EFFICIENT ESTIMATION OF ERLANG MIXTURES USING iSCAD PENALTY WITH INSURANCE APPLICATION

BY

CUIHONG YIN AND X. SHELDON LIN

ABSTRACT

The Erlang mixture model has been widely used in modeling insurance losses due to its desirable distributional properties. In this paper, we consider the problem of efficient estimation of the Erlang mixture model. We present a new thresholding penalty function and a corresponding EM algorithm to estimate model parameters and to determine the order of the mixture. Using simulation studies and a real data application, we demonstrate the efficiency of the EM algorithm.

KEYWORDS

Erlang mixture, EM algorithm, iSCAD penalty, VaR, TVaR.

1. INTRODUCTION

In this paper, we consider the estimation of a (univariate) Erlang mixture model with density:

$$h(x;\boldsymbol{\alpha},\boldsymbol{\gamma},\theta)=\sum_{j=1}^{m}\alpha_j\frac{x^{\gamma_j-1}e^{-x/\theta}}{\theta^{\gamma_j}(\gamma_j-1)!},\ x>0, \tag{1.1}$$

where $\theta>0$ is a common scale parameter, $\boldsymbol{\gamma}=(\gamma_1,\gamma_2,\ldots,\gamma_m)$ with $\gamma_1<\gamma_2<\cdots<\gamma_m$ are the integer shape parameters, and $\boldsymbol{\alpha}=(\alpha_1,\ldots,\alpha_m)$ are the mixing weights.

The Erlang mixture has been widely used to model insurance losses. In insurance ruin theory, when insurance loss severity is modeled using an Erlang mixture, many quantities of interest such as the probability of infinite ruin, the Laplace transform of the time of ruin random variable may be expressed analytically. See Lin and Willmot (2000), Tsai and Willmot (2002), Landriault and Willmot (2009), Barges *et al.* (2013), and references therein. More recently the focus of using an Erlang mixture model in insurance is on fitting the model to real insurance loss data. Using an Erlang mixture model to fit insurance loss

Astin Bulletin 46(3), 779-799. doi: 10.1017/asb.2016.14

data is very appealing due to many of its desirable distributional properties. The distribution function and moments have an analytical form. As a result, risk measures such as value-at-risk (VaR) and tail VaR (TVaR) can be calculated easily. Any positive distribution can be approximated by an Erlang mixture to any given accuracy in the sense of weak convergence (Tijms approximation; See Tijms (2003)), especially in the situation that data exhibits multi-modal behavior. Further, there is a stable and fast expectation-maximization (EM) algorithm that can fit an Erlang mixture to data. Also, the Erlang mixture is weakly identifiable in the sense of Teicher (1963), i.e. two Erlang mixtures of form (1.1) are equal if and only if their scale parameter, weights and shape parameters are equal, respectively. The identifiability ensures that the EM algorithm converges to a unique distribution. See Lee and Lin (2010), Cossette *et al.* (2012), Cossette *et al.* (2013), Porth *et al.* (2014), Verbelen *et al.* (2015a) and references therein. In particular, Verbelen *et al.* (2015a) apply an Erlang mixture to censored and truncated insurance data and use the model to calculate the net premium of reinsurance contracts. More recently, a multivariate version of the Erlang mixture model (1.1) was proposed in Lee and Lin (2012). The multivariate model not only inherits most of the desirable distributional properties but also offers a non-copula approach for dependence modeling. Also see Hashorva and Ratovomirija (2015), Verbelen, Antonio and Claeskens (2015), Willmot and Woo (2015), and Badescu *et al.* (2015).

Although a fast EM algorithm is available for the estimation of the model parameters, i.e. the common scale parameter and mixing weights, the shape parameter of each of the Erlang components is not estimated by the EM algorithm. In order to include all possible Erlang distributions for component selection, one must start a large number of components in an Erlang mixture when running the EM algorithm. Over-fitting could be an issue in this situation. To maintain goodness of fit and avoid over-fitting at the same time, an ad hoc method for shape parameter selection and BIC are used. See Lee and Lin (2010) and Verbelen *et al.* (2015a). Several issues arise. First, the ad hoc method requires repeated runs of the EM algorithm, which can be computationally burdensome. Second, the chosen shape parameters are often suboptimal in terms of the order of the mixture (see the discussions in Section 4 of this paper). Also using BIC often results in a poor fit of a model to the sparse right tail of the data, a major shortcoming in insurance loss modeling and risk measure calculation. Finally, statistical properties of the corresponding estimators cannot be obtained under the ad hoc approach.

Using penalty functions such as the L_1 penalty in Lasso (Tibshirani (1996)) and the smoothly clipped absolute deviation (SCAD) penalty (Fan and Li (2001)) are popular in regression analysis, especially for sparse GLMs. The application of the latter to a likelihood can produce a sparse set of non-zero unbiased coefficients in a linear model and hence reduce model complexity. Due to some similarities between linear models and mixture models, this approach

https://doi.org/10.1017/asb.2016.14 Published online by Cambridge University Press

may be applicable to mixture modeling. Chen and Khalili (2008) applied the approach to the estimation of a finite Gaussian mixture. They developed a new penalty function based on the SCAD penalty and showed that the resulting estimators for the order of the mixture and its mixing weights are consistent. Motivated by Chen and Khalili (2008), in this paper we propose a new penalty function, termed as iSCAD, for the estimation of the Erlang mixture (1.1). The modification is due to the fact that the penalty in Chen and Khalili (2008) is not applicable in our situation. We show in the later sections that the aforementioned issues can be resolved satisfactorily. In a separate paper, Yin and Lin (2016), we prove that the corresponding estimators based on our approach are consistent.

This paper is organized as follows, In Section 2, some distributional properties including VaR and TVaR are presented and discussed. In Section 3, we introduce the iSCAD penalty and present an EM algorithm for the iSCAD penalized likelihood. Two simulation studies are conducted in Section 4 to illustrate how to apply the EM algorithm and to demonstrate the efficiency of the EM algorithm by comparing it with the existing estimation methods. In Section 5, we apply the EM algorithm to medical insurance claims in the Society of Actuaries' Large Claims Database. A brief conclusion is given in the final section.

2. Left-truncated distribution and risk measures

Insurance loss/claim data are mostly left truncated with known truncation points (in the form of a deductible or retention limit). See the data sets in Beirlant *et al.* (2006) and Verbelen *et al.* (2015a). Hence model fitting is essentially fitting a model to left-truncated data and to estimate the model parameters accordingly. In this section, we provide the analytic expressions of the left truncated distribution of (1.1) and the two risk measures, the VaR and TVaR, to be used in the later sections.

