Fitting the Erlang Mixture Model to Data via a GEM-CMM Algorithm

Wenyong Gui[a], Rongtan Huang[a], and X. Sheldon Lin[*b]

[a]School of Mathematical Sciences, Xiamen University, Xiamen,China
[b]Department of Statistical Sciences, University of Toronto, Toronto, M5S 3G3, Canada

Abstract The Erlang mixture model with common scale parameter is flexible and analytically tractable. As such, it is a useful model to fit insurance loss data and to calculate quantities of interest for insurance risk management. In this paper, we propose a generalized expectation-maximization (GEM) algorithm along with a clusterized method of moments (CMM) to estimate the model parameters. The GEM algorithm not only estimates the mixing weights and scale parameter of the model but also estimates the shape parameters of the model using a local search method. The CMM method enables to produce quality initial estimates for the GEM algorithm. As a result, the proposed approach provides an efficient algorithm that can fit the model to the body and the tail of truncated and censored loss data well and converges fast. We examine the performance of the proposed approach through several simulation studies and apply it to fit the Erlang mixture model to two real loss data sets.

Keywords Erlang mixture model; Insurance loss data; Generalized EM algorithm; Clusterized method of moments; Local search method

1 Introduction

The class of Erlang mixtures with common scale parameter has been shown to have many desirable distributional properties for insurance valuation and risk management. The class is dense in the space of positive continuous distributions in the sense of weak convergence and hence any loss distribution can be approximated by an Erlang mixture to any accuracy. The survival function and moments of an Erlang mixture can be expressed explicitly, which enables us to calculate risk measures such as VaR and TVaR easily. As a result, Erlang mixtures have recently been found in many insurance applications. See Klugman and Rioux (2006), Lee and Lin (2010), Willmot and Lin (2011), Cossette et al. (2012), Cossette et al. (2013), Hashorva and Ratovomirija (2015), Willmot and Woo (2015), Miljkovic and Grün (2016), Yin and Lin (2016), and references therein. A multivariate version of the Erlang mixture is proposed in Lee and Lin (2012) and further studied by Verbelen et al. (2016) in a truncated and censored data case.

*Corresponding Author
e-mail addresses: 790369790@qq.com (W. Gui), rthuang@xmu.edu.cn (R. Huang), sheldon@utstat.utoronto.ca (X.S. Lin)

Another advantage of the use of Erlang mixtures is the existence of a simple EM algorithm for parameter estimation. See Lee and Lin (2010) and Verbelen et al. (2015). However, there are some drawbacks with the existing EM algorithm. First, as an iterative algorithm the EM algorithm is sensitive to its initial estimates. The choice of initial estimates is critical for the fast convergence of the algorithm. Initial estimates based on the Tijms approximation (Tijms, 1994) in Lee and Lin (2010) might not be ideal. Second, if loss data have long right tail, the algorithm might not be able to fit the tail effectively since the shape parameters are estimated using an ad-hoc procedure. Some efforts have been made to improve the EM algorithm by introducing a penalized likelihood (see Yin and Lin (2016)).

In this paper, we propose a new method called GEM-CMM algorithm for parameter estimation to tackle both aforementioned issues. Insurance losses are often censored and truncated due to policy modifications such as deductibles (left truncation) and policy limits (right censoring). When fitting a statistical model to insurance loss data, it is necessary to take truncation and censoring into consideration. In this paper, we consider fitting the Erlang mixture model to truncated and censored loss data. We propose the use of a generalized EM (GEM) algorithm along with a data-driven clusterized method of moments (CMM) for initialization to estimate key parameters of the model including the shape parameters. Under this approach, the model can fit both the body and the tail of the data well, which is confirmed by simulation studies and real data applications in this paper.

This paper is organized as follows. In Section 2, we provide a brief overview of the Erlang mixture and the currently used EM algorithm for the mixture. In Section 3, a GEM algorithm is proposed to improve the EM algorithm by maximizing the likelihood function of the data and at the meantime by adopting a local search method (Givens and Hoeting, 2013) to estimate the shape parameters in the M-step of the EM algorithm. In Section 4, a CMM method is presented to obtain high quality initial estimates for the GEM algorithm so that the algorithm converges rapidly. The initialization involves clustering data using the K-means method and the application of the method of moments for each cluster. In Section 5, the parameters are adjusted to further improve the fitting. In Section 6, we test the efficiency of our algorithm through several simulation studies. We then apply the algorithm to fit the model to real data in Section 7. The results in both sections show that the model fits the data well. We conclude in Section 8 with some remarks on future research.

2 Erlang mixtures and associated EM algorithm

In this section, we recall the definition of the univariate Erlang mixture model with common scale parameter and an EM algorithm for parameter estimation.
An Erlang distribution has probability density function (pdf)

$$f(x|m,\theta) = \frac{x^{m-1}e^{-x/\theta}}{\theta^m(m-1)!}, x > 0, \tag{2.1}$$

where m is a positive integer and $\theta > 0$. Its survival function is given by

$$\overline{F}(x|m,\theta) = e^{-x/\theta} \sum_{n=0}^{m-1} \frac{x^n}{\theta^n n!}. \tag{2.2}$$

An M-component Erlang mixture with common scale parameter has pdf

$$h(x|\Phi) = \sum_{u=1}^{M} \alpha_u f(x|m_u,\theta) = \sum_{u=1}^{M} \alpha_u \frac{x^{m_u-1}e^{-x/\theta}}{\theta^{m_u}(m_u-1)!}, \tag{2.3}$$

where the parameters are $\Phi = \{\alpha_u, m_u, \theta, u = 1,\cdots,M\}$ with weight constraints $\alpha_u > 0$, $\sum_{u=1}^{M} \alpha_u = 1$, and scale parameter $\theta > 0$. We denote its distribution function as $H(x|\Phi)$ and the survival function as $\overline{H}(x|\Phi) = 1 - H(x|\Phi)$.

Lee and Lin (2010) show that the class of univariate Erlang mixtures is dense in the space of positive continuous distributions in the sense of weak convergence. There are explicit expressions for many distributional quantities such as the moments, the distribution function and the characteristic function. Risk measures such as value-at-risk (VaR) and tail VaR (TVaR) can be easily calculated as well.

Borrowing the notation in Verbelen et al. (2015) for censoring and truncation, denote the common truncation range of the data as (t^l, t^r). Let $\boldsymbol{X} = (X_1,\cdots,X_n)$ be the underlying random sample under truncation and its realization with censoring be $\boldsymbol{x} = (x_1,\cdots,x_n)$. For each data point $x_v, v = 1,\cdots,n$, we denote its censoring interval as (l_v, r_v), i.e., the realization of X_v insider the interval is censored. Thus, x_v is determined as follows:

$$\begin{aligned}
&\text{uncensored:} && t^l \le l_v = r_v = x_v \le t^r,\\
&\text{left censored:} && t^l = l_v < x_v = r_v < t^r,\\
&\text{right censored:} && t^l < l_v = x_v < r_v = t^r.
\end{aligned}$$

The density function of X_v is given by

$$g(x_v|t^l,t^r,\Psi) = \sum_{u=1}^{M} \beta_u p(x_v|t^l,t^r,m_u,\theta), t^l \le x_v \le t^r, \tag{2.4}$$

where $\beta_u = \alpha_u \dfrac{F(t^r|m_u,\theta) - F(t^l|m_u,\theta)}{H(t^r|\Phi) - H(t^l|\Phi)}$ and $p(x_v|t^l,t^r,m_u,\theta) = \dfrac{f(x_v|m_u,\theta)}{F(t^r|m_u,\theta) - F(t^l|m_u,\theta)}$. Thus, the model parameters are re-parametrized to be $\Psi = \{\beta_u, m_u, \theta, u = 1,\cdots,M\}$.

An EM algorithm for parameter estimation for the Erlang mixture model is presented in Lee and Lin (2010) and extended in Verbelen et al. (2015) whose EM algorithm deals with truncated and censored data. In both papers, the set of parameters to be estimated are mixing weights and the scale parameter: $\Phi_{-\mathbf{m}} = \{\theta, \alpha_u, u = 1,\cdots,M\}$, or $\Psi_{-\mathbf{m}} = \{\theta, \beta_u, u = 1,\cdots,M\}$ in the truncation and censoring case with relationship

$$\alpha_u = c\frac{\beta_u}{F(t^r|m_u,\theta) - F(t^l|m_u,\theta)}, u = 1,2,\cdots,M, \tag{2.5}$$

where c is a normalizing constant such that $\sum_{u=1}^{M} \alpha_u = 1$.

