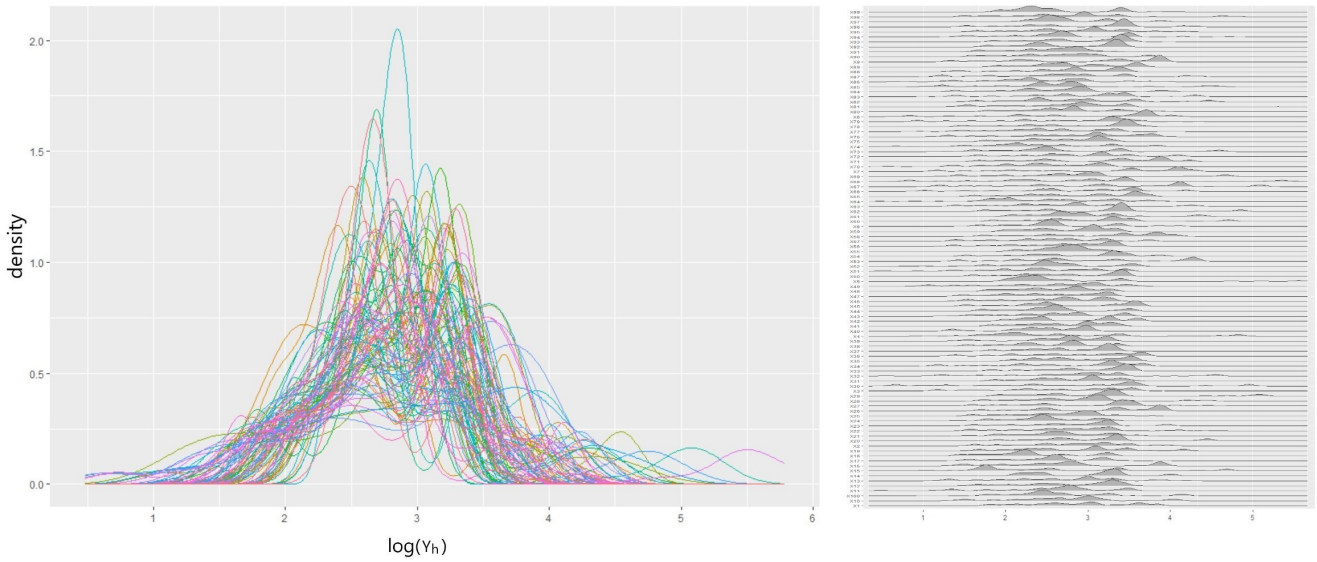

Figure 7. LogN-DPM with the **PnCdemand** dataset, with the last 100 in-sample predictive densities (scenarios) overlaid together.

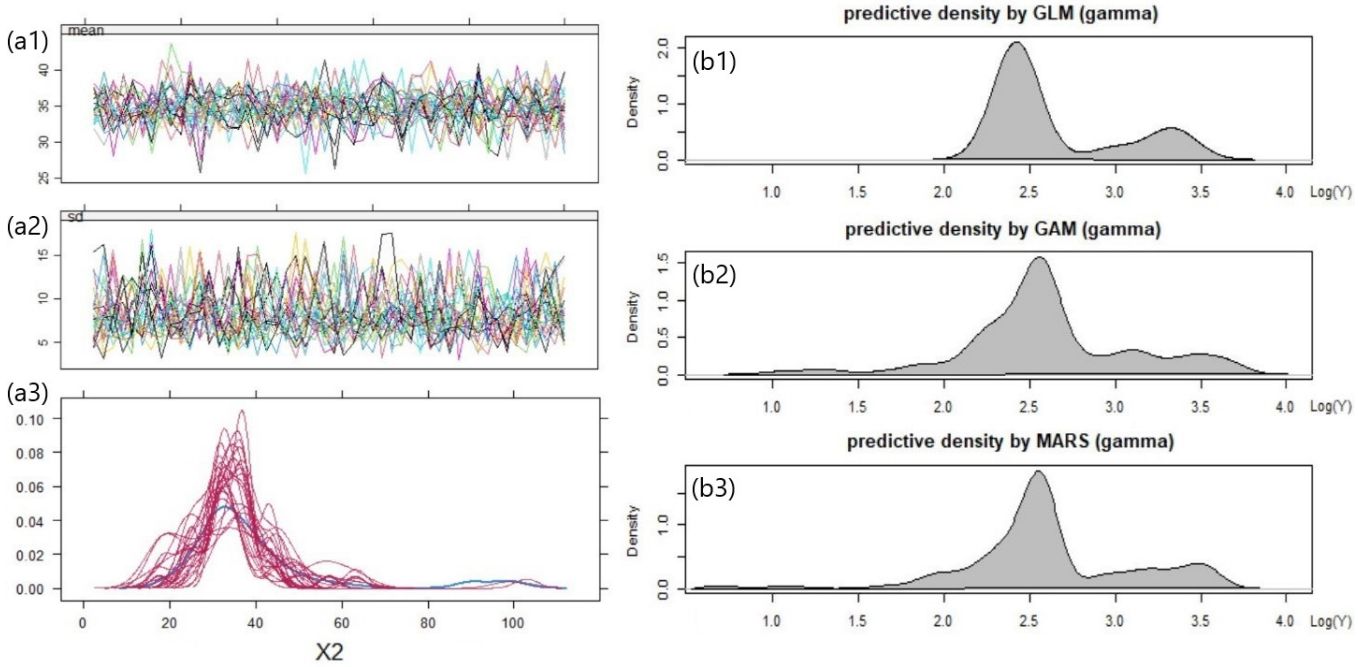

Figure 8. MICE trace plots (**a1**,**a2**), the imputation comparison plot for the MAR covariate (**a3**), and in-sample predictive densities produced from GLM, GAM, and MARS (**b1**–**b3**) with the **PnCdemand** dataset.

The MICE runs multiple imputation chains and selects the imputation values from the final iteration. This process results in multiple candidate datasets. The trace plots (a1,a2) monitor the imputation mean and variance for the missing values in the dataset. In the covariate distribution plot (a3), the density of the observed covariate, shown in blue, is compared with the ones of the imputed covariate for each imputed dataset, shown in red. The parameter inferences for the rival models were performed based on the imputed datasets tied to convergence (see Shah et al. 2014). The gamma distribution was chosen to fit the rival models as Y_h was continuous and positively skewed with a constant coefficient of variation. The gamma-based predictive density plots (b1,b2,b3) estimated with GLM, GAM, and MARS look similar, showing unusual bumps near the right tail.

In Figure 9, a histogram of the outcome data in the test set is displayed. The posterior mean densities for the out-of-sample predictions produced with our DPM along with the rival models' density estimates are overlaid on the histogram. Judging from the plot, one can say that our DPM model generated the best approximation. While the rival models generated smooth, mounded curves to make predictions, our DPM captured all possible peaks and bumps, which was closer to the actual situation.

According to Table 1, our DPM obtained the highest SSPE compared with other rival models. At first glance, our DPM might seem like a failure. However, upon closer inspection, it becomes evident that the presence of outliers greatly influenced its performance. Remarkably, our DPM excelled at capturing these outliers, leading to the highest SSPE. This is evidenced by the lowest SAPE of our DPM. In other words, as the SAPE weights all the individual differences equally, we can assume that the rival models tend to pay too much attention to the most probable data points and miss the majority of outliers. This can be mainly due to the insufficient sample size as well. However, our DPM had good performance under small sample sizes as long as there was sufficient prior knowledge available. From the perspective of CTE, Table 1 shows that our DPM proposed a heavier tail than other rival models, which reflects that our DPM captured more uncertainties given the small sample size.

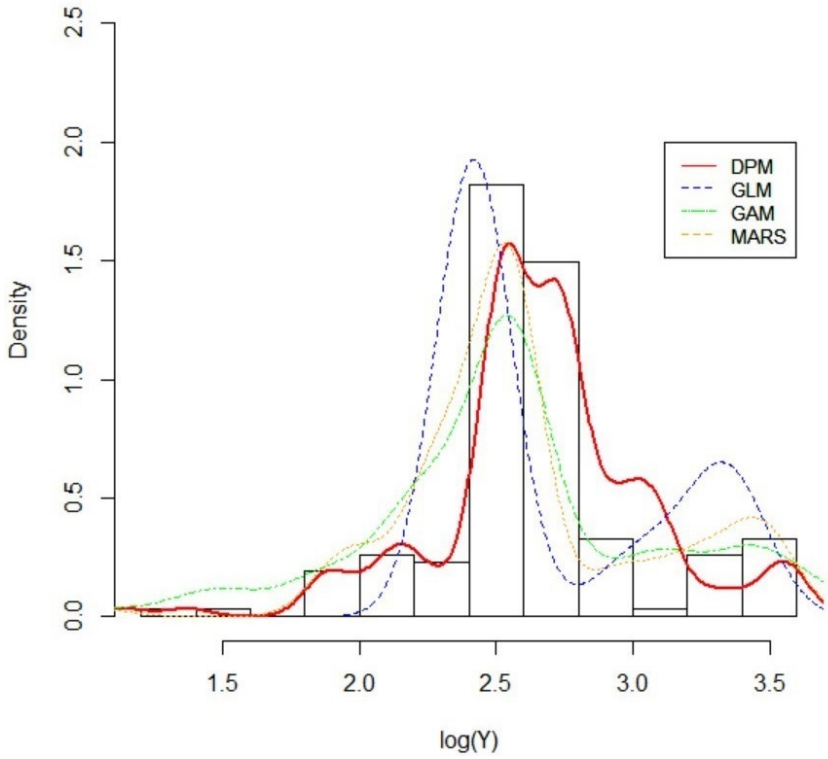

Figure 9. A histogram of the observed loss Y_h on the log scale and the out-of-sample predictive densities for the typical class of a policy in the **PnCdemand** dataset.