For the notational simplicity, we denote an Erlang density as

$$f(x;\gamma,\theta)=\frac{x^{\gamma-1}e^{-x/\theta}}{\theta^{\gamma}(\gamma-1)!}.$$

Its survival function is given by

$$\overline{F}(x;\gamma,\theta)=\sum_{k=0}^{\gamma-1}e^{-x/\theta}\frac{(x/\theta)^k}{k!}.$$

https://doi.org/10.1017/asb.2016.14 Published online by Cambridge University Press

Let l be a truncation point. Thus, the density of left-truncated Erlang mixture is

$$\begin{aligned} h(x; l, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \theta) &= \frac{h(x; \boldsymbol{\alpha}, \boldsymbol{\gamma}, \theta)}{\overline{H}(l; \boldsymbol{\alpha}, \boldsymbol{\gamma}, \theta)} = \sum_{j=1}^{m} \alpha_j \frac{f(x; \gamma_j, \theta)}{\overline{H}(l; \boldsymbol{\alpha}, \boldsymbol{\gamma}, \theta)} \\ &= \sum_{j=1}^{m} \alpha_j \frac{\overline{F}(l; \gamma_j, \theta)}{\overline{H}(l; \boldsymbol{\alpha}, \boldsymbol{\gamma}, \theta)} \frac{f(x; \gamma_j, \theta)}{\overline{F}(l; \gamma_j, \theta)} \\ &= \sum_{j=1}^{m} \pi_j f(x; l, \gamma_j, \theta), \quad x > l, \end{aligned} \tag{2.1}$$

where $\overline{H}(x; \boldsymbol{\alpha}, \boldsymbol{\gamma}, \theta)$ is the survival function of $h(x; \boldsymbol{\alpha}, \boldsymbol{\gamma}, \theta)$, $\pi_j = \alpha_j \frac{\overline{F}(l; \gamma_j, \theta)}{\overline{H}(l; \boldsymbol{\alpha}, \boldsymbol{\gamma}, \theta)}$, and $f(x; l, \gamma_j, \theta) = \frac{f(x; \gamma_j, \theta)}{\overline{F}(l; \gamma_j, \theta)}$. We remark that $\overline{H}(x; \boldsymbol{\alpha}, \boldsymbol{\gamma}, \theta)$ may be expressed in terms of Erlang densities explicitly. See Lee and Lin (2010).

Obviously, (2.1) is a mixture of left-truncated Erlangs with truncation point l and its survival function is given by

$$\overline{H}(x; l, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \theta) = \sum_{j=1}^{m} \pi_j \frac{\overline{F}(x; \gamma_j, \theta)}{\overline{F}(l; \gamma_j, \theta)}, \quad x \geq l. \tag{2.2}$$

VaR and TVaR are two commonly used risk measures that estimate how much an insurance/investment portfolio might lose in a time period. Given a security level p, VaR is a probability of ruin measure and is defined as the $100p$-th percentile of the loss distribution of the portfolio. TVaR is a cost of ruin measure and is defined as the expected loss given that the loss is greater than the corresponding VaR.

In order to calculate risk measures VaR and TVaR based on truncated data, we re-express the left-truncated survival function (2.2) in terms of Erlang densities:

$$\overline{H}(x; l, \boldsymbol{\alpha}, \boldsymbol{\gamma}, \theta) = \frac{\theta}{\overline{H}(l; \boldsymbol{\alpha}, \boldsymbol{\gamma}, \theta)} \sum_{j=1}^{\gamma_m} Q_j f(x; j, \theta), \quad x \geq l, \tag{2.3}$$

where $Q_j = \sum_{k=j}^{\gamma_m} \alpha_k^*$, $j = 1, \ldots, \gamma_m$, in which $\alpha_k^* = \alpha_j$ if $k = \gamma_j$ and $\alpha_k^* = 0$ otherwise. The derivation is almost identical to that in Lee and Lin (2010) and is omitted here.

Value at risk at security level p, VaR_p, can be calculated by solving the following equation for x:

$$\frac{\theta}{\overline{H}(l; \boldsymbol{\alpha}, \boldsymbol{\gamma}, \theta)} \sum_{j=1}^{\gamma_m} Q_j f(x; j, \theta) = 1 - p. \tag{2.4}$$

https://doi.org/10.1017/asb.2016.14 Published online by Cambridge University Press

Similarly, when the loss random variable X follows the Erlang mixture (1.1), the stop-loss premium or the net premium of the excess of loss (in the context of re-insurance) at retention level $R > l$ may be written as

$$\mathbb{E}((X-R)_+|X>l) = \frac{\theta^2}{\overline{H}(l;\boldsymbol{\alpha},\boldsymbol{\gamma},\theta)} \sum_{j=1}^{\gamma_m} Q_j^* f(R;j,\theta) \tag{2.5}$$

where $Q_j^* = \sum_{k=j}^{\gamma_m} Q_k$, $j = 1, \ldots, \gamma_m$.

Tail VaR at security level p, TVaR_p, is given by

$$\mathrm{TVaR}_p = \mathbb{E}(X|X>\mathrm{VaR}_p) = \frac{\theta^2}{\overline{H}(\mathrm{VaR}_p;\boldsymbol{\alpha},\boldsymbol{\gamma},\theta)} \sum_{j=1}^{\gamma_m} Q_j^* f(\mathrm{VaR}_p;j,\theta) + \mathrm{VaR}_p. \tag{2.6}$$

3. iSCAD AND ASSOCIATED EM ALGORITHM

There are a number of similarities between a generalized linear model and a mixture model due to their linear structure. An effective way to increase sparsity and accuracy of a GLM in variable selection is to use a thresholding penalty. See Donoho and Johnstone (1994), Fan and Li (2001) and references therein. This idea is applicable to mixture modeling to select component distributions in a mixture model. One may use a thresholding penalty to penalize a likelihood of the mixture model in such a way that any weight estimate below a given threshold is reset to zero. As a result, the order of a mixture model is minimized. The SCAD penalty proposed in Fan and Li (2001) stands out from other forms of penalty mainly due to its three desirable properties in estimation: unbiasedness, continuity and sparsity. Borrowing the idea from the SCAD penalty, Chen and Khalili (2008) introduce a new penalty function, called MSCAD, to determine the order and mixing distribution of Gaussian mixtures as the MSCAD allows the EM algorithm to cluster and merge component distributions. However, the MSCAD is not applicable to Erlang mixtures as the shape parameter of an Erlang is an integer which makes merging two Erlang distributions impossible.

