Journal of Operational Risk **10**(1), 1–43

Research Paper

Modeling correlated frequencies with application in operational risk management

Andrei L. Badescu, Lan Gong, X. Sheldon Lin and Dameng Tang

Department of Statistical Sciences, University of Toronto, 100 St George Street, Toronto, ON M5S 3G3, Canada; email: sheldon@utstat.utoronto.ca

(Received June 5, 2014; revised August 22, 2014; accepted September 8, 2014)

ABSTRACT

In this paper, we propose a copula-free approach for modeling correlated frequency distributions using an Erlang-based multivariate mixed Poisson distribution. We investigate some of the properties possessed by this class of distributions and derive a tailor-made expectation-maximization algorithm for fitting purposes. The applicability of the proposed distribution is illustrated in an operational risk management context, where this class is used to model the operational loss frequencies and their complex dependence structure in a high-dimensional setting. Furthermore, by assuming that operational loss severities follow the mixture of Erlang distributions, our approach leads to a closed-form expression for the total aggregate loss distribution and its value-at-risk can be calculated easily by any numerical method. The efficiency and accuracy of the proposed approach are analyzed using a modified real operational loss data set.

Keywords: advanced measurement approach; correlated frequency; Erlang-based mixture model; EM algorithm; operational risk management; value at risk.

Corresponding author: X. Sheldon Lin

Print ISSN 1744-6740 | Online ISSN 1755-2710

1 INTRODUCTION

Modeling dependent financial loss events is of paramount importance for the sound management of any risks. While market and credit risks have been managed in a mature and systematic way, financial institutions have not paid enough attention to operational risk, which is at least equally important. In June 1999, the Basel Committee on Banking Supervision (BCBS) highlighted the importance of operational risk in banks, by advocating an explicit regulatory capital charge for "other risks". In January 2001, the BCBS narrowed down these "other risks" by drafting the first definition of operational risk, namely "the risk of loss resulting from inadequate or failed internal processes, people and systems or from external events" (see Basel Committee on Banking Supervision 1998, 1999, 2000). Since then, the attention paid to quantifying and managing this risk type has increased substantially.

When quantifying operational risk, banks are expected to move toward the advanced measurement approach (AMA). This complex approach allows a bank to calculate its regulatory capital using internal models, based on internal risk variables and profiles instead of exposure proxies, such as the gross income. One of the methods used to quantify operational risk under the AMA is the loss distribution approach (LDA), which allows banks to determine both the regulatory capital (RCAP) and economic capital (ECAP) estimates (see Aue and Kalkbrener 2006). Under this approach, banks disaggregate loss events into a number of units of measure (UOMs) according to different combinations of business lines and event types. For each UOM, the severity distribution is considered, together with the frequency distribution at which the losses occur under a finite time horizon, in order to obtain an aggregate loss distribution for that unit. These loss distributions are further summed among all UOMs to arrive at the total aggregate loss distribution at the top of the house (TOH) level.

Given the very complex characteristics of the data sets involved in operational risk, identifying the correct dependence structure is of paramount importance in practice (see Chaudhury (2010) for an extensive discussion). The dependence structure in operational risk modeling can alternatively be described via individual severities, via aggregate losses at the UOM levels or via frequencies across the UOMs. The first alternative is sometimes considered undesirable because of its inconsistency with the LDA framework and the difficulties encountered when simulating a multivariate severity distribution, due to the enormous range of possible values. Some banks adopted the second alternative. Frachot *et al* (2004) deemed this assumption very unlikely to be used in practice, as the UOMs correlation is rather weak, even with a strong frequency correlation, as long as severities are independent. Moreover, by using data taken from the Operational Riskdata eXchange Association (ORX), Cope and Antonini (2008) reported that, empirically, the aggregate losses at the UOM level exhibit strong independence behavior.

Similar problems occur in other areas such as non-life insurance, where we are interested in modeling the aggregate losses of an insurer that runs through several correlated lines of business. As with the operational risk problem, the dependence structure among the business lines can be introduced via claim amounts or via their frequency distribution. An essential step in modeling such insurance applications is the proper identification of the dependence structure exhibited by the data sets available.

Motivated by the aforementioned issues, in this paper we propose a dependence structure introduced via frequency distributions. Specifically, we assume that the correlated frequencies are modeled by an Erlang-based multivariate mixed Poisson distribution. We choose its mixing distribution from the versatile class of multivariate Erlang mixtures with a common scale parameter (see Lee and Lin (2012) for an extensive discussion of the class).

The multivariate mixed Poisson distribution has a very intuitive explanation and is a natural extension of the multivariate Poisson distribution, which has been widely used by actuaries for frequency modeling. If we regard the intensity parameter of a Poisson distribution as an individual trait within an insurance portfolio, then from the insurer's perspective it would be ideal if this parameter is constant so that the portfolio is homogeneous. However, for some practical reasons such as adverse selection, moral hazards or losses stemming from various sources, insurers generally take on heterogeneous risks. Thus, the mixed Poisson distribution gives a better practical interpretation of such phenomena.

In practice, financial and insurance losses are often left-truncated. For example, in the operational losses data we use in our numerical illustration, we record only losses exceeding the threshold of US$30 000. The multivariate loss frequencies are thus modified accordingly and it is desirable and even necessary that both the original and the modified frequencies belong to the same class of distributions, regardless of the truncation probabilities. In the one-dimensional case, it is well-known that the class of mixed Poisson distribution is the only one which has this property (see Klugman *et al* 2013, Theorem 8.2). We can easily extend this result to the multivariate case by following a similar argument. In this regard, it is justifiable to choose the class of multivariate mixed Poisson distributions.

The proposed class of distributions is also flexible, to capture a wide range of dependence structures, in the sense that every multivariate mixed Poisson distribution is the limit (under weak convergence) of a series of distributions in this class (see also the remarks after Corollary 2.4). One distinct advantage of using the proposed class, as opposed to using a certain copula, is that the former allows us to capture a much broader range of dependence structures while the dependence structure of each specific copula family is predetermined. In addition, since operational loss data is of very high dimension, it is rather challenging to fit data using one of the commonly used

copulas. In contrast, we can easily use a tailor-made expectation-maximization (EM) algorithm to fit the proposed mixed Poisson distribution to multivariate frequency data. As is illustrated later, the EM algorithm enables us to still fit high-dimensional data with relative ease.

Due to the conditional independence structure and the mathematical tractability of the class of mixed Erlang distributions with a common scale parameter, the proposed class of multivariate mixed Poisson distribution has other desirable properties. The class is closed when taking marginals of any dimensions. Under the choice of this particular frequency distribution, the associated multivariate compound distribution (see Sundt and Vernic 2009) can be calculated recursively when the severities follow discrete distributions. Thus, when compounding this class of frequency distributions with severities that are of mixed Erlang type, we can obtain a closed-form expression for the total aggregate loss distribution, and its value-at-risk (VaR) can easily be calculated by any numerical method, which means that we can keep the use of Monte Carlo simulation to a minimal or even zero level. Fitting data with complicated features, such as multiple modes, could be successfully addressed by using such a distribution. Unlike the multivariate phase-type distributions, the proposed Erlang-based class of multivariate mixed Poisson distribution is identifiable. As demonstrated in the paper, this important property makes this class an ideal candidate for statistical inference.

The paper is organized as follows. Section 2 introduces the class of Erlang-based multivariate Poisson mixtures and some of its properties and in Section 3 we present the EM algorithm for parameter estimation. As a direct application, we use this class of distributions to model operational loss frequencies and the dependence structure therein. We calibrate our model in Section 4, using modified real operational loss data, and show the precision of the fitting results. In Section 5, we validate our model using various methods, in particular by analyzing the effectiveness of the fitting results at the TOH level. Section 6 gives a closed-form expression for the total aggregate loss distribution and presents some directions for future research.

2 ERLANG-BASED MULTIVARIATE MIXED POISSON DISTRIBUTIONS

In this section, we introduce the class of Erlang-based multivariate mixed Poisson distributions and present some of its properties. Unless stated otherwise, in this paper we set

$$\boldsymbol{m} = (m_1, \ldots, m_k), \quad \boldsymbol{\beta} = (\beta_{\boldsymbol{m}} = \beta_{m_1,\ldots,m_k};\ m_i = 1, 2, \ldots;\ i = 1, 2, \ldots, k),$$

where each $\beta_{\boldsymbol{m}} \geqslant 0$ and $\sum_{m_1} \cdots \sum_{m_k} \beta_{\boldsymbol{m}} = 1$. We also define $\boldsymbol{N} = (N_1, \ldots, N_k)$, which can be understood as the loss frequencies in the k risk cells when our model is applied in an operational risk management context.

Suppose that $\boldsymbol{N}$ is a k-variate counting random vector that has a multivariate mixed Poisson distribution. For nonnegative integers $n_1, n_2, \ldots, n_k$, we define its probability function as

$$\begin{aligned}&\mathbb{P}(N_1 = n_1, \ldots, N_k = n_k)\\ &\quad = \int_0^\infty \cdots \int_0^\infty \left(\prod_{j=1}^{k} \mathrm{e}^{-\lambda_j} \frac{(\lambda_j)^{n_j}}{n_j!} \right) \pi(\lambda_1, \ldots, \lambda_k; \boldsymbol{\beta}, \theta)\, \mathrm{d}\lambda_1 \cdots \mathrm{d}\lambda_k, \end{aligned} \tag{2.1}$$

where the mixing distribution is a multivariate Erlang mixture with a common scale parameter θ:

$$\pi(\lambda_1, \ldots, \lambda_k; \boldsymbol{\beta}, \theta) = \sum_{m_1=1}^{\infty} \cdots \sum_{m_k=1}^{\infty} \beta_{\boldsymbol{m}} \prod_{j=1}^{k} \frac{{\lambda_j}^{m_j - 1}\, \mathrm{e}^{-\lambda_j/\theta}}{\theta^{m_j}\,(m_j - 1)!}. \tag{2.2}$$

Here and throughout the rest of the paper, we assume the mixture is finite, ie, the number of nonzero $\beta_{\boldsymbol{m}}$ is finite; we write the finite mixture in the form (2.2) for notational convenience. As a finite mixture, this distribution can easily be simulated (see, for example, Klugman *et al* 2012, Section 20.2.1). We further denote the probability generating function of a discrete k-dim random vector $\boldsymbol{N}$ by $P_{\boldsymbol{N}}(\boldsymbol{z})$, which is defined as $\mathbb{E}(\prod_{j=1}^{k} z_j^{N_j})$. Using the mixing components of the form given in (2.2), we can easily see that $\boldsymbol{N}$ follows a k-variate Pascal mixture distribution (a negative binomial distribution with a positive integer shape parameter) with a probability function given by

$$\mathbb{P}(N_1 = n_1, \ldots, N_k = n_k) = \sum_{m_1=1}^{\infty} \cdots \sum_{m_k=1}^{\infty} \beta_{\boldsymbol{m}} \prod_{j=1}^{k} p(n_j; m_j, \theta), \tag{2.3}$$

where

$$p(n_j; m_j, \theta) = \binom{n_j + m_j - 1}{m_j - 1} \left(\frac{1}{1+\theta} \right)^{m_j} \left(\frac{\theta}{1+\theta} \right)^{n_j}. \tag{2.4}$$

2.1 Distributional properties

Lee and Lin (2012) proved that the class of k-variate Erlang mixtures is dense in the space of all k-variate positive continuous distributions in the sense of weak convergence. Combining this fact with Theorem 2.1 of Grandell (1997) leads to the following proposition, which guarantees the theoretical versatility of our proposed model.

PROPOSITION 2.1 *The class of multivariate Pascal mixtures is dense in the space of multivariate mixed Poisson distributions in the sense of weak convergence.*

A useful representation of this class of distributions is given in the following theorem.

THEOREM 2.2 *If $\boldsymbol{N}$ is a multivariate Pascal mixture with probability function given by (2.3), the marginal random variables N_j are compound geometrically distributed, ie, $N_j = \sum_{i=1}^{M_j} G_{ij}$ in distribution for $j = 1, \ldots, k$, where M_j is the primary counting random variable and G_{ij} are independent and identically distributed (iid) geometric random variables with mean θ. In addition, $\boldsymbol{M} = (M_1, \ldots, M_k)$ has a joint probability function*

$$\mathbb{P}(\boldsymbol{M} = \boldsymbol{m}) = \beta_{\boldsymbol{m}}. \tag{2.5}$$

PROOF The proof is straightforward by comparing their probability generating functions (PGFs). □

The following two corollaries can easily be obtained by virtue of Theorem 2.2.

COROLLARY 2.3 *If $\boldsymbol{N}$ is a multivariate Pascal mixture of the form (2.3), each marginal component N_j is a univariate Pascal mixture with mixing weights given by*

$$\beta_{m_j}^{(j)} = \sum_{m_l, l \neq j; m_l = 1,2,\ldots} \beta_{\boldsymbol{m}}. \tag{2.6}$$

Furthermore, any p-variate marginal is a p-variate Pascal mixture.

COROLLARY 2.4

(1) *Any marginal component N_j has mean and variance*

$$\mathbb{E}(N_j) = \theta \mathbb{E}(M_j), \tag{2.7}$$

$$\operatorname{var}(N_j) = \theta(1+\theta)\mathbb{E}(M_j) + \theta^2 \operatorname{var}(M_j), \tag{2.8}$$

respectively, where M_j has probability function (2.6).

(2) *(Covariance invariance.) The covariance of any marginal pair (N_j, N_l) of $\boldsymbol{N}$ is given by*

$$\theta^2 \operatorname{cov}(M_j, M_l) = \theta^2 \Bigg[\sum_{m_j=1}^{\infty} \sum_{m_l=1}^{\infty} \beta_{m_j, m_l}^{(j,l)} m_j m_l - \Bigg(\sum_{m_j=1}^{\infty} \beta_{m_j}^{(j)} m_j \Bigg) \Bigg(\sum_{m_l=1}^{\infty} \beta_{m_l}^{(l)} m_l \Bigg) \Bigg], \tag{2.9}$$

where

$$\beta_{m_j, m_l}^{(j,l)} = \sum_{m_i, i \neq (j,l), m_i = 1,2,\ldots} \beta_{\boldsymbol{m}}. \tag{2.10}$$

REMARK 2.5 Owing to the covariance invariance property of our proposed class of distributions, we can use it to model a wide range of dependence structures by choosing appropriate corresponding mixing distributions. For example, assume $N_1 = \sum_{i=1}^{M} G_{i1}$ and $N_2 = \sum_{i=1}^{M} G_{i2}$, where G_{ij} are geometrically with mean of θ. Then the Pearson's correlation coefficient between N_1 and N_2 equals

$$\frac{\theta^2 \operatorname{var}(M)}{\theta(1+\theta)\mathbb{E}(M) + \theta^2 \operatorname{var}(M)} \to \frac{\operatorname{var}(M)}{\mathbb{E}(M) + \operatorname{var}(M)}, \quad \theta \to \infty.$$

If we let the mean of M go to infinity, then

$$\frac{\operatorname{var}(M)}{\mathbb{E}(M) + \operatorname{var}(M)} \to 1,$$

which means that the correlation coefficient of N_1 and N_2 goes to 1 (perfectly correlated). On the other hand, if we let (M_1, M_2) have joint probabilities $\mathbb{P}(M_1 = 1, M_2 = 2) = \mathbb{P}(M_1 = 2, M_2 = 1) = 1/2$, direct calculation shows that the correlation coefficient between the associated N_1 and N_2 equals $-(1+7\theta)^{-2}$, which has a limit of -1, when θ goes to zero (perfectly negatively correlated).