An EM algorithm treats the data as an incomplete data set by introducing a set of latent variables $\mathbf{Z} = (\mathbf{Z}_1, \mathbf{Z}_2, \cdots, \mathbf{Z}_n)$, where $\mathbf{Z}_v = (Z_{v1}, \cdots, Z_{vM})$, $v = 1, 2, \cdots, n$, with

$$Z_{vu} = \begin{cases} 1, & \text{if } X_v \text{ comes from the } u\text{th component density } f(x_v|m_u, \theta), \\ 0, & \text{otherwise.} \end{cases} \tag{2.6}$$

In the following we present the standard EM algorithm for the Erlang mixture. Note that since the truncation points $t^l = 0$ and $t^r = \infty$ imply no truncation, the algorithm also works for non-truncated data.

E-Step: Assuming that in the kth iteration of the E-step, the current parameter values are $\Psi_{-\mathbf{m}}^{(k)} = \{\theta^{(k)}, \beta_u^{(k)}, u = 1, \cdots, M\}$, the expectation of the complete log-likelihood given the data and the current parameters $\Psi_{-\mathbf{m}}^{(k)}$ is

$$\begin{aligned} Q(\Psi_{-\mathbf{m}}|\Psi_{-\mathbf{m}}^{(k)}) &= \sum_{v=1}^{n}\sum_{u=1}^{M} z_{vu}^{(k)}(\ln\beta_u - \tfrac{1}{\theta}E[X_v|Z_{vu} = 1, l_v, r_v, t^l, t^r, \Psi_{-\mathbf{m}}^{(k)}] \\ &\quad -m_u \ln\theta - \ln(F(t^r|m_u, \theta) - F(t^l|m_u, \theta))), \end{aligned} \tag{2.7}$$

where

$$z_{vu}^{(k)} = \frac{\alpha_u^{(k)}\widetilde{f}(l_v, r_v|m_u, \theta^{(k)})}{\sum_{w=1}^{M}\alpha_w^{(k)}\widetilde{f}(l_v, r_v|m_w, \theta^{(k)})}, v = 1, \cdots, n, u = 1, \cdots, M, \tag{2.8}$$

with

$$\widetilde{f}(l_v, r_v|m_u, \theta^{(k)}) = \begin{cases} f(x_v|m_u, \theta^{(k)}), & l_v = r_v = x_v, \\ F(r_v|m_u, \theta^{(k)}) - F(l_v|m_u, \theta^{(k)}), & l_v < r_v. \end{cases}$$

The expectation $E[X_v|Z_{vu} = 1, l_v, r_v, t^l, t^r, \Psi_{-\mathbf{m}}^{(k)}]$ is equal to x_v if it is uncensored. Here, we omit the terms that do not contain mixing weights and the scale parameter.

M-step: The mixing weights and the common scale parameter are updated by

$$\beta_u^{(k+1)} = \frac{1}{n}\sum_{v=1}^{n} z_{vu}^{(k)}, u = 1, \cdots, M, \tag{2.9}$$

$$\theta^{(k+1)} = \frac{\sum_{v=1}^{n}\sum_{u=1}^{M} z_{vu}^{(k)} E[X_v|Z_{vu} = 1, l_v, r_v, t^l, t^r, \Psi_{-\mathbf{m}}^{(k)}] - T^{(k+1)}}{n\sum_{u=1}^{M} m_u\beta_u^{(k+1)}}, \tag{2.10}$$

where

$$T^{(k+1)} = n\sum_{u=1}^{M}\beta_u^{(k+1)} \left.\frac{(t^l)^{m_u}e^{-t^l/\theta} - (t^r)^{m_u}e^{-t^r/\theta}}{\theta^{m_u-1}(m_u-1)!(F(t^r|m_u, \theta) - F(t^l|m_u, \theta))}\right|_{\theta=\theta^{(k)}}. \tag{2.11}$$

The detailed derivation and initialization can be found in Lee and Lin (2010) and Verbelen et al. (2015). However, some issues arise: first, the initial estimates play an important role to the fast convergence of the EM algorithm. The initialization method based on the Tijms approximation is, although justifiable,

not computationally efficient. Second, the ad-hoc method for shape parameters adjustment requires a large number of runs of the EM algorithm, which adds greatly the computation burden. To improve the fit, we propose a generalized EM (GEM) algorithm with shape parameters being taken into consideration and a clusterized method of moments (CMM) to obtain high quality initial values for Erlang mixtures in the next two sections.

3 A GEM algorithm for parameter estimation

In this section, we extend the standard EM algorithm such that it also estimates the shape parameters. The following approach is motivated by Givens and Hoeting (2013).

Recalling the definition of latent variables $\mathbf{Z}_v, v = 1, \cdots, n$ in Section 2, the corresponding complete random sample $(X_1, \mathbf{Z}_1), (X_2, \mathbf{Z}_2), \cdots, (X_n, \mathbf{Z}_n)$ contains all uncensored observations and latent variables. Then we have:
(a) $(X_1, \mathbf{Z}_1), (X_2, \mathbf{Z}_2), \cdots, (X_n, \mathbf{Z}_n)$ are independent and identically distributed;
(b) $\mathbf{Z}_v \sim Mult_M(1, \boldsymbol{\alpha}), v = 1, \cdots, n$, that is, $\mathbf{Z}_v$ are multinomially distributed with the number of trials being one and event probabilities $\alpha_1, \cdots, \alpha_M$;
(c) The conditional distribution of X_v given $\mathbf{Z}_v = \mathbf{z}_v$ is

$$L(x_v|\mathbf{z}_v, \Phi) = \prod_{u=1}^{M} \{f(x_v|m_u, \theta)\}^{z_{vu}} = \prod_{u=1}^{M} \left\{ \frac{x_v^{m_u-1} e^{-x_v/\theta}}{\theta^{m_u}(m_u-1)!} \right\}^{z_{vu}}, v = 1, \cdots, n.$$

We now present the GEM algorithm.

E-step: Suppose that in the kth iteration of the E-step, the current parameter values are $\Psi^{(k)} = \{\beta_u^{(k)}, m_u^{(k)}, \theta^{(k)}, u = 1, \cdots, M\}$. The conditional expectation for the complete log-likelihood given the current parameter values and data is given by

$$\begin{aligned} Q(\Psi|\Psi^{(k)}) = \sum_{v=1}^{n}\sum_{u=1}^{M} z_{vu}^{(k)} \Big[& \ln \beta_u + (m_u - 1) E[\ln(X_v)|Z_{vu} = 1, l_v, r_v, t^l, t^r, \Psi^{(k)}] \\ & - \frac{1}{\theta} E[X_v|Z_{vu} = 1, l_v, r_v, t^l, t^r, \Psi^{(k)}] - \ln(m_u - 1)! \\ & - m_u \ln\theta - \ln(F(t^r|m_u, \theta) - F(t^l|m_u, \theta)) \Big], \end{aligned} \tag{3.1}$$

where the posterior probability $z_{vu}^{(k)}$ has the same form as (2.8) but with m_u being replaced by $m_u^{(k)}$. The expectations in (3.1) are given in the following:
(i) for a censored data point (l_v, r_v),

$$E[X_v|Z_{vu} = 1, l_v, r_v, t^l, t^r, \Psi^{(k)}] = \frac{\theta^{(k)} m_u^{(k)} (F(r_v|m_u^{(k)} + 1, \theta^{(k)}) - F(l_v|m_u^{(k)} + 1, \theta^{(k)}))}{F(r_v|m_u^{(k)}, \theta^{(k)}) - F(l_v|m_u^{(k)}, \theta^{(k)})},$$

and

$$E(\ln(X_v)|Z_{vu}=1,l_v,r_v,t^l,t^r,\Psi^{(k)})$$
$$=\frac{(\ln l_v\overline{F}(l_v|m_u^{(k)},\theta^{(t)})-\ln r_v\overline{F}(r_v|,m_u^{(k)},\theta^{(k)}))+\sum_{n=0}^{m_u^{(k)}-1}\frac{1}{n}[F(r_v|n,\theta^{(k)})-F(l_v|n,\theta^{(k)})]}{F(r_v|m_u^{(k)},\theta^{(k)})-F(l_v|m_u^{(k)},\theta^{(k)})};$$

(ii) for an uncensored data point x_v,

$$E[X_v|Z_{vu}=1,l_v,r_v,t^l,t^r,\Psi^{(k)}]=x_v,$$

and

$$E[\ln(X_v)|Z_{vu}=1,l_v,r_v,t^l,t^r,\Psi^{(k)}]=\ln x_v.$$

M-step: We update the parameters by maximizing

$$\Psi^{(k+1)}=\arg\max_{\Psi}Q(\Psi|\Psi^{(k)}). \tag{3.2}$$

The updates for mixing weights $\beta_u^{(k+1)},u=1,\cdots,M$ and common scale parameter $\theta^{(k+1)}$ have the same forms as those in Section 2 with m_u being replaced by $m_u^{(k+1)}$.