Table 1. The comparison of out-of-sample modeling results based on the dataset **PnCdemand**.

Model	AIC	SSPE	SAPE	10% CTE	50% CTE	90% CTE	95% CTE
Ga-GLM	830.56	268.6	139.8	6.5	13.8	54.5	78.0
Ga-MARS	830.58	267.2	138.2	6.1	13.0	57.2	71.1
Ga-GAM	845.94	266.7	136.1	6.2	13.3	58.1	72.2
LogN-DPM	-	272.0	134.7	6.4	13.8	59.3	79.3

5.4. Result with LGPIF Data

For this dataset, a training set of a response and covariate pair $(S, \boldsymbol{X})$ with $n = 4529$ records and a test set of a response and covariate pair $(S', \boldsymbol{X}')$ with $m = 1110$ records were constructed. We implemented the following DPM:

$$
\begin{aligned}
S_h|x_{1h},x_{2h},\boldsymbol{\beta}_j,\sigma_j^2,\xi_j,\tilde{\boldsymbol{\beta}}_j & \\
&\sim \delta(\boldsymbol{X}_h^T\tilde{\boldsymbol{\beta}}_j)\ \mathbb{1}(S_h=0) + \left[1-\delta(\boldsymbol{X}_h^T\tilde{\boldsymbol{\beta}}_j)\right]\ LogSN\left(\boldsymbol{X}_h^T\boldsymbol{\beta}_j,\ \sigma_j^2,\ \xi_j\right)\\
x_{1h}|\pi_j &\sim Bern\left(\pi_j\right)\\
x_{2h}|\mu_j,\tau_j^2 &\sim N\left(\mu_j,\ \tau_j^2\right)\\
\{\boldsymbol{\theta}_j,\ \boldsymbol{w}_j\} &\sim G\\
G &\sim DP\left(\alpha,\ G_0\right)
\end{aligned}
$$

As the outcome S_h, *total losses*, for a policy h in this dataset was considered to be distributed with the sum of the log-normal densities, a log skew-normal likelihood was chosen to approximate this convolution (see Li 2008). The covariate x_1, *Fire5*, was subject to missingness under the MAR condition, and the internalized imputation process illustrated in Figure 3 resolved this issue without creating imputed datasets. As the outcome S_h exhibited zero inflation, we employed a two-part model using a sigmoid and indicator function. Our DPM Gibbs sampler described in Algorithm A2 produced the posterior

parameters of $\boldsymbol{\theta}_j, \boldsymbol{w}_j$ with 10,000 iterations until convergence. Figure 10 reveals the resulting scenarios of the clustering mixture. In the plot, there are 100 predictive densities suggested by our DPM, each of which stands for the convergence of the estimation results.

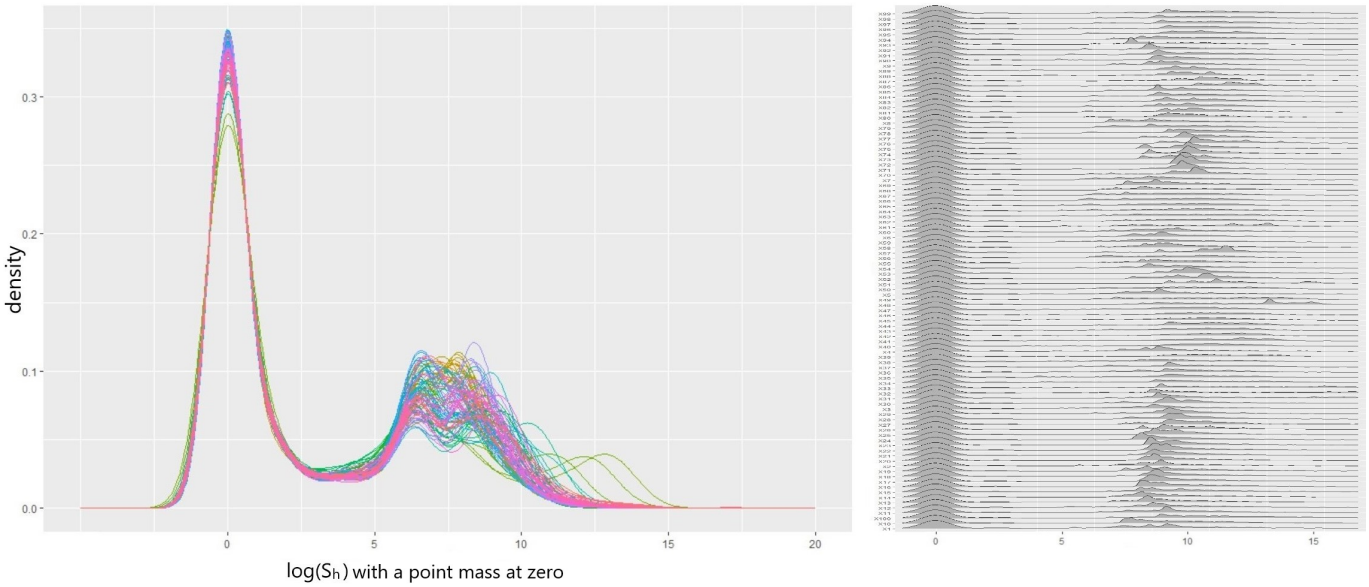

Figure 10. LogSN-DPM with the LGPIF dataset, with the last 100 in-sample predictive densities (scenarios) overlaid together.

The output of the MICE and the resulting predictive densities from the rival models are displayed in Figure 11. The rival models were built upon a Tweedie distribution due to its ability to account for a large number of zero losses and the flexibility to capture the unique loss patterns of the different classes of policyholders. According to the plot, all three rival models reasonably captured zero inflation, but the GAM tended to suggest more bumps that indicated a need for further assessment of the prediction uncertainty.

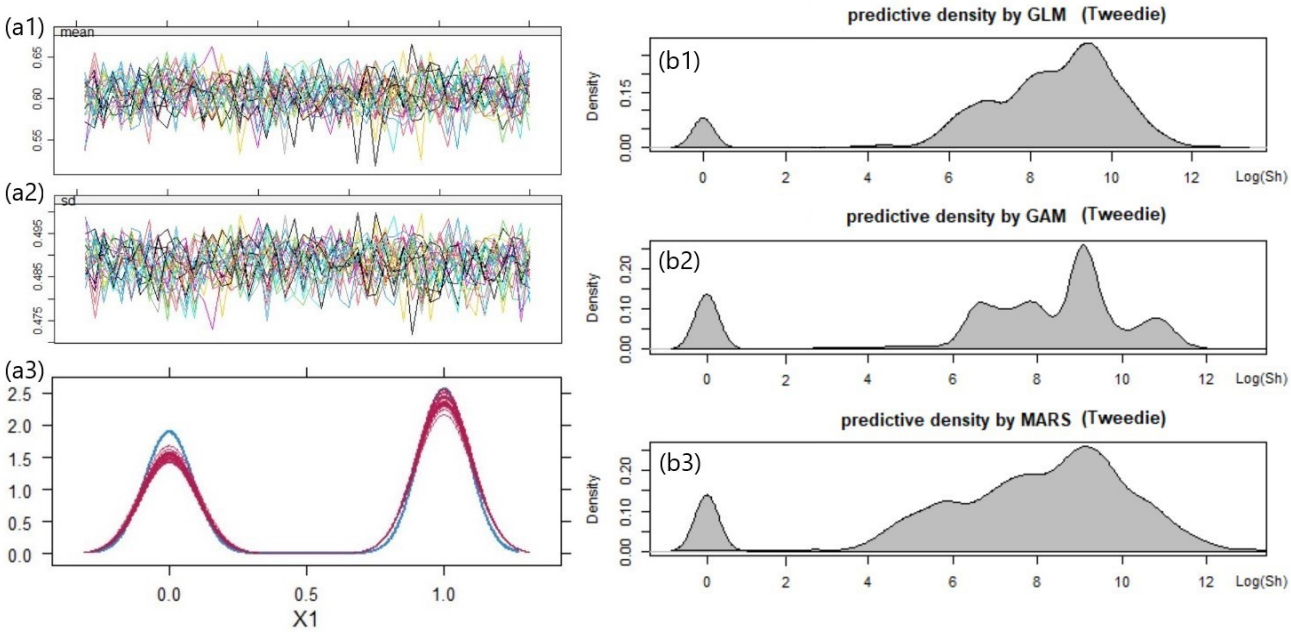

Figure 11. MICE trace plots (**a1**,**a2**), the imputation comparison plot for the MAR covariate (**a3**), and in-sample predictive densities produced from GLM, GAM, and MARS (**b1**–**b3**) with the LGPIF dataset.