Another important property that this class of distributions possesses is identifiability. It is well-known that not every class of mixture distributions is identifiable, ie, if two mixtures have the same distribution function, the coefficients and components are not necessarily the same. An example of a nonidentifiable class of distributions is given by the equally versatile class of phase-type distributions. However, identifiability is preserved when we consider the proposed class of distributions, making the class very appealing for statistical inference purposes. We first restate a modified version of Theorem 2 of Teicher (1963) on the identifiability of finite mixtures, in which the coefficients of the component distributions are arbitrary positive numbers. Note that for notational convenience all finite summations are expressed here as infinite ones.

PROPOSITION 2.6 *Let $\mathcal{F} = \{F_\xi\}$ be a family of distributions with transforms $\phi_\xi(t)$ defined for $t \in \mathcal{S}_{\phi_\xi}$ such that the mapping $M : F_\xi \to \phi_\xi$ is linear and one-to-one. Suppose that there exists a total ordering of $\mathcal{F}$ such that $F_{\xi_1} \prec F_{\xi_2}$ implies*

(1) $\mathcal{S}_{\phi_{\xi_1}} \subseteq \mathcal{S}_{\phi_{\xi_2}}$,

(2) *the existence of some $t_1 \in \bar{\mathcal{S}}_{\phi_{\xi_1}}$ (t_1 being independent of ϕ_{ξ_2}) such that $\lim_{t \to t_1} \phi_{\xi_2}(t)/\phi_{\xi_1}(t) = 0$.*

Then, if for any x,

$$\sum_{m=1}^{\infty} \beta_m F_{\xi_m}(x) = \sum_{m=1}^{\infty} \beta'_m F_{\xi_m}(x),$$

we must have $\beta_m = \beta'_m$.

THEOREM 2.7 *The class of multivariate Pascal finite mixtures taking the form of (2.3) is identifiable.*

PROOF For any nonnegative integers n_j, $j = 1, 2, \ldots, k$, assume that

$$\sum_{m_1=1}^{\infty} \cdots \sum_{m_k=1}^{\infty} \beta_{\boldsymbol{m}} \prod_{j=1}^{k} p(n_j; m_j, \theta) = \sum_{m_1=1}^{\infty} \cdots \sum_{m_k=1}^{\infty} \beta'_{\boldsymbol{m}} \prod_{j=1}^{k} p(n_j; m_j, \theta).$$

Now we use mathematical induction to prove $\beta_{\boldsymbol{m}} = \beta'_{\boldsymbol{m}}$.

(1) When $k = 1$, the statement can be proven by showing that the class of Pascal $p(n; m, \theta)$ distributions meets the conditions in Proposition 2.6. Here only parameter m is related to the mixtures and parameter θ is fixed. The transform ϕ_m of $p(n; m, \theta)$ is chosen to be its PGF, ie,

$$\phi_m(z) = P(z; m, \theta) = \frac{1}{(1 + \theta - \theta z)^m}, \quad 0 < z < \frac{1 + \theta}{\theta}.$$

We denote the domain of the PGF by

$$\mathcal{S}_m = \left(0, \frac{1 + \theta}{\theta}\right)$$

and order Pascal distributions by

$$p(n; m_1, \theta) \prec p(n; m_2, \theta) \iff m_1 > m_2.$$

Under this choice, it is easy to see that

$$\lim_{z \to (1+\theta)/\theta} \frac{\phi_{m_2}(z)}{\phi_{m_1}(z)} = 0.$$

Since $\mathcal{S}_m$ does not change with respect to m, by Proposition 2.6, we conclude that $\beta_m = \beta'_m$ for all m if $\sum_{m=1}^{\infty} \beta_m p(n; m, \theta) = \sum_{m=1}^{\infty} \beta'_m p(n; m, \theta)$ holds for any nonnegative integers n.

(2) We assume that the statement holds for $k = k_0$, ie, if, for any nonnegative integers n_j, $j = 1, 2, \ldots, k_0$,

$$\sum_{m_1=1}^{\infty} \cdots \sum_{m_{k_0}=1}^{\infty} \beta_{\boldsymbol{m}} \prod_{j=1}^{k_0} p(n_j; m_j, \theta) = \sum_{m_1=1}^{\infty} \cdots \sum_{m_{k_0}=1}^{\infty} \beta'_{\boldsymbol{m}} \prod_{j=1}^{k_0} p(n_j; m_j, \theta),$$

then $\beta_{\boldsymbol{m}} = \beta'_{\boldsymbol{m}}$.

(3) Let $k = k_0+1$. Assume that for any nonnegative integers n_j, $j = 1, 2, \ldots, k_0+1$,

$$\sum_{m_1=1}^{\infty} \cdots \sum_{m_{k_0+1}=1}^{\infty} \beta_{\boldsymbol{m}} \prod_{j=1}^{k_0+1} p(n_j; m_j, \theta) = \sum_{m_1=1}^{\infty} \cdots \sum_{m_{k_0+1}=1}^{\infty} \beta'_{\boldsymbol{m}} \prod_{j=1}^{k_0+1} p(n_j; m_j, \theta).$$

Equivalently, for any n_{k_0+1} and any fixed $n_1, \ldots, n_{k_0}$, we have

$$\begin{aligned}\sum_{m_{k_0+1}=1}^{\infty} \Big(\sum_{m_1=1}^{\infty} \cdots \sum_{m_{k_0}=1}^{\infty} \beta_{\boldsymbol{m}} \prod_{j=1}^{k_0} p(n_j; m_j, \theta) \Big) p(n_{k_0+1}; m_{k_0+1}, \theta) \\ = \sum_{m_{k_0+1}=1}^{\infty} \Big(\sum_{m_1=1}^{\infty} \cdots \sum_{m_{k_0}=1}^{\infty} \beta'_{\boldsymbol{m}} \prod_{j=1}^{k_0} p(n_j; m_j, \theta) \Big) p(n_{k_0+1}; m_{k_0+1}, \theta).\end{aligned}$$

By (1), this implies

$$\sum_{m_1=1}^{\infty} \cdots \sum_{m_{k_0}=1}^{\infty} \beta_{\boldsymbol{m}} \prod_{j=1}^{k_0} p(n_j; m_j, \theta) = \sum_{m_1=1}^{\infty} \cdots \sum_{m_{k_0}=1}^{\infty} \beta'_{\boldsymbol{m}} \prod_{j=1}^{k_0} p(n_j; m_j, \theta).$$

According to our induction assumption for the case of $k = k_0$, we conclude that $\beta_{\boldsymbol{m}} = \beta'_{\boldsymbol{m}}$. □

REMARK 2.8 Note that the identifiability result in the above theorem ensures that the estimation procedure discussed in Section 3 will produce a sequence of distributions that will converge to the model/target distribution.

COROLLARY 2.9 *If $\boldsymbol{N}$ is a multivariate Pascal mixture of form (2.3), the marginal random variables $N_1, \ldots, N_k$ are mutually independent if and only if*

$$\beta_{\boldsymbol{m}} = \prod_{j=1}^{k} \beta_{m_j}^{(j)}. \tag{2.11}$$

PROOF If (2.11) holds, it is obvious that $N_1, \ldots, N_k$ are mutually independent. Conversely, if $N_1, \ldots, N_k$ are mutually independent, then, for any nonnegative integers $n_1, \ldots, n_k$, $\mathbb{P}(N_1 = n_1, \ldots, N_k = n_k) = \prod_{i=1}^{k} \mathbb{P}(N_i = n_i)$, ie,

$$\sum_{m_1=1}^{\infty} \cdots \sum_{m_k=1}^{\infty} \beta_{\boldsymbol{m}} \prod_{q=1}^{k} p(n_j; m_j, \theta) = \sum_{m_1=1}^{\infty} \cdots \sum_{m_k=1}^{\infty} \Big(\prod_{j=1}^{k} \beta_{m_j}^{(j)} \Big) \prod_{q=1}^{k} p(n_j; m_j \theta).$$

By Theorem 2.7, (2.11) holds. □

2.2 Discrete compound multivariate Pascal mixtures

In this subsection we give a general result on calculating compound multivariate Pascal sums, when the loss severities follow a general discrete distribution. To this end, we consider a k-variate compound distribution, $\boldsymbol{S} = (S_1, \ldots, S_k)$, with multivariate mixed Pascals distributed frequencies $\boldsymbol{N} = (N_1, \ldots, N_k)$ and generally distributed univariate severities $X_1, \ldots, X_k$ such that

$$S_i = X_{i1} + X_{i2} + \cdots + X_{iN_i}, \quad i = 1, \ldots, k, \tag{2.12}$$

where $X_{i1}, X_{i2}, \ldots$ are iid replicates from X_i, independent of N_i. Simple algebra shows that the PGF of $\boldsymbol{S}$ can be expressed as

$$P_{(S_1,\ldots,S_k)}(z_1, \ldots, z_k) = P_{(N_1,\ldots,N_k)}(P_{X_1}(z_1), \ldots, P_{X_k}(z_k)). \tag{2.13}$$

After expanding and plugging the probability function of the multivariate Pascal mixture distribution, we can further simplify (2.13) as follows

$$\begin{aligned}
&\sum_{n_1=0}^{\infty} \cdots \sum_{n_k=0}^{\infty} \mathbb{P}(N_1 = n_1, \ldots, N_k = n_k)(\mathbb{P}_{X_1}(z_1))^{n_1} \cdots (\mathbb{P}_{X_k}(z_k))^{n_k} \\
&\quad = \sum_{n_1=0}^{\infty} \cdots \sum_{n_k=0}^{\infty} \sum_{m_1=1}^{\infty} \cdots \sum_{m_k=1}^{\infty} \beta_{\boldsymbol{m}} \prod_{j=1}^{k} p(n_j; m_j, \theta)(\mathbb{P}_{X_1}(z_1))^{n_1} \cdots (\mathbb{P}_{X_k}(z_k))^{n_k} \\
&\quad = \sum_{m_1=1}^{\infty} \cdots \sum_{m_k=1}^{\infty} \beta_{\boldsymbol{m}} \sum_{n_1=0}^{\infty} \cdots \sum_{n_k=0}^{\infty} \prod_{j=1}^{k} [p(n_j; m_j, \theta)(P_{X_j}(z_j))^{n_j}] \\
&\quad = \sum_{m_1=1}^{\infty} \cdots \sum_{m_k=1}^{\infty} \beta_{\boldsymbol{m}} \prod_{j=1}^{k} \Bigg[\sum_{n_j=0}^{\infty} p(n_j; m_j, \theta)(P_{X_j}(z_j))^{n_j} \Bigg] \\
&\quad = \sum_{m_1=1}^{\infty} \cdots \sum_{m_k=1}^{\infty} \beta_{\boldsymbol{m}} \prod_{j=1}^{k} P_{T_j}(P_{X_j}(z_j)) \\
&\quad \triangleq \sum_{m_1=1}^{\infty} \cdots \sum_{m_k=1}^{\infty} \beta_{\boldsymbol{m}} \prod_{j=1}^{k} P_{C_j}(z_j),
\end{aligned} \tag{2.14}$$

where, for $j = 1, \ldots, k$, C_j is a compound random variable with the counting random variables T_j and severity amounts X_j. Since T_j is a Pascal random variable with distribution of form (2.4) and X_j is a discrete random variable, the distribution of C_j can be calculated recursively using Panjer's recursion (see, for example, Klugman *et al* 2012, Theorem 7.1). This result is summarized in the following proposition.

PROPOSITION 2.10 *Suppose $\boldsymbol{S}$ is a k-variate compound distribution with counting distribution of form (2.3) and discrete severities $X_1, \ldots, X_k$. The joint probability*

function of $\boldsymbol{S}$ *is given by*

$$\mathbb{P}(S_1 = s_1, \ldots, S_k = s_k) = \sum_{m_1=1}^{\infty} \cdots \sum_{m_k=1}^{\infty} \beta_{\boldsymbol{m}} \prod_{j=1}^{k} \mathbb{P}(C_j = s_j), \tag{2.15}$$

where, for $j = 1, \ldots, k$, C_j *is a compound random variable with counting distribution of form (2.4) and discrete severities* X_j.

Note that the case of continuous severity distribution is investigated in detail in Section 6, by assuming severities that are mixed Erlang distributed. We decided to treat this separately, as in this particular case we obtain a closed-form solution for the aggregate loss at TOH level.

3 PARAMETRIC ESTIMATION: AN EXPECTATION-MAXIMIZATION ALGORITHM

Although the class of multivariate Pascal mixtures has many desirable properties as indicated in the previous section, we still need an efficient algorithm to fit it to frequency data. A natural choice for estimating the parameters of a finite mixture model is the EM algorithm (McLachlan and Peel 2000). Ghitany *et al* (2012) proposed an EM algorithm for the class of multivariate mixed Poisson distribution, which is more general than our proposed class as it has no restriction on the mixing distribution and incorporates regression covariates. In this section we derive a tailor-made EM algorithm for fitting frequencies from multivariate mixtures of Pascal distributions. Compared with Ghitany *et al* (2012), our EM algorithm has explicit formulas and its maximization-step (M-step) always converges to a unique global maximum.

Before proceeding to the steps involved in the EM algorithm, we note that it is common in practice to encounter some modifications of the real loss amounts. For example, when banks compile their operational loss data, they usually only record those losses whose amounts exceed a certain threshold level. As a result, the recorded loss amount is left-truncated and equals $X \mid X > d$, assuming that X is the original loss amount and d is the recording threshold. A similar scenario is encountered by actuaries when dealing with franchise deductibles. In the context of multivariate compound distribution (2.12), where $X_{i1}, X_{i2}, \ldots$ are the original loss amounts of the ith policy and N_i are the corresponding frequencies, we denote the recording threshold of the ith policy by d_i. Consequently, the number of recorded losses of the ith policy are

$$N_i^{\text{Record}} = \sum_{j=1}^{N_i} 1_{\{X_{ij} > d_i\}}.$$

The following result states that $\boldsymbol{N}^{\text{Record}} = (N_1^{\text{Record}}, \ldots, N_k^{\text{Record}})$ still belongs to the class of multivariate Pascal mixtures of the form (2.3), with different θ parameters reflecting the truncation points.