Noting that the scale parameter depends on the shape parameters, we propose a local search method to find optimal shape parameters to maximize $Q(\Psi|\Psi^{(k)})$ and then update the scale parameter. First, we replace the mixing weights and the common scale parameter in expression (3.1) with the new values and consider the expression as a function of shape parameters only. In this case, we may rewrite the formula (3.1) as

$$\begin{aligned}Q^*(\mathbf{m})=\sum_{v=1}^{n}\sum_{u=1}^{M}z_{vu}^{(k)}\Big[&\ln\beta_u^{(k+1)}+(m_u-1)E[\ln(X_v)|Z_{vu}=1,l_v,r_v,t^l,t^r,\Psi^{(k)}]\\&-\frac{1}{\widetilde{\theta}}E[X_v|Z_{vu}=1,l_v,r_v,t^l,t^r,\Psi^{(k)}]-\ln(m_u-1)!\\&-m_u\ln\widetilde{\theta}-\ln(F(t^r|m_u,\widetilde{\theta})-F(t^l|m_u,\widetilde{\theta}))\Big],\end{aligned} \tag{3.3}$$

where $\mathbf{m}=(m_1,m_2,\cdots,m_M)$ and

$$\widetilde{\theta}=(\sum_{v=1}^{n}\sum_{u=1}^{M}z_{vu}^{(k)}E[X_v|Z_{vu}=1,l_v,r_v,t^l,t^r,\Psi^{(k)}]-T^{(k+1)})/(n\sum_{u=1}^{M}m_u\beta_u^{(k+1)}),$$

where $T^{(k+1)}$ is given in (2.11).

To maximize $Q^*(\mathbf{m})$, we apply an iterative method called the 3-optimal method, similar to that in Givens and Hoeting (2013). Denote

$$\delta_u^+=Q^*(\mathbf{m}+\mathbf{e}_u)-Q^*(\mathbf{m})$$

and

$$\delta_u^- = Q^*(\mathbf{m} - \mathbf{e}_u) - Q^*(\mathbf{m}),$$

where $\mathbf{e}_u$ is an $M-$length vector with the uth entry equals to 1 and others 0. Then the shape parameters are adjusted by $\mathbf{m}^{[t]} = \mathbf{m}^{[t-1]} + \Delta\mathbf{m}$ with $\mathbf{m}^{[0]} = \mathbf{m}^{(k)}$. Here,

$$\Delta\mathbf{m} = \underbrace{(0, \cdots, 0, \Delta m_u, 0, \cdots, 0)}_{u\text{th entry is } \Delta m_u}, \ t = Ms + u, \tag{3.4}$$

where s is an integer satisfies $t = Ms + u$ for $1 \le u \le M$, and Δm_u is calculated by

$$\Delta m_u = \begin{cases} 1, & \max\{\delta_u^+, \delta_u^-\} > \varepsilon,\ \delta_u^+ > \delta_u^-, \\ -1, & \max\{\delta_u^+, \delta_u^-\} > \varepsilon,\ \delta_u^- > \delta_u^+,\ m_u > 1, \quad u = 1, \cdots, M, \\ 0, & \text{others}, \end{cases} \tag{3.5}$$

where $\varepsilon \ge 0$ is a predefined threshold. This process for searching the shape parameters repeats until the parameters do not change any more. Afterwards, the common scale parameter is updated according to (2.10).

Finally, given the final estimates $\widehat{\Psi} = \{\widehat{\theta}, \widehat{\beta}_u, \widehat{m}_u, u = 1, 2, \cdots, M\}$, the estimates of the original mixing weights are

$$\widehat{\alpha}_u = c\frac{\widehat{\beta}_u}{F(t^r|\widehat{m}_u, \widehat{\theta}) - F(t^l|\widehat{m}_u, \widehat{\theta})}, u = 1, 2, \cdots, M, \tag{3.6}$$

where c is a normalizing constant such that $\sum_{u=1}^{M} \widehat{\alpha}_u = 1$.

The above GEM algorithm optimizes the parameters for the Erlang mixtures with the fixed number of components M. In order to reach a satisfactory fitting result, one often needs to start with a large number of components, which may lead to overfitting. Hence, we further apply a forward selection approach to select the least possible number of components using the cross-validation or the CV method. We split the data set into a training set denoted by S_T and a validation set denoted by S_V. Assume the fitted density function depending on the training set S_T is $\widehat{h}(\cdot|\widehat{\Phi})$, where $\widehat{\Phi}$ are the estimated parameters. For a data point represented by (l_v, r_v) with truncation (t^l, t^r), we introduce the score function

$$CV(l_v, r_v) = \begin{cases} \ln \widehat{h}(x_v|\widehat{\Phi}) - \ln(\widehat{H}(t^r|\widehat{\Phi}) - \widehat{H}(t^l|\widehat{\Phi})), & l_v = r_v = x_v, \\ \ln(\widehat{H}(r_v|\widehat{\Phi}) - \widehat{H}(l_v|\widehat{\Phi})) - \ln(\widehat{H}(t^r|\widehat{\Phi}) - \widehat{H}(t^l|\widehat{\Phi})), & l_v < r_v, \end{cases}$$

where $\widehat{H}(\cdot|\widehat{\Phi})$ is the distribution function of the fitted model, and define the score function on the validation set S_V as

$$CV(S_V) = \sum_{(l_v, r_v) \in S_V} CV(l_v, r_v). \tag{3.7}$$

The rationale for such a score function to judge the adequacy of fit of a statistical model can be found in Silverman (1986). We adopt the 10-fold cross-validation for this purpose. The detailed procedure is given as follows.

(1) Randomly partition the data set into 10 equal sized groups;

(2) Of the 10 groups, a single group is retained as the validation data for testing the model, and the remaining 9 groups are used as training data. Estimate the score function using the fitted model;
(3) Repeat the cross-validation process 10 times (the folds), with each of the 10 groups being used exactly once as the validation data and calculate the score function;
(4) The 10 results from the folds can then be averaged to produce a single estimation for the score function.

This procedure is repeated with different numbers of components such that the number of components is determined to maximize the average of the score function.

4 A CMM algorithm for parameter initialization

The EM algorithm as an iterative algorithm highly depends on initial estimates. In this section we propose a new method which combines the method of moments and K-means clustering to obtain high quality initial estimates for the GEM algorithm. We call this method the clusterized method of moments, or CMM for short. We measure the quality of an algorithm according to the run time of the algorithm.
For the random sample $(X_1, \mathbf{Z}_1), \cdots, (X_n, \mathbf{Z}_n)$, we have the following properties: for $v = 1, \cdots, n, u = 1, \cdots, M$,

$$E[Z_{vu}] = P(Z_{vu} = 1) = \alpha_u, \tag{4.1}$$

$$E[X_v | Z_{vu} = 1] = \theta m_u \triangleq \mu_u,$$

$$E[X_v Z_{vu}] = \alpha_u \mu_u, \tag{4.2}$$

$$E[X_v] = \theta \sum_{u=1}^{M} \alpha_u m_u = \sum_{u=1}^{M} \alpha_u \mu_u, \tag{4.3}$$

$$E[X_v^2] = \theta^2 \sum_{u=1}^{M} \alpha_u (m_u + m_u^2) = \theta E[X_v] + \sum_{u=1}^{M} \alpha_u \mu_u^2. \tag{4.4}$$

For convenience, we re-parametrize Φ as $\Phi' = \{\theta, \alpha_u, \mu_u, u = 1, \cdots, M\}$. In this case, the shape parameters are then estimated by $m_u = \lceil \mu_u / \theta \rceil$, where $\lceil x \rceil$ is the ceiling function of x.

Properties (4.1)-(4.4) are used as a basis to estimate the initial parameters as follows.