In this section, we propose a new thresholding penalty function termed as iSCAD, which is a function of individual weights of a to-be-estimated Erlang mixture. The iSCAD penalty function for each weight π_j is defined as

$$\begin{aligned} P_{\varepsilon,\lambda}(\pi_j) = {} & \lambda\{\ln\frac{a\lambda+\varepsilon}{\varepsilon} + \frac{a^2\lambda^2}{2} - \frac{a\lambda}{a\lambda+\varepsilon}\} I(\pi_j > a\lambda) \\ & + \lambda\{\ln\frac{\pi_j+\varepsilon}{\varepsilon} - \frac{\pi_j^2}{2} + (a\lambda - \frac{1}{a\lambda+\varepsilon})\pi_j\} I(\pi_j \le a\lambda), \end{aligned} \tag{3.1}$$

where $I(\cdot)$ is the indicator function; λ serves as a tuning/thresholding parameter such that when an estimated weight is below λ, it is set to be 0. The plot of an

https://doi.org/10.1017/asb.2016.14 Published online by Cambridge University Press

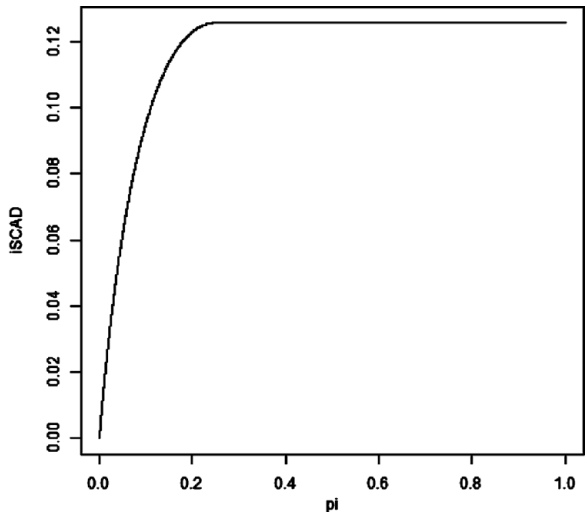

FIGURE 1: Plot of iSCAD penalty with $\lambda = 0.2$ and $\varepsilon = 0.09$.

iSCAD penalty is given in Figure 1. To apply the iSCAD penalty to a likelihood, we assume that λ is a function of n with conditions: $\lambda \to 0, n\lambda \to \infty$, as $n \to \infty$. The conditions ensure that the order estimator is consistent (Yin and Lin (2016)). In general, we let $\lambda = \frac{C(m)}{\sqrt{n}}$, where $C(m)$ is a decreasing function of the order m; $a = \frac{m}{m-\lambda} > 1$ to ensure that the estimator $\hat{\pi}_j$ of π_j is continuous; parameter $\varepsilon > 0, \varepsilon \to 0$, as $n \to \infty$, to ensure that the range of π_j includes 0. The choice of λ and ε will be discussed later in this section and Section 4.

It is clear that the first derivative of the iSCAD penalty function with respect to π_j is

$$\begin{aligned} P'_{\varepsilon,\lambda}(\pi_j) &= \lambda\{\frac{1}{\pi_j + \varepsilon} - \pi_j + (a\lambda - \frac{1}{a\lambda + \varepsilon})\} I(\pi_j \le a\lambda) \\ &= \lambda(a\lambda - \pi_j)\{1 + \frac{1}{(\pi_j + \varepsilon)(a\lambda + \varepsilon)}\} I(\pi_j \le a\lambda). \end{aligned} \tag{3.2}$$

Hence, $P_{\varepsilon,\lambda}(\pi_j)$ and $P'_{\varepsilon,\lambda}(\pi_j)$ are continuous in π_j. Moreover, $P_{\varepsilon,\lambda}(\pi_j)$ is increasing in π_j with $P_{\varepsilon,\lambda}(0) = 0$. $P'_{\varepsilon,\lambda}(\pi_j)$ is decreasing in π_j with $P'_{\varepsilon,\lambda}(1) = 0$. The last property also implies that iSCAD is concave.

The tuning parameter will ensure the sparsity of the mixture as it serves as a lower bound of the mixing weights. This property is crucial to avoid over-fitting and to maintain fitting accuracy at the same time. Moreover, the structure of the iSCAD penalty and its derivative will result in the unbiasedness and continuity in estimation of the mixing distribution as will be seen later in this section. The consistency of the estimator of the order of a mixture is always a challenging problem. We provide a lengthy proof to show that the estimator of the order

https://doi.org/10.1017/asb.2016.14 Published online by Cambridge University Press

of the Erlang mixture under the iSCAD penalty is consistent in Yin and Lin (2016).

In the remainder of this section, we present an EM algorithm to maximize the penalized likelihood of an Erlang mixture. With the iSCAD penalty, the algorithm is able to estimate both the order of the mixture and the mixing distribution, and to induce sparsity of the model.

Suppose that $\boldsymbol{X} = (X_1, \ldots, X_n)$ is a random sample of size n from a sparse Erlang mixture with density $h(x; l, \boldsymbol{\gamma}, \boldsymbol{\phi}) = \sum_{j=1}^{m} \pi_j f(x; l, \gamma_j, \theta)$. Let $\boldsymbol{x} = (x_1, \ldots, x_n)$ be its left-truncated observations. The parameters in the model that is to be estimated are $\boldsymbol{\phi} = (\pi_1, \ldots, \pi_m, \theta)$.

We begin with a "large" model that includes all possible Erlang components:

$$h_0(x; l, \boldsymbol{\gamma}_0, \boldsymbol{\phi}) = \sum_{j=1}^{M} \pi_j f(x; l, \gamma_j^0, \theta).$$

In general, we initially set $\gamma_j^0 = j$ for all j and M is chosen such that $\theta^{(0)} M$ is greater than or equal to the largest sample point, where $\theta^{(0)}$ is the initial scale parameter in the EM algorithm.

The log-likelihood function is

$$\ell_n(\boldsymbol{\phi}) = \sum_{i=1}^{n} \ln(h_0(x_i; l, \boldsymbol{\gamma}_0, \boldsymbol{\phi})) = \sum_{i=1}^{n} \ln(\sum_{j=1}^{M} \pi_j f(x_i; l, \gamma_j^0, \theta)). \tag{3.3}$$

The log-likelihood function with iSCAD penalty is then given by

$$\ell_{n,P}(\boldsymbol{\phi}) = \ell_n(\boldsymbol{\phi}) - n \sum_{j=1}^{M} P_{\varepsilon,\lambda}(\pi_j). \tag{3.4}$$

To apply an EM algorithm, a standard approach is to introduce the following unobservable component-indicator random vectors: $\boldsymbol{Z} = (\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_n)$, where $\boldsymbol{Z}_i = (Z_{ij} | i = 1, \ldots, n, j = 1, \ldots, M)$, with

$$Z_{ij} = \begin{cases} 1 & \text{if observation } x_i \text{ comes from } j\text{th component density } f(x_i; \gamma_j^0, \theta) \\ 0 & \text{otherwise.} \end{cases} \tag{3.5}$$

The log-likelihood function of the complete sample $(\boldsymbol{x}, \boldsymbol{Z})$ is then

$$\ell_n(\boldsymbol{\phi}; \boldsymbol{x}, \boldsymbol{Z}) = \sum_{i=1}^{n} \sum_{j=1}^{M} Z_{ij} \ln(\pi_j f(x_i; l, \gamma_j^0, \theta)). \tag{3.6}$$

https://doi.org/10.1017/asb.2016.14 Published online by Cambridge University Press

The log-likelihood function of the complete data $(\boldsymbol{x}, \boldsymbol{Z})$ with iSCAD penalty is

$$\ell_{n,P}(\boldsymbol{\phi}; \boldsymbol{x}, \boldsymbol{Z}) = \ell_n(\boldsymbol{\phi}; \boldsymbol{x}, \boldsymbol{Z}) - n\sum_{j=1}^{M} P_{\varepsilon,\lambda}(\pi_j). \tag{3.7}$$

Suppose that we have completed the kth iteration in the EM algorithm with estimates $\boldsymbol{\phi}^{(k)} = (\pi_1^{(k)}, \ldots, \pi_M^{(k)}, \theta^{(k)})$. Then the E-Step and the M-Step can be obtained as follows.

