Journal of Computational and Applied Mathematics 386 (2021) 113216

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

Fitting multivariate Erlang mixtures to data: A roughness penalty approach

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

[a] College of Computer Science and Artificial Intelligence, Wenzhou University, Wenzhou, 325035, China
[b] School of Mathematical Sciences, Xiamen University, Xiamen, China
[c] Department of Statistical Sciences, University of Toronto, Toronto, M5G 1X6, Canada

ARTICLE INFO

Article history:
Received 14 May 2020
Received in revised form 19 September 2020

Keywords:
Multivariate Erlang mixtures
Truncated and censored data
GECM algorithm
Roughness penalty

ABSTRACT

The class of multivariate Erlang mixtures with common scale parameter has many desirable properties and has widely been used in insurance loss modeling. The parameters of a multivariate Erlang mixture are normally estimated using an expectation–maximization (EM) algorithm as shown in Lee and Lin (2012) and Verbelen et al. (2016). However, when fitting the mixture to data of high dimension, the fitted density surface is often not smooth (with deep peaks and valleys) and the tail fitting may also be rather unsatisfactory. In this paper, we propose a generalized expectation conditional maximization (GECM) algorithm that maximizes a penalized likelihood with a proposed roughness penalty. The roughness penalty is based on integrated squared second derivative of the density function of aggregate data, which is used in functional data analysis. We illustrate the performance of the proposed method through some numerical experiments and real data applications.

1. Introduction

Mixture distributions have been widely used in data classification and segmentation applications, and in modeling data that exhibit clustering behavior. The most common mixture model in statistics is the Gaussian mixture. See [1,2] and references therein. However, in many practical applications and in particular insurance and economic applications, data are mostly positive. See [3,4] and examples in these papers. In those situations a Gaussian mixture is often not a suitable model to fit the data and we may want to consider positive mixtures. One class of positive mixtures that has recently attracted much attention in insurance modeling is the class of Erlang mixtures with common scale parameter. See [3,5]. Their applications can be found in [6–13] and references therein.

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

$$g(\mathbf{x}|\Phi) = \sum_{u=1}^{M} \alpha_u \prod_{j=1}^{k} f(x_j|m_{uj}, \theta), \ x_j > 0, j = 1, \ldots, k, \tag{1.1}$$

where

$$f(x|m, \theta) = \frac{x^{m-1} e^{-x/\theta}}{\theta^m (m-1)!} \tag{1.2}$$

* Corresponding author.
E-mail addresses: 20180171@wzu.edu.cn (W. Gui), rthuang@xmu.edu.cn (R. Huang), sheldon@utstat.utoronto.ca (X.S. Lin).

https://doi.org/10.1016/j.cam.2020.113216
0377-0427/

is the Erlang density with scale parameter $\theta > 0$ and integer shape parameter m. The parameters of the mixture are $\Phi = \{\theta, \alpha_u, m_{uj}, u = 1, \ldots, M; j = 1, \ldots, k\}$. Note that the distribution function of $f(x|m, \theta)$ can be explicitly written as

$$F(x|m, \theta) = 1 - e^{-x/\theta} \sum_{n=0}^{m-1} \frac{x^n}{\theta^n n!}. \tag{1.3}$$

Lee and Lin [5] showed that this class of multivariate Erlang mixtures is dense in the space of positive continuous multivariate distributions in the sense of weak convergence. Hence, theoretically any positive data can be fitted by such an Erlang mixture to any accuracy. They also showed that the mixture has many desirable properties: there are explicit expressions for many distributional quantities such as the moments and marginal distributions; any dependent structure can be modeled and dependence measures such as the Kendall's tau and Spearman's rho also have explicit expressions. The availability of explicit expressions is due to the use of an integer shape parameter in the Erlang distribution instead of a real shape parameter in a gamma distribution.