(1) We apply the following K-means clustering method to group the data into M groups so that Group $u, u = 1, \cdots, M$, represents data from the uth component distribution of the mixture. The goal of this K-means clustering method is to find the values of $\mathbf{z}_1, \cdots, \mathbf{z}_n$ and centers $\nu_1, \cdots, \nu_M$, by minimizing the expression

$$J = \sum_{v=1}^{n} \sum_{u=1}^{M} z_{vu} (x_v - \nu_u)^2. \tag{4.5}$$

Assuming the centers and the values of $\mathbf{z}_1, \cdots, \mathbf{z}_n$ in the previous step are $\nu_1^{(k-1)}, \cdots, \nu_M^{(k-1)}$ and $\mathbf{z}_1^{(k-1)}, \cdots, \mathbf{z}_n^{(k-1)}$, the new values are determined using the following procedure (Bishop, 2006):

(a) new centers $\nu_u, \cdots, \nu_M$ are obtained by

$$\nu_u = \frac{\sum_{v=1}^{n} z_{vu}^{(k-1)} x_v}{\sum_{v=1}^{n} z_{vu}^{(k-1)}}. \tag{4.6}$$

(b) with the new centers $\nu_1, \cdots, \nu_M$ in (a), the values of $\mathbf{z}_1, \cdots, \mathbf{z}_n$ are given by

$$z_{vu} = \begin{cases} 1, & u = \arg\min_v (x_v - \nu_u)^2, \\ 0, & \text{otherwise}, \end{cases} \quad v = 1, \cdots, n, u = 1, \cdots, M. \tag{4.7}$$

The data points are then re-assigned to the clusters by minimizing J in (4.5). The procedure is repeated until there are no further changes in the assignments.

(2) According to (4.1), the mixing weights are estimated by

$$\widehat{\alpha}_u = \frac{\sum\limits_{v=1}^{n} z_{vu}}{n} = \frac{n_u}{n}, u = 1, 2, \cdots, M, \tag{4.8}$$

where $n_u = \sum\limits_{v=1}^{n} z_{vu}$ is the number of the data points clustered into the uth group.

(3) According to (4.2), the mean parameters are estimated by

$$\widehat{\mu}_u = \frac{\sum\limits_{v=1}^{n} x_v z_{vu}}{n} / \widehat{\alpha}_u = \frac{\sum\limits_{v=1}^{n} x_v z_{vu}}{n_u}, u = 1, 2, \cdots, M. \tag{4.9}$$

(4) According to (4.4), the common scale parameter is estimated by

$$\widehat{\theta} = \left(\overline{x^2} - \sum_{u=1}^{M} \widehat{\alpha}_u \widehat{\mu}_u^2 \right) / \overline{x}, \tag{4.10}$$

where $\overline{x} = \frac{1}{n} \sum\limits_{v=1}^{n} x_v$, $\overline{x^2} = \frac{1}{n} \sum\limits_{v=1}^{n} x_v^2$.
From the above initial estimates, we have the following equation

$$\sum_{v=1}^{n} (x_v - \overline{x})^2 = \sum_{u=1}^{M} \widehat{\alpha}_u (\widehat{\mu}_u - \overline{x})^2 + \widehat{\theta} \overline{x}. \tag{4.11}$$

This is the decomposition formula of the sum of squared errors. The term on the left hand side represents the sum of squared errors which will not be influenced by the parameters. The first term on the right hand side represents the sum of squared errors among groups while the second term represents the sum of squared errors within each of the groups. When fitting an Erlang mixture to a positive continuous distribution, a smaller θ is desirable (see Lee and Lin (2010)), as long as overfitting is avoided. It implies that we should try to make the sum of squares within each of the groups as small as possible. The K-means clustering is

exactly such a method aiming to classify data to minimize the within-cluster sum of squared errors, and hence is a reasonable method to solve the clustering issue.
Sometimes a large $\hat{\theta}$ might lead to the situation that many shape parameters may be equal to 1, which often appears when data have a long tail. In this situation we may modify the estimate by

$$\hat{\theta}^* = \min\{\hat{\theta}, \min\{\widehat{\mu}_u; u = 1, \cdots, M\}\}. \tag{4.12}$$

At last, the initial shape parameters are estimated by

$$\hat{m}_u = \lceil \widehat{\mu}_u / \hat{\theta}^* \rceil, u = 1, 2, \cdots, M. \tag{4.13}$$

By taking truncation into account, we transform the initial values for the mixing weights by

$$\widehat{\beta}_u = \widehat{\alpha}_u \frac{F(t^r|\widehat{m}_u, \widehat{\theta}) - F(t^l|\widehat{m}_u, \widehat{\theta})}{H(t^r|\widehat{\Phi}) - (t^l|\widehat{\Phi})}, u = 1, 2, \cdots, M. \tag{4.14}$$

5 Further parameter adjustments

In order to further improve the model fitting, we may adjust the parameters by matching the first two moments of the fitted model with the corresponding empirical moments.

5.1 Data with no truncation and censoring

As shown in Lee and Lin (2010), if X has an Erlang mixture of form (2.3), then it can be rewritten as a random sum with a compound exponential distribution, i.e.,

$$X = \sum_{i=1}^{N} E_i, \tag{5.1}$$

where N is the primary counting random variable with probability function $P(N = m_u) = \alpha_u, u = 1, \cdots, M$, and $E_i, i = 1, 2, \cdots,$ are iid exponential random variables with mean θ.

It is easy to see

$$E[X] = \theta E[N], \ E[X^2] = \theta^2 (E[N] + E[N^2]), \tag{5.2}$$

where $E[N] = \sum_{u=1}^{M} \alpha_u m_u$, and $E[N^2] = \sum_{u=1}^{M} \alpha_u m_u^2$.

We now introduce an additional tuning parameter s to adjust the parameters. For notational convenience, we suppose that the estimated parameters via the GEM algorithm are $\Phi = \{\theta, \alpha_u, m_u, u = 1, \cdots, M\}$. The common scale parameter is to be adjusted by $\theta^* = s\theta$ and the shape parameters by $m_u^* = \lceil m_u / s \rceil$ so that the first moment remains the same. The Erlang mixture is then fitted to the data with new parameters $\Phi^* = \{\theta^*, \alpha_u, m_u^*, u = 1, \cdots, M\}$ such that the first two moments of the model with corresponding sample moments are matched with a properly chosen tuning parameter s.

The corresponding random variable X^* again has an Erlang mixture of form (2.3) with parameters Φ^*, and may be rewritten as $X^* = \sum_{i=1}^{N^*} E_i^*$, where N^* has probability function $P(N^* = m_u^*) = \alpha_u, u = 1, \cdots, M$, and $E_i^*, i = 1, 2, \cdots$, are iid exponential random variables with mean θ^*. Thus, we have

$$E[X^*] = \theta^* E[N^*] \approx \theta E[N],\ E[X^{*2}] \approx \theta^2 (sE[N] + E[N^2]). \tag{5.3}$$

Here, an approximation instead of an equation is used because the new parameters $m_u^* = \lceil m_u/s \rceil \approx m_u/s, u = 1, \cdots, M$. Now, the squared coefficient of variation of X^*, denoted by $c_{X^*}^2$, is given by

$$c_{X^*}^2 = \frac{Var(X^*)}{E[X^*]^2} \approx \frac{s}{E[N]} + \frac{E[N^2]}{E[N]^2} - 1. \tag{5.4}$$

Similarly, for the observed data, the squared sample coefficient of variation denoted by c_x^2 is

$$c_x^2 = \frac{\overline{x^2}}{\overline{x}^2} - 1, \tag{5.5}$$

where $\overline{x^2} = \frac{1}{n} \sum_{v=1}^{n} x_v^2$, $\overline{x} = \frac{1}{n} \sum_{v=1}^{n} x_v$.

We match the squared coefficients of variation, i.e. $c_{X^*}^2 = c_x^2$, which results in

$$\frac{\overline{x^2}}{\overline{x}^2} = \frac{s}{E[N]} + \frac{E[N^2]}{E[N]^2}. \tag{5.6}$$

The tuning parameter s thus is estimated by

$$\widehat{s} = E[N] \left(\frac{\overline{x^2}}{\overline{x}^2} - \frac{E[N^2]}{E[N]^2} \right). \tag{5.7}$$

We may modify the tuning parameter by

$$s^* = \min\{\widehat{s}, \min\{m_u\}, u = 1, 2, \cdots, M\}, \tag{5.8}$$

for the same reason as that to (4.12).

With the estimated value s^*, we finally adjust the scale parameter and the shape parameters can be adjusted by

$$\theta^* = s^* \theta,\ m_u^* = \lceil m_u/s^* \rceil, u = 1, 2, \cdots, M. \tag{5.9}$$

5.2 Truncated and censored data

Unlike data with no truncation and no censoring, the problem becomes slightly more complex.

Again consider new parameters $\Phi^* = \{\theta^*, \alpha_u, m_u^*, u = 1, \cdots, M\}$, where $\theta^* = s\theta$ and $m_u^* = \lceil m_u/s \rceil$.