PROPOSITION 3.1 *If $\boldsymbol{N}$ is a multivariate Pascal mixture of form (2.3) and we denote the survival function of the losses in line i with truncation point d_i by $\bar{F}_{X_i}(d_i)$, then*

$$\mathbb{P}(N_1^{\text{Record}} = n_1, \ldots, N_k^{\text{Record}} = n_k) = \sum_{m_1=1}^{\infty} \cdots \sum_{m_k=1}^{\infty} \beta_{\boldsymbol{m}} \prod_{j=1}^{k} p(n_j; m_j, \theta \bar{F}_{X_j}(d_j)). \tag{3.1}$$

PROOF By a similar calculation to that in (2.14), we can see that $\boldsymbol{N}$ has PGF

$$\mathbb{E}\Big(\prod_{j=1}^{k} z_j^{N_j} \Big) = \sum_{m_1} \cdots \sum_{m_k} \beta_{\boldsymbol{m}} \prod_{j=1}^{k} [1 - \theta(z_j - 1)]^{m_j}. \tag{3.2}$$

On the other hand, $\boldsymbol{N}^{\text{Record}}$ has PGF

$$\begin{aligned}
\mathbb{E}\Big(\prod_{j=1}^{k} z_j^{N_j^{\text{Record}}} \Big) &= \mathbb{E}\Big\{ \mathbb{E}\Big[\prod_{j=1}^{k} z_j^{N_j^{\text{Record}}} \;\Big|\; N_1, \ldots, N_k \Big] \Big\} \\
&= \mathbb{E}\Big[\prod_{j=1}^{k} \mathbb{E}\Big(z_j^{N_j^{\text{Record}}} \mid N_j \Big) \Big] \\
&= \mathbb{E}\Big\{ \prod_{j=1}^{k} [1 + \bar{F}_{X_j}(d_j)(z_j - 1)]^{N_j} \Big\} \\
&= \sum_{m_1} \cdots \sum_{m_k} \beta_{\boldsymbol{m}} \prod_{j=1}^{k} [1 - \theta \bar{F}_{X_j}(d_j)(z_j - 1)]^{m_j},
\end{aligned} \tag{3.3}$$

which is the PGF of the right-hand side of (3.1). □

REMARK 3.2 The class of distributions given by (2.3) is a special case of the general class introduced by (3.1) and can be obtained by assuming $d_j = 0$, $j = 1, \ldots, k$.

We now present a tailor-made EM algorithm to fit the multivariate Pascal mixture to data:

$$\mathbb{P}(N_1 = n_1, \ldots, N_k = n_k) = \sum_{m_1=1}^{\infty} \cdots \sum_{m_k=1}^{\infty} \beta_{\boldsymbol{m}} \prod_{j=1}^{k} p(n_j; m_j, c_j \theta) \tag{3.4}$$

where $c_j \in [0,1]$, $j = 1,\ldots,k$. The EM algorithm proposed here is similar to the one in Lee and Lin (2012) for the case of multivariate Erlang mixtures. The R codes that we developed for implementing this EM algorithm are included in Appendix A.

Assume we have k-variate data $\boldsymbol{x} = (\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n)$, where $\boldsymbol{x}_i = (x_{i1},\ldots,x_{ik})$ and we wish to fit (3.4) to the data. In reality, only a finite number of $\boldsymbol{m} = (m_1,\ldots,m_k)$ are nonzero. Here, we treat both the number of mixtures and all the shape parameters m_j of Pascals distributions as prespecified, which will be further adjusted, as illustrated below. The parameters to be estimated are the mixing weights $\beta_{\boldsymbol{m}}$ and θ and we denote them by $\boldsymbol{\Phi}$. The observed likelihood of $\boldsymbol{x}$ equals

$$\mathcal{L}(\boldsymbol{\Phi};\boldsymbol{x}) = \prod_{i=1}^{n}\Bigg(\sum_{m_1=1}^{\infty}\cdots\sum_{m_k=1}^{\infty}\beta_{\boldsymbol{m}}\prod_{j=1}^{k}p(x_{ij};m_j,c_j\theta)\Bigg),$$

and the corresponding loglikelihood equals

$$l(\boldsymbol{\Phi};\boldsymbol{x}) = \sum_{i=1}^{n}\ln\Bigg(\sum_{m_1=1}^{\infty}\cdots\sum_{m_k=1}^{\infty}\beta_{\boldsymbol{m}}\prod_{j=1}^{k}p(x_{ij};m_j,c_j\theta)\Bigg). \tag{3.5}$$

The EM algorithm treats the original data as incomplete by introducing unobservable mixing component indicator vectors $\boldsymbol{Z} = (\boldsymbol{Z}_1,\ldots,\boldsymbol{Z}_n)$, where $\boldsymbol{Z}_i = (Z_{i\boldsymbol{m}} \mid \boldsymbol{m} = (m_1,\ldots,m_k), m_i = 1,2,\ldots, i = 1,\ldots,k)$, as follows:

$$Z_{i\boldsymbol{m}} = \begin{cases} 1 & \text{if observation } \boldsymbol{x}_i \text{ comes from } \boldsymbol{m}\text{th component } \displaystyle\prod_{j=1}^{k}p(x_{ij};m_j,c_j\theta), \\ 0 & \text{otherwise.} \end{cases} \tag{3.6}$$

For $i = 1,\ldots,n$, $\boldsymbol{Z}_i$ describes which component density generated observation $\boldsymbol{x}_i$. In other words, the conditional density of $\boldsymbol{X}_i$ given $Z_{i\boldsymbol{m}}$ can be written as $\prod_{j=1}^{k}p(x_{ij};m_j,c_j\theta)$ with $\boldsymbol{m} = (m_1,\ldots,m_k)$. Thus, the complete likelihood function of $(\boldsymbol{x},\boldsymbol{Z})$ is given by

$$\mathcal{L}(\boldsymbol{\Phi};\boldsymbol{x},\boldsymbol{Z}) = \prod_{i=1}^{n}\prod_{m_1=1}^{\infty}\cdots\prod_{m_k=1}^{\infty}\Bigg(\beta_{\boldsymbol{m}}\prod_{j=1}^{k}p(x_{ij};m_j,c_j\theta)\Bigg)^{Z_{i\boldsymbol{m}}},$$

and the corresponding loglikelihood function is given by

$$l(\boldsymbol{\Phi};\boldsymbol{x},\boldsymbol{Z}) = \sum_{i=1}^{n}\sum_{m_1=1}^{\infty}\cdots\sum_{m_k=1}^{\infty}Z_{i\boldsymbol{m}}\Bigg(\ln\beta_{\boldsymbol{m}} + \sum_{j=1}^{k}\ln p(x_{ij};m_j,c_j\theta)\Bigg). \tag{3.7}$$

3.1 Expectation-step

In the lth iteration of the expectation-step (E-step), given the observed data $\boldsymbol{x}$ and the current estimate $\boldsymbol{\Phi}^{(l-1)}$, it follows from Bayes's formula that the posterior probability of $\boldsymbol{\Phi}$ is

$$z_{i\boldsymbol{m}}^{(l)} = \frac{\beta_{\boldsymbol{m}}^{(l-1)} \prod_{j=1}^{k} p(x_{ij}; m_j, c_j\theta^{(l-1)})}{\sum_{m_1=1}^{\infty} \cdots \sum_{m_k=1}^{\infty} \beta_{\boldsymbol{m}}^{(l-1)} \prod_{j=1}^{k} p(x_{ij}; m_j, c_j\theta^{(l-1)})}. \tag{3.8}$$

Taking the conditional expectation of the complete loglikelihood function (3.7) given the observed data $\boldsymbol{x}$ and the current estimate $\boldsymbol{\Phi}^{(l-1)}$ for $\boldsymbol{\Phi}$, we obtain

$$\begin{aligned} &Q(\boldsymbol{\Phi}; \boldsymbol{\Phi}^{(l-1)}) \\ &\quad = \mathbb{E}(l(\boldsymbol{\Phi}; \boldsymbol{x}, \boldsymbol{Z}) \mid \boldsymbol{x}; \boldsymbol{\Phi}^{(l-1)}) \\ &\quad = \sum_{i=1}^{n} \sum_{\boldsymbol{m}} z_{i\boldsymbol{m}}^{(l)} \Bigg\{ \ln \beta_{\boldsymbol{m}} + \sum_{j=1}^{k} \Bigg(\ln \binom{x_{ij} + m_j - 1}{m_j - 1} \\ &\qquad\qquad + m_j \ln\left(\frac{1}{1 + c_j\theta}\right) + x_{ij} \ln\left(\frac{c_j\theta}{1 + c_j\theta}\right)\Bigg)\Bigg\}. \end{aligned} \tag{3.9}$$

3.2 Maximization-step

In this step we maximize Q in (3.9) with respect to $\beta_{\boldsymbol{m}}$ and θ:

$$\boldsymbol{\Phi}^{(l)} = \arg\max_{\boldsymbol{\Phi}} (\boldsymbol{\Phi} \mid \boldsymbol{\Phi}^{(l-1)}), \tag{3.10}$$

over $\boldsymbol{\Phi} = (\beta_{\boldsymbol{m}}, \theta)$ with $\beta_{\boldsymbol{m}} > 0$, $\sum_{\boldsymbol{m}} \beta_{\boldsymbol{m}} = 1$ and $\theta > 0$. Using Lagrange multipliers, it is easy to see that

$$\beta_{\boldsymbol{m}}^{(l)} = \frac{\sum_{i=1}^{n} z_{i\boldsymbol{m}}^{(l)}}{n}, \tag{3.11}$$

which is the average of all posterior probabilities over all observations. The estimate of the scale parameter $\theta^{(l)}$ satisfies

$$\theta^{(l)} = \sum_{i=1}^{n} \sum_{j=1}^{k} \frac{1}{n} \frac{x_{ij}}{1 + c_j\theta^{(l)}} \Bigg(\sum_{m_1=1}^{\infty} \cdots \sum_{m_k=1}^{\infty} \beta_{\boldsymbol{m}}^{(l)} \sum_{j=1}^{k} \frac{m_j c_j}{1 + c_j\theta^{(l)}} \Bigg)^{-1}, \tag{3.12}$$

which will be shown below to have a unique solution.

Let

$$\bar{x}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij},$$
$$\tilde{\beta}_j = \sum_{m_1=1}^{\infty}\cdots\sum_{m_k=1}^{\infty} m_j \beta_{\boldsymbol{m}}.$$

Then (3.12) becomes

$$\theta^{(l)} = \left(\sum_{j=1}^{k}\frac{\bar{x}_j}{1+c_j\theta}\right)\left(\sum_{j=1}^{k}\frac{\tilde{\beta}_j c_j}{1+c_j\theta}\right)^{-1}.$$

Moving everything to the left-hand side of the equation produces

$$\sum_{j=1}^{k}\left[\frac{\tilde{\beta}_j c_j\theta - \bar{x}_j}{1+c_j\theta}\right] = 0. \tag{3.13}$$

Since $\tilde{\beta}_j \geqslant 0$, $c_j \geqslant 0$, $\bar{x}_j \geqslant 0$, (3.13) is a monotonous function of θ, increasing from $-\sum_{j=1}^{k}\bar{x}_j \leqslant 0$ to $\sum_{j=1}^{k}\tilde{\beta}_j \geqslant 0$ as θ goes from 0 to ∞. Hence, (3.13) and thus (3.12) have a unique solution. Since (3.13) is an increasing function, (3.12) can be solved numerically with very little computational cost.

The E-step and M-step are iterated until the difference $l(\boldsymbol{\Phi}^{(l)};\boldsymbol{x}) - l(\boldsymbol{\Phi}^{(l-1)};\boldsymbol{x})$ is smaller than a predefined tolerance level. Since the number of mixtures and all the shape parameters m_j are predetermined, our fitting results may be suboptimal. However, finding the optimal fitting results means searching over $\mathbb{N}^{kM}$, where M is the number of mixtures (it remains a parameter itself to be determined, see below), which is a tedious task, if possible to complete at all. Here, we suggest using a similar searching procedure proposed by Lee and Lin (2010), which can easily be modified for our multivariate case, so we omit the details. Beside the shape parameters, the number of mixtures M should also be chosen carefully, as too many mixtures can lead to overfitting, which should always be avoided in statistics. Again we refer to a selection procedure based on information criteria proposed by Lee and Lin (2010). Their backward selection strategy greatly improves the efficiency of the algorithm and can be easily applied to our case.

Generally speaking, a good initialization is crucial for the EM algorithm. For example, Lee and Lin (2010, 2012) proposed high-quality initial values for the parameters through the Tijms (2003) approximation, which states that each positive continuous random variable can be approximated by a mixture of Erlang distributions with the same scale parameter. However, in our case, such an approximation does not exist.

TABLE 1 Number of losses categorized by business lines and event types.

Number of losses	EDPM	EF
TS	12 738 (UOM 1)	546 (UOM 2)
CS	4 620 (UOM 3)	3 501 (UOM 4)
CB	1 176 (UOM 5)	9 433 (UOM 6)
RB	2 494 (UOM 7)	342 (UOM 8)

Here we suggest initializing the parameters as

$$\beta = 0.5, \qquad \alpha_{\boldsymbol{m}} = \frac{1}{M^k},$$

where M is the maximum number of mixtures in all marginals. Based on our experiments, this EM algorithm is robust in the sense that the final fitted result is insensitive to different selections of β and $\alpha_{\boldsymbol{m}}$.

4 MODEL CALIBRATION

In this section, we aim to test the versatility of the proposed multivariate Pascal mixture model by fitting it to modified operational loss data. The data was collected by a North American financial institution from April 2007 until March 2012, including recognition data and net loss. The recording threshold is US$30 000? ie, the real losses were left-truncated at US$30 000 to produce the recorded losses. The data covers two event types (execution, delivery and process management (EDPM) and external fraud (EF)) and four business lines (trading and sales (TS), card services (CS), commercial banking (CB) and retail brokerage (RB)) and thus can be categorized into eight UOMs according to BCBS (see Table 1). We can see that some UOMs, such as UOM 2 and UOM 8, have very small number of losses, which makes any statistical inference fairly challenging. For confidentiality, we have modified the original data by resampling from it while still keeping the same sample size. By doing this the main characteristics of the original data are preserved.

The degree of the tail-heaviness varies greatly from UOM to UOM. We outline some features of the losses as follows:

- the ratio of the maximum value to the mean ranges from 44 to 492;
- the skewness and kurtosis span from 10 to 47 and from 105 to 2935, respectively;
- the percentage of the data points smaller than 0.1% of the maximum value varies from 0% to 91%.

According to Proposition 3.1, the observed loss frequency here is modified with information about the loss severities. Consequently, before fitting the original frequency distribution, we need to model the severities and fit them to the severity data first.