An expectation–maximization (EM) algorithm was presented in [5], to estimate the parameters of the multivariate Erlang mixture. Verbelen et al. [9] extended the EM algorithm to handle truncated and censored data. However, there are two main issues especially when fitting the mixture to data of high dimension: the fitted density surface is often not smooth (with deep peaks and valleys) and the tail fitting is rather unsatisfactory. The roughness is often due to the need to capture the heaviness of the tail with a high number of Erlang components. In this paper, we propose a generalized expectation conditional maximization (GECM) algorithm that maximizes a penalized likelihood with a roughness penalty that is the integrated squared second derivative of the density function of aggregate data. This penalty function is widely used in univariate functional data analysis. See [14]. Due to the specific form of the Erlang density and its closeness in convolution, the penalty function is applicable and works well in our multivariate case, as shown in Section 2.2. In this paper, we conduct several simulations studies and apply the algorithm to real data sets. Our simulation studies and real data applications show that the algorithm is able to accurately fit the multivariate Erlang mixture with a relatively small number of components to data very well and overcomes the aforementioned drawbacks of the existing algorithms.

This paper is organized as follows. In Section 2, we introduce the roughness penalty in the context of the multivariate Erlang mixture and present the GECM algorithm for truncated and censored data. The algorithm may be viewed as an extension of the GEM-CMM (generalized EM algorithm along with clusterized method of moments) algorithm in [11] that estimates the parameters of the uni-variate Erlang mixture. In Section 3, a moment-matching based initialization strategy for the GECM algorithm is provided. In Section 4, we test the efficiency of the GECM algorithm through several simulation studies with different data characteristics. In Section 5, we fit the multivariate Erlang mixture to several real data sets and the results show that the model can fit data of different types well. Moreover, we compare the fitness of the fitted model with and without roughness penalty and show that the roughness penalty plays an important role in the improvement of the fitting. We conclude in Section 6 with some closing remarks.

2. A GECM algorithm for parameter estimation

In this section, we propose a GECM algorithm to fit the multivariate Erlang mixture to truncated and censored data. Many real data sets are of this type including left truncated and right censored insurance claims where left truncation represents policy deductible and right censoring is interpreted as policy limit/maximum covered loss. This proposed algorithm is different from the EM algorithms in [5,9] in the way that the algorithm enables us to estimate not only the mixing weights and the scale parameter but also the shape parameters, which results in a smooth fitted density and a better fit to the tail of the data.

2.1. Data type and notation

Using similar notation to that in [9], we assume that the truncation range of a data set, denoted as $[\mathbf{t}^l, \mathbf{t}^r]$ where $\mathbf{t}^l = (t_1^l, \ldots, t_k^l)$ is the left truncation point and $\mathbf{t}^r = (t_1^r, \ldots, t_k^r)$ is the right truncation point, is the same for all the data points. For each multivariate data point, let $\mathbf{x}_v = (x_{v1}, \ldots, x_{vk})$, $v = 1, \ldots, n$, be the true value, $\mathbf{c}_v^l = (c_{v1}^l, \ldots, c_{vk}^l)$ be the left censoring point and $\mathbf{c}_v^r = (c_{v1}^r, \ldots, c_{vk}^r)$ the right censoring point. The censoring status for the jth dimension of the vth observation is determined as follows,

$$\begin{aligned}
&\text{uncensored:} && t_j^l \le c_{vj}^l = x_{vj} = c_{vj}^r \le t_j^r,\\
&\text{left censored:} && t_j^l < x_{vj} < c_{vj}^l < c_{vj}^r \le t_j^r,\\
&\text{right censored:} && t_j^l \le c_{vj}^l < c_{vj}^r < x_{vj} < t_j^r,\\
&\text{interval censored:} && t_j^l \le c_{vj}^l < x_{vj} < c_{vj}^r \le t_j^r.
\end{aligned}$$

We remark that strictly speaking, for uncensored data points, the true values are not equal to the left and right censoring points. We make such a modification to distinguish the uncensored data from interval censored data because the censoring points for uncensored data will not be used in our derivations.

2.2. Roughness penalty function

We now introduce a roughness penalty function that penalizes the log-likelihood of the data. As described in [14], one way to smooth a function's roughness is to apply the integrated squared second derivative as a penalty on an objective function in an optimization scheme.