E-step:

$$\begin{aligned} Q(\boldsymbol{\phi} \mid \boldsymbol{\phi}^{(k)}) &= \mathbb{E}(\ell_{n,P}(\boldsymbol{\phi}; \boldsymbol{x}, \boldsymbol{Z})|\boldsymbol{x}, \boldsymbol{\phi}^{(k)}) \\ &= \sum_{i=1}^{n}\sum_{j=1}^{M}[\ln(\pi_j) - \frac{x_i}{\theta} - \gamma_j^0 \ln(\theta) \\ &\quad - \overline{F}(l; \gamma_j^0, \theta)]q(\gamma_j^0 \mid x_i, \boldsymbol{\phi}^{(k)}) - n\sum_{j=1}^{M} P_{\varepsilon,\lambda}(\pi_j), \end{aligned} \tag{3.8}$$

where $q(\gamma_j^0 \mid x_i, \boldsymbol{\phi}^{(k)})$ is the probability of the observation x_i coming from the jth component:

$$q(\gamma_j^0 \mid x_i, \boldsymbol{\phi}^{(k)}) = \frac{\pi_j^{(k)} f(x_i; l, \gamma_j^0, \theta^{(k)})}{\sum_{j=1}^{M} \pi_j^{(k)} f(x_i; l, \gamma_j^0, \theta^{(k)})}. \tag{3.9}$$

M-step: The MLE of π_j, $j = 1, \ldots, M$ and θ can be obtained as

$$\hat{\boldsymbol{\phi}}^{(k+1)} = \arg\max_{\boldsymbol{\phi}}\{Q(\boldsymbol{\phi} \mid \boldsymbol{\phi}^{(k)})\}.$$

Denote $\bar{q}_j^{(k)} \triangleq \frac{\sum_{i=1}^{n} q(\gamma_j^0|x_i,\boldsymbol{\phi}^{(k)})}{n}$. The Lagrange method leads to an explicit expression of the $(k+1)$st estimate of π_j

$$\hat{\pi}_j^{(k+1)} = \bar{q}_j^{(k)} I(\bar{q}_j^{(k)} > a\lambda) + \frac{M}{\lambda}(\bar{q}_j^{(k)} - \lambda)_+ I(\bar{q}_j^{(k)} \le a\lambda). \tag{3.10}$$

It follows from the first term of formula (3.10) that the estimator for a mixing weight is unbiased when it is bounded away from zero and from the second term that the sparsity is achieved as λ serves as a lower bound for all the mixing weights. It is straightforward to verify that with $a = \frac{M}{M-\lambda}$, the estimates are continuous in $\boldsymbol{x}$.

https://doi.org/10.1017/asb.2016.14 Published online by Cambridge University Press

The $(k+1)$st estimate of the scale parameter θ is given by

$$\hat{\theta}^{(k+1)} = \frac{\frac{1}{n}\sum_{i=1}^{n} x_i - t^{(k)}}{\sum_{j=1}^{M} \gamma_j^0 \bar{q}_j^{(k)}}, \tag{3.11}$$

where

$$t^{(k)} = \sum_{j=1}^{M} \bar{q}_j^{(k)} \frac{l^{\gamma_j^0} e^{-l/\theta}}{\theta^{\gamma_j^0 - 1}(\gamma_j^0 - 1)!\, \overline{F}(l; \gamma_j^0, \theta)}\Bigg|_{\theta=\theta^{(k)}}. \tag{3.12}$$

The derivation is almost identical to that in Verbelen *et al.* (2015a).

The iteration of the EM-steps continues until $|Q(\phi^{(k)}) - Q(\phi^{(k-1)})|$ is below a pre-specified error bound. Let $\hat{\theta}$ and $\hat{\pi}_j$'s be the estimated values in the final EM step and the estimate of the order of mixture

$$\hat{m} = \#\{\hat{\pi}_j | \hat{\pi}_j \neq 0,\, j = 1, \ldots, M\}.$$

For notational cleanness, we rename the shape parameters $\hat{\gamma} = \{\gamma_j^0 | \hat{\pi}_j \neq 0,\, j = 1, \ldots, M\}$ as $\hat{\gamma} = (\hat{\gamma}_1, \ldots, \hat{\gamma}_{\hat{m}})$ in the increasing order and the corresponding mixing weights $\hat{\boldsymbol{\pi}} = (\hat{\pi}_1, \ldots, \hat{\pi}_{\hat{m}})$. Finally, the estimates of the original weight parameters $\hat{\boldsymbol{\alpha}} = (\hat{\alpha}_1, \ldots, \hat{\alpha}_{\hat{m}})$ are obtained as

$$\hat{\alpha}_j = c\frac{\hat{\pi}_j}{\overline{F}(l; \hat{\gamma}_j, \hat{\theta})} \tag{3.13}$$

where c is a normalizing constant such that $\sum_{j=1}^{\hat{m}} \hat{\alpha}_j = 1$.

We remark that in order to achieve the required goodness of fit, the above EM algorithm needs to be applied iteratively a few times with increasing values of the tuning parameter. See Section 4 for more details.

4. Simulation studies

In this section, we examine the goodness of fit and efficiency of the EM algorithm in Section 3 through simulation studies. In particular, we pay close attention to how the EM algorithm determines the order of an Erlang mixture when dealing with various simulated data.

In the statistical literature, the order of a mixture is often determined using a BIC type penalty. See Yakowitz and Spragins (1968), Kass and Wasserman (1995), Keribin (2000) and references therein. In Lee and Lin (2010) and Verbelen *et al.* (2015a), BIC is also used to determine the order of an Erlang mixture. In the former, an EM algorithm with BIC penalty is applied without adjusting the shape parameters of the Erlang mixture, which is similar to the approach in this paper except the use of different penalties (BIC vs. iSCAD). The latter applies the EM algorithm repeatedly and every time the shape parameters are

https://doi.org/10.1017/asb.2016.14 Published online by Cambridge University Press

TABLE 1

COMPARISON OF PARAMETER ESTIMATES γ, α, θ, BIC AND RUNTIME.