4.1 Modeling and calibrating the severity distributions

Due to the analytical properties and computational tractability of the class of finite mixtures of Erlang distributions (Willmot and Woo 2007; Lee and Lin 2010), we assume that the original severity distribution before left-truncation follows such a distribution. Mathematically, for the ith UOM, the loss severity has a density given by

$$f_i(x) = \sum_{j=1}^{\infty} \alpha_{ij} \frac{x^{j-1}\, \mathrm{e}^{-x/\theta_i}}{\theta_i^j\, (j-1)!}. \tag{4.1}$$

In order to estimate the parameters for this left-truncated severity data, we use an EM algorithm proposed by Verbelen *et al* (2015). The fitting results reflect the different scales among the UOMs. For example, the number of components in these mixtures ranges from 5 to 34 while the biggest and smallest values of the estimated scale parameter θ are 48 724.27 and 6957.52, respectively.

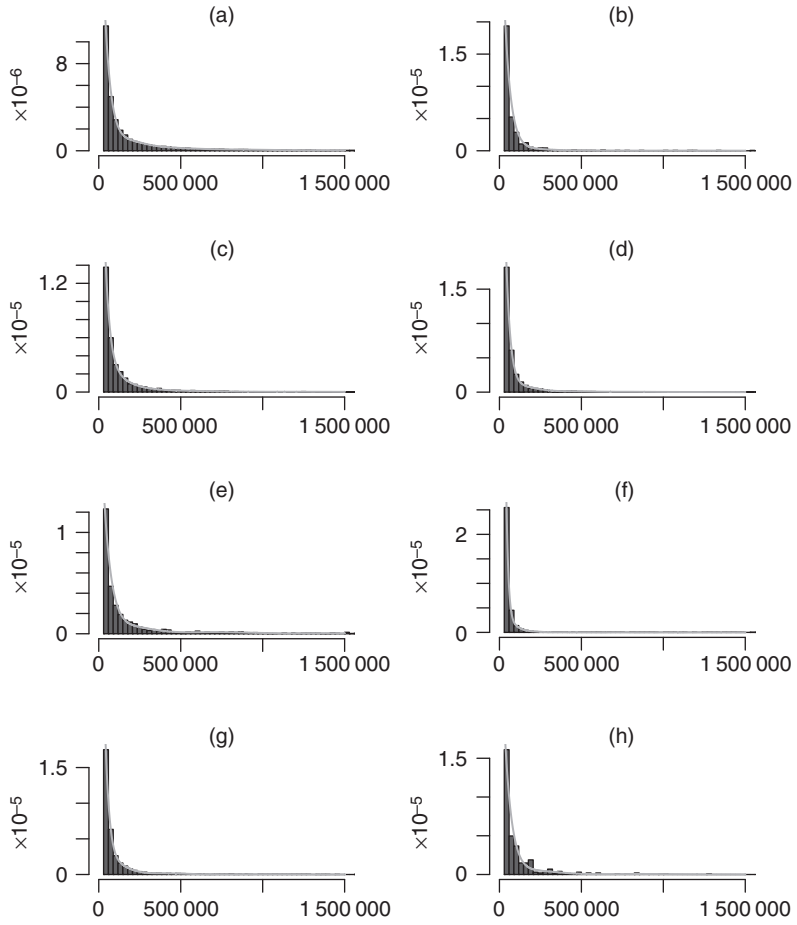
To examine the goodness-of-fit of each UOM, we produced their data histograms (Figure 1 on the next page), P–P plots (Figure 2 on page 19) and Q–Q plots (Figure 3 on page 20) with fitted densities. We also compared the empirical moments with the fitted moments (Table 2 on page 21). Considering that the fittings are done for small sample size with heavy-tailed data, these results all indicate that the fitting effect is quite satisfactory.

4.2 Calibrating the multivariate frequency distribution

The frequency data is separated into weekly bins in order to ensure that the number of data points in each bin is sufficient for statistical analysis. Since the data was collected from April 2007 to March 2012, this would mean that we have 261 weekly frequency data points, each being an eight-dimensional vector describing the frequencies of the eight UOMs. Following our framework, we model the frequency distributions across all the UOMs as a whole and take the correlations between the UOMs into consideration. This could be seen as an ambitious task. However, as we will show, our multivariate frequency model not only fits all the marginals quite well, but also captures the dependence structures among the UOMs.

We denote the left-truncation threshold by d. If we model the frequencies of the losses without left-truncation by our proposed multivariate Pascal mixture distribution of the form (2.3), then according to Proposition 3.1, the observed frequencies follow

FIGURE 1 Data versus fitted severity distributions: histograms.

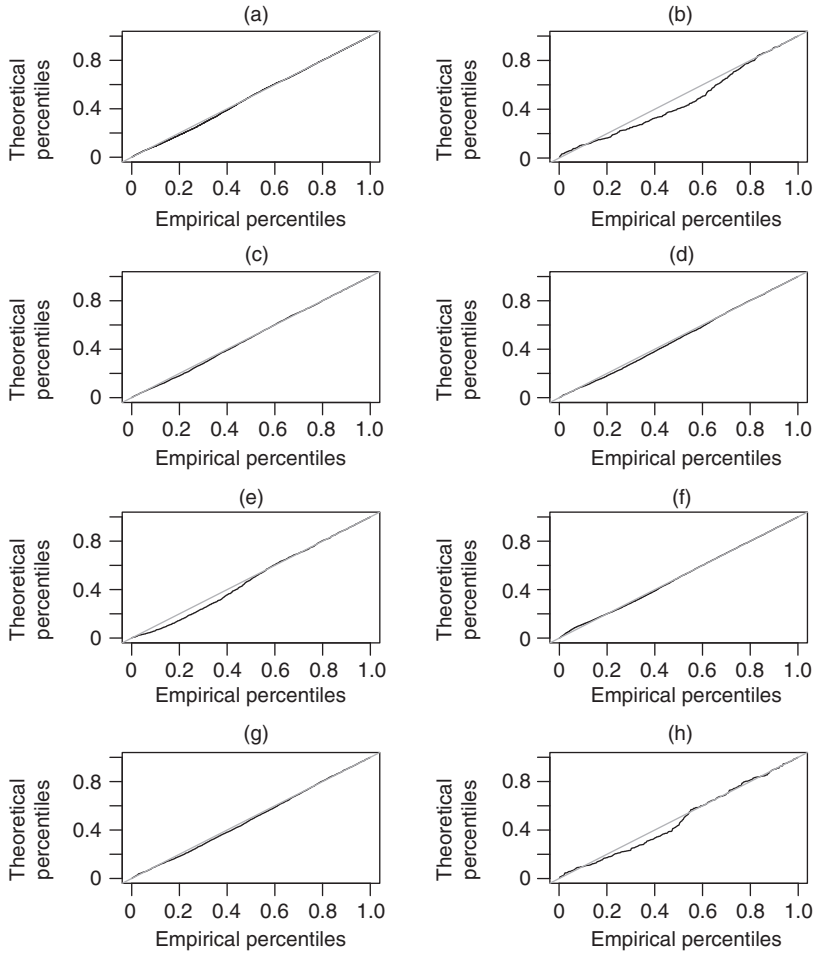


(a) UOM 1. (b) UOM 2. (c) UOM 3. (d) UOM 4. (e) UOM 5. (f) UOM 6. (g) UOM 7. (h) UOM 8.

the modified distribution given by (3.1), with all the d_j equal to d. Despite the fact that $\bar{F}_{X_j}(d)$ are unknown here, we can replace them by their estimators $\hat{\bar{F}}_{X_j}(d)$, which are calculated from the fitted severity distributions obtained in the previous subsection. Our fitting results show that these estimates have values contained in the vector of $(0.60, 0.49, 0.51, 0.39, 0.63, 0.45, 0.40, 0.55)$ (we present only the first two decimals).

Using the tailor-made EM algorithm shown in Section 3, we find a fitted distribution with thirty-two components. We check the goodness-of-fit in terms of the

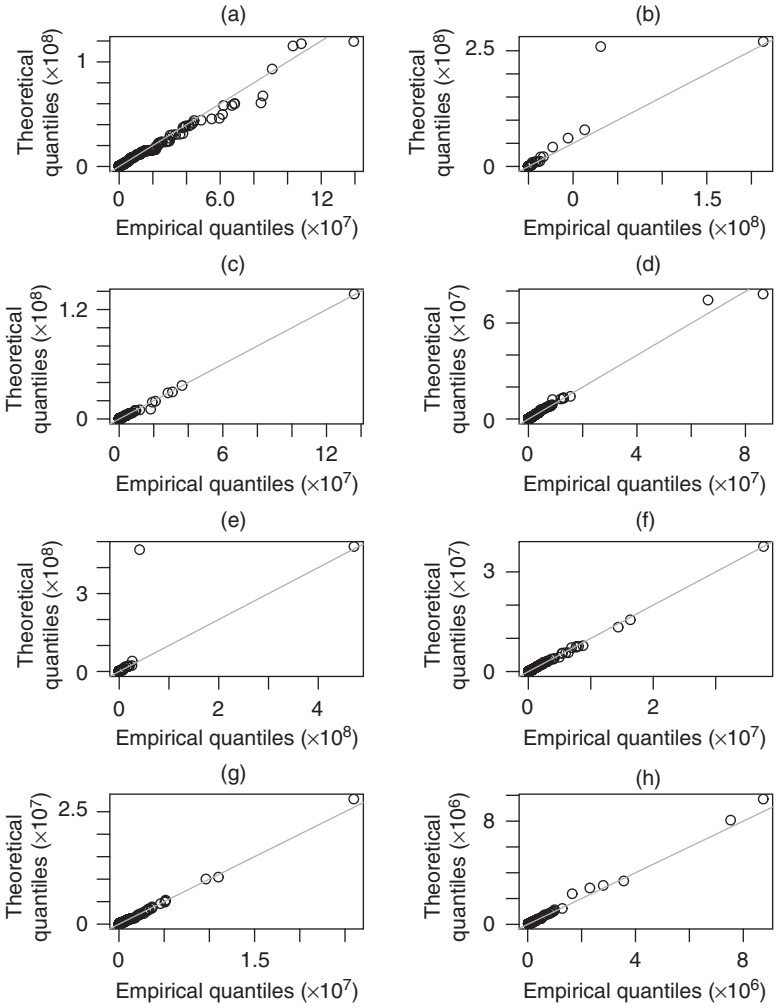
FIGURE 2 Data versus fitted severity distributions: P–P plots.



(a) UOM 1. (b) UOM 2. (c) UOM 3. (d) UOM 4. (e) UOM 5. (f) UOM 6. (g) UOM 7. (h) UOM 8.

eight marginals with various statistical tools. In Figure 4 on page 22, we compare the observed frequencies (gray) with fitted frequencies (black). These frequencies are the numbers of losses that fall into a series of predefined intervals. For example, for UOM 5 these intervals are $[0, 2), [2, 4), [4, 6), \ldots, [18, 20)$. We can observe that the fitting effect is very good. Figure 5 on page 23 and Figure 6 on page 24 give another two angles of comparison by P–P plots and Q–Q plots. The panels for UOM 2 and UOM 8 in Figure 5 indicate that for these two UOMs, the probabilities of zero frequency are

FIGURE 3 Data versus fitted severity distributions: Q–Q plots.



(a) UOM 1. (b) UOM 2. (c) UOM 3. (d) UOM 4. (e) UOM 5. (f) UOM 6. (g) UOM 7. (h) UOM 8.

approximately 0.6 and 0.4, respectively. We also note that panels for UOMs 2, 5 and 8 have significant stairway shapes. This is because for these three UOMs the frequency data is relatively clustered, so there are multiple jumps when drawing P–P plots. This explains why there are fewer points for these three UOMs in Q–Q plots as well.

An important point here is that our frequency model can capture the dependence structure within high-dimensional data very well. We calculated Kendall's τ between

TABLE 2 Data versus fitted severity distributions: moments.

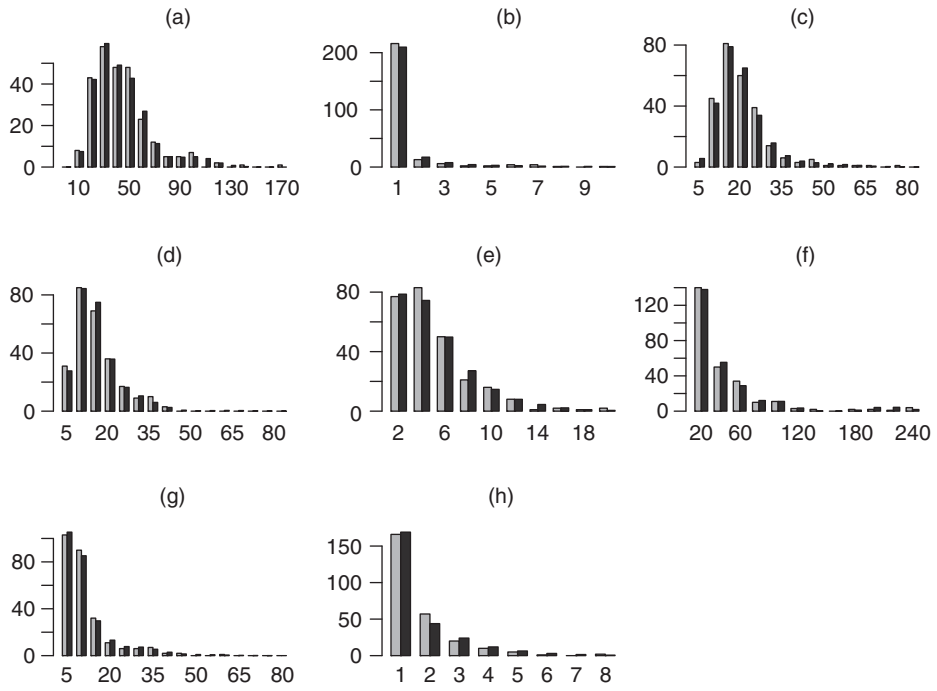
UOM		Moments				
		1st	2nd	3rd	4th	5th
1	Empirical value	608358.9	1.21E+13	8.15E+20	7.59E+28	8.13E+36
	Fitted value	608358.9	1.19E+13	7.82E+20	6.89E+28	6.83E+36
	Relative error (%)	0	−1.15	−4.09	−9.14	−15.96
2	Empirical value	1148695	1.53E+14	3.51E+22	8.92E+30	2.33E+39
	Fitted value	1148695	1.53E+14	3.51E+22	8.93E+30	2.33E+39
	Relative error (%)	0	−0.10	0.09	0.17	0.24
3	Empirical value	276199.7	5.31E+12	5.72E+20	7.47E+28	1.01E+37
	Fitted value	276199.7	5.30E+12	5.71E+20	7.46E+28	1.01E+37
	Relative error (%)	0	−0.19	−0.09	−0.05	0.01
4	Empirical value	215033.4	3.90E+12	2.73E+20	2.16E+28	1.75E+36
	Fitted value	215033.4	3.78E+12	2.53E+20	1.89E+28	1.43E+36
	Relative error (%)	0	−3.03	−7.24	−12.44	−18.40
5	Empirical value	994853.5	1.95E+14	8.94E+22	4.21E+31	1.99E+40
	Fitted value	994853.5	1.94E+14	8.92E+22	4.20E+31	1.98E+40
	Relative error (%)	0	−0.22	−0.22	−0.26	−0.30
6	Empirical value	87803.33	2.84E+11	6.84E+18	2.27E+26	8.19E+33
	Fitted value	87802.99	2.84E+11	6.84E+18	2.28E+26	8.23E+33
	Relative error (%)	0	−0.12	0.03	0.26	0.49
7	Empirical value	159181	5.03E+11	8.36E+18	1.94E+26	4.86E+33
	Fitted value	159181	5.02E+11	8.37E+18	1.95E+26	4.90E+33
	Relative error (%)	0	−0.19	0.12	0.44	0.86
8	Empirical value	197276.9	5.14E+11	3.48E+18	2.73E+25	2.23E+32
	Fitted value	197276.9	5.09E+11	3.46E+18	2.72E+25	2.23E+32
	Relative error (%)	0	−0.98	−0.52	−0.14	0.13

UOMs for the data and the fitted distributions and they are given in Table 3 on page 24, where the upper triangular entries are empirical estimates and the lower triangular entries are estimates from the fitted model.