The overall out-of-sample prediction comparison is made in the histogram overlayed with predictive density curves generated from the four models in Figure 12. From the plot, it is apparent that the posterior predictive density proposed by our DPM best explained the new samples, while other rival models kept producing multiple peaks.

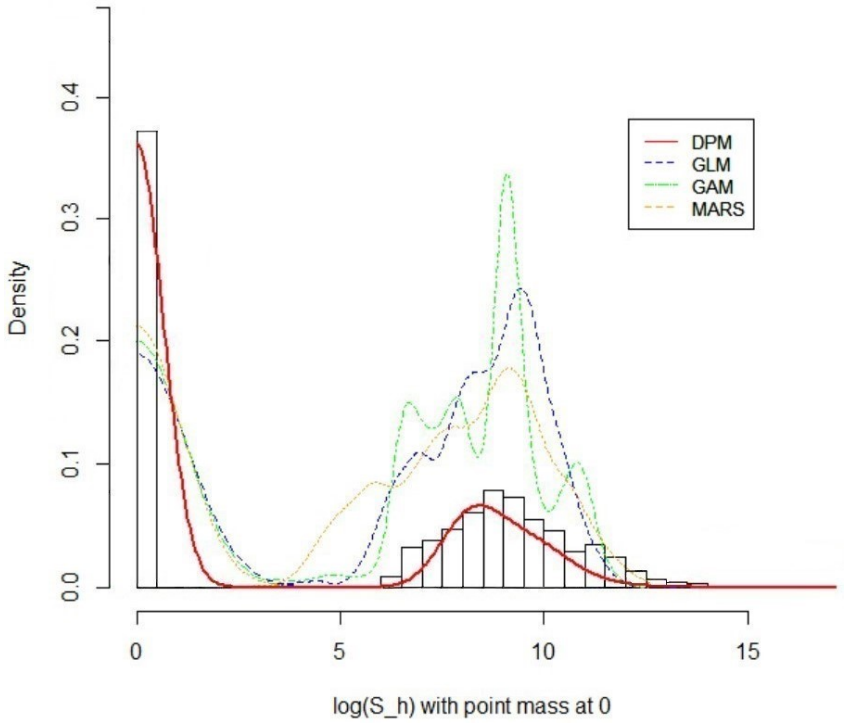

Figure 12. A histogram of the observed total loss S_h on the log scale and the out-of-sample predictive densities for the typical class of a policy in the LGPIF dataset.

The improved prediction performance of our DPM is confirmed by the smallest SAPE in Table 2. However, as for the SSPE, our DPM showed the second-highest performance, being slightly lower than that of the GAM. This is mainly due to the ability of our DPM to capture outliers more often, which is heavily penalized by the SSPE that squares the term. In terms of CTE, all three rival models suggested a similar level of tailedness, reflecting the knowledge obtained from the observed data. However, our DPM went beyond this and proposed a much heavier tail. This was because our DPM accommodated the presence of outliers and shaped the tail behavior based on the combined knowledge of the prior parameters and the observations available.

Table 2. The comparison of out-of-sample modeling results based on the LGPIF dataset.

Model	AIC	SSPE	SAPE	10% CTE	50% CTE	90% CTE	95% CTE
Tweedie-GLM	26,270.3	2.04×10^{14}	89,380,707	955.9	12,977.2	133,374.4	340,713.1
Tweedie-MARS	24,721.4	1.99×10^{14}	88,594,850	961.7	10,391.0	129,409.2	355,112.6
Tweedie-GAM	21,948.9	1.95×10^{14}	88,213,987	989.4	13,026.2	140,199.5	398,263.1
LogSN-DPM	-	1.98×10^{14}	83,864,890	975.3	13,695.1	147,486.6	425,682.6

6. Discussion

This paper proposes a novel DPM framework for actuarial practice to model total losses with the incorporation of MAR covariates. Both the log-normal and log skew-normal DPM presented overall good empirical performances in capturing the shape of the distribution, out-of-sample prediction, and the estimation of the tailedness. This suggests

that it is worth considering our DPM framework in order to avoid various model risks or biases in insurance claim analysis.

6.1. Research Questions

Regarding **RQ1**, we proposed a DPM framework to address the within-cluster heterogeneity emerging from the inclusion of covariates. By allowing for an infinite number of clustering scenarios determined by the observations as well as prior knowledge, our DPM outperformed the rival methods in drawing the lines for the cluster membership. This can be assessed by examining the homogeneity of the resulting clusters. In our case, we fit cluster-wise GLMs (based on gamma and Tweedie distributions) to the data points within each resulting cluster to compare the goodness-of-fit, and the consistent AICs across all clusters endorse the benefits of the DPM. Similarly, our rival methods, such as GAM or MARS, can capture heterogeneity by using customized smooth functions across different subsets of the data, but we observed some statistically insignificant smooth terms, indicating the presence of heterogeneity in the cluster.

In terms of **RQ2**, we suggest incorporating the imputation steps into the parameter and cluster membership update process in the DPM Gibbs sampler by leveraging the joint distribution of the observed outcomes and missing covariates. This approach allows the imputed values to be consistent with the observed data, preserving the correlation structure within the dataset. In order to make a comparison of our approach with an existing alternative, we additionally employed a chained equation technique. The multiple sets of imputed values simulated from both approaches were investigated, and the results show that our DPM Gibbs sampler did not represent a significant improvement over the chained equation because their average estimates of the imputed values were closer to each other. However, we feel that this result was mainly due to the relatively low dimensionality of the datasets we used and their simple data structure. The specific characteristics or dependencies in the data and the complexity of the missing patterns would give different results in practice.

As for **RQ3**, we fit a log skew-normal density to the aggregate loss outcomes. In order to assess its performance, one can consider minimax approximation, least squares approximation, log-shifted gamma approximation, etc. as the competitors. Li (2008) provided a useful comparison between these competitors by overlaying the cumulative density curves for each technique, but the experiments were grounded in the simulated log-normal data with the predefined parameters and assumptions, which cannot be easily controlled in real-world scenarios. Therefore, we feel that the choice of the best approximation technique should be made based on the identification of the specific characteristics of the dataset. In our case, each summand in our dataset was significantly different from each other in magnitude (the minimax approach was inappropriate), and the LGPIF data had a large volume of data smaller than five (the log-shifted gamma was inappropriate). Therefore, we chose a log skew-normal density that was relatively simple while giving an accurate approximation at the lower region of the distribution.

6.2. Future Work

There are several concerns with our log skew-normal DPM framework:

(a) **Dimensionality:** First, in our analysis, we only used two covariates (binary and continuous) for simplicity. Hence, more complex data should be considered. As the number of covariates grows, the likelihood components (covariate models) to describe the covariates grow, which results in the shrinking of the cluster weights. Therefore, using more covariates might enhance the level of sensitivity and accuracy in the creation of cluster memberships. However, it can also introduce more noise or hidden structures that render the resulting predictive distributions unstable. In this sense, further research on the problem of high dimensional covariates in the DPM framework would be worthwhile.

(b) **Measurement error:** Second, although our focus in this article was the MAR covariate, mismeasured covariates is an equally significant challenge that impairs the proper model development in insurance practice. For example, Aggarwal et al. (2016) pointed out that "model risk" mainly arises due to missingness and measurement error in variables, leading to flawed risk assessments and decision making. Thus, further investigation is necessary to explore the specialized construction of the DPM Gibbs sampler for mismeasured covariates, aiming to prevent the issue of model risk.

(c) **Sum of the log skew-normal:** Third, as an extension to the approximation of total losses S_h (the sum of individual losses) for a policy, we recommend researching ways to approximate the sum of total losses $\tilde{S}$ across entire policies. In other words, we pose the following question: "How do we approximate the sum of log skew-normal random variables?" From the perspective of an executive or an entrepreneur whose concern is the total cash flow of the firm, nothing might be more important than the accurate estimation of the sum of total losses in order to identify the insolvency risk or to make important business decisions.