By adapting the idea, we introduce a roughness penalty function on the distribution of the aggregate $S = X_1 + \cdots + X_k$ of random sample point $(X_1, \ldots, X_k)$. Due to Property 5.1 in [5], if the joint distribution of $(X_1, \ldots, X_k)$ has a multivariate Erlang mixture with density (1.1), then the aggregate random variable S has a univariate Erlang mixture with density

$$f_S(x) = \sum_{u=1}^{M} \alpha_u f(x|m_{u1} + \cdots + m_{uk}, \theta). \tag{2.1}$$

As a result, the integrated squared second derivative, as the roughness penalty function, of S is given by

$$PEN_2 = \int_0^\infty [f_S''(x)]^2 dx = \int_0^\infty \left\{ \sum_{u=1}^{M} \alpha_u f''(x|m_{u1} + \cdots + m_{uk}, \theta) \right\}^2 dx. \tag{2.2}$$

It is easy to check that the second derivative of Erlang density $f(x|m, \theta)$ may be written as

$$f''(x|m, \theta) = \frac{1}{\theta^2}[f(x|m-2, \theta) - 2f(x|m-1, \theta) + f(x|m, \theta)] \tag{2.3}$$

with $f(x|m-i, \theta) = 0$ if $m \le i$, $i = 1, 2$. Moreover, for any pair of integers m_1 and m_2,

$$\int_0^\infty f(x|m_1, \theta) f(x|m_2, \theta) dx = \frac{1}{\theta} \frac{(m_1 + m_2 - 2)!}{(m_1 - 1)!(m_2 - 1)!} \frac{1}{2^{m_1+m_2-1}}. \tag{2.4}$$

Hence, the roughness penalty function may be re-expressed as

$$\begin{aligned} PEN_2 &= \frac{1}{\theta^4} \int_0^\infty \left\{ \sum_{u=1}^{M} \alpha_u [f(x|\widetilde{m}_u - 2, \theta) - 2f(x|\widetilde{m}_u - 1, \theta) + f(x|\widetilde{m}_u, \theta)] \right\}^2 dx \\ &= \frac{1}{\theta^5} \boldsymbol{\alpha}' \boldsymbol{P} \boldsymbol{\alpha}, \end{aligned} \tag{2.5}$$

where $\widetilde{m}_u = m_{u1} + \cdots + m_{uk}$, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_M)$, $\boldsymbol{P}$ is an $M \times M$ matrix with the (u, w)th entry:

$$p_{uw} = \sum_{i=0}^{2} \sum_{j=0}^{2} (-1)^{i+j} \binom{2}{i} \binom{2}{j} \frac{(\widetilde{m}_u + \widetilde{m}_w - i - j - 2)!}{(\widetilde{m}_u - i - 1)!(\widetilde{m}_w - j - 1)!} \frac{1}{2^{\widetilde{m}_u + \widetilde{m}_w - i - j - 1}}. \tag{2.6}$$

Taken truncation into consideration, it is easy to check that the roughness penalty is adjusted as

$$PEN_2^* = \frac{1}{\theta^5} \boldsymbol{\beta}' \boldsymbol{P}^* \boldsymbol{\beta}, \tag{2.7}$$

where $\boldsymbol{P}^*$ is an $M \times M$ matrix with the (u, w)th entry

$$p_{uw}^* = p_{uw} \frac{\left[\sum_{u=1}^{M} \alpha_u \prod_{j=1}^{k} F(t_j^r|m_{uj}, \theta) - F(t_j^l|m_{uj}, \theta) \right]^2}{\alpha_u \alpha_w \prod_{j=1}^{k} \left[F(t_j^r|m_{uj}, \theta) - F(t_j^l|m_{uj}, \theta) \right] \prod_{j=1}^{k} \left[F(t_j^r|m_{wj}, \theta) - F(t_j^l|m_{wj}, \theta) \right]},$$

and the mixing weights $\boldsymbol{\beta}$ are defined according to (2.14) below.