Although not shown explicitly here, we can also calculate the relative errors of fitted values with respect to empirical values, which are summarized below:

- 32.1% of the relative errors are less than 5%;
- 28.6% of them are between 5% and 10%;
- 17.9% of them are between 10% and 15%;

FIGURE 4 Data versus fitted frequency distributions: barplots.



(a) UOM 1. (b) UOM 2. (c) UOM 3. (d) UOM 4. (e) UOM 5. (f) UOM 6. (g) UOM 7. (h) UOM 8.

- 14.3% of them are between 15% and 20%;
- 7.1% of them (only two) are above 20%.

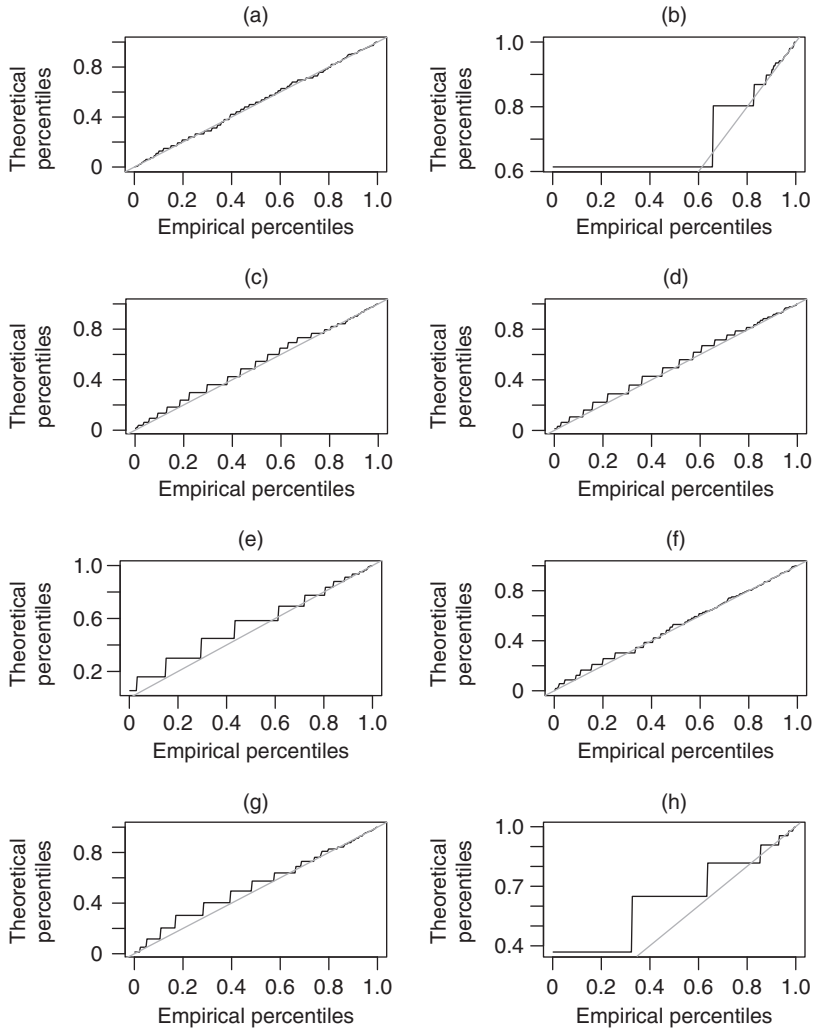
The biggest two relative errors are the 22.6% between UOMs 4 and 8, and the 20.6% between UOMs 5 and 8. Considering that there are only 342 losses for UOM 8, these estimates are quite satisfactory.

5 MODEL VALIDATION AND FITTING AGGREGATE LOSSES AT THE TOP OF THE HOUSE LEVEL

In this section, we validate our model using both in-sample and out-sample techniques. In particular, we examine the fitness of our model at the TOH level.

The validation process of a model consists of three parts: a review of the data, assumptions and model outputs. In this paper we deem that the data used here is both appropriate and valid. Thus, we focus on the other two parts using statistical tools.

FIGURE 5 Data versus fitted frequency distributions: P–P plots.

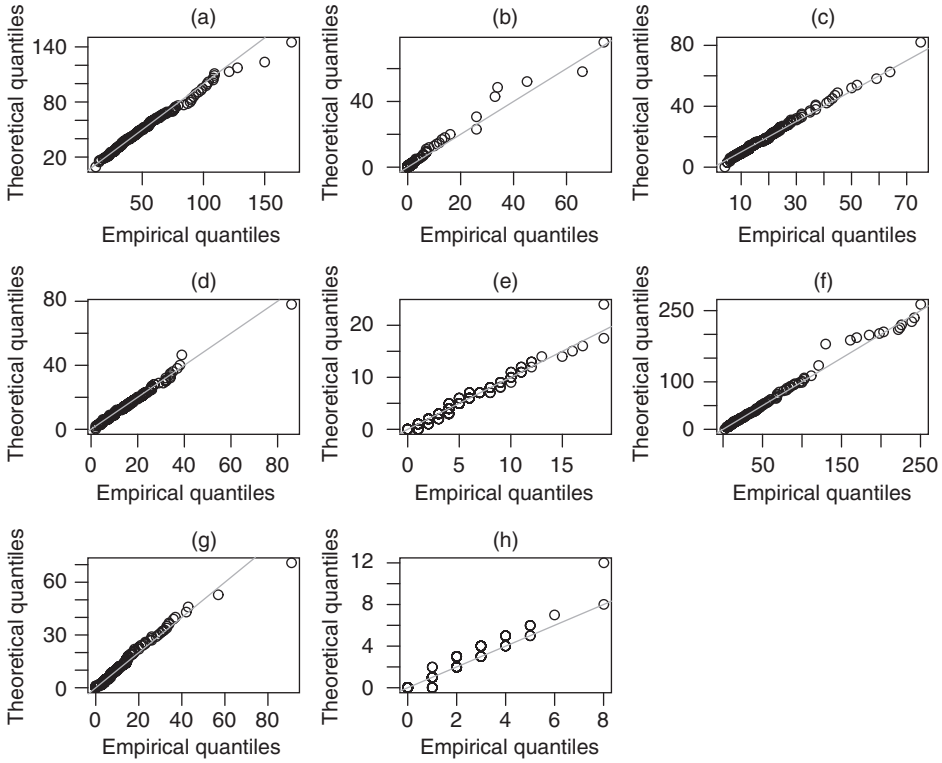


(a) UOM 1. (b) UOM 2. (c) UOM 3. (d) UOM 4. (e) UOM 5. (f) UOM 6. (g) UOM 7. (h) UOM 8.

5.1 Assumption validation test

In our model we assume that the severities among different UOMs are independent. We test this assumption by calculating their Kendall's τ, which is more suitable than the correlation coefficient when measuring the dependence of two differently scaled data sets. For two given UOM, Kendall's τ is calculated as that of the two

FIGURE 6 Data versus fitted frequency distributions: Q–Q plots.



(a) UOM 1. (b) UOM 2. (c) UOM 3. (d) UOM 4. (e) UOM 5. (f) UOM 6. (g) UOM 7. (h) UOM 8.

TABLE 3 Data versus fitted frequency distributions: Kendall's τ.

	UOM 1	**UOM 2**	**UOM 3**	**UOM 4**	**UOM 5**	**UOM 6**	**UOM 7**	**UOM 8**
UOM 1	1.0000	0.2549	0.1060	0.3234	0.0821	0.2735	0.3055	0.1352
UOM 2	0.2496	1.0000	0.1925	0.2539	0.1580	0.2563	0.2412	0.1653
UOM 3	0.0932	0.1643	1.0000	0.1836	0.1988	0.1442	0.2557	0.0544
UOM 4	0.3272	0.2523	0.1960	1.0000	0.1059	0.3454	0.2715	0.1019
UOM 5	0.0846	0.1442	0.1815	0.1024	1.0000	0.1766	0.1123	0.0453
UOM 6	0.2806	0.2780	0.1582	0.3592	0.1673	1.0000	0.3242	0.1790
UOM 7	0.3234	0.2821	0.2238	0.2814	0.0982	0.2999	1.0000	0.1640
UOM 8	0.1140	0.1608	0.0640	0.0788	0.0546	0.1584	0.1386	1.0000

TABLE 4 Kendall's τ between severities from different UOMs.

	UOM 1	UOM 2	UOM 3	UOM 4	UOM 5	UOM 6	UOM 7	UOM 8
UOM 1	1	0.032	0.080	0.078	0.049	0.062	0.026	0.029
UOM 2		1	−0.028	0.108	−0.048	0.136	0.094	0.145
UOM 3			1	0.004	0.081	0.008	−0.002	0.040
UOM 4				1	0.091	0.072	0.081	0.058
UOM 5					1	0.093	0.030	0.023
UOM 6						1	0.051	0.059
UOM 7							1	0.017
UOM 8								1

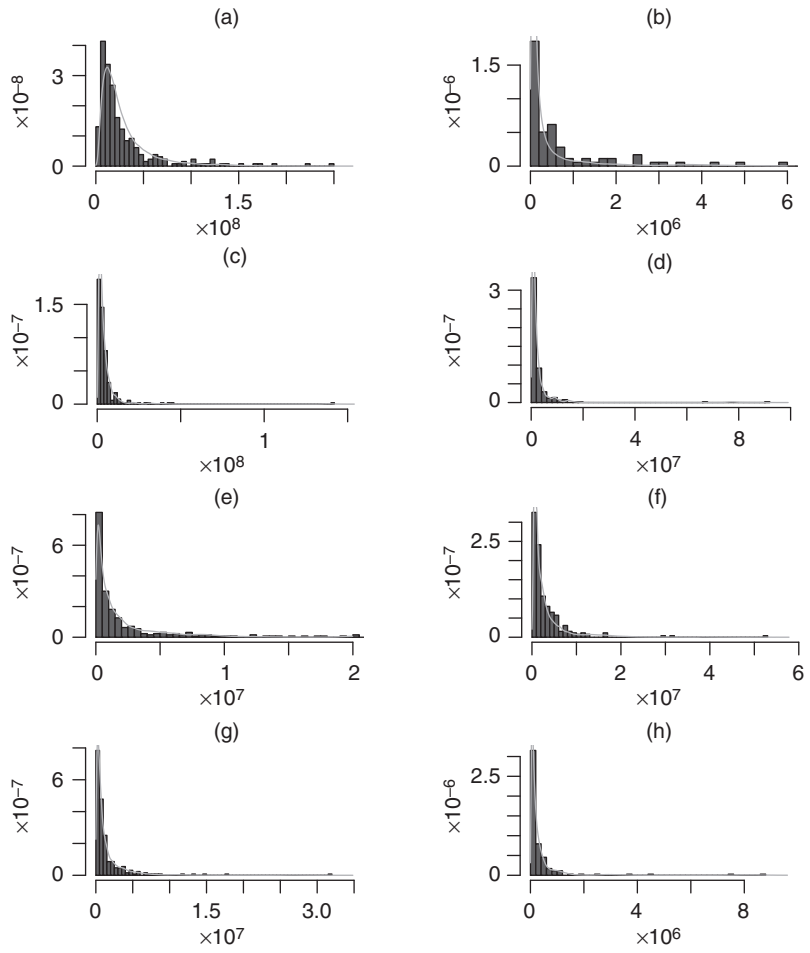
average weekly loss severities given that at least one loss occurred in both UOMs. The results are summarized in Table 4. The small numbers reconfirm that the independence assumption is reasonable.

5.2 In-sample validation test

Since the compound distribution of random sum (2.12) depends on its frequency distribution, a poor fitting to the frequency data would result in general in a poor fitting of the compound distribution as well. This suggests that for model validation purposes, we should assess the performance of our model at the aggregate loss distribution level. Figure 7 on the next page, Figure 8 on page 27 and Figure 9 on page 28 compare data and fitted aggregate loss distributions of each UOM, on a weekly basis, by means of histograms, P–P plots and Q–Q plots. Here, for each UOM, we simulate one million scenarios from the marginal frequency distribution of the fitted multivariate (weekly) frequency distribution and then simulate corresponding numbers of losses from the fitted severity distributions. All graphs indicate a satisfactory performance of our model. Note again that there are flat segments in the P–P plots for UOMs 2 and 8, which is due to the fact that the probabilities that there is no loss occurring in these two UOMs are roughly 0.6 and 0.4, respectively.

In addition, we can also compare the empirical and fitted VaR, which is of paramount importance in OR modeling, of the weekly aggregate losses. Here we choose VaRs at a confidence level of 99.9% and 99.95% respectively, as they are usually the confidence level required for the regulatory capital (RC) and the economic capital (EC). We summarize the relevant results in Table 5 on page 29 and Table 6 on page 29. Note that because of the scarcity of data any percentile greater than 99% is unstable regardless of the underlying distributions. As a result, it is almost impossible to get reliable estimates for the 99.9% and 99.95% percentiles. Nevertheless, about half of the relative errors are still below 10%, which shows the predictive power of our frequency model.

FIGURE 7 Data versus fitted aggregate loss distributions: histograms.