(d) **Scalability:** Lastly, we suggest investigating the scalability of the posterior simulation with our DPM Gibbs sampler. As shown in our empirical study on the **PnCdemand** dataset, our DPM framework produced reliable estimates with relatively small sample sizes ($n \leq 160$). This was because our DPM framework actively utilized significant prior knowledge in posterior inference rather than heavily relying on the actual features of the data. In the result from the LGPIF dataset, our DPM exhibited stable performance at a sample size $n = 4529$ as well. However, a sample size of over 10,000 was not explored in this paper. With increasing amounts of data, our DPM framework raises the question of computational efficiency due to the growing demand for computational resources or degradation in performance (see Ni et al. 2020). This is an important consideration, especially in scenarios where the insurance loss information is expected to grow over time.

Author Contributions: All authors contributed substantially to this work. Conceptualization, M.K.; methodology, M.K., D.L., M.B. and M.C.; software, M.K. and D.L.; validation, M.K., M.B. and M.C.; formal analysis, M.K.; investigation, M.K., M.B. and M.C.; resources, M.K.; data curation, M.K.; writing—original draft preparation, M.K.; writing—review and editing, M.K., M.B. and M.C.; visualization, M.K.; supervision, M.B. and M.C.; project administration, M.B. and M.C.; funding acquisition, M.C. All authors have read and agreed to the published version of this manuscript.

Funding: This research was conducted with financial support from the Science Foundation Ireland under Grant Agreement No.13/RC/2106 P2 at the ADAPT SFI Research Centre at DCU. ADAPT, the SFI Research Centre for AI-Driven Digital Content Technology, is funded by the Science Foundation Ireland through the SFI Research Centres Programme.

Data Availability Statement: Data and implementation details are available at https://github.com/mainkoon81/Paper2-Nonparametric-Bayesian-Approach01 (accessed on 25 May 2023).

Acknowledgments: We extend our appreciation to the Editor, Associate Editor, and anonymous referee for their thorough review of this paper and their valuable suggestions, which have greatly contributed to its enhancement.

Conflicts of Interest: The authors declare no conflict of interest.

Variable Definitions

The following variables and functions are used in this manuscript:

$i = 1, \ldots, N_h$	Observation index i in a policy h
$h = 1, \ldots, H$	Policy index h with a total policy number H
$j = 1, \ldots, J$	Cluster index for J clusters
s_h	Cluster index $j = 1, \ldots, J$ for observation h
n_j	Number of observations in cluster j

n_j^{-h}	Number of observations in cluster j where observation h was removed from
Y_{ih}	Individual loss i in a policy observation h
S_h	Outcome variable which is ΣY_{ih} in a policy observation h.
$\tilde{S}$	Outcome variable which is ΣS_h across entire policies
$\boldsymbol{X}_h$	Vector of covariates (including $\boldsymbol{x}_1, \boldsymbol{x}_2$) for a policy observation h
$\boldsymbol{x}_1$	Vector of covariate (Fire5)
$\boldsymbol{x}_2$	Vector of covariate (Ln(coverage))
x_{1h}	Individual value of covariate (Fire5) for a policy observation h
x_{2h}	Individual value of covariate (Ln(coverage)) for a policy observation h
$p_0(\cdot)$	Parameter model (for prior)
$p(\cdot)$	Parameter model (for posterior)
$f_0(\cdot)$	Data model (for continuous cluster)
$f(\cdot)$	Data model (for discrete cluster)
$\delta(\cdot)$	Logistic sigmoid function—expit(·)—to allow for a positive probability of the zero outcome
$\boldsymbol{\theta}_j$	Set of parameters—$\boldsymbol{\beta}, \sigma^2, \xi$—associated with $f(\Sigma Y \vert \boldsymbol{X})$ for cluster j
$\boldsymbol{w}_j$	Set of parameters—π, μ, τ—associated with $f(\boldsymbol{X})$ for cluster j
$\boldsymbol{\omega}_j$	Cluster weights (mixing coefficient) for cluster j
$\boldsymbol{\beta}_0, \Sigma_0$	Vector of initial regression coefficients and variance-covariance matrix (i.e., $\hat{\sigma}^2(\boldsymbol{X}^T\boldsymbol{X})^{-1} = \boldsymbol{X}^T\boldsymbol{X}(\Sigma Y - \Sigma\hat{Y})^T(\Sigma Y - \Sigma\hat{Y})/(n-p)$) obtained from the baseline multivariate gamma regression of $\Sigma\hat{Y} > 0$
$\boldsymbol{\beta}_j$	Regression coefficient vector for a mean outcome estimation
σ_j^2	Cluster-wise variation value for the outcome
ξ_j	Skewness parameter for log skew-normal outcome
$\tilde{\boldsymbol{\beta}}_0, \tilde{\Sigma}_0$	Vector of initial regression coefficients and variance-covariance matrix obtained from the baseline multivariate logistic regression of $\Sigma\hat{Y} = 0$
$\tilde{\boldsymbol{\beta}}_j$	Regression coefficient vector for a logistic function to handle zero outcomes
π_j	Proportion parameter for Bernoulli covariate
μ_j, τ_j	Location and spread parameter for Gaussian covariate
α	Precision parameter that controls the variance of the clustering simulation. For instance, a larger α allows selecting more clusters.
G_0	Prior joint distribution for all parameters in the DPM: $\beta, \sigma^2, \xi, \pi, \mu, \tau$, and α. It allows all continuous, integrable distributions to be supported while retaining theoretical properties and computational tractability such as asymptotic consistency and efficient posterior estimation.
a_0, b_0	Hyperparameters for inverse gamma density of σ_j^2
c_0, d_0	Hyperparameters for Beta density of π_j
ν_0	Hyperparameters for Student's t density of ξ_j
μ_0, τ_0^2	Hyperparameters for Gaussian density of μ_j
e_0, γ_0	Hyperparameters for inverse gamma density of τ_j^2
g_0, h_0	Hyperparameters for gamma density of α
η	Random probability value for gamma mixture density of the posterior on α
π_η	Mixing coefficient for gamma mixture density of the posterior on α

Appendix A. Parameter Knowledge

Appendix A.1. Prior Kernel for Distributions of Outcome, Covariates, and Precision

$$p_0(\boldsymbol{\beta}_j|\boldsymbol{\beta}_0,\ \Sigma_0):\ \boldsymbol{MVN}(\boldsymbol{\beta}_0,\ \sigma_j^2\Sigma_0)^{*} \ \propto\ e^{\{(\boldsymbol{\beta}_j-\boldsymbol{\beta}_0)^T\Sigma_0^{-1}(\boldsymbol{\beta}_j-\boldsymbol{\beta}_0)\}},\quad p_0(\sigma_j^2|a_0,\ b_0):\boldsymbol{InvGa}(a_0,\ b_0)\ \propto\ (\sigma_j^2)^{-(a_0+1)}\cdot e^{-b_0/\sigma_j^2}$$

$$p_0(\xi_j|\nu_0):\ \boldsymbol{T}(\nu_0)\ \propto\ \left(\frac{\xi_j^2}{\nu_0}+1\right)^{-(\nu_0+1)/2},\qquad p_0(\tilde{\boldsymbol{\beta}}_j|\tilde{\boldsymbol{\beta}}_0,\ \tilde{\Sigma}_0):\boldsymbol{MVN}(\tilde{\boldsymbol{\beta}}_0,\ \tilde{\Sigma}_0)^{*}\ \propto\ e^{\{(\tilde{\boldsymbol{\beta}}_j-\tilde{\boldsymbol{\beta}}_0)^T\tilde{\Sigma}_0^{-1}(\tilde{\boldsymbol{\beta}}_j-\tilde{\boldsymbol{\beta}}_0)\}}$$

$$p_0(\pi_j|c_0,\ d_0):\ \boldsymbol{\beta}(c_0,\ d_0)\ \propto\ \pi_j^{(c_0-1)}\cdot(1-\pi_j)^{(d_0-1)},\qquad p_0(\mu_j|\mu_0,\ \tau_0^2):\boldsymbol{N}(\mu_0,\ \tau_0^2)\ \propto\ e^{-\frac{1}{2}(\mu_j-\mu_0)^2/\tau_0^2}$$

$$p_0(\tau_j^2|e_0,\ \gamma_0):\ \boldsymbol{InvGa}(e_0,\ \gamma_0)\ \propto\ (\tau_j^2)^{-(e_0+1)}\cdot e^{-\gamma_0/\tau_j^2},\qquad p_0(\alpha|g_0,\ h_0):\ \boldsymbol{Ga}(g_0,\ h_0)\ \propto\ \alpha^{(g_0-1)}\cdot e^{-\alpha\cdot h_0}$$

* $\boldsymbol{\beta}_0, \Sigma_0 \sim$ gamma regression, $\tilde{\boldsymbol{\beta}}_0, \tilde{\Sigma}_0 \sim$ logistic regression.