Method	m	γ	α	θ	BIC	Total Runtime
Model	7	(8,20,40,65, 95,130,170)	(0.143, 0.143, 0.143, 0.143, 0.143, 0.143, 0.143)	1	–	–
I	7	(8,20,40,65, 95,133,172)	(0.131, 0.146, 0.138, 0.151, 0.152, 0.135, 0.147)	0.98517	25,745.39	5.04 mins
II	7	(9,21,42,68, 100,137,179)	(0.136, 0.141, 0.138, 0.152, 0.149, 0.133, 0.151)	0.94364	25,744.23	90.6 mins
III	7	(8,21,40,65, 95,131,172)	(0.144,0.137, 0.136, 0.151, 0.148, 0.134, 0.151)	0.98582	25,721.77	3.92 mins

adjusted in a systematical way. In following examples, we compare numerical results from these three approaches. We refer to the approach in Lee and Lin (2010) as Method I, that in Verbelen *et al.* (2015a) as Method II and ours as Method III.

Example 4.1. In this example, data are generated from an Erlang mixture with equally weighted seven components. The shape parameters are $\gamma = (8, 20, 40, 65, 95, 130, 170)$ and the scale parameter $\theta = 1$. The weights $\alpha_j = 1/7 = 0.143$, $j = 1, \ldots, 7$. 2,500 data points ($n = 2{,}500$) are generated assuming a left truncation of $l = 1$. The purpose of this example is to examine the quality of estimation given that there is a large number of modes, as well as the computing time. We start with a "large" over-fitting model of 207 components (i.e. $M = 207$ and $\gamma_j^0 = j$, $j = 1, 2, \ldots, M$) as described in Section 3. The Tijms approximation is used for the initial estimates for all three methods.

In this example, the tuning parameter λ takes the form of $\lambda = \frac{c(1+\sqrt{m})}{m^{3/2}\sqrt{n}}$ or $C(m) = c(\frac{1}{m} + \frac{1}{m^{3/2}})$, with $c = 30$ and $\varepsilon = \lambda^{3/2}$. The choice of the value of ε in connection with λ is to ensure the consistency of the estimators, which is discussed in Yin and Lin (2016). We then apply the EM algorithm five times consecutively with the estimates from the previous application being the initial estimates of the following application: $M = 207 \xrightarrow{Tijm's\ Approx.} (\hat{m} = 196, \lambda = 0.00328) \xrightarrow{1^{st}\ Appl.} (\hat{m}_2 = 45, \lambda_2 = 0.01532) \xrightarrow{2^{nd}\ Appl.} (\hat{m}_3 = 22, \lambda_3 = 0.03309) \xrightarrow{3^{rd}\ Appl.} (\hat{m}_4 = 15, \lambda_4 = 0.05033) \xrightarrow{4^{th}\ Appl.} (\hat{m}_5 = 11, \lambda_5 = 0.07099) \xrightarrow{5^{th}\ Appl.} \hat{m} = 7$.

Table 1 presents the estimation results from the three methods.

As shown in Table 1, all the three methods work well in estimating the weights and the order of the mixture. Because adjusting the shape parameters requires a large number of applications of the EM algorithm, Method II has a significantly longer runtime. To confirm the goodness of fit, Figure 2 shows the three fitted densities and the true density.

https://doi.org/10.1017/asb.2016.14 Published online by Cambridge University Press

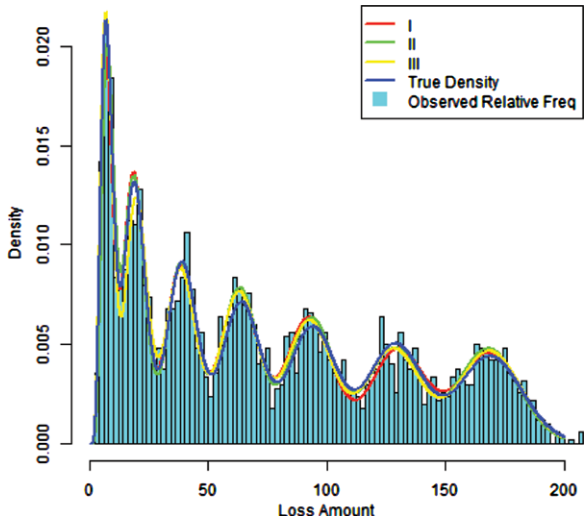

FIGURE 2: Graphical comparison of the densities of the fitted mixtures by Method I, II and III, the true density, and the histogram of the sample. (Color online)

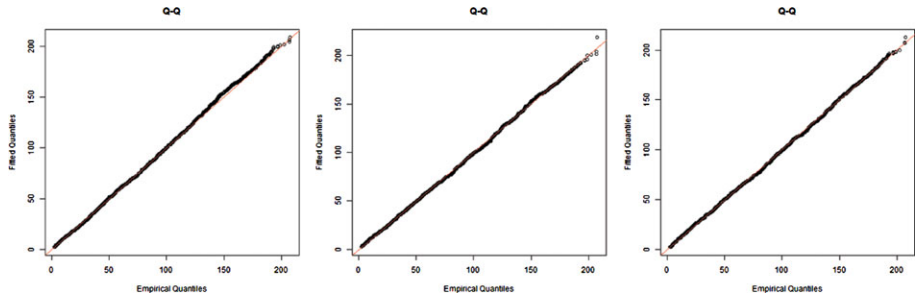

FIGURE 3: Q–Q plots of the fitted model by Methods I, II and III. (Color online)

The Q–Q plots in Figure 3 show that Method I gives slightly poor body (as expected), the fit to the tail by Method II is not as good as that by Method III. In other words, not only is Method III more efficient in terms of runtime but also it provides better fit to data both in the body and tail. This advantage will become clearer in the real data application in the next section.

Although it is proven in Yin and Lin (2016) that the order estimator using the iSCAD penalty is consistent, we also want to test how efficient our EM algorithm is in terms of order selection. We adapt a procedure used in Chen and Khalili (2008) in which a large number of replications are created and each of them is used to estimate the parameter values and the order by the three methods. 100 replications are generated and each of them again contains $n = 2{,}500$ simulated left truncated data points from the seven-component Erlang mixture. The results are presented in Table 2.

https://doi.org/10.1017/asb.2016.14 Published online by Cambridge University Press

TABLE 2

COMPARISON OF THE ACCURACY OF THE ORDER ESTIMATES.

Method	m	Frequency of m
Model	7	1
I	(7,8,9,10,11,12,15)	(0.64,0.16,0.07,0.06,0.04,0.02,0.01)
II	(7,8)	(0.99,0.01)
III	7	1

TABLE 3

COMPARISON OF PARAMETER ESTIMATES γ, α, θ, BIC AND RUNTIME.

Method	m	γ	α	θ	BIC	Total Runtime
Model	2	(5,10)	(0.5,0.5)	(1,2)	–	–
I	4	(5, 17, 23, 27)	(0.514, 0.310, 0.116, 0.060)	1.02517	32,932.50	5.57195 mins
II	4	(5, 14, 20, 28)	(0.501, 0.118, 0.309, 0.072)	1.00553	32,924.21	31.93998 mins
III	3	(5, 17, 25)	(0.514, 0.331, 0.154)	1.03002	32,919.19	0.8045 mins

The estimation results from the three methods are given in Table 2. It is clear from Table 2 that Methods I and II tend to over-estimate the order of a mixture. The over-estimation by method I is significant with a high frequency of 36%. Method II performs much better, which shows that it is necessary to adjust shape parameters if one is to use BIC. However, the iSCAD penalty approach is superior to both and is able to select the correct order every time.