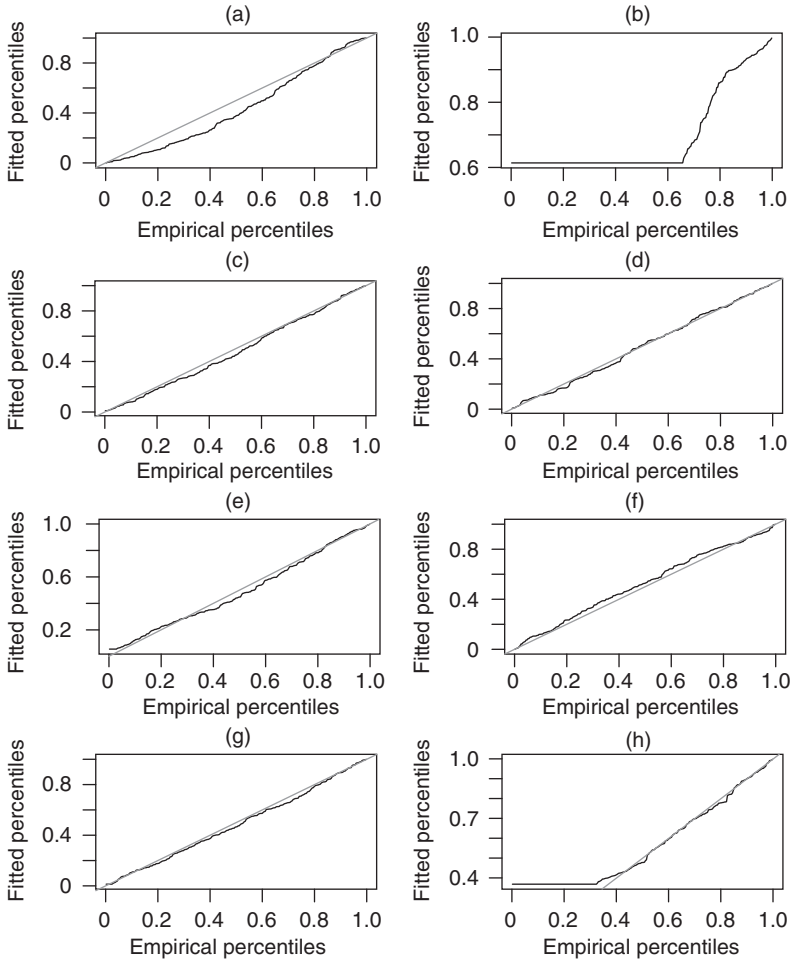


(a) UOM 1. (b) UOM 2. (c) UOM 3. (d) UOM 4. (e) UOM 5. (f) UOM 6. (g) UOM 7. (h) UOM 8.

5.3 Out-sample validation test

Because of the scarcity of the operational loss data, we use all the data available to effectively calibrate the model, which means a hold-out data set is not possible. Here we suggest generating an out-sample by bootstrapping (see, for example, Hall 1995). Although this resampling process cannot increase the amount of information contained in the original data, it does reduce the effects of random sample errors when the number of resamplings is large.

FIGURE 8 Data versus fitted aggregate loss distributions: P–P plots.

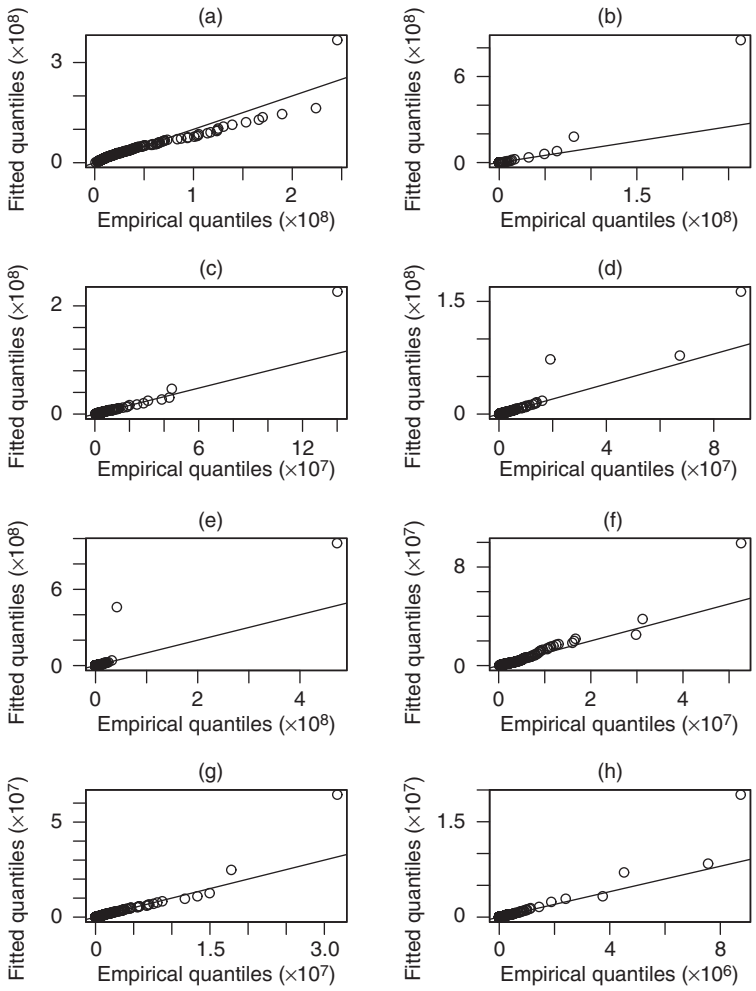


(a) UOM 1. (b) UOM 2. (c) UOM 3. (d) UOM 4. (e) UOM 5. (f) UOM 6. (g) UOM 7. (h) UOM 8.

Our frequency data can be expressed as 261 vector points, each representing the eight numbers of weekly losses at each UOM. We propose the following resampling procedure.

- First we resample the frequency vectors randomly with replacement, with the same size of 261 vector points. We resample the vectors directly because this could preserve the dependence structure among the frequencies.

FIGURE 9 Data versus fitted aggregate loss distributions: Q–Q plots.



(a) UOM 1. (b) UOM 2. (c) UOM 3. (d) UOM 4. (e) UOM 5. (f) UOM 6. (g) UOM 7. (h) UOM 8.

- Assume that the number of losses for each cell obtained in the first step is $n_1, n_2, \ldots, n_8$. We resample n_i numbers of loss amounts with replacement from the severity data of the ith UOM.
- The statistic of interest is computed from the resample obtained from the first two steps. We repeat this routine 100 000 times to get a more precise estimate of the bootstrap distribution of the statistic.

TABLE 5 Data versus fitted weekly aggregated loss distributions: VaR at 99.9%.

	VaR (millions)		
	Empirical value	**Fitted value**	**Relative error (%)**
UOM 1	239.77	200.93	−16.20
UOM 2	215.98	282.42	30.76
UOM 3	115.16	142.51	23.75
UOM 4	84.15	82.23	−2.28
UOM 5	360.56	478.21	32.63
UOM 6	47.01	49.49	5.28
UOM 7	28.07	29.63	5.56
UOM 8	8.44	9.29	10.17

TABLE 6 Data versus fitted weekly aggregated loss distributions: VaR at 99.95%.

	VaR (millions)		
	Empirical value	**Fitted value**	**Relative error (%)**
UOM 1	242.58	220.96	−8.91
UOM 2	239.64	317.38	32.43
UOM 3	127.58	145.90	14.35
UOM 4	87.10	85.45	−1.90
UOM 5	416.64	482.29	15.76
UOM 6	49.78	53.57	7.62
UOM 7	29.87	31.55	5.62
UOM 8	8.59	9.71	13.07

We can validate our fitted model in terms of VaR of the total aggregate loss at the TOH level as this is related to the regulatory capitals required by the Basel II accord, especially at the level of 99.9% and 99.95%, as mentioned earlier. We should note that all the results here are on a weekly basis and serve to validate our fitted model. We use the fitted severity model for each UOM and the fitted multivariate frequency model to simulate one million scenarios for the total aggregate losses at the TOH level, in the same way as how we bootstrap from the original data. Since our model is a mixture, simulation is straightforward and can be performed very fast and easily. In Table 7 on the next page, we summarize the predicted values (based on the simulated distribution from our fitted model) and the confidence intervals (based on bootstrap resamples) of VaRs at various levels. For visual convenience, the same results are illustrated in

TABLE 7 VaRs and their confidence intervals (in millions) based on bootstrap resamples.

Percentile	Predicted value	Lower bound of 90% CI	Upper bound of 90% CI
92.00	103.851	86.490	119.963
93.00	110.103	91.490	127.770
94.00	117.593	96.544	137.268
95.00	127.124	102.686	149.005
96.00	138.597	110.951	163.122
97.00	152.735	120.089	183.463
98.00	173.758	124.167	226.768
99.00	235.249	112.955	377.254
99.90	537.507	241.453	634.856
99.95	572.806	248.675	658.224

Figure 10 on the facing page. We can see that for the first eight levels from 92% to 99%, the predicted values are in the middle of the corresponding 90% confidence interval. These confidence intervals are relatively narrow, indicating that the original data provides sufficient information for prediction. When there is not enough data, as in the case of the last two VaRs, the confidence intervals are wide. We observe that our predicted values at these two levels are a bit over the middle of the confidence intervals.

REMARK 5.1 Since the LDA framework implicitly assumes that the operational risk data is stationary over time, a duration component can be injected into the mixed Poisson distribution (2.1),

$$\begin{aligned}&\mathbb{P}(N_1(t)=n_1,\ldots,N_k(t)=n_k)\\&\quad=\int_{(0,\infty)^k}\left(\prod_{j=1}^{k}\mathrm{e}^{-\lambda_j t}\frac{(\lambda_j t)^{n_j}}{n_j!}\right)\pi(\lambda_1,\ldots,\lambda_k;\boldsymbol{\beta},\theta)\,\mathrm{d}\lambda_1,\ldots,\mathrm{d}\lambda_k,\end{aligned}\tag{5.1}$$

where $\pi(\lambda_1,\ldots,\lambda_k)$ is the same as in (2.2) and t is duration. It can be rewritten as

$$\mathbb{P}(N_1(t)=n_1,\ldots,N_k(t)=n_k)=\sum_{m_1=1}^{\infty}\cdots\sum_{m_k=1}^{\infty}\beta_{\boldsymbol{m}}\prod_{j=1}^{k}p(n_j;m_j,t\theta),\tag{5.2}$$

where $p(n_j,m_j,t\theta)$ is the Pascal distribution function defined in (2.4). Consequently, if we choose one week as the unit of time and fit the weekly frequency first, we only need to change the scale parameter θ in the fitted model to 52θ to obtain a yearly frequency distribution. We can then calculate the required one-year RC or EC

FIGURE 10 VaRs of total aggregate loss at TOH: predicted values and confidence intervals.

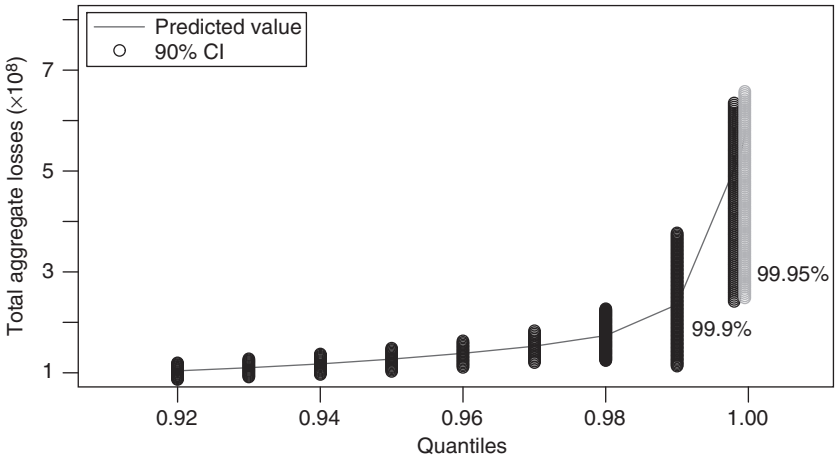


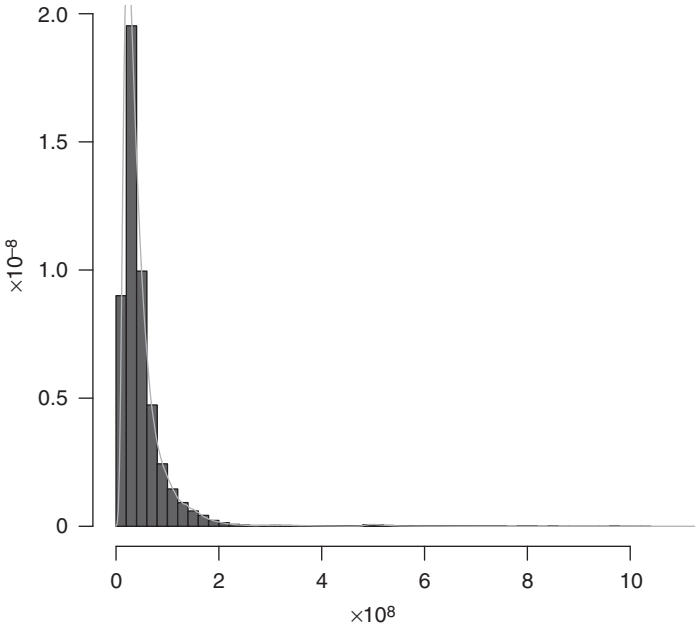
TABLE 8 Predicted RC and EC using the fitted model at individual UOM and TOH level.

VaRs (millions)	99.9% (RC)	99.95% (EC)
UOM 1	4 261.54	4 339.86
UOM 2	4 137.95	4 396.47
UOM 3	1 015.84	1 026.36
UOM 4	805.58	859.79
UOM 5	1 559.24	1 743.21
UOM 6	1 103.11	1 127.74
UOM 7	481.66	499.88
UOM 8	76.48	79.17
TOH	10 262.01	10 435.94

with serial correlations being taken into account. For example, based on our fitted weekly frequency distribution, we can easily predict one-year RCs and ECs at both the individual UOM level and the TOH level, as shown in Table 8.

We can also validate the model by fitting the total aggregate losses at the TOH level itself. Based on the previous bootstrap resamples and one million simulated scenarios, we draw histograms (Figure 11 on the next page) and Q–Q plots (Figure 12 on page 33), which show that our model performs very well. We test the goodness of

FIGURE 11 Total aggregate losses at TOH level: histograms.



fit of the total aggregate losses at the TOH level. As shown in Table 9 on the facing page, the fitted distribution passes the Kolmogorov–Smirnov test with a significant margin.

Finally in this section, we examine the impact of the dependency of frequencies on RC and EC. Assume instead that the losses across the UOMs are independent. In this case, we can either fit a univariate Pascal mixture (ie, $k = 1$ in (2.3)) to the number of losses in each UOM and the multivariate independent model is the product of those marginal distributions (Method 1), or we can use Corollary 2.3 to obtain the marginal distributions without going through the fitting procedure (Method 2). Another advantage of Method 2 is that both the dependent and independent models have identical marginal distributions. Since our dependent model fits the data well, the values of RC and EC under Method 2 are expected to be very close to those under Method 1. As the frequencies are positively correlated, as shown in Table 3 on page 24, we can also anticipate that the total aggregate losses at the TOH level will be relatively light-tailed, implying that the values of RC and EC should be smaller. Our numerical results in Table 10 on the facing page reconfirm both presumptions.

FIGURE 12 Total aggregate losses at TOH level: Q–Q plot.