Appendix A.2. Posterior Inference for Outcome, Covariates, and Precision

Algorithm A1 Posterior inference $\boldsymbol{\theta}_j^* = \{\boldsymbol{\beta}_j^*, \sigma_j^{2*}, \xi_j^*, \tilde{\boldsymbol{\beta}}_j^*\}$

Require: initialize $\boldsymbol{\theta}_j^{(old)}$: $\begin{cases} \boldsymbol{\beta}_j \sim \boldsymbol{MVN}(\boldsymbol{\beta}_0,\ \sigma_j^2 \Sigma_0) \\ \sigma_j^2 \sim \boldsymbol{IG}(a_0,\ b_0) \\ \xi_j \sim \boldsymbol{T}(\nu_0) \\ \tilde{\boldsymbol{\beta}}_j \sim \boldsymbol{MVN}(\tilde{\boldsymbol{\beta}}_0,\ \tilde{\Sigma}_0) \end{cases}$

repeat
  for $j = 1, \cdots, J$ **do** ▷ Assume J cluster memberships.
    Sample $\boldsymbol{\theta}^{(new)}$ from the proposal densities $\boldsymbol{q}$: ▷ Choose priors as $\boldsymbol{q}$.
    $\boldsymbol{\beta}_j^{(new)} \sim \boldsymbol{q}_\beta,\ \ \sigma_j^{2(new)} \sim \boldsymbol{q}_{\sigma^2},\ \ \xi_j^{(new)} \sim \boldsymbol{q}_\xi,\ \ \tilde{\boldsymbol{\beta}}_j^{(new)} \sim \boldsymbol{q}_{\tilde{\beta}}$
    for $\boldsymbol{\theta}_j^{(new)} = \{\boldsymbol{\beta}_j^{(new)}, \sigma_j^{2(new)}, \xi_j^{(new)}, \tilde{\boldsymbol{\beta}}_j^{(new)}\}$ **do**
      Compute the transition ratio using the outcome models:

$$Ratio_\theta = \frac{\prod_{h=1}^{H} f(S_h|\boldsymbol{X},\ \boldsymbol{\theta}_j^{(new)}) \cdot p_0(\boldsymbol{\theta}_j^{(new)}) \cdot \boldsymbol{q}_\theta(\boldsymbol{\theta}_j^{(old)})}{\prod_{h=1}^{H} f(S_h|\boldsymbol{X},\ \boldsymbol{\theta}_j^{(old)}) \cdot p_0(\boldsymbol{\theta}_j^{(old)}) \cdot \boldsymbol{q}_\theta(\boldsymbol{\theta}_j^{(new)})}$$

      Sample $U \sim \boldsymbol{Unif}(0,1)$
      if $U < Ratio_\theta$ **then** $\boldsymbol{\theta}_j^* = \boldsymbol{\theta}_j^{(new)}$ otherwise $\boldsymbol{\theta}_j^* = \boldsymbol{\theta}_j^{(old)}$
      end if
    end for
    Record $\boldsymbol{\theta}_j^*$
  end for
until M posterior samples ($\boldsymbol{\theta}_{j=1,\cdots,J}^*$) obtained. ▷ M is a sufficient sample size

$$p(\pi_j|c_0, d_0,\ S,\ \boldsymbol{x}_1) : \boldsymbol{\beta}(c_{new},\ d_{new}) \qquad p(\mu_j|\mu_0,\ \tau_0^2,\ S,\ \boldsymbol{x}_2) :\ \boldsymbol{N}(\mu_{new},\ \tau_{new}^2)$$

$$\begin{cases} c_{new} = c_0 + \sum_{h=1}^{n_j} x_{1h} \\ d_{new} = d_0 + n_j - \sum_{h=1}^{n_j} x_{1h} \end{cases} \qquad \begin{cases} \mu_{new} = (n_j \overline{x_2} + \mu_0)/(n_j + 1) \\ \tau_{new}^2 = \tau_j^2/(n_j + 1) \end{cases}$$

$$p(\tau_j^2|e_0, \gamma_0,\ S,\ \boldsymbol{x}_2) :\ \boldsymbol{InvGa}(e_{new},\ \gamma_{new}) \qquad p(\alpha|g_0,\ h_0,\ h,\ J,\ \eta,\ \pi_\eta) :\ \pi_\eta \boldsymbol{Ga}(g_0 + J,\ h_0 - log(\eta)) + (1 - \pi_\eta)\boldsymbol{Ga}(g_0 + J - 1,\ h_0 - log(\eta))$$

$$\begin{cases} e_{new} = e_0 + n_j/2 \\ \gamma_{new} = \gamma_0 + \frac{1}{2}\{\frac{n_j}{n_j+1} \cdot (\overline{x_2} - \mu_0)^2 + \sum_{h=1}^{n_j} (x_{2h} - \overline{x_2})^2\} \end{cases} \qquad \begin{cases} \eta|\alpha, h \sim \boldsymbol{\beta}(\alpha + 1,\ h) \\ \pi_\eta = \dfrac{g_0 + J - 1}{g_0 + J - 1 + h(h_0 - \log(\eta))} \end{cases}$$

Appendix B. Baseline Inference Algorithm for the DPM

Once we obtain decent parameter samples from the posterior distributions, the posterior predictive density can be computed via the DPM Gipps sampling. The basic inference algorithm is described below. Note that the modification details for the missing data imputation are provided in Section 3.5. In every iteration, the algorithm updates the cluster memberships based on the parameter samples and observed data at hand, which leads to the recalculation of the cluster parameters. In the sampler, the state is the collection of membership indices $(s_1, \cdots, s_H)$ and parameters $\{\alpha^*, (\boldsymbol{\theta}_1^*, \cdots, \boldsymbol{\theta}_J^*), (\boldsymbol{w}_1^*, \cdots, \boldsymbol{w}_J^*)\}$, where $\boldsymbol{\theta}_j^*$ refers to the parameter associated with cluster j.

Algorithm A2 DPM Gibbs sampling for new cluster development

Require: Starting state $(s_1, \cdots, s_H), \alpha, (\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_J), (\boldsymbol{w}_1, \cdots, \boldsymbol{w}_J)$

repeat
  for $h = 1, \cdots, H$ **do**
    (1) Update cluster memberships:
      ▷ Take s_h and compute the Cl probabilities using the joint model.
    if $s_h = j$ **then**
      for $j = 1, \cdots, J$ **do**
        $P(s_h = j) = p(s_h|s_{-h}) \cdot f(x_{1h}, x_{2h}|\boldsymbol{w}_j) \cdot f(S_h|x_{1h}, x_{2h}, \boldsymbol{\theta}_j)$
          ▷ for observation h entering into existing discrete clusters.
      end for
    else if $s_h = J + 1$ **then**
      $P(s_h = J + 1) = p(s_h|s_{-h}) \cdot f_0(x_{1h}, x_{2h}) \cdot f_0(S_h|x_{1h}, x_{2h})$
        ▷ for observation h entering into a new continuous cluster.
    end if
    Draw a Cl index from a multinomial $\{1, 2, \cdots, J + 1\}$
      ▷ with probabilities $\big(P(s_h = 1), P(s_h = 2), \cdots, P(s_h = J + 1)\big)$:Polya Urn.
    if the Cl index $= J + 1$ **then**
      Record $(\boldsymbol{\theta}_1, \cdots, \boldsymbol{\theta}_{J+1}), (\boldsymbol{w}_1, \cdots, \boldsymbol{w}_{J+1})$
    end if

    (2) Update parameters:
      ▷ $(\boldsymbol{\theta}_j, \alpha, \boldsymbol{w}_j)$ for each cluster based on the posterior densities.
    for $j = 1, \cdots, J + 1$ **do**
      Sample $\boldsymbol{w}_j^*$ from the posterior: $p(\boldsymbol{w}|\boldsymbol{X}_h)$.
    end for
    Sample α^* from the posterior: $p(\alpha|J + 1)$.
    for $j = 1, \cdots, J + 1$ **do**
      Sample $\boldsymbol{\theta}_j^*$ from the posterior: $p(\boldsymbol{\theta}|S_j, \boldsymbol{X}_h)$.
    end for
    Record $(\boldsymbol{\theta}_1^*, \cdots, \boldsymbol{\theta}_{J+1}^*), (\boldsymbol{w}_1^*, \cdots, \boldsymbol{w}_{J+1}^*)$
  end for
  Record α^*

  for $h = 1, \cdots, H$ **do**
    (3) Compute the log-likelihood: $\sum_{h=1}^{n} log[f(\boldsymbol{X}_h|\boldsymbol{w}_j^*) f(S_h|\boldsymbol{X}_h, \boldsymbol{\theta}_j^*)]$
      ▷ the function is to eventually stabilize after a large number of iterations.
  end for
until M posterior samples $(\boldsymbol{\theta}_j^*, \alpha^*, \boldsymbol{w}_j^*)$ obtained. ▷ M is a sufficient sample size