For an uncensored random variable X^* comes from an Erlang mixture with the new parameters, we have

$$E[X^*|t^l < X^* < t^r] = \frac{\theta}{H(t^r|\Phi^*) - H(t^l|\Phi^*)} \sum_{u=1}^{M} \alpha_u m_u^* [F(t^r|m_u^* + 1, \theta^*) - F(t^l|m_u^* + 1, \theta^*)], \tag{5.10}$$

and

$$E[X^{*2}|t^l < X^* < t^r] = \frac{(\theta^*)^2}{H(t^r|\Phi^*) - H(t^l|\Phi^*)} \sum_{u=1}^{M} \alpha_u m_u^*(m_u^* + 1)[F(t^r|m_u^* + 2, \theta^*) - F(t^l|m_u^* + 2, \theta^*)]. \tag{5.11}$$

We then compute the tuning parameter s by minimizing the following objective function:

$$\begin{aligned} O(s) \quad &= \left\{ \sum_{v;x_v \text{ uncensored}} (x_v - E[X^*|t^l < X^* < t^r]) \right\}^2 \\ &+ \left\{ \sum_{v;x_v \text{ uncensored}} (x_v^2 - E[X^{*2}|t^l < X^* < t^r]) \right\}^2. \end{aligned} \tag{5.12}$$

The optimization problem can be solved easily by a numerical procedure. The remaining procedure is the same as data with no truncation.

After adjusting the parameters, we set the new parameters as initial values and then apply the GEM algorithm again. In other words, we repeat the aforementioned adjusting procedure and the GEM algorithm until the increment of the likelihood function (the decrement of the corresponding BIC is twice the increment) is less than a pre-specified threshold.

6 Simulation studies

This section provides simulation studies to illustrate the efficiency of the proposed method in the previous sections. The simulated and real data sets used in this and next sections and the R code of the proposed algorithm are available at http://www.utstat.utoronto.ca/~sheldon/DataandCodes.html for interested readers to download, test and implement for other positive data.

The first simulation study involves fitting the Erlang mixture to simulated data from a mixture of two gamma distributions with different scale and shape parameters. The main purpose of this study is to see how the Erlang mixture fits a multi-modal distribution outside its class. The second study involves fitting the model to simulated data from a Pareto distribution. Since the Pareto distribution has a long right tail, the main purpose of this study is to examine whether the algorithm can capture the long tail. In our simulation studies, we also compare the run time of our method with those in Lee and Lin (2010) and Verbelen et al. (2015). The run time of an algorithm is used to measure the "quality" of initial values. The results show that our initial values can lead to a faster convergence, which indicates that we obtain quality initial values.

6.1 Mixture of two gamma distributions

2000 data points are generated from a mixture of two gamma distributions with shape parameters 10 and 15. The corresponding scale parameters are 0.5 and 1, having weights of 0.4 and 0.6, respectively. Hence, this distribution is not in the class of the Erlang mixture with common scale parameter. We consider left

truncation with different truncation points t^l in this example. The data points are removed if they are less than t^l and the remaining data are used to fit the model. To illustrate the effect of different truncation points, we let $t^l = 0$, $t^l = 3$ and $t^l = 5$, respectively. The estimated parameter values are given in Table 1.

Table 1: Estimated parameter values of fitted Erlang mixtures

truncation point	u	α_u	m_u	θ
$t^l = 0$	1	0.4361	7	0.7621
	2	0.5639	20	
$t^l = 3$	1	0.4422	8	0.6882
	2	0.5578	22	
$t^l = 5$	1	0.4525	8	0.6707
	2	0.5475	23	

From Table 1, we can see, as expected, that the model has different parameter values for different truncation points. But even with a high truncation point, the estimated parameter values remain fairly close to the ones without truncation. Figure 1 shows the efficiency of the models for different truncation points in which we compare the true density, the density of the fitted Erlang mixtures with truncated points $t^l = 0, 3, 5$, respectively, and the histogram of all 2000 points. From Figure 1, one may observe that all the fitted Erlang mixtures fit the data reasonably well: the curves almost overlap the true density curve. Figure 2 shows the PP-plot and QQ-plot for $t^l = 3$, the plots indicate that the fitted model can fit data well in body and in tail. However in practice, as the left truncation point increases we caution that the model may not fit the original data well due to the loss of information. The greater the truncation point is, the more information is lost.

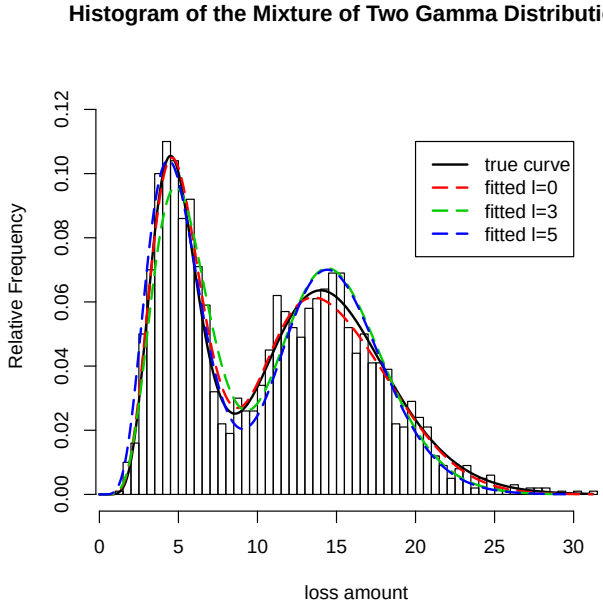

Figure 1: Densities of fitted Erlang mixtures

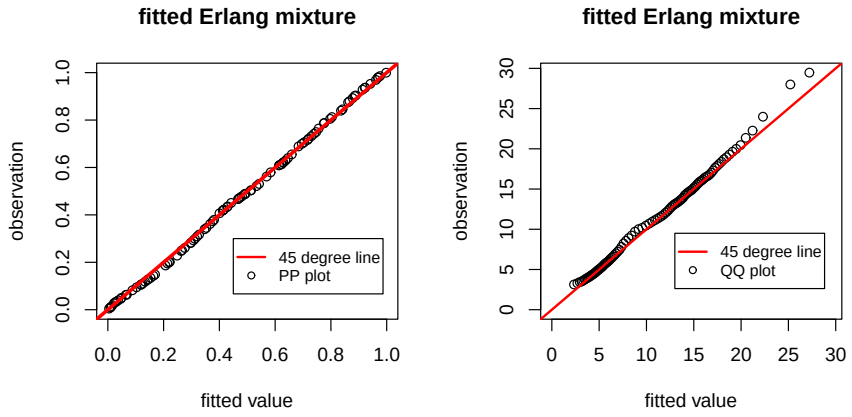

Figure 2: PP-plot and QQ-plot of fitted mixture at $t^l = 3$

Another aspect of the proposed algorithm is to reduce the computing time. In Table 2, we compare the run time of our approach (denoted by Method I) with the run time of the original modified EM algorithm in Verbelen et al. (2015) (denoted by Method II) for fitting the mixture of two gammas with $t^l = 3$. It shows, in addition to the improvement of BIC, that the current approach reduces the run time significantly.

Table 2: Comparison of two methods

methods	Method I	Method II
components	2	2
log-likelihood	-5697.725	-5826.801
BIC	11433.309	11691.61
run time	0.2502379 secs	5.047358 secs

6.2 Fitting data from Pareto distributions

The Pareto distribution has a long (right) tail. In this study we use an Erlang mixture to fit data generated from the Type II Pareto distribution with shape parameter α and scale parameter λ:

$$p(x|\alpha, \lambda) = \frac{\alpha\lambda^\alpha}{(x+\lambda)^{\alpha+1}}, \ x \geq 0. \tag{6.1}$$

First, 2000 data points are generated from a Type II Pareto with shape parameter 1.5 and scale parameter 10. Denote the data as $(x_1, \cdots, x_{2000})$. We then fit the data using an Erlang mixture. A 7-component Erlang mixture is obtained to fit the data and its parameter estimates are given in Table 3.

Table 3: Parameter estimates of fitted 7-component Erlang mixture

u	α_u	m_u	θ
1	0.7517	1	5.2309
2	0.1707	5	
3	0.0513	13	
4	0.0158	25	
5	0.0046	44	
6	0.0035	83	
7	0.0025	170	

We compare our results with those from the kernel estimation, a common universal statistical method, and then with those from the EM algorithm in Lee and Lin (2010). Figure 3 shows the PP-plots and QQ-plots, from which we observe that the Erlang mixture is able to fit the body and the tail of the Pareto, even though the Erlang mixture and the Pareto have completely different tails. This is because the impact of large values will be captured when we adopt the K-means method to cluster the data. Also the PP-plot of the kernel estimation indicates the fitness is unsatisfactory at the left tail, a drawback of the kernel estimation method.