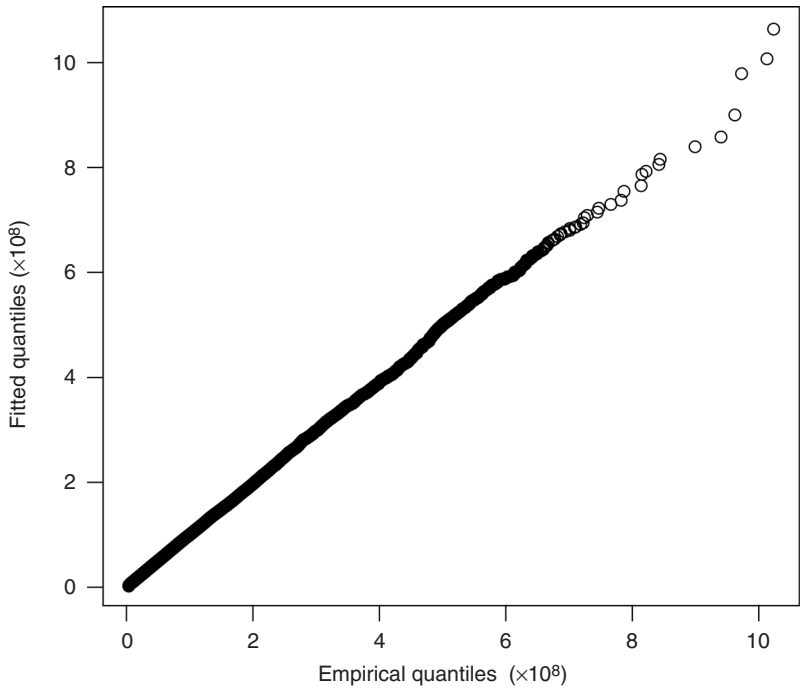


TABLE 9 Statistical test for fitness of model to resampled data at TOH level.

Test	Statistics	p-value	Accepted at 5% significant level?
KS Test	0.0593	0.3171	Yes

TABLE 10 Predicted RC and EC at TOH level: dependent frequencies versus independent frequencies.

	99.9% (RC)		99.95% (EC)	
VaRs (millions)	**Method 1**	**Method 2**	**Method 1**	**Method 2**
Independent frequencies	7 203.93	7 275.97	7 566.06	7 626.59
Dependent frequencies	10 262.01		10 435.94	

Moreover, these results show that the dependence has a very significant impact on the RC and EC calculation. When assuming independence, the values of RC and EC are approximately 30% lower than those when dependence is incorporated into the model.

6 FURTHER RESULTS AND CONCLUDING REMARKS

In this section, we assume the individual losses to be mixed Erlang distributed and derive a closed-form expression for the total aggregate losses at the TOH level. To this end, we define the density of a k-variate zero-modified Erlang mixture $\boldsymbol{X}$ as

$$f_{\boldsymbol{X}}(\boldsymbol{x}) = \sum_{m_1=0}^{\infty} \cdots \sum_{m_k=0}^{\infty} \gamma_{\boldsymbol{m}} \prod_{j=1}^{k} q(x_j; m_j, \theta), \tag{6.1}$$

where

$$q(x_j; m_j, \theta) = \frac{x_j^{m_j-1}\, \mathrm{e}^{-x_j/\theta}}{\theta^{m_j}\,(m_j-1)!}. \tag{6.2}$$

Here $q(x; m, \theta)$ is defined as 1 when both x and m are 0. Therefore, it is easy to show that the MGF of $\boldsymbol{X}$ is

$$M_{\boldsymbol{X}}(\boldsymbol{z}) = P_{\boldsymbol{M}}\left(\frac{1}{1-\theta z_1}, \ldots, \frac{1}{1-\theta z_k}\right), \tag{6.3}$$

where $\boldsymbol{M}$ has probability function

$$\mathbb{P}(\boldsymbol{M} = \boldsymbol{m}) = \gamma_{\boldsymbol{m}}, \quad m_j = 1, 2, \ldots, \ j = 1, \ldots, k. \tag{6.4}$$

It is also quite direct to show that $X = X_1 + \cdots + X_k$ is a zero-modified univariate Erlang mixture with density

$$f_X(x) = \begin{cases} \gamma_0 & \text{if } x = 0, \\ \sum_{m=1}^{\infty} \gamma_m q(x; m, \theta) & \text{if } x > 0, \end{cases} \tag{6.5}$$

where

$$\gamma_m = \sum_{m_1+\cdots+m_k=m} \gamma_{\boldsymbol{m}}, \quad m = 0, 1, \ldots. \tag{6.6}$$

Now, let us assume there are k UOMs and aim to find the joint distribution of the aggregate losses of each UOM, ie, $\boldsymbol{S} = (S_1, \ldots, S_k)$. Their frequencies $\boldsymbol{N}$ have a joint probability function of the form of (2.3) and, for the ith UOM, the severity X_i follows an Erlang mixture distribution with density (4.1). For all the UOMs, of course, the scale parameters θ_i will not necessarily have the same value and we denote their

minimum value by θ, ie, $\theta = \min\{\theta_1, \ldots \theta_k\}$. Using a result from Willmot and Woo (2007), the severity density of the ith UOM can be reexpressed as

$$f_{X_i}(x) = \sum_{j=1}^{\infty} \tilde{\alpha}_{ij} \frac{x^{j-1}\, \mathrm{e}^{-x/\theta}}{\theta^j (j-1)!}, \tag{6.7}$$

where the modified weights are

$$\tilde{\alpha}_{ij} = \sum_{k=1}^{j} \alpha_{ik} \binom{j-1}{k-1} \left(\frac{\theta}{\theta_i}\right)^k \left(1 - \frac{\theta}{\theta_i}\right)^{j-k}. \tag{6.8}$$

It is easy to check that its MGF equals

$$M_{X_i}(z_i) = P_{\tilde{W}_i}\left(\frac{1}{1-\theta_i z_i}\right), \tag{6.9}$$

where $\tilde{W}_i$ is a discrete random variable with probability function

$$\mathbb{P}(\tilde{W}_i = j) = \tilde{\alpha}_{ij}. \tag{6.10}$$

It is also easy to see that the MGF of $\boldsymbol{S}$ satisfies

$$M_{(S_1,\ldots,S_k)}(z_1, \ldots, z_k) = P_{(N_1,\ldots,N_k)}(M_{X_1}(z_1), \ldots, M_{X_k}(z_k)). \tag{6.11}$$

Plugging (6.9) into (6.11), we get

$$\begin{aligned} M_{(S_1,\cdot,S_k)}(z_1, \ldots, z_k) &= P_{(N_1,\ldots,N_k)}\left(P_{\tilde{W}_1}\left(\frac{1}{1-\theta z_1}\right), \ldots, P_{\tilde{W}_k}\left(\frac{1}{1-\theta z_k}\right)\right) \\ &= P_{(L_1,\ldots,L_k)}\left(\frac{1}{1-\theta z_1}, \ldots, \frac{1}{1-\theta z_k}\right), \end{aligned} \tag{6.12}$$

where L_i is a compound random variable with counting distribution N_i and severity distribution $\tilde{W}_i$. Here $\tilde{W}_i$ corresponds to the transformed mixing weights, ie, $\mathbb{P}(\tilde{W}_i = j) = \tilde{\alpha}_{ij}$. By Proposition 2.10, its joint probability function can be calculated recursively using Panjer's recursion method. Comparing (6.12) with (6.3) reveals that $\boldsymbol{S}$ is a zero-modified k-variate Erlang mixture with the scale parameter being θ and the mixing distribution being $\boldsymbol{L} = (L_1, \ldots, L_k)$. As a result of (6.5), $S = S_1 + \cdots + S_k$ follows a zero-modified Erlang mixture distribution with scale parameter θ and mixing weights calculated according to (6.6). According to results in Lee and Lin (2010), risk measures including VaR can easily be calculated using any numerical method.

Theoretically, the above closed-form expression for the total aggregate losses at the TOH level gives us the luxury of avoiding the use of simulation altogether. This

feature is very important because the simulation method has many drawbacks, such as a high degree of complexity, assumptional in-transparency and requiring important resources (eg, staff, time and IT). Nevertheless, there exist some computational issues when implementing the above approach. By transforming the original severity density function by (6.7), the new density function has infinite terms. Although the remote terms have slim probabilities, they are important for describing the heavy tail of the severities. We can adopt many criteria to decide after how many terms the infinite series is cut off, such as matching of moments and matching of VaRs. In our experiments, this results in tens of thousands of terms. Consequently, when we use Panjer's recursion to calculate the distribution of L_i, the computational time is relatively long. Beside this, no matter how many terms are kept, the rounding error will accumulate, as we are calculating a compound distribution here. However, the numerical implementation of the closed-form expression for the aggregate losses at TOH level represents the topic of our future research investigations.