Appendix C. Development of the Distributional Components for the DPM

Appendix C.1. Derivation of the Distribution of Precision α

In Section 4.1, the parameter model (posterior) of the precision term α is defined as

$$\begin{aligned} p(\alpha|J) &\propto p_0(\alpha) \cdot \alpha^{J-1} \cdot (\alpha + n) \cdot \boldsymbol{\beta}(\alpha + 1, n) \\ p(\alpha|J, \eta, g_0, h_0) &\propto \pi_\eta \boldsymbol{Ga}(g_0 + J,\ h_0 - log(\eta)) + (1 - \pi_\eta) \boldsymbol{Ga}(g_0 + J - 1,\ h_0 - log(\eta)) \end{aligned}$$

To derive this, we first derive the distribution of the number of clusters given the precision parameter $p(J|\alpha)$. Consider a trivial example where we want to determine the number of clusters that $n = 5$ observations fall into. One possible arrangement would be that observations 1, 2, and 5 form new clusters, while observations 3 and 4 join an existing cluster (note that the order is important):

- Observation 1 forms a new cluster with a probability = $\frac{\alpha}{\alpha}$

- Observation 2 forms a new cluster with a probability = $\frac{\alpha}{\alpha+1}$
- Observation 3 enters into an existing cluster with a probability = $\frac{2}{\alpha+2}$
- Observation 4 enters into an existing cluster with a probability = $\frac{3}{\alpha+3}$
- Observation 5 forms a new cluster with a probability = $\frac{\alpha}{\alpha+4}$

In this example, we have $J = 3$ clusters. We want to find the probability of this arrangement. The probability is the following:

$$\left(\frac{\alpha}{\alpha}\right)\left(\frac{\alpha}{\alpha+1}\right)\left(\frac{2}{\alpha+2}\right)\left(\frac{3}{\alpha+3}\right)\left(\frac{\alpha}{\alpha+4}\right) \propto \frac{\alpha^3}{\alpha(\alpha+1)(\alpha+2)(\alpha+3)(\alpha+4)}$$
$$= \alpha^3 \frac{\Gamma(\alpha)}{\Gamma(\alpha+5)}$$

Hence, the probability of observing J clusters amongst a sample size of n is given by

$$p(J|\alpha) \propto \alpha^J \frac{\Gamma(\alpha)}{\Gamma(\alpha+n)}$$

This is also considered the likelihood function. The posterior on α is proportional to the likelihood times the prior $p_0(\alpha)$:

$$p(\alpha|J) \propto p(J|\alpha)p_0(\alpha)$$
$$\propto \alpha^J \frac{\Gamma(\alpha)}{\Gamma(\alpha+n)} p_0(\alpha)$$

The beta function $\boldsymbol{\beta}(x, y)$ is defined as follows:

$$\boldsymbol{\beta}(x, y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}$$

We can find the beta function of $\alpha + 1$ and n as follows:

$$\boldsymbol{\beta}(\alpha+1, n) = \frac{\Gamma(\alpha+1)\Gamma(n)}{\Gamma(\alpha+1+n)}$$
$$\propto \frac{\alpha\Gamma(\alpha)}{(\alpha+n)\Gamma(\alpha+n)}$$
$$\frac{\Gamma(\alpha)}{\Gamma(\alpha+n)} \propto \boldsymbol{\beta}(\alpha+1, n)\frac{\alpha+n}{\alpha}$$

Thus, the posterior simplifies to the following:

$$p(\alpha|J) \propto \alpha^J \cdot \boldsymbol{\beta}(\alpha+1, n) \cdot \frac{\alpha+n}{\alpha} \cdot p_0(\alpha)$$
$$\propto p_0(\alpha) \cdot \alpha^{J-1} \cdot (\alpha+n) \cdot \boldsymbol{\beta}(\alpha+1, n)$$

Now, under the $\boldsymbol{Ga}(g_0, h_0)$ prior for α, by substituting $p_0(\alpha)$ with $\boldsymbol{Ga}(g_0, h_0)$, then

$$p(\alpha|J, \eta, g_0, h_0) \propto \alpha^{g_0+j-2} \cdot (\alpha+n) \cdot e^{-\alpha(h_0 - log(\eta)}$$
$$\propto \pi_\eta \boldsymbol{Ga}(g_0+J,\ h_0 - log(\eta)) + (1-\pi_\eta)\boldsymbol{Ga}(g_0+J-1,\ h_0 - log(\eta))$$

Appendix C.2. Outcome Data Model of S_h Development with the MAR Covariate x_1 for the Discrete Clusters

Prior to the outcome parameter estimation, the missing covariates should be imputed first to obtain the complete covariate model beforehand. In this study, if the binary covariate

x_{1h} is the only covariate with missingness, then we develop the imputation model to impute the binary covariate x_{1h} by taking the steps below and then update $\boldsymbol{\beta}, \sigma^2, \xi, \tilde{\boldsymbol{\beta}}$ based on the posterior sampling detailed in Algorithm A1 in Appendix A. The imputation model for x_{1h} is approximated by the joint values

$$f(x_{1h}|S_h, x_{2h}, \boldsymbol{\beta}_j, \sigma_j, \xi_j, \tilde{\boldsymbol{\beta}}_j, \pi_j) \propto f(S_h,\ x_{1h}|x_{2h}, \boldsymbol{\beta}_j, \sigma_j, \xi_j, \tilde{\boldsymbol{\beta}}_j, \pi_j)$$

where

$$\begin{aligned} f(S_h,\ x_{1h}|x_{2h}, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j, \pi_j) &= f(S_h|x_{1h}, x_{2h}, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j) \cdot f_{Bern}(x_{1h}|\pi_j) \\ &= \delta(\boldsymbol{X}_h^T \tilde{\boldsymbol{\beta}}_j)\ \mathbb{1}(S_h = 0) \cdot \pi_j^{x_{1h}} (1-\pi_j)^{1-x_{1h}} + \left[1 - \delta(\boldsymbol{X}_h^T \tilde{\boldsymbol{\beta}}_j)\right] \frac{2}{\sigma_j S_h} \\ &\cdot \phi\left(\frac{\log S_h - \boldsymbol{X}^T \boldsymbol{\beta}_j}{\sigma_j}\right) \cdot \Phi\left(\xi_j \frac{\log S_h - \boldsymbol{X}^T \boldsymbol{\beta}_j}{\sigma_j}\right) \cdot \pi_j^{x_{1h}} (1-\pi_j)^{1-x_{1h}} \end{aligned}$$

which serves as the joint density that we can use to sample the imputation values. For example, we have

$$\begin{aligned} f_{Bern}(x_{1h} = 1|S_h, x_{2h}, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j, \pi_j) &\propto f(S_h,\ x_{1h} = 1|x_{2h}, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j, \pi_j) \\ &= \delta(\tilde{\beta}_{j0} + \tilde{\beta}_{j1} + \tilde{\beta}_{j2} x_{2h})\ \mathbb{1}(S_h = 0) \cdot \pi_j + \left[1 - \delta(\tilde{\beta}_{j0} + \tilde{\beta}_{j1} + \tilde{\beta}_{j2} x_{2h})\right] \frac{2}{\sigma_j S_h} \\ &\cdot \phi\left(\frac{\log S_h - (\beta_{j0} + \beta_{j1} + \beta_{j2} x_{2h})}{\sigma_j}\right) \cdot \Phi\left(\xi_j \frac{\log S_h - (\beta_{j0} + \beta_{j1} + \beta_{j2} x_{2h})}{\sigma_j}\right) \pi_j \end{aligned}$$

$$\begin{aligned} f_{Bern}(x_{1h} = 0|S_h, x_{2h}, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j, \pi_j) &\propto f(S_h,\ x_{1h} = 0|x_{2h}, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j, \pi_j) \\ &= \delta(\tilde{\beta}_{j0} + \tilde{\beta}_{j2} x_{2h})\ \mathbb{1}(S_h = 0) \cdot (1-\pi_j) + \left[1 - \delta(\tilde{\beta}_{j0} + \tilde{\beta}_{j2} x_{2h})\right] \frac{2}{\sigma_j S_h} \\ &\cdot \phi\left(\frac{\log S_h - (\beta_{j0} + \beta_{j2} x_{2h})}{\sigma_j}\right) \cdot \Phi\left(\xi_j \frac{\log S_h - (\beta_{j0} + \beta_{j2} x_{2h})}{\sigma_j}\right) \cdot (1-\pi_j) \end{aligned}$$