Example 4.2. In this Example, we generate data from a generalized Erlang mixture that has different scale parameters. In particular, the Erlang mixture under consideration in this example has two components with shape parameters $(\gamma_1, \gamma_2) = (5, 10)$ and scale parameters $(\theta_1, \theta_2) = (1, 2)$, respectively. The components are equally weighted. Clearly this mixture is not from the class of Erlang mixtures considered in this paper and hence is not recoverable. The main purpose of this study is to see how well each of the three methods performs in terms of model selection. Again we assume a left truncated point of $l = 1$ but 5, 000 data points are generated from the mixture.

As in Example 4.1, we choose $\lambda = \frac{c(1+\sqrt{m})}{m^{3/2}\sqrt{n}}$ with $c = 20$ and $\varepsilon = \lambda^{3/2}$. Again, the EM algorithm is applied consecutively until the order reaches the lowest possible value of 3. This time, it takes three times with the following values: $M = 50 \xrightarrow{Tijm's\ Approx.} (\hat{m} = 25, \lambda = 0.01358) \xrightarrow{1^{st}\ Appl.} (\hat{m}_2 = 11, \lambda_2 = 0.03347) \xrightarrow{2^{nd}\ Appl.} (\hat{m}_3 = 6, \lambda_3 = 0.06639) \xrightarrow{3^{rd}\ Appl.} (\hat{m}_3 = 5, \lambda_3 = 0.08187) \xrightarrow{4^{th}\ Appl.} \hat{m} = 3$.

Table 3 shows the estimation results.

https://doi.org/10.1017/asb.2016.14 Published online by Cambridge University Press

TABLE 4

COMPARISON OF THE ACCURACY OF THE ORDER ESTIMATES.

Method	m	Frequency of m
Model	2	1
I	(3, 4, 5, 6, 7, 8, 9, 10, 11)	(0.18, 0.23, 0.11, 0.14, 0.17, 0.03, 0.09, 0.04, 0.01)
II	(3, 4, 5, 7)	(0.39, 0.51, 0.09, 0.01)
III	(3, 4, 5)	(0.53, 0.35, 0.12)

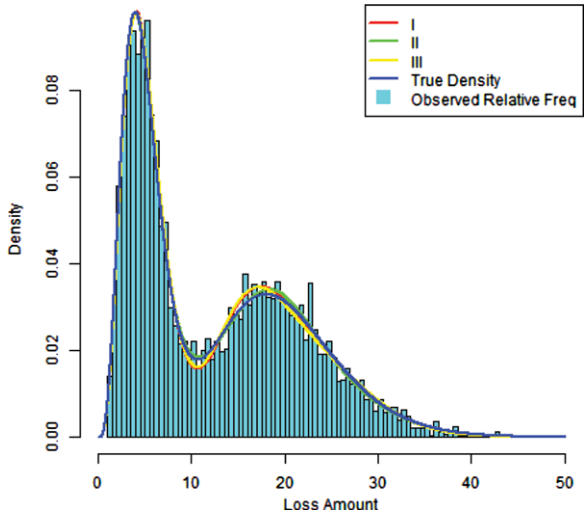

FIGURE 4: Graphical comparison of the densities of the fitted mixtures by Method I, II and III, the true density, and the histogram of the sample. (Color online)

Table 3 shows that in this situation both Methods I and II are outperformed by Method III in terms of model selection and runtime, as the latter requires only three components to fit the data and the runtime is significantly shorter. The graphical comparison of the density of the fitted models and the histogram of the sample in Figure 4 reconfirms the goodness of fit by Method III.

Also similar to Example 4.1, we now perform the procedure of running 100 replications to see how the algorithm works in terms of model selection. The results are provided in Table 4.

Table 4 shows that Method III tends to have smaller order than that by the other methods.

5. APPLICATION TO A GROUP MEDICAL INSURANCE CLAIMS DATA

In this section, we apply the Erlang mixture and the EM algorithm with iSCAD to the SOA Group Medical Insurance Large Claims Database that can be found

https://doi.org/10.1017/asb.2016.14 Published online by Cambridge University Press

TABLE 5

COMPARISON OF BIC AND RUNTIME BY METHODS I, II AND III.

case	m	BIC	time
I	19	1712244	2.440208 hours
II	19	1711722	13.4219 hours
III	19	1711570	0.143 hours

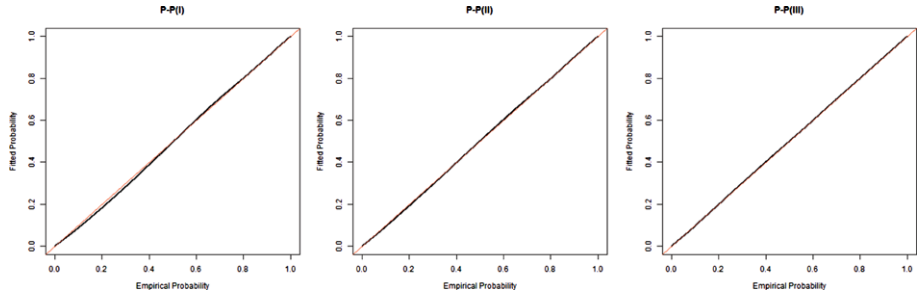

FIGURE 5: P–P plots of the fitted models by Methods I, II and III. (Color online)

in Beirlant *et al.* (2006). The data set contains 75,789 claims in 1991 that are left truncated at \$25,000. Cebrian *et al.* (2003) use the generalized Pareto distribution (GPD) to fit the data and compare it to the commonly used gamma, log-normal, and log-gamma distributions. They conclude that the GPD is superior to those traditional parametric models (see Figure 8 of Cebrian *et al.* (2003)). They also show that the GPD fits the data very well with high threshold but not as well when fitting the entire data set (see Figure 7 of Cebrian *et al.* (2003)).

In this section, we fit the Erlang mixture to the entire data set and show that the Erlang mixture can simultaneously fit both the body and tail of the data well. In this example, the tuning parameter λ takes the form of $\lambda = \frac{c(1+m^3)}{m^4\sqrt{n}}$ or $C(m) = c(\frac{1}{m} + \frac{1}{m^4})$, with $c = 0.0845$ and $\varepsilon = \lambda^{3/2}$, and We apply EM algorithm two times with the following values: $M = 38 \xrightarrow{Tijm's\ Approx.} (\hat{m}_1 = 23, \lambda = 1.334632e-05) \xrightarrow{1^{st}\ Appl.} (\hat{m}_2 = 21, \lambda_2 = 1.461777e-05) \xrightarrow{2^{nd}\ Appl.} \hat{m} = 19$. For comparison, we also fit the Erlang mixture to data using Methods I and II.