APPENDIX A. R CODES FOR FITTING THE FREQUENCIES

```
#################################################################
## EM algorithm for Multivariate Negative Binomial Mixtures ##
#################################################################

## Libraries

library(MASS)

## Distribution function of Negative Binomial (r,beta)

pNB <- function(n, r, beta){
  #pNB <- gamma(n+r)/gamma(r)/factorial(n)*(1/(1+beta))^r*(beta/(1+beta))^n
  pNB <- dnbinom(n, size=r, prob=1/(1+beta))
  return(pNB)
}

## Initial values

multiNB.initial <- function(data, num.mixtures){
  beta <- 1
  r <- matrix(NA, nrow=num.mixtures, ncol=length(data[1,]))
  for (i in 1:num.mixtures){
    r[i,]=sample(seq(1:num.mixtures), length(data[1,]), replace=TRUE)
  }
  alpha <- rep(1,num.mixtures)/num.mixtures
  list(beta = beta, r = r, alpha = alpha)
}

## Another way to initialize

nb.initialize<- function(d, res, cutoff){
  lambda <- res/max(d)
  location <- matrix(as.integer(d*lambda + 1), , length(d[1,]))
  sortl <- order(location[,1], location[,2], location[,3], location[,4],
  location[,5],location[,6],location[,7],location[,8])
  location <- location[sortl,]
  temp = rep(1,length(location[,1]))
  for (i in length(location[,1]):2){
    if (all(location[i,]- location[i-1,] ==0 )) {
      temp[i-1] = temp[i-1]+temp[i]
      temp[i]=0
    }
```

```
  }
  location <- location[temp!=0,]
  temp <- temp[temp!=0]
  if (cutoff <= res){
    sortw <- order(temp)[(length(temp)-cutoff+1 ):length(temp)]
    location <- location[sortw,]
    alpha <- temp[sortw]
    alpha <- alpha/sum(alpha)
    init <- list(alpha = alpha, beta = 1/lambda, r = location)
  }
  else {
    print("Bad Input")
    init <- NA
  }
  init
}

## Log likelihood

multiNB.loglikelihood.OR <- function(x, r, beta, alpha, c){ # x:
data; r: size; c: coefs before beta
  # matrix containing densities products
  x.densities <- matrix(1,nrow=length(x[,1]),ncol=length(alpha))
  for (k in 1:length(alpha)){
    for (j in 1:length(x[1,])){
      x.densities[,k]=pNB(x[,j],r[k,j],c[j]*beta)*x.densities[,k]
    }
  }
  # matrix containing alpha*density
  x.components <- sweep(x.densities,2,alpha,FUN="*")
  # likelihood
  likelihood.contribution <- rowSums(x.components)
  # loglikelihood
  loglikelihood.contribution <- ifelse(likelihood.contribution>0,
                                       log(likelihood.contribution), -1000)
  multiNB.loglikelihood <- sum(loglikelihood.contribution)
  return(multiNB.loglikelihood)
}

## z_{ij}^{k}: posterior probabilities

multiNB.z.OR <- function(x, r, beta, alpha, c){
  # matrix containing densities products
  x.densities <- matrix(1,nrow=length(x[,1]),ncol=length(alpha))
  for (k in 1:length(alpha)){
    for (j in 1:length(x[1,])){
      x.densities[,k]=pNB(x[,j],r[k,j],c[j]*beta)*x.densities[,k]
    }
  }
  # matrix containing alpha*density
  x.components <- sweep(x.densities,2,alpha,FUN="*")
  multiNB.z <- sweep(x.components, 1, rowSums(x.components), FUN="/")
  # in case all z_{ij}^{k} for j=1,...,M are numerically 0
  multiNB.z[is.nan(multiNB.z)] = 1/length(alpha)
  return(multiNB.z)
}

## An isolated recursive step for beta

recu.beta <- function(data, r, beta, alpha, c){
  component1 <- sweep(data, 2, 1+c*beta, "/")
  numerator <- sum(component1)/length(data[,1])
  component2 <- sweep(r, 2, c/(1+c*beta), "*")
  denominator <- sum(alpha*rowSums(component2))
  recu.beta <- numerator/denominator
  recu.beta
}

## EM algorithm

multiNB.em.OR <- function(data, r, beta, alpha, c, eps=1e-03,
print=TRUE){
```

```
  n <- length(data[,1])
  iteration <- 1
  loglikelihood <- multiNB.loglikelihood.OR(data, r, beta, alpha, c)
  old.loglikelihood <- -Inf
  history.loglikelihood <- loglikelihood
  while(loglikelihood - old.loglikelihood > eps){
    old.loglikelihood <- loglikelihood
    # E step
    z <- multiNB.z.OR(data, r, beta, alpha, c)
    # M step
    alpha <- colSums(z)/n
    beta <- recu.beta(data,r,beta,alpha,c)
    iteration <- iteration + 1
    loglikelihood <- multiNB.loglikelihood.OR(data, r, beta, alpha, c)
    if(print) print(loglikelihood)
    history.loglikelihood <- c(history.loglikelihood, loglikelihood)
  }
  k <- dim(data)[2]
  list(alpha = alpha, r = r, beta = beta, loglikelihood = loglikelihood,
       history.loglikelihood = history.loglikelihood, iteration = iteration,
       AIC=-2*loglikelihood+2*((k+1)*length(alpha)+1),
       BIC=-2*loglikelihood+((k+1)*length(alpha)+1)*log(length(data)))
}

## Shape adjustments

multiNB.shapes.OR <- function(data, r, beta, alpha, c, eps=1e-03,
print=TRUE, alpha.limit = 0.0001){
  num.mixture <- length(r[,1]) # number of mixtures
  k <- length(data[1,]) # dimension of data
  # Try increasing the sizes r
  iteration <- 1
  if (print) cat("Try increasing the sizes", "\n")
  for (j in 1:k){
    for (i in 1:num.mixture){
      if (print) cat("Consider r(",i,j,")","\n")
      improve <- T
      while ((improve == T) && (alpha[i] > alpha.limit)){
        fit <- multiNB.em.OR(data,r,beta,alpha,c,print=FALSE)
        loglikelihood <- fit$loglikelihood
        new.r <- r
        new.r[i,j] <- new.r[i,j]+1
        new.fit <- multiNB.em.OR(data,new.r,beta,alpha,c,print=FALSE)
        new.loglikelihood <- new.fit$loglikelihood
        if (new.loglikelihood - loglikelihood > eps){
          r <- new.r
          iteration <- iteration + 1
          beta <- new.fit$beta
          alpha <- new.fit$alpha
        }
        else {improve <- F}
      }
    }
  }
  if (print) cat("Try decreasing the sizes","\n")
  for (j in 1:k){
    for (i in 1:num.mixture){
      if (print) cat("Consider r(",i,j,")","\n")
      improve <- T
      while ((improve == T) && (r[i,j] >= 2) && (alpha[i] > alpha.limit)){
        fit <- multiNB.em.OR(data,r,beta,alpha,c,print=FALSE)
        loglikelihood <- fit$loglikelihood
        new.r <- r
        new.r[i,j] <- new.r[i,j]-1
        new.fit <- multiNB.em.OR(data,new.r,beta,alpha,c,print=FALSE)
        new.loglikelihood <- new.fit$loglikelihood
        if (new.loglikelihood - loglikelihood > eps){
          r <- new.r
          iteration <- iteration + 1
          beta <- new.fit$beta
          alpha <- new.fit$alpha
        }
```

```
        else {improve <- F}
      }
    }
  }
  list(iteration=iteration,alpha=alpha, r=r, beta=beta,
       loglikelihood=loglikelihood,
       AIC=-2*loglikelihood+2*((dim(data)[2]+1)*length(alpha)+1),
       BIC=-2*loglikelihood+((dim(data)[2]+1)*length(alpha)+1)*log(length(data)))
}

## Reduction of Number of Mixtures based on AIC/BIC

multiNB.IC.OR <- function(data, r, beta, alpha, c, criterium="BIC",
eps=1e-03,
                          print=TRUE, alpha.limit = 0.0001){
  fit <- multiNB.shapes.OR(data, r, beta, alpha, c, eps, print=F, alpha.limit)
  loglikelihood <- fit$loglikelihood
  IC <- fit[[criterium]]
  r <- fit$r
  beta <- fit$beta
  alpha <- fit$alpha
  num.mixture <- length(alpha)
  if(print){
    cat("Number of mixtures = ", num.mixture, ",", "\n", "r = ", "\n")
    print(r)
    cat("alpha = ", alpha, "\n")
    cat("beta = ", beta, ",", criterium, " = ", IC, "\n", "\n")
  }
  improve <- T
  while((improve == T) && (length(alpha) > 1)){
    new.r <- r[alpha != min(alpha),]
    new.alpha <- alpha[alpha != min(alpha)]
    new.alpha <- new.alpha/sum(new.alpha)
    if (is.vector(new.r)){
      if (length(data[1,])==1){
        new.r <- matrix(new.r,ncol=1)
      }
      else {new.r <- matrix(new.r,nrow=1)}
    }
    fit <- multiNB.shapes.OR(data, new.r, beta, new.alpha, c, eps, print=F,
                             alpha.limit)
    new.IC <- fit[[criterium]]
    if(new.IC < IC){
      IC <- new.IC
      loglikelihood <- fit$loglikelihood
      r <- fit$r
      beta <- fit$beta
      alpha <- fit$alpha
      num.mixture <- length(alpha)
      if(print){
        cat("Number of mixtures = ", num.mixture, ",", "\n", "r = ", "\n")
        print(r)
        cat("alpha = ", alpha, "\n")
        cat("beta = ", beta, ",", criterium, " = ", IC, "\n", "\n")
      }
    } else{improve <- F}
  }
  list(Num.mixtures = num.mixture, alpha = alpha, r = r, beta = beta,
       loglikelihood = loglikelihood,
       AIC=-2*loglikelihood+2*((dim(data)[2]+1)*length(alpha)+1),
       BIC=-2*loglikelihood+((dim(data)[2]+1)*length(alpha)+1)*log(n))
}

############################################################
## Find r and alpha of a specified marginal distribution ##
############################################################

par.marginal.OR <- function(alpha, r, beta, c, dim){ # dim tells of
which dimension the marginal dist is
  r.marginal <- r[,dim]
  alpha.marginal <- tapply(alpha,factor(r.marginal),sum)
  r.marginal <- sort(unique(r.marginal))
```

```
  beta.marginal <- beta*c[dim]
  list(alpha.marginal = alpha.marginal, r.marginal= r.marginal,
  beta.marginal = beta.marginal)
}

## Distribution function of Mixture of Negative Binomials

NB.density <- function(x, beta, r, alpha){
  p <- 1/(1+beta)
  x.densities <- outer(x, r, dnbinom, prob=p)
  x.components <- sweep(x.densities, 2, alpha, FUN="*")
  NB.density <- rowSums(x.components)
  return(NB.density)
}

## CDF of Negative Binomial Mixtures

NB.cdf <- function(x, beta, r, alpha){
  NB.cdf <- numeric(length(x))
  for (i in 1:length(x)){
    for (j in 0:x[i]){
      NB.cdf[i] = NB.cdf[i] + NB.density(j,beta,r,alpha)
    }
  }
  return(NB.cdf)
}

#########################################################
## Barplot, P-P, Q-Q for empirical freq vs fitted freq ##
#########################################################

NB.barplot <- function(data, fitteddis, br, mainlabel){
  sim_fitted <- rnbinom(100000,size=sample(fitteddis$r.marginal,100000,
                                           replace=T,
                                           prob=fitteddis$alpha.marginal),
                        prob=1/(1+fitteddis$beta.marginal))
  freq_obs <- table(cut(data,br=br))
  freq_fit <- table(cut(sim_fitted,br=br))
  freq_fit <- freq_fit/100000*length(data)
  barname <- br[seq(2,length(br))]
  barplot(rbind(freq_obs,freq_fit),beside=T,names.arg=barname,
          col=c("red","blue"),main=mainlabel)
}

## P-P plot for empirical data VS fitted NB mixtures

NB.pp_plot <- function(data, beta, r, alpha, mainlabel, xlabel,
ylabel){
  y <- NB.cdf(data,beta,r,alpha)
  plot(ppoints(length(data)),sort(y),main=mainlabel,xlab=xlabel,
       ylab=ylabel,type="l",lwd=2)
  abline(0,1,col="red",lwd=1.5)
}

## Q-Q plot for empirical data VS fitted NB mixtures

NB.qq_plot <- function(data, beta, r, alpha, mainlabel, xlabel,
ylabel){
  if (length(alpha)>1){
    fittednum <- rnbinom(10000000,size=sample(r,length(data),replace=T,alpha),
                         prob=1/(1+beta))
  }
  else fittednum <- rnbinom(10000000,size=r,prob=1/(1+beta))
  empirinum <- data
  qqplot(empirinum,fittednum,main=mainlabel,xlab=xlabel,ylab=ylabel)
  abline(0,1,col="red")
}

#############################################
## Functions related to moments calculation ##
#############################################
```

```
## Calculate the factorial moments of Negative Binomial distribution

NB.factorial.moment <- function(k, r, beta){
  moment <- beta^k
  for (i in 1:k){
    moment <- moment*(r+i-1)
  }
  moment
}

## Calculate the raw moments of Negative Binomial distribution

NB.raw.moments <- function(k, r, beta){ # calculate the kth raw
moments
  facmom1 <- NB.factorial.moment(1,r,beta)
  facmom2 <- NB.factorial.moment(2,r,beta)
  facmom3 <- NB.factorial.moment(3,r,beta)
  facmom4 <- NB.factorial.moment(4,r,beta)
  facmom5 <- NB.factorial.moment(5,r,beta)
  if (k == 1) moment <- facmom1
  if (k == 2) moment <- facmom2 + facmom1
  if (k == 3) moment <- facmom3 + 3*facmom2 + facmom1
  if (k == 4) moment <- facmom4 + 6*facmom3 + 7*facmom2 + facmom1
  if (k == 5) moment <- facmom5 + 10*facmom4 + 25*facmom3 + 15*facmom2 + facmom1
  moment
}

## Calculate the raw moments of NB mixtures

multiNB.moments <- function(k, r, alpha, beta){
  moment <- sum(alpha*NB.raw.moments(k,r,beta))
  moment
}

##############################################################################
## Generate random pairs from a multivariate NB mixtures with different betas ##
##############################################################################

rmultiNB.mix.OR <- function(n, r, beta, alpha, c, index1, index2){
  sim <- matrix(NA,n,2)
  M <- length(alpha)
  for (i in 1:n){
    index <- sample(M,1,replace=TRUE,alpha)
    size <- r[index,]
    sim[i,1] <- rnbinom(1,size[index1],1/(1+c[index1]*beta))
    sim[i,2] <- rnbinom(1,size[index2],1/(1+c[index2]*beta))
  }
  sim
}

randomFreq.OR <- function(n, r, beta, alpha, c){
  sim <- matrix(NA,n,8)
  M <- length(alpha)
  for (i in 1:n){
    index <- sample(M,1,replace=TRUE,alpha)
    size <- r[index,]
    for (j in 1:8){
      sim[i,j] <- rnbinom(1,size[j],1/(1+c[j]*beta))
    }
  }
  sim
}

## Generate random pairs from a multivariate NB mixtures

rnbinom.mix <- function(n, r, beta, alpha, index1, index2){
  M <- length(alpha)
  sim <- matrix(NA,n,2)
  cumalpha <- cumsum(alpha)
  cum <- numeric(M+1)
  cum[seq(2,M+1)]=cumalpha
  prob <- runif(n)
```

```
  for (i in 1:n){
    for (j in 1:M){
      if ((prob[i]>=cum[j])&&(prob[i]<cum[j+1])) size <- r[j,]
    }
    sim[i,1] <- rnbinom(1,size[index1],1/(1+beta))
    sim[i,2] <- rnbinom(1,size[index2],1/(1+beta))
  }
  sim
}

################################################################
## Calculate Kendall's tau of fitted distribution (OR version) ##
################################################################

multiNB.kendall.OR <- function(alpha, r, beta, c, index1, index2){
  num <- 1000
  tau <- numeric(num)
  for (sim in 1:num){
    datasim <- rmultiNB.mix.OR(1000,r,beta,alpha,c,index1,index2)
    tau[sim] <- cor(datasim[,1],datasim[,2],method="kendall")
  }
  mean(tau)
}

#############################################################################
## Calculate correlation coefficients of fitted distribution (OR.version) ##
#############################################################################

multiNB.corr.OR <- function(alpha, r, beta, c, index1, index2){
  num <- 1000
  corr <- numeric(num)
  for (sim in 1:num){
    datasim <- rmultiNB.mix.OR(1000,r,beta,alpha,c,index1,index2)
    corr[sim] <- cor(datasim[,1],datasim[,2],method="pearson")
  }
  mean(corr)
}
```

DECLARATION OF INTEREST

Andrei Badescu and X. Sheldon Lin are supported by grants from the Natural Sciences and Engineering Research Council of Canada (NSERC).

ACKNOWLEDGEMENTS

The authors thank Professor Gordon Willmot for his comments on the mixed Poisson distribution, and Professor Pengfei Li for pointing out the references on identifiability of finite mixtures. We also thank the referees for their criticism and suggestions, which helped us improve the paper greatly.

REFERENCES

Artzner, P., Delbaen, F., Eber, J. M., and Heath, D. (1999). Coherent measures of risk. *Mathematical Finance* **9**, 203–228.

Aue, F., and Kalkbrener, M. (2006). LDA at work: Deutsche Bank's approach to quantifying operational risk. *The Journal of Operational Risk* **1**(4), 49–93.

Basel Committee on Banking Supervision, Risk Management Group (1998). Framework for the internal controls systems in banking organizations. Bank for International Settlements, September.

Basel Committee on Banking Supervision, Risk Management Group (1999). Enhancing corporate governance for banking organizations. Bank for International Settlements, September.

Basel Committee on Banking Supervision, Risk Management Group (2000). Other risks: operational risk discussion paper. Report BS/00/27, Bank for International Settlements, April.

Chaudhury, M. (2010). A review of the key issues in operational risk capital modeling. *The Journal of Operational Risk* **5**(3), 37–66.

Cope, E., and Antonini, G. (2008). Observed correlations and dependency among operational risk losses in the ORX consortium database. *The Journal of Operational Risk* **3**(4), 47–76.

Efron, B. (1979). Bootstrap methods: another look at the jackknife. *Annals of Statistics* **7**, 1–26.

Embrechts, P., Furrer, H., and Kaufmann, R. (2003). Quantifying regulatory capital for operational risk. *Derivatives Use, Trading and Regulation* **9**(3), 217–233.

Frachot, A., Roncalli, T., and Salomon, E. (2004). The correlation problem in operational risk. Working Paper, Groupe de Recherche Operationnelle, Crédit Lyonnais.

Ghitany, M. E., Karlis, D., Al-Mutairi, D. K., and Al-awadhi, F. A. (2012). An EM algorithm for multivariate mixed Poisson regression models and its application. *Applied Mathematical Sciences* **6**, 6843–6856.

Grandell, J. (1997). *Mixed Poisson Processes*. Chapman & Hall, London.

Hall, P. (1995). *The Bootstrap and Edgeworth Expansion*. Springer.

Joe, H. (1997). *Multivariate Models and Dependence Concepts*. Chapman & Hall, London.

Klugman, S., Panjer, H. H., and Willmot, G. E. (2012). *Loss Models: From Data to Decision*, 4th edn. Wiley.

Klugman, S., Panjer, H. H., and Willmot, G. E. (2013). *Loss Models: Further Topics*. Wiley.

Lee, S. C. K., and Lin, X. S. (2010). Modeling and evaluating insurance losses via mixtures of Erlang distributions. *North American Actuarial Journal* **14**(1), 107–130.

Lee, S. C. K., and Lin, X. S. (2012). Modeling dependent risks with multivariate Erlang mixtures. *ASTIN Bulletin* **42**(1), 153–180.

McLachlan, G., and Peel, D. (2000). *Finite Mixture Models*. Wiley Series in Probability and Statistics. Wiley.

Steutel, F. W., and Van Eenige, M. J. A. (1997). Note on the approximation of distributions on Z_+ by mixtures of negative binomial distributions. *Stochastic Models* **13**(2), 271–274.

Sundt, B., and Vernic, R. (2009). *Recursions for Convolutions and Compound Distributions with Insurance Applications*. Springer.

Teicher, H. (1963). Identifiability of finite mixtures. *Annals of Mathematical Statistics* **34**(4), 1147–1659.

Tijms, H. C. (2003). *A First Course in Stochastic Models*. Wiley.

Verbelen, R., Gong, L., Antonio, K., Badescu, A. L., and Lin, X. S. (2015). Fitting mixtures of Erlangs to censored and truncated data using the EM algorithm. *ASTIN Bulletin*, under revision.

Willmot, G. E., and Woo, J.-K. (2007). On the class of Erlang mixtures with theoretic applications. *North American Actuarial Journal* **11**(2), 99–115.