Then, we can impute x_{1h} with the values sampled from $Bern(\pi_{x_1}^*)$, where

$$\pi_{x_1}^* = \frac{f(S_h,\ x_{1h} = 1|x_{2h}, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j, \pi_j)}{f(S_h,\ x_{1h} = 1|x_{2h}, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j, \pi_j) + f(S_h,\ x_{1h} = 0|x_{2h}, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j, \pi_j)}$$

Note that in R, the computation can be difficult when the numerator is too small. We suggest the following tricks:

$$\begin{aligned} p_1 &= f(S_h,\ x_{1h} = 1|x_{2h}, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j, \pi_j) \\ p_0 &= f(S_h,\ x_{1h} = 0|x_{2h}, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j, \pi_j) \\ \pi_{x_1}^* &= \frac{e^{log(p_1)}}{e^{log(p_1)} + e^{log(p_0)}} \cdot \frac{e^{-log(p_1)}}{e^{-log(p_1)}} = \frac{1}{1 + e^{log(p_0) - log(p_1)}} \end{aligned}$$

Finally, the outcome model that is required to compute the parameter $\boldsymbol{\theta} = \{\boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j\}$ in the Metropolis–Hastings algorithm in Algorithm A1 is obtained by summing the joint values of S_h and x_{1h} (marginalize) out of the MAR covariate x_{1h}, shown in Equation (10), as illustrated below:

$$\begin{aligned} f(S_h|x_{2h}, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j, \pi_j) &= \sum_{x_{1h}=0}^{1} f(S_h, x_{1h}|x_{2h}, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j, \pi_j) \\ &= f(S_h,\ x_{1h} = 1|x_{2h}, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j, \pi_j) + f(S_h,\ x_{1h} = 0|x_{2h}, \boldsymbol{\beta}_j, \sigma_j^2, \xi_j, \tilde{\boldsymbol{\beta}}_j, \pi_j) \end{aligned}$$

Appendix C.3. Covariate Data Model of x_2 Development with the MAR Covariate x_1 for the Continuous Clusters

The parameter-free distributions $f_0(y|x)$ and $f_0(x)$ as data models for continuous clusters are needed to calculate the probabilities of cluster membership and for the post-processing calculations for prediction in the DPM. However, when MAR covariates are present, it gives extra complexity in specifying the distribution to integrate out the parameters. Recall that the integrals we are attempting to find are the following:

$$f_0(x_i) = \int f(x_i|\boldsymbol{w})\ dG_0(\boldsymbol{w}) = \int f(x_i|\boldsymbol{w})\ p(\boldsymbol{w})\ d\boldsymbol{w}$$

If the binary covariate x_1 is missing, then we will need to replace the distribution $f(x|\boldsymbol{w})$ with the continuous distribution (Gaussian) of x_2, which is $f(x_2|\mu_j, \tau_j^2)$. The derivation of the parameter-free distributions $f_0(x_1)$ and $f_0(x_2)$ for the continuous cluster is as shown below:

$$\begin{aligned}
&f_0(x_1)\\
&\quad= \int f(x_1|\pi)\ p(\pi)\ d\mu\ d\pi\\
&\quad= \int \pi^{x_1}(1-\pi)^{1-x_1}\frac{1}{\boldsymbol{\beta}(c_0,d_0)}\pi^{(c_0-1)}(1-\pi)^{(d_0-1)}d\pi\\
&\quad= \frac{1}{\boldsymbol{\beta}(c_0,d_0)}\int \pi^{(x_1+c_0-1)}(1-\pi)^{(1-x_1+d_0-1)}d\pi\\
&\quad= \frac{\boldsymbol{\beta}(x_1+c_0,\ 1-x_1+d_0)}{\boldsymbol{\beta}(c_0,d_0)}\cdot \underbrace{\int \frac{\pi^{(x_1+c_0-1)}(1-\pi)^{(1-x_1+d_0-1)}}{\boldsymbol{\beta}(x_1+c_0,\ 1-x_1+d_0)}\ d\pi}_{=1,\text{ beta distribution}}
\end{aligned}$$

$$\begin{aligned}
&f_0(x_2)\\
&= \iint f(x_2|\mu,\tau^2)p(\mu|\tau^2)p(\tau^2)\ d\mu\ d\tau^2\\
&= \iint \frac{1}{\sqrt{2\pi\tau^2}}\exp\left\{-\frac{1}{2\tau^2}(x_2-\mu)^2\right\}\times\frac{1}{\sqrt{2\pi\tau^2}}\exp\left\{-\frac{1}{2\tau^2}(\mu-\mu_0)^2\right\}\\
&\qquad\times\frac{\gamma_0^{e_0}}{\Gamma(e_0)}\left(\tau^2\right)^{-e_0-1}e^{-\gamma_0/\tau^2}\ d\mu\ d\tau^2\\
&= \frac{\gamma_0^{e_0}}{2\pi\Gamma(e_0)}\iint\left(\tau^2\right)^{-e_0-2}\exp\left\{-\frac{1}{2\tau^2}(x_2-\mu)^2-\frac{1}{2\tau^2}(\mu-\mu_0)^2-\frac{\gamma_0}{\tau^2}\right\}\ d\mu\ d\tau^2
\end{aligned}$$

The first step is to integrate with respect to μ. First, we'll simplify the exponent:

$$\begin{aligned}
&-\frac{1}{2\tau^2}(x_2-\mu)^2-\frac{1}{2\tau^2}(\mu-\mu_0)^2-\frac{\gamma_0}{\tau^2}\\
&\qquad= -\frac{1}{2\tau^2}\left[x_2^2-2x_2\mu+\mu^2+\mu^2-2\mu_0\mu+\mu_0^2\right]-\frac{\gamma_0}{\tau^2}\\
&\qquad= -\frac{1}{2\tau^2}\left[2\mu^2-2(x_2+\mu_0)\mu\right]-\frac{1}{2\tau^2}\left[x_2^2+\mu_0^2\right]-\frac{\gamma_0}{\tau^2}\\
&\qquad= -\frac{2}{2\tau^2}\left[\mu^2-(x_2+\mu_0)\mu+\frac{(x_2+\mu_0)^2}{4}\right]+\frac{1}{\tau^2}\left(\frac{(x_2+\mu_0)^2}{4}\right)\\
&\qquad\quad-\frac{x_2^2+\mu_0^2}{2\tau^2}-\frac{\gamma_0}{\tau^2}\\
&\qquad= -\frac{1}{2(\tau^2/2)}\left(\mu-\frac{x_2+\mu_0}{2}\right)^2+\frac{(x_2+\mu_0)^2}{4\tau^2}-\frac{x_2^2+\mu_0^2}{2\tau^2}-\frac{\gamma_0}{\tau^2}
\end{aligned}$$

The integrand will have the kernel of a normal distribution for μ with a mean $\frac{x_2+\mu_0}{2}$ and variance $\frac{\tau^2}{2}$:

$$\begin{aligned} & f_0(x_2) \\ &= \frac{\gamma_0^{e_0}}{2\pi\Gamma(e_0)} \int \underbrace{\sqrt{2\pi(\tau^2/2)}}_{\text{term from } \mu \text{ integral}} \times \left(\tau^2\right)^{-e_0-2} \times \exp\left\{ \frac{(x_2+\mu_0)^2}{4\tau^2} - \frac{x_2^2+\mu_0^2}{2\tau^2} - \frac{\gamma_0}{\tau^2} \right\} d\tau^2 \\ &= \frac{\gamma_0^{e_0}}{2\sqrt{\pi}\Gamma(e_0)} \int \left(\tau^2\right)^{-e_0-3/2} \exp\left\{ -\frac{1}{\tau^2} \left(-\frac{x_2^2+2x_2\mu_0+\mu_0^2}{4} + \frac{x_2^2+\mu_0^2}{2} + \gamma_0 \right) \right\} d\tau^2 \\ &= \frac{\gamma_0^{e_0}}{2\sqrt{\pi}\Gamma(e_0)} \int \left(\tau^2\right)^{-e_0-1/2-1} \exp\left\{ -\frac{1}{\tau^2} \left(\frac{(x_2^2-\mu_0)^2}{4} + \gamma_0 \right) \right\} d\tau^2 \end{aligned}$$

The integrand is the kernel of an inverse gamma distribution with the shape parameter $e_0 + \frac{1}{2}$ and scale parameter $\frac{(x_2^2-\mu_0)^2}{4} + \gamma_0$:

$$f_0(x_2) = \frac{\gamma_0^{e_0}}{2\sqrt{\pi}\Gamma(e_0)} \times \Gamma(e_0+1/2) \left(\frac{(x_2^2-\mu_0)^2}{4} + \gamma_0 \right)^{-e_0-1/2}$$

As shown above, a closed-form expression can be determined, but this is not always the case since it can be extremely complicated. To simplify, we instead might have to consider a Monte Carlo integral.