We may use the rth order derivative as the roughness penalty and in this case the roughness penalty without considering truncation is given by

$$PEN_r = \frac{1}{\theta^{2r+1}} \boldsymbol{\alpha}' \boldsymbol{P} \boldsymbol{\alpha}, \tag{2.8}$$

where the (u, w)th entry of the matrix $\boldsymbol{P}$ is given by

$$p_{uw} = \sum_{i=0}^{r}\sum_{j=0}^{r}(-1)^{i+j}\begin{pmatrix} r \\ i \end{pmatrix}\begin{pmatrix} r \\ j \end{pmatrix}\frac{(\widetilde{m}_u + \widetilde{m}_w - i - j - 2)!}{(\widetilde{m}_u - i - 1)!(\widetilde{m}_w - j - 1)!}\frac{1}{2^{\widetilde{m}_u+\widetilde{m}_w-i-j-1}}. \tag{2.9}$$

We remark that the above explicit formulas uniquely hold for the multivariate Erlang mixture, which is one of the main reasons that the integrated squared second derivative penalty is chosen.

2.3. The GECM algorithm

In this subsection, we propose a GECM algorithm to maximize the penalized log-likelihood with roughness penalty (2.7).

The log-likelihood of a censored and truncated sample as in Section 2.1 is given by

$$\begin{aligned} l(\Phi) = &\sum_{v=1}^{n}\ln\left\{\sum_{u=1}^{M}\alpha_u\prod_{j=1}^{k}\widetilde{f}(x_{vj}|m_{uj},\theta)\right\} \\ &- \sum_{i=1}^{n}\ln\left\{\sum_{u=1}^{M}\alpha_u\prod_{j=1}^{k}(F(t_j^r|m_{uj},\theta) - F(t_j^l|m_{uj},\theta))\right\}, \end{aligned} \tag{2.10}$$

where

$$\widetilde{f}(x_{vj}|m_{uj},\theta) = \begin{cases} f(x_{vj}|m_{uj},\theta), & t_j^l \le c_{vj}^l = x_{vj} = c_{vj}^r \le t_j^r, \\ F(c_{vj}^l|m_{uj},\theta) - F(t_j^l|m_{uj},\theta), & t_j^l < x_{vj} < c_{vj}^l < c_{vj}^r \le t_j^r, \\ F(t_j^r|m_{uj},\theta) - F(c_{vj}^r|m_{uj},\theta), & t_j^l \le c_{vj}^l < c_{vj}^r < x_{vj} < t_j^r, \\ F(c_{vj}^r|m_{uj},\theta) - F(c_{vj}^l|m_{uj},\theta), & t_j^l \le c_{vj}^l < x_{vj} < c_{vj}^r \le t_j^r. \end{cases} \tag{2.11}$$

A random variable comes from the M-component Erlang mixture means that the variable is selected by chance from the given M Erlang distributions (components) according to given probabilities (mixing weights). As usual, introduce latent random vectors $\mathbf{Z} = (\mathbf{Z}_1, \mathbf{Z}_2, \ldots, \mathbf{Z}_n)$, where $\mathbf{Z}_v = (Z_{v1}, \ldots, Z_{vM})$, $v = 1, 2, \ldots, n$ with

$$Z_{vu} = \begin{cases} 1, & \text{observation } \mathbf{X}_v \text{ comes from the } u\text{th component}, \\ 0, & \text{others}. \end{cases} \tag{2.12}$$

The log-likelihood of the (complete) sample is

$$l(\Phi|\mathbf{z}) = \sum_{v=1}^{n}\sum_{u=1}^{M} z_{vu}\ln\left(\beta_u f(\mathbf{x}_v|\mathbf{t}^l,\mathbf{t}^r,\mathbf{m}_u,\theta)\right), \tag{2.13}$$

where

$$\beta_u = \frac{\alpha_u\prod_{j=1}^{k}\left[F(t_j^r|m_{uj},\theta) - F(t_j^l|m_{uj},\theta)\right]}{\sum_{w=1}^{M}\alpha_w\prod_{j=1}^{k}\left[F(t_j^r|m_{wj},\theta) - F(t_j^l|m_{wj},\theta)\right]}, \tag{2.14}$$

and

$$f(\mathbf{x}_v|\mathbf{t}^l,\mathbf{t}^r,\mathbf{m}_u,\theta) = \prod_{j=1}^{k}\frac{f(x_{vj}|m_{uj},\theta)}{F(t_j^r|m_{uj},\theta) - F(t_j^l|m_{uj},\theta)}. \tag{2.15}$$

The penalized log-likelihood is then given by

$$l_p(\Phi|\mathbf{z}) = l(\Phi|\mathbf{z}) - \frac{\lambda}{\theta^5}\boldsymbol{\beta}'\boldsymbol{P}^*\boldsymbol{\beta}, \tag{2.16}$$

where λ is a tuning parameter.