Table 5 shows Method III is obviously superior to Methods I and II in terms of runtime and BIC. For completeness, we provides the estimated parameter values by these three methods in Tables A1–A3 in the appendix.

The three panels in each of the following figures show the P–P plots and Q–Q plots of the fitted Erlang mixtures.

The P–P plots by all three methods in Figures 5 are satisfactory, implying that the fitted Erlang mixtures fit the body of the data well. However, the Q–Q plots in Figure 6 tell a different story. Although Methods I and II produce

https://doi.org/10.1017/asb.2016.14 Published online by Cambridge University Press

TABLE 6

COMPARISON OF VARS OF THE FITTED MIXTURES TO THE NON-PARAMETRIC VAR.

Confidence Level	Non-Parametric	I	II	III
80.0%	69,332	69,314	69,761	69,420
85.0%	81,456	81,785	81,765	81,523
90.0%	101,846	102,423	101,775	101,942
95.0%	147,563	146,411	147,080	147,114
97.5%	205,397	206,069	205,467	206,102
98.5%	259,236	258,455	259,205	258,599
99.0%	305,970	307,031	306,947	305,963
99.5%	406,225	407,364	407,582	409,180
99.9%	721,119	727,059	727,538	730,098
99.95%	970,505	989,074	971,888	971,057
99.99%	1,701,388	1,267,732	1,770,397	1,775,483
99.995%	1,963,024	1,335,436	2,005,718	1,985,626
99.997%	2,089,817	1,378,500	2,195,287	2,096,864
99.999%	3,734,111	1,459,130	2,394,833	3,967,590

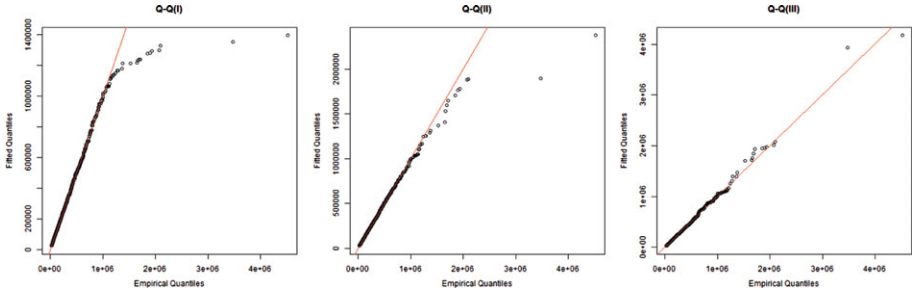

FIGURE 6: Q–Q plots of the fitted models by Methods I, II and III. (Color online)

reasonable Q–Q plots, Method III is significantly better in terms of right tail fitting.

We can easily calculate VaR and TVaR of the fitted Erlang mixtures using formulas (2.4) and (2.6) in Section 2. Non-parametric VaR and TVaR are calculated using the empirical distribution. More precisely, the non-parametric VaR at security level p is the solution of $F_n(\text{VaR}_p) = p$ for where $F_n(x) = \frac{\sum_{i=1}^{n} I(x_i \le x)}{n}$, and $\text{TVaR}_p = \frac{\sum_{i=1}^{n}(x_i \cdot I(x_i > VaR_p))}{\sum_{i=1}^{n} I(x_i > VaR_p)}$. Using non-parametric VaR and TVaR as benchmarks, we show in Tables 6 and 7 that Method III produces a much more accurate estimate for both VaR and TVaR, especially at very high security levels. In other words, the iSCAD penalty is more powerful than the BIC penalty in capturing the right tail heaviness of data.

https://doi.org/10.1017/asb.2016.14 Published online by Cambridge University Press

TABLE 7

COMPARISON OF TVARS OF THE FITTED MIXTURES TO THE NON-PARAMETRIC TVAR.

Confidence Level	Non-Parametric	I	II	III
80.0%	136,265	135,638	136,146	136,236
85.0%	156,692	155,820	156,411	156,645
90.0%	189,648	188,167	189,246	189,600
95.0%	258,456	255,501	257,789	258,351
97.5%	345,564	339,489	343,951	345,350
98.5%	422,794	412,847	420,273	422,383
99.0%	494,014	479,048	489,937	493,205
99.5%	637,748	608,219	629,990	636,246
99.9%	1,151,879	1,009,962	1,110,367	1,150,762
99.95%	1,458,602	1,164,513	1,376,645	1,457,699
99.99%	2,447,259	1,355,116	2,062,714	2,463,663
99.995%	3,043,534	1,411,431	2,255,474	3,041,870
99.997%	3,365,433	1,449,224	2,359,357	3,715,832
99.999%	4,518,420	1,520,749	2,499,270	4,051,506

6. CONCLUSION

In this paper, we present a new thresholding penalty function and a corresponding EM algorithm for estimation of Erlang mixtures. Using simulation studies and a real data application, we have demonstrated the efficiency of the EM algorithm in estimating the model parameters and in determining the order of the mixture. Moreover, in Yin and Lin (2016), we prove that the order estimator is consistent. In the future, we will explore the use of the proposed penalty for other non-Gaussian mixtures and application to loss data from property and casualty insurance.

ACKNOWLEDGEMENTS

This research was partly supported by a grant from the Natural Sciences and Engineering Research Council of Canada (NSERC). C. Yin thanks the Graduate School of the Xiamen University for its financial support during her PhD studies. Both authors would also like to thank Professor Pengfei Li of the University of Waterloo, Professor Rongtan Huang of the Xiamen University, and two anonymous referees for their valuable comments and suggestions on this paper.

REFERENCES

BADESCU, A.L., GONG, L., LIN, X.S. and TANG, D. (2015) Modeling correlated frequencies with application in operational risk management. *Journal of Operational Risk,* **10**(1), 1–43.

https://doi.org/10.1017/asb.2016.14 Published online by Cambridge University Press

BARGES, M., LOISEL, S. and VENEL, X. (2013) On finite-time ruin probabilities with reinsurance cycles influenced by large claims. *Scandinavian Actuarial Journal,* **2013**(3), 163–185.

BEIRLANT, J., GOEGEBEUR, Y., SEGERS, J. and TEUGELS, J. (2006) *Statistics of Extremes: Theory and Applications*. John Wiley & Sons: Chichester, England.

CEBRIAN, A. C., DENUIT, M. and LAMBERT, P. (2003) Generalized Pareto fit to the society of actuaries large claims database. *North American Actuarial Journal,* **7**(3), 18–36.

CHEN, J. and KHALILI, A. (2008) Order selection in finite mixture models with a nonsmooth penalty. *Journal of the American Statistical Association,* **103**(484), 1674–1683.

CHEN, J. and LI, P. (2009) Hypothesis test for normal mixture models: The EM approach. *Annals of Statistics,* **37**(5A), 2523–2542.

CHEN, J., LI, P. and FU, Y. (2012) Inference on the order of a normal mixture. *Journal of the American Statistical Association,* **107**(499), 1096–1105.