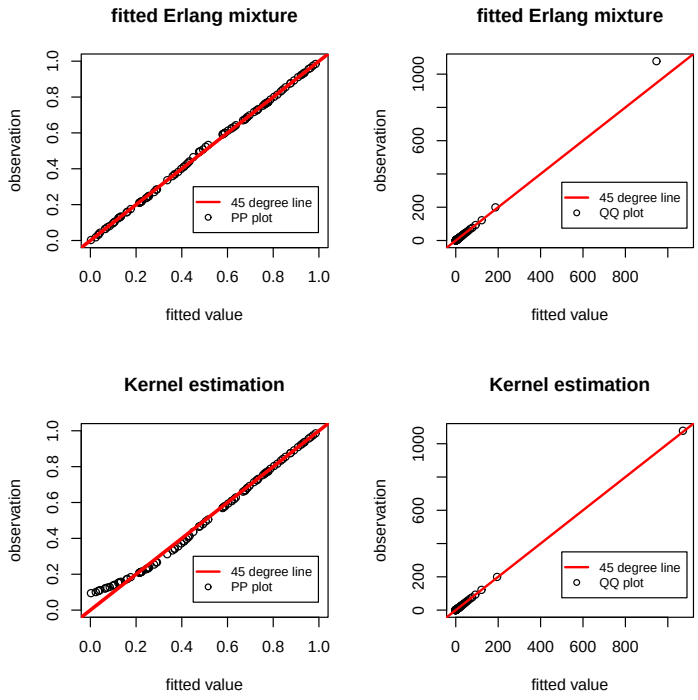

Figure 3: PP-plots and QQ-plots for fitted model vs. kernel estimation/standard EM

We also perform several common statistical tests on the fitness. The tests we use in this section are the Kolmogorov-Smirnov test, the Anderson Darling test and the Cramer-von Mises test. Table 4 summarizes the results of three common goodness-of-fit tests. All the tests indicate a good fit to the simulated data.

Table 4: Goodness-of-fit tests for fitted Erlang mixture to Pareto data

Test	Statistic	p-value	Accepted at 5% significant level
K-S	0.0196	0.4287	Yes
A-D	0.4917	0.7553	Yes
Cv-M	0.1044	0.5639	Yes

To test the efficiency of the proposed algorithm, finally we run both EM algorithms. The results show that the run time of our method is significantly shorter (Table 5).

Table 5: Comparison of BICs and run times

methods	Method I	Method II	Kernel
components	7	7	2000
log-likelihood	-7186.021	-7222.83	-7389.431
BIC	14486.167	14559.67	14778.86
run time	3.31312 secs	182.61384 secs	—

Next, we generate 2000 right random censoring points denoted as $(y_1, \cdots, y_{2000})$, from another Type II Pareto distribution with shape parameter 0.1 and scale parameter 15. For $v = 1, \cdots, 2000$, data point x_v is censored if $x_v > y_v$ and in this case $l_v = y_v$. The resulting data set now contains 1875 uncensored data points and 125 right censored data points. We use an Erlang mixture to fit the censored data. The estimated parameters are given in Table 6.

Table 6: Parameter estimates of fitted 7-component Erlang mixture

u	α_u	m_u	θ
1	0.7848	1	5.6715
2	0.1503	5	
3	0.0409	12	
4	0.0142	22	
5	0.0050	41	
6	0.0027	72	
7	0.0020	141	

From Table 6, we observe that the parameters of the fitted model are closed to the results for the uncensored data. Furthermore, Figure 4 shows the PP-plot and QQ-plot of the fitted model. As expected, although the model cannot fit the tail of the data as well as the complete data, the results in Table 7 still indicate good fitness to the original generated data.

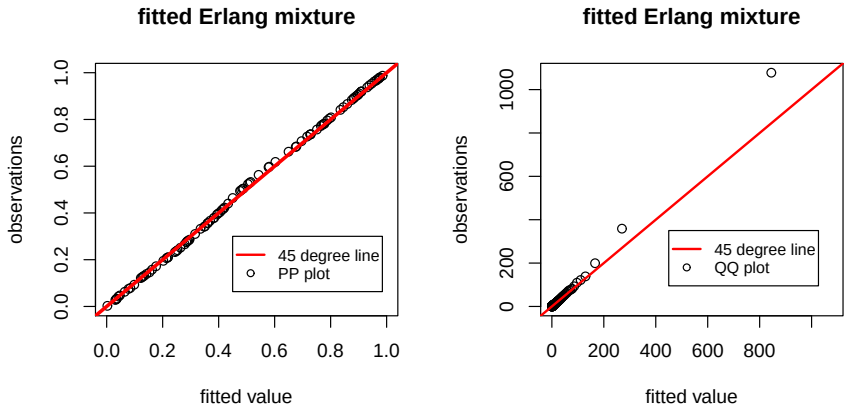

Figure 4: PP-plot and QQ-plot for fitted model

Table 7 summarizes the results of three common goodness-of-fit tests. All the tests do not reject the results at the 5% significant level.

Table 7: Tests for fitness of Erlang mixture to censored data

Test	Statistic	p-value	Accepted at 5% significant level
K-S	0.0200	0.4052	Yes
A-D	1.1571	0.2845	Yes
Cv-M	0.1998	0.2679	Yes

7 Real data applications

In this section, we consider applications to two real insurance loss data sets.

7.1 Danish fire data

We consider a Danish data set that contains 2167 fire losses from 1980 to 1990. Only losses exceeding 1 million Danish Krone were recorded. The data have been adjusted for inflation to reflect the 1985 values and are expressed in millions of Danish Krone. This data set has been widely studied (e.g. McNeil (1997), Drees and Müller (2008) and Embrechts et al. (2013)).

Using 1 million Danish Krone as the monetary unit, the data are left truncated with truncation point $t^l = 1$. The fitting procedure with the 10-fold cross-validation leads to a 5-component Erlang mixture with the estimated parameters given in Table 8.

Table 8: Estimated parameters of fitted 5-component Erlang mixture

u	α_u	m_u	θ
1	0.9467	1	1.03693
2	0.0369	6	
3	0.0138	17	
4	0.0020	44	
5	0.0006	174	

In the left panel of Figure 5, we compare the truncated data with the fitted truncated density on the logarithmic scale. In the right panel of Figure 5 the fitted truncated survival curve and the Kaplan-Meier curve are given. It is clear that both curves are very close to each other.

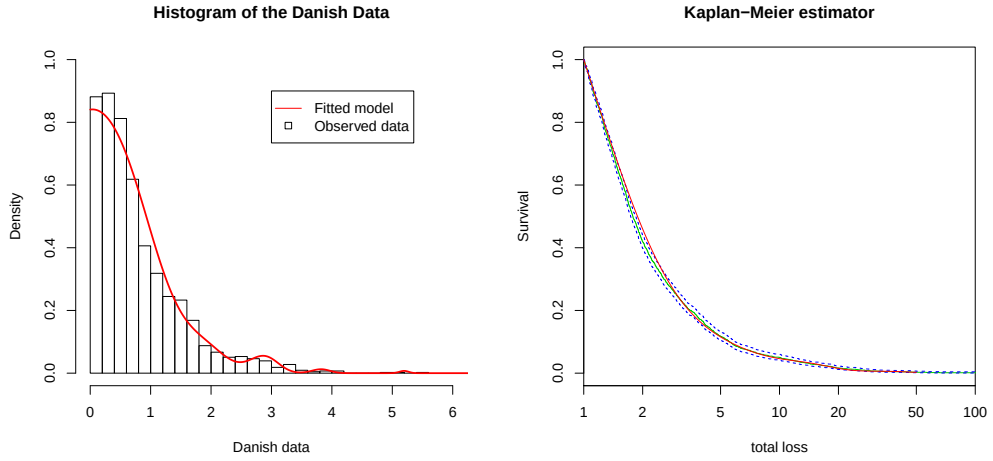

Figure 5: Left panel: histogram and fitted density on logarithmic scale; Right panel: empirical and fitted survival curves.

We remark that McNeil (1997) used the generalized Pareto distribution (GPD) to fit the same data set and tested the tail fitness by considering policy payouts with lower attachment ($r = 50$) and upper attachment ($R = 200$) that will be described in (7.1). The parameter estimates of the GPD model for this data set are $\xi = 0.61, \mu = 1, \sigma = 0.93$ (see McNeil (1997)). They showed that the GPD is superior to some traditional parametric models such as the truncated log-normal and the ordinary Pareto distribution in terms of tail fitting by examining the expected payouts with high attachment points and low attachment ratios. Their approach also indicates indirectly that the GPD fits the data well in the tail. However, the GPD does not fit the body of the data well as evidenced by poor approximation for low attachment ratios, as shown in Table 11. In following, we will show that the Erlang mixture model can fit both the body and the tail of the data well when comparing with the GPD by (a) calculating VaR and TVaR for the empirical distribution, the fitted Erlang mixture and the GPD at a wide range of confidence levels; and (b) calculating the expected payouts, given different lower and upper attachment points.