In the following, we present the E-Step and M-Step of the GECM algorithm.

E-Step: Given that the estimated parameter values from the last iteration t are $\Phi^{(t)} = \{\beta_u^{(t)}, \mathbf{m}_u^{(t)}, \theta^{(t)}, u = 1, \ldots, M\}$, the expectation of the complete data log-likelihood is given by

$$\begin{aligned} Q(\Phi|\Phi^{(t)}) = &\sum_{v=1}^{n}\sum_{u=1}^{M}\sum_{j=1}^{k} z_{vu}\Bigg[(\ln\beta_u - \frac{1}{\theta}E(X_{vj}|Z_{vu}=1, x_{vj}, m_{uj}^{(t)}, \theta^{(t)}) \\ &- m_{uj}\ln\theta + (m_{uj}-1)E(\ln(X_{vj})|Z_{vu}=1, x_{vj}, m_{uj}^{(t)}, \theta^{(t)}) - \ln(m_{uj}-1)!)\Bigg] \\ &- \sum_{v=1}^{n}\sum_{u=1}^{M} z_{vu}\sum_{j=1}^{k}\ln(F(t_j^r|m_{uj},\theta) - F(t_j^l|m_{uj},\theta)) - \frac{\lambda}{\theta^5}\boldsymbol{\beta}'\boldsymbol{P}^*\boldsymbol{\beta}. \end{aligned} \tag{2.17}$$

Here, the posterior probability z_{vu} of the latent variable Z_{vu} as shown in (2.12) is given by

$$z_{vu} = \frac{\alpha_u^{(t)} p(\mathbf{x}_v|\mathbf{m}_u^{(t)}, \theta^{(t)})}{\sum_{w=1}^{M}\alpha_w^{(t)} p(\mathbf{x}_v|\mathbf{m}_w^{(t)}, \theta^{(t)})}, v = 1, \ldots, n, u = 1, \ldots, M, \tag{2.18}$$

where $p(\mathbf{x}_v|\mathbf{m}_u, \theta) = \prod_{j=1}^{k}\widetilde{f}(x_{vj}|m_{uj}, \theta)$.

In (2.17), the expected values of X_{vj} and $\ln(X_{vj})$ equal to x_{vj} and $\ln x_{vj}$ for uncensored data points but we need to compute the expected values of X_{vj} and $\ln(X_{vj})$ conditioning on the censoring and truncation points and the current parameters. For a left censored data point, we have

$$E(X_{vj}|Z_{vu}=1, x_{vj}, m_{uj}^{(t)}, \theta^{(t)}) = \frac{\theta^{(t)} m_{uj}^{(t)}(F(c_{vj}^l|m_{uj}^{(t)}+1, \theta^{(t)}) - F(t_j^l|m_{uj}^{(t)}+1, \theta^{(t)}))}{F(c_{vj}^l|m_{uj}^{(t)}, \theta^{(t)}) - F(t_j^l|m_{uj}^{(t)}, \theta^{(t)})}, \tag{2.19}$$

and

$$\begin{aligned} &E(\ln(X_{vj})|Z_{vu}=1, x_{vj}, m_{uj}^{(t)}, \theta^{(t)}) \\ =&\frac{(\ln t_j^l \overline{F}(t_j^l|m_{uj}^{(t)}, \theta^{(t)}) - \ln c_{vj}^l \overline{F}(c_{vj}^l|, m_{uj}^{(t)}, \theta^{(t)})) + \sum_{n=0}^{m_{uj}^{(t)}-1}\frac{1}{n}[F(c_{vj}^l|n, \theta^{(t)}) - F(t_j^l|n, \theta^{(t)})]}{F(c_{vj}^l|m_{uj}^{(t)}, \theta^{(t)}) - F(t_j^l|m_{uj}^{(t)}, \theta^{(t)})}, \end{aligned} \tag{2.20}$$

and similarly, the expectations for right and interval censored data points can be obtained as well.