References

Aggarwal, Ankur, Michael B. Beck, Matthew Cann, Tim Ford, Dan Georgescu, Nirav Morjaria, Andrew Smith, Yvonne Taylor, Andreas Tsanakas, Louise Witts, and et al. 2016. Model risk–daring to open up the black box. *British Actuarial Journal* 21: 229–96. [CrossRef]

Antoniak, Charles E. 1974. Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The Annals of Statistics* 2: 1152–74. [CrossRef]

Bassetti, Federico, Roberto Casarin, and Fabrizio Leisen. 2014. Beta-product dependent pitman–yor processes for bayesian inference. *Journal of Econometrics* 180: 49–72. [CrossRef]

Beaulieu, Norman C., and Qiong Xie. 2003. Minimax approximation to lognormal sum distributions. Paper present at the 57th IEEE Semiannual Vehicular Technology Conference, VTC 2003-Spring, Jeju, Republic of Korea, April 22–25; Piscataway: IEEE, vol. 2, pp. 1061–65.

Billio, Monica, Roberto Casarin, and Luca Rossini. 2019. Bayesian nonparametric sparse var models. *Journal of Econometrics* 212: 97–115.

Blackwell, David, and James B. MacQueen. 1973. Ferguson distributions via pólya urn schemes. *The Annals of Statistics* 1: 353–55. [CrossRef]

Blei, David M., and Peter I. Frazier. 2011. Distance dependent chinese restaurant processes. *Journal of Machine Learning Research* 12: 2461–88.

Braun, Michael, Peter S. Fader, Eric T. Bradlow, and Howard Kunreuther. 2006. Modeling the "pseudodeductible" in insurance claims decisions. *Management Science* 52: 1258–72. [CrossRef]

Browne, Mark J., JaeWook Chung, and Edward W. Frees. 2000. International property-liability insurance consumption. *The Journal of Risk and Insurance 67*: 73–90.

Cairns, Andrew J. G., David Blake, Kevin Dowd, Guy D. Coughlan, and Marwa Khalaf-Allah. 2011. Bayesian stochastic mortality modelling for two populations. *ASTIN Bulletin: The Journal of the IAA* 41: 29–59.

Diebolt, Jean, and Christian P. Robert. 1994. Estimation of finite mixture distributions through bayesian sampling. *Journal of the Royal Statistical Society: Series B (Methodological)* 56: 363–75. [CrossRef]

Escobar, Michael D., and Mike West. 1995. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association* 90: 577–88. [CrossRef]

Ferguson, Thomas S. 1973. A bayesian analysis of some nonparametric problems. *The Annals of Statistics* 1: 209–30. [CrossRef]

Furman, Edward, Daniel Hackmann, and Alexey Kuznetsov. 2020. On log-normal convolutions: An analytical–numerical method with applications to economic capital determination. *Insurance: Mathematics and Economics* 90: 120–34. [CrossRef]

Gelman, Andrew, and Jennifer Hill. 2007. *Data Analysis Using Regression and Multilevel/Hierarchical Models*. Cambridge: Cambridge University Press.
Gershman, Samuel J., and David M. Blei. 2012. A tutorial on bayesian nonparametric models. *Journal of Mathematical Psychology* 56: 1–12. [CrossRef]
Ghosal, Subhashis. 2010. The dirichlet process, related priors and posterior asymptotics. *Bayesian Nonparametrics* 28: 35.
Griffin, Jim, and Mark Steel. 2006. Order-based dependent dirichlet processes. *Journal of the American statistical Association* 101: 179–94. [CrossRef]
Griffin, Jim, and Mark Steel. 2011. Stick-breaking autoregressive processes. *Journal of Econometrics* 162: 383–96. [CrossRef]
Hannah, Lauren A., David M. Blei, and Warren B. Powell. 2011. Dirichlet process mixtures of generalized linear models. *Journal of Machine Learning Research 12*: 1923–53.
Hogg, Robert V., and Stuart A. Klugman. 2009. *Loss Distributions*. Hoboken: John Wiley & Sons.
Hong, Liang, and Ryan Martin. 2017. A flexible bayesian nonparametric model for predicting future insurance claims. *North American Actuarial Journal* 21: 228–41. [CrossRef]
Hong, Liang, and Ryan Martin. 2018. Dirichlet process mixture models for insurance loss data. *Scandinavian Actuarial Journal* 2018: 545–54. [CrossRef]
Huang, Yifan, and Shengwang Meng. 2020. A bayesian nonparametric model and its application in insurance loss prediction. *Insurance: Mathematics and Economics* 93: 84–94. [CrossRef]
Kaas, Rob, Marc Goovaerts, Jan Dhaene, and Michel Denuit. 2008. *Modern Actuarial Risk Theory: Using R*. Berlin and Heidelberg: Springer Science & Business Media, vol. 128.
Lam, Chong Lai Joshua, and Tho Le-Ngoc. 2007. Log-shifted gamma approximation to lognormal sum distributions. *IEEE Transactions on Vehicular Technology* 56: 2121–29. [CrossRef]
Li, Xue. 2008. A Novel Accurate Approximation Method of Lognormal Sum Random Variables. Ph.D. thesis, Wright State University, Dayton, OH, USA.
Neal, Radford M. 2000. Markov chain sampling methods for dirichlet process mixture models. *Journal of Computational and Graphical Statistics* 9: 249–65.
Neuhaus, John M., and Charles E. McCulloch. 2006. Separating between-and within-cluster covariate effects by using conditional and partitioning methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68: 859–72. [CrossRef]
Ni, Yang, Yuan Ji, and Peter Müller. 2020. Consensus monte carlo for random subsets using shared anchors. *Journal of Computational and Graphical Statistics* 29: 703–14. [CrossRef]
Quan, Zhiyu, and Emiliano A. Valdez. 2018. Predictive analytics of insurance claims using multivariate decision trees. *Dependence Modeling* 6: 377–407. [CrossRef]
Richardson, Robert, and Brian Hartman. 2018. Bayesian nonparametric regression models for modeling and predicting healthcare claims. *Insurance: Mathematics and Economics* 83: 1–8. [CrossRef]
Rodriguez, Abel, and David B. Dunson. 2011. Nonparametric bayesian models through probit stick-breaking processes. *Bayesian Analysis (Online)* 6: 145–78.
Roy, Jason, Kirsten J. Lum, Bret Zeldow, Jordan D. Dworkin, Vincent Lo Re, III, and Michael J. Daniels. 2018. Bayesian nonparametric generative models for causal inference with missing at random covariates. *Biometrics* 74: 1193–202. [CrossRef]
Sethuraman, Jayaram. 1994. A constructive definition of dirichlet priors. *Statistica Sinica* 4: 639–650.
Shah, Anoop D., Jonathan W. Bartlett, James Carpenter, Owen Nicholas, and Harry Hemingway. 2014. Comparison of random forest and parametric imputation models for imputing missing data using mice: A caliber study. *American Journal of Epidemiology* 179: 764–74. [CrossRef] [PubMed]
Shahbaba, Babak, and Radford Neal. 2009. Nonlinear models using dirichlet process mixtures. *Journal of Machine Learning Research* 10: 1829–50.
Shams Esfand Abadi, Mostafa. 2022. Bayesian Nonparametric Regression Models for Insurance Claims Frequency and Severity. Ph.D. thesis, University of Nevada, Las Vegas, NV, USA.
Si, Yajuan, and Jerome P. Reiter. 2013. Nonparametric bayesian multiple imputation for incomplete categorical variables in large-scale assessment surveys. *Journal of Educational and Behavioral Statistics* 38: 499–521. [CrossRef]
Suwandani, Ria Novita, and Yogo Purwono. 2021. Implementation of gaussian process regression in estimating motor vehicle insurance claims reserves. *Journal of Asian Multicultural Research for Economy and Management Study* 2: 38–48. [CrossRef]
Teh, Yee Whye. 2010. Dirichlet Process. In *Encyclopedia of Machine Learning*. Berlin and Heidelberg: Springer Science & Business Media, pp. 280–87.
Ungolo, Francesco, Torsten Kleinow, and Angus S. Macdonald. 2020. A hierarchical model for the joint mortality analysis of pension scheme data with missing covariates. *Insurance: Mathematics and Economics* 91: 68–84. [CrossRef]
Zhao, Lian, and Jiu Ding. 2007. Least squares approximations to lognormal sum distributions. *IEEE Transactions on Vehicular Technology* 56: 991–97. [CrossRef]