7.1.1 VaR and TVaR

We calculate VaR and TVaR using the three models and the corresponding nonparametric results are used as a benchmark. We show in Table 9 and Table 10 that both the Erlang mixture and the GPD can fit the data well for high confidence levels. As expected, the Erlang mixture produces more accurate estimates at relatively low confidence levels while the GPD more accurate at high confidence levels. The real advantage of the use of an Erlang mixture is that it can capture not only the body but also the tail of the data as shown in Subsection 7.1.2, while the GPD model fails to do so.

Table 9: Comparison of VaRs from different models

Confidence Level	Empirical	Erlang	GPD
80.0%	3.4782	3.4329	3.5448
85.0%	4.3254	4.3241	4.2595
90.0%	5.5415	5.4249	5.6863
95.0%	9.9726	9.7347	8.9548
97.5%	16.268	16.644	13.943
99.0%	26.043	22.852	24.777
99.5%	36.824	40.607	38.093
99.95%	151.77	164.98	156.79

Table 10: Comparison of TVaRs from different models

Confidence Level	Empirical	Erlang	GPD
80.0%	9.9613	9.7456	8.7008
85.0%	12.001	11.734	10.645
90.0%	15.565	15.214	14.065
95.0%	24.082	23.445	22.351
97.5%	35.472	34.176	35.076
99.0%	58.586	56.502	62.796
99.5%	87.591	84.866	96.910
99.95%	207.83	194.62	201.23

7.1.2 Expected payouts

Let X be an insurance loss and a policy on X has a loss layer with lower and upper attachment points r and R, respectively. The payout Z on X is then given by

$$Z = \begin{cases} 0, & 0 < X < r, \\ X - r, & r \leq X < R, \\ R - r, & X \geq R. \end{cases} \tag{7.1}$$

For this policy modification, we may calculate the expected payout $E[Z|X > t^l]$. Denote the distribution function of the left truncated losses as $F_{X^l}(x) = P(X \leq x|X > t^l)$ and the corresponding density function

$f_{X^l}(x)$. Then, for payout layer (r, R) we have

$$E[Z|X > t^l] = \int_r^R (x - r) f_{X^l}(x) dx + (R - r)(1 - F_{X^l}(R)). \tag{7.2}$$

The results from these three different models are presented in Table 11. Using the non-parametric estimates of $E[Z|X > t^l]$ as a benchmark, it is shown that the Erlang mixture provides better approximations, which indicates the Erlang mixture fits both the body and the tail of the data. Again as expected, the GPD model provides good approximations when the lower attachment point r is much larger than the truncation point t^l, but it is not the case for small values of r.

Table 11: Expected policy payout from different models

R	r/R	Empirical	Erlang mixture	GPD
20	0.00	1.976	2.104	1.288
	0.25	0.654	0.658	0.502
	0.50	0.299	0.342	0.215
	0.75	0.112	0.129	0.081
	0.95	0.017	0.018	0.018
30	0.00	2.088	2.175	1.383
	0.25	0.552	0.544	0.421
	0.50	0.224	0.200	0.176
	0.75	0.076	0.067	0.065
	0.95	0.011	0.012	0.010
50	0.00	2.182	2.205	1.476
	0.25	0.398	0.329	0.325
	0.50	0.139	0.129	0.133
	0.75	0.051	0.047	0.049
	0.95	0.008	0.006	0.009
100	0.00	2.265	2.274	1.565
	0.25	0.221	0.218	0.221
	0.50	0.083	0.089	0.088
	0.75	0.035	0.034	0.032
	0.95	0.007	0.007	0.007
200	0.00	2.356	2.384	1.622
	0.25	0.174	0.190	0.146
	0.50	0.091	0.079	0.058
	0.75	0.024	0.019	0.021
	0.95	0.005	0.002	0.003

7.2 Loss and ALAE insurance data

In this subsection, we consider an insurance data set from the US Insurance Services Office (ISO) that comprises of 1500 non-life insurance claims of which both the indemnity payment or loss as well as the allocated loss adjustment expense (ALAE) are recorded. For each claim, the policy limit, i.e., the maximal claim amount, of the contract is also recorded. This data set has been studied in Frees and Valdez (1998),

Klugman and Parsa (1999) and Verbelen et al. (2016).

We first consider fitting to the loss part of the data. Certain features of the data indicate that it is long tailed. For example the maximum value of the data is 59 times the mean; 20.7% of the data points are categorized as outliers if the 1.5 IQR rule is used. See the box-plot in Figure 6.

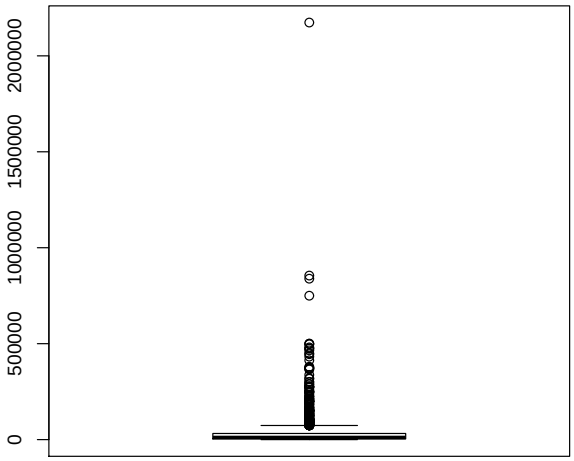

Figure 6: Box-plot of loss data

Also note that the loss data are right censored. Although only 34 of the 1500 losses are right censored, censoring should not be ignored when fitting the model to the data. Using our algorithm, a 7-component Erlang mixture is selected with the parameter values given in Table 12.

Table 12: Estimated parameters of fitted 7-component Erlang mixture

u	α_u	m_u	θ
1	0.7036	1	9463.258
2	0.1755	5	
3	0.0725	12	
4	0.0308	27	
5	0.0136	49	
6	0.0033	96	
7	0.0007	230	

To check the fitness of the fitted model to data, we compare the the fitted survival curve with the Kaplan-Meier survival curve, as shown in the left panel of Figure 7. It is clear that the two curves are very close to each other, which indicates a good fit.

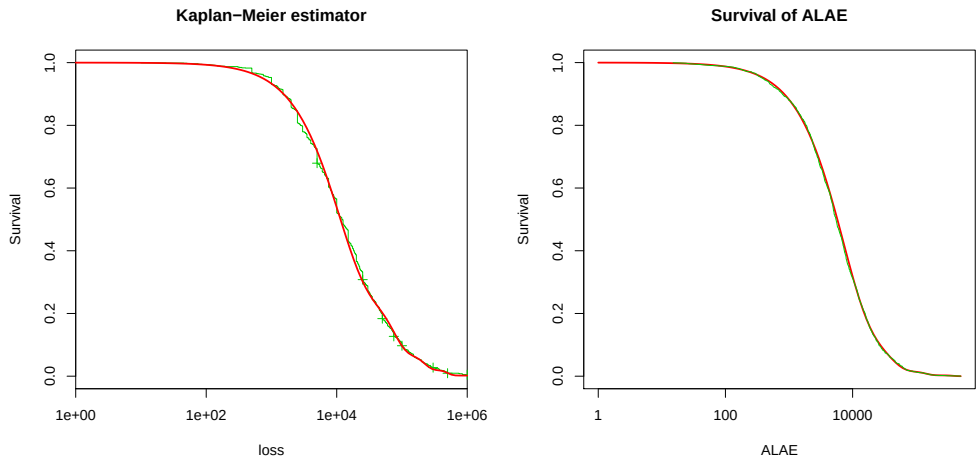

Figure 7: Empirical and fitted survival curves

We now use another Erlang mixture to fit the ALAE part of the data. The ALAE data are complete. A 4-component Erlang mixture is selected to fit the data and Table 13 shows the estimated parameters.

Table 13: Estimated parameters of fitted 4-component Erlang mixture

u	α_u	m_u	θ
1	0.9064	1	7177.032
2	0.0793	6	
3	0.0117	20	
4	0.00003	53	

The right panel in Figure 7 shows the fitted survival curve and the empirical survival curve, which visually confirms that the model fits the ALAE data well. The three common goodness-of-fit tests presented in Table 14 show the fited model is not rejected at the 5% significant level.