COSSETTE, H., MAILHOT, M. and MARCEAU, E. (2012) TVaR-based capital allocation for multivariate compound distributions with positive continuous claim amounts. *Insurance: Mathematics and Economics,* **50**(2), 247–256.

COSSETTE, H., COTE, M. P., MARCEAU, E. and MOUTANABBIR, K. (2013) Multivariate distribution defined with Farlie-Gumbel-Morgenstern copula and mixed Erlang marginals: aggregation and capital allocation. *Insurance: Mathematics and Economics,* **52**(3), 560–572.

DONOHO, D. L. and JOHNSTONE, I. M. (1994) Threshold selection for wavelet shrinkage of noisy data. In *Engineering Advances: New Opportunities for Biomedical Engineers, Proceedings of the 16th Annual International Conference of the IEEE*, A24–A25. Baltimore, MD.

FAN, J. and LI, R. (2001) Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association,* **96**(456), 1348–1360.

FRANK, L. E. and FRIEDMAN, J. H. (1993) A statistical view of some chemometrics regression tools. *Technometrics,* **35**(2), 109–135.

GONG, L., BADESCU, A. L. and CHEUNG, E. C. (2012) Recursive methods for a multi-dimensional risk process with common shocks. *Insurance: Mathematics and Economics,* **50**(1), 109–120.

HASHORVA, E. and RATOVOMIRIJA, G. (2015) On Sarmanov mixed Erlang risks in insurance applications. *ASTIN Bulletin,* **45**(01), 175–205.

KASS, R. E. and WASSERMAN, L. (1995) A reference Bayesian test for nested hypotheses and its relationship to the Schwarz criterion. *Journal of the American Statistical Association,* **90**(431), 928–934.

KERIBIN, C. (2000) Consistent estimation of the order of mixture models. *The Indian Journal of Statistics*, Series A **62**(1), 49–66.

LANDRIAULT, D. and WILLMOT, G.E. (2009) On the joint distributions of the time to ruin, the surplus prior to ruin, and the deficit at ruin in the classical risk model. *North American Actuarial Journal,* **13**(2), 252–270.

LEE, S.C. 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. and LIN, X.S. (2012) Modeling dependent risks with multivariate Erlang mixtures. *ASTIN Bulletin,* **42**(01), 153–180.

LIN, X.S. and WILLMOT, G.E. (2000) The moments of the time of ruin, the surplus before ruin, and the deficit at ruin. *Insurance: Mathematics and Economics,* **27**(1), 19–44.

PORTH, L., ZHU, W. and TAN, K.S. (2014) A credibility-based Erlang mixture model for pricing crop reinsurance. *Agricultural Finance Review,* **74**(2), 162–187.

TEICHER, H. (1963) Identifiability of finite mixtures. *Annals of Mathematical Statistics,* **34**(4), 1265–1269.

TIBSHIRANI, R. (1996) Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, Series B (Methodological), **58**(1), 267–288.

TIJMS, H.C. (2003) *A First Course in Stochastic Models*. John Wiley and Sons.

TSAI, C.C.L. and WILLMOT, G. E. (2002) On the moments of the surplus process perturbed by diffusion. *Insurance: Mathematics and Economics,* **31**(3), 327–350.

VERBELEN, R., ANTONIO, K,. BADESCU, A., GONG, L. and LIN, X.S. 2015. Fitting mixtures of Erlangs to censored and truncated data using the EM algorithm. *ASTIN Bulletin*, **45**(3), 729–758.

https://doi.org/10.1017/asb.2016.14 Published online by Cambridge University Press

VERBELEN, R., ANTONIO, K. and CLAESKENS, G. (2016) Multivariate mixtures of Erlangs for density estimation under censoring. *Lifetime Data Analysis*, to appear.
WILLMOT, G.E. and WOO, J.K. (2015) On some properties of a class of multivariate Erlang mixtures with insurance applications. *ASTIN Bulletin,* **45**(1), 151–173.
YAKOWITZ, S.J. and SPRAGINS, J.D. (1968) On the identifiability of finite mixtures. *The Annals of Mathematical Statistics*, **39**(1), 209–214.
YIN, C. and LIN, X.S. (2016) On the consistency of the order of Erlang mixtures, working paper.

CUIHONG YIN
School of Mathematical Sciences
Xiamen University
Xiamen, China
E-Mail: 460279857@qq.com

X. SHELDON LIN (Corresponding author)
School of Mathematical Sciences
Xiamen University
Xiamen, China
Department of Statistical Sciences
University of Toronto
Toronto, Ontario, Canada M5S 3G3
E-Mail: sheldon@utstat.utoronto.ca

https://doi.org/10.1017/asb.2016.14 Published online by Cambridge University Press

APPENDIX : PARAMETER VALUES OF ERLANG MIXTURE ESTIMATED BY THREE METHODS

TABLE A1

ESTIMATES OF SHAPE, WEIGHT AND SCALE PARAMETERS BY METHOD I.

γ_j	α_j	θ
1	0.9503691692	16621.91
4	0.0003052692	
5	0.0166746763	
6	0.0210814331	
7	0.0014179692	
8	0.0001729291	
9	0.0001442196	
10	0.0002659201	
11	0.0009092656	
12	0.0028866802	
13	0.0016789623	
14	0.0010294484	
15	0.0005081389	
19	0.0001214971	
21	0.0009733723	
23	0.0005471604	
32	0.0006582970	
53	0.0001392742	
70	0.0001163179	

https://doi.org/10.1017/asb.2016.14 Published online by Cambridge University Press

TABLE A2

ESTIMATES OF SHAPE, WEIGHT AND SCALE PARAMETERS BY METHOD II.

γ_j	α_j	θ
1	6.670521e-01	13008.11
4	6.492856e-05	
5	2.438996e-01	
6	1.464163e-02	
10	2.03565e-04	
11	3.21249e-02	
12	1.996671e-02	
13	7.804933e-05	
19	1.492888e-03	
20	1.22534e-02	
21	9.353468e-04	
30	6.340282e-05	
31	2.739182e-03	
32	1.914173e-03	
45	1.633296e-03	
68	5.30461e-04	
89	2.48693e-04	
136	1.12031e-04	
174	4.568729e-05	

https://doi.org/10.1017/asb.2016.14 Published online by Cambridge University Press

TABLE A3

ESTIMATES OF SHAPE, WEIGHT AND SCALE PARAMETERS BY METHOD III.

γ_j	α_j	θ
6	4.357146e-01	3574.662
12	2.082148e-01	
15	1.391404e-01	
20	7.625732e-02	
25	3.480783e-02	
30	3.982403e-02	
40	2.684655e-02	
50	1.425068e-02	
65	1.152380e-02	
85	6.516467e-03	
115	3.886971e-03	
155	1.721645e-03	
200	4.708100e-04	
250	3.426637e-04	
300	2.451458e-04	
370	8.071734e-05	
480	7.057723e-05	
550	5.856789e-05	
1100	2.638905e-05	

https://doi.org/10.1017/asb.2016.14 Published online by Cambridge University Press