Table 14: Goodness-of-fit tests for ALAE data

Test	Statistic	p-value	Accepted at 5% significant level
K-S	0.014	0.932	Yes
A-D	0.3303	0.9138	Yes
Cv-M	0.0222	0.9945	Yes

Again we can easily calculate the VaR and TVaR using the fitted Erlang mixture. The VaR and TVaR for both loss data and ALAE data at different confidence levels are given in Table 15.

Table 15: VaRs and TVaRs for Loss data and ALAE data

Confidence Level	Loss data		ALAE data	
	VaR	TVaR	VaR	TVaR
80.0%	48294	158691	15259	42438
85.0%	67431	192464	19377	50869
90.0%	94375	248753	27101	64919
95.0%	186029	372979	43978	95399
97.5%	280790	513761	62638	138926
99.0%	532897	723280	130817	218914
99.5%	612838	874711	169001	289372
99.95%	2056012	2245767	425133	457493

8 Concluding remarks

In this paper, we develop a GEM-CMM algorithm to fit the Erlang mixture model to truncated and censored loss data. The purpose of this paper is to address two critical issues when using an EM algorithm to fit the Erlang mixture model to loss data: obtaining the optimal or suboptimal values for the model shape parameters and seeking quality initial estimates. A GEM algorithm with is developed for the former and a CMM method is proposed to obtain quality initial estimates. Further improvement is achieved by adjusting the estimated parameters to match first two moments of the model with corresponding sample moments. We test the efficiency through several simulation studies and two real data applications. The results show that indeed our approach is capable to obtain quality initial values, which leads to fewer iterations to find the estimates. The run time is greatly reduced when comparing with the methods proposed in other papers such as Lee and Lin (2010) and Verbelen et al. (2015).

We intend to extend the approach proposed in this paper to multivariate mixture models. Quality initial estimates would be even more critical in a high dimensional setting due to the sparseness of multivariate data. Further, the shape parameters of the component distributions in a multivariate mixture model are often scarcely located and the local search method in this paper might have advantage to locate their optimal or suboptimal values. With the efficiency of the proposed algorithm, we also expect the run time would be more significantly reduced.

Acknowledgments

This research was partly supported by grants from the Natural Sciences and Engineering Research Council of Canada (RGPIN-2017-06684) and the Natural Science Foundation of China (No. 11471272). W. Gui wishes to thank the Graduate School of the Xiamen University for its financial support during his PhD studies.

A Summary of the GEM-CMM Algorithm

Algorithm GEM-CMM algorithm

1. Let $M = 2$ and CV$=-\infty$
2. **do**

{**Initial step**}

(a) Apply the K-means algorithm to cluster the data into M groups and assign values to the latent variables $\mathbf{z}_1, \cdots, \mathbf{z}_n$ according to (2.6).

(b) Compute mixing weights, $\widehat{\alpha}_u = \sum_{v=1}^{n} z_{vu}/n = n_u/n, u = 1, 2, \cdots, M$

(c) Compute the mean parameters, $\widehat{\mu}_u = \sum_{v=1}^{n} x_v z_{vu}/n_u, u = 1, 2, \cdots, M$

(d) Compute the common scale parameter, $\hat{\theta}^* = \min\{(\overline{x^2} - \sum_{u=1}^{M} \widehat{\alpha}_u \widehat{\mu}_u^2)/\overline{x}, \widehat{\mu}_1, \cdots, \widehat{\mu}_M\}$

(e) Compute the shape parameters, $\hat{m}_u = \lceil \widehat{\mu}_u/\hat{\theta}^* \rceil, u = 1, 2, \cdots, M$

(f) Transform the initial mixing weights according to (4.14)

{**EM algorithm**}

while log-likelihood improves **do**

{E-Step}

Compute the conditional expectation as in (3.1) and the posterior probability as in (2.7)

{M-step}

(a) Update the mixing weights as in (2.9)

(b) Compute the conditional expectation $Q^*(\mathbf{m})$ as in(3.3)

do

(b1) Derive the increment of shape parameters $\Delta\mathbf{m}$ as in (3.5)

(b2) Update the shape parameters by $\mathbf{m} \leftarrow \mathbf{m} + \Delta\mathbf{m}$

until no shape parameters change any more

(c) Update the scale parameter as in (2.10)

end while

Transform weights β to α using (3.6)

Compute log-likelihood and CV for Φ

$M \leftarrow M + 1$

until the CV doesn't improve any more

3. Output the number of components M and $\Phi = \{\alpha_u, m_u, \theta\}, u = 1, \cdots, M$

References

Bishop, C.M. (2006). *Pattern Recognition and Machine Learning*. Springer, 2006.

Chen, J., Li, P., & 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., & 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., Côté, M.P., Marceau, E., & 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.

Drees, H., & Müller, P. (2008). Fitting and validation of a bivariate model for large claims. *Insurance: Mathematics and Economics*, 42(2), 638-650.

Dufresne, D. (2007). Fitting combinations of exponentials to probability distributions. *Applied Stochastic Models in Business and Industry*, 23(1), 23-48.

Embrechts, P., Klüppelberg, C., & Mikosch, T. (2013). *Modelling Extremal Events: for Insurance and Finance*. Springer, New York.

Feldmann, A., & Whitt, W. (1998). Fitting mixtures of exponentials to long-tail distributions to analyze network performance models. *Performance Evaluation*, 31, 245-279.

Frees, E.W., & Valdez, E.A. (1998). Understanding relationships using copulas. *North American Actuarial Journal*, 2(1), 1-25.

Givens, G.H., & Hoeting, J.A. (2013). *Computational Statistics*. Wiley, Hoboken, NJ.

Hamerly, G., & Elkan, C. (2002). Alternatives to the k-means algorithm that find better clusterings. *In Proceedings of the Eleventh International Conference on Information and Knowledge Management*, 600-607.

Hashorva, E., & Ratovomirija, G. (2015). On Sarmanov mixed Erlang risks in insurance applications. *Astin Bulletin*, 45(1), 175-205.

Kasahara, H., & Shimotsu, K. (2015). Testing the number of components in normal mixture regression models. *Journal of the American Statistical Association*, 110(512), 1632-1645.

Keatinge, C.L. (1999). Modeling losses with the mixed exponential distribution. *Proceedings of the Casualty Actuarial Society*, 86, 654-698.

Klugman, S. A., & Parsa, R. (1999). Fitting bivariate loss distributions with copulas. *Insurance: Mathematics and Economics*, 24, 139-148.

Klugman, S., & Rioux, J. (2006). Toward a unified approach to fitting loss models. *North American Actuarial Journal*, 10(1), 63-83.

Lee, S.C., & 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., & Lin, X.S. (2012). Modeling dependent risks with multivariate Erlang mixtures. *Astin Bulletin*, 42(1), 153-180.

MacKay, D. J. C. (2003). *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, Cambridge, UK.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1(14), 281-297.

McLachlan, G., & Krishnan, T. (2007). *The EM Algorithm and Extensions*. John Wiley & Sons, New York.

McNeil, A.J. (1997). Estimating the tails of loss severity distributions using extreme value theory. *Astin Bulletin*, 27(1), 117-137.

Miljkovic, T., & Grün, B. (2016). Modeling loss data using mixtures of distributions. *Insurance Mathematics & Economics*, 70, 387-396.

Rosenblatt, M. (1956). Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27, 832-837.

Seo, B., & Kim, D. (2012). Root selection in normal mixture models. *Computational Statistics and Data Analysis*, 56(8), 2454-2470.

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*. CRC press.

Tijms, H. C. (1994). *Stochastic Models: an Algorithmic Approach*. John Wiley & Sons, Chichester, England.

Verbelen, R., Gong, L., Antonio, K., Badescu, A., & Lin, X.S. (2015). Fitting mixtures of Erlangs to censored and truncated data using the EM algorithm. *Astin Bulletin*, 45(3), 729-758.

Verbelen, R., Antonio, K., & Claeskens, G. (2016). Multivariate mixtures of Erlangs for density estimation under censoring. *Lifetime Data Analysis*, 22(3), 429-455.

Willmot, G.E., & Lin, X.S. (2011). Risk modelling with the mixed Erlang distribution. *Applied Stochastic Models in Business and Industry*, 27(1), 2-16.

Willmot, G.E., & Woo, J.K. (2015). On some properties of a class of multivariate Erlang mixtures with insurance applications. *Astin Bulletin*, 45(1), 151-173.

Yao, W. (2010). A profile likelihood method for normal mixture with unequal variance. *Journal of Statistical Planning and Inference*, 140(7), 2089-2098.

Yin, C., & Lin, X.S. (2016). Efficient estimation of Erlang mixtures using iSCAD penalty with insurance application. *Astin Bulletin*, 46(3), 779-799.