M-Step: Update the parameters by maximization:

$$\Phi^{(t+1)} = \arg\max Q(\Phi|\Phi^{(t)}). \tag{2.21}$$

Instead of using the M-step in traditional EM algorithm it is computationally simpler to implement the following conditional maximization steps, i.e., the CM-steps and a local search method for estimating shape parameters in each step.

CM-step 1: The mixing weights are obtained by the following formula:

$$\beta_u^{(t+1)} = \frac{1}{N}\left\{\sum_{v=1}^{n} z_{vu} - \frac{2\theta^5}{\lambda}\beta_u^{(t)}\sum_{j=1}^{M}\beta_j^{(t)} p_{uj}^*\right\}, u = 1, \ldots, M, \tag{2.22}$$

where $N = n - \frac{2\theta^5}{\lambda}\boldsymbol{\beta}^T\boldsymbol{P}^*\boldsymbol{\beta}|_{\boldsymbol{\beta}=\boldsymbol{\beta}^{(t)}}$ and the entries in matrix $\boldsymbol{P}^*$ are calculated with previous shape parameters.

CM-step 2: The scale parameter is updated by solving the equation:

$$\begin{aligned} &n\theta^5\sum_{u=1}^{M}\sum_{j=1}^{k} m_{uj}^{(t)}\beta_u^{(t+1)} - \theta^4\sum_{v=1}^{n}\sum_{u=1}^{M}\sum_{j=1}^{k} z_{vu}E(X_{vj}|Z_{vu}=1, x_{vj}, m_{uj}^{(t)}, \theta^{(t)}) \\ &+ n\theta T - 5\lambda\boldsymbol{\beta}'P^*\boldsymbol{\beta} = 0, \end{aligned} \tag{2.23}$$

where

$$T = \sum_{v=1}^{n}\sum_{j=1}^{k}\sum_{m=1}^{M} z_{vu}\frac{(t_j^l)^{m_{uj}^{(t)}}e^{-t_j^l/\theta} - (t_j^r)^{m_{uj}^{(t)}}e^{-t_j^r/\theta}}{\theta^{m_{uj}^{(t)}-1}(m_{uj}^{(t)}-1)!(F(t_j^r|m_{uj}^{(t)}, \theta) - F(t_j^l|m_{uj}^{(t)}, \theta))}\Bigg|_{\theta=\theta^{(t)}}. \tag{2.24}$$

CM-step 3: Noting that the shape parameters are constrained to positive numbers, we adopt a local search method (see [11,15]) to find optimal shape parameters to maximize $Q(\Phi|\Phi^{(t)})$. We first replace the mixing weights and the common scale parameter in (2.17) with the new ones from the above steps and now the objective function Q is treated as a function of shape parameters only:

$$\begin{aligned} Q^*(\mathbf{m}) = &\sum_{v=1}^{n}\sum_{u=1}^{M}\sum_{j=1}^{k} z_{vu}\Bigg[(\ln\beta_u^{(t+1)} - \frac{1}{\theta^{(t+1)}}E(X_{vj}|Z_{vu}=1, x_{vj}, m_{uj}, \theta^{(t+1)}) \\ &- m_{uj}\ln\theta^{(t+1)} + (m_{uj}-1)E(\ln(X_{vj})|Z_{vu}=1, x_{vj}, m_{uj}, \theta^{(t+1)}) - \ln(m_{uj}-1)!)\Bigg] \\ &- \sum_{v=1}^{n}\sum_{u=1}^{M} z_{vu}\sum_{j=1}^{k}\ln(F(t_j^r|m_{uj}, \theta^{(t+1)}) - F(t_j^l|m_{uj}, \theta^{(t+1)})) - \lambda\frac{[\boldsymbol{\beta}^{(t+1)}]'\boldsymbol{P}^*\boldsymbol{\beta}^{(t+1)}}{[\theta^{(t+1)}]^5}, \end{aligned} \tag{2.25}$$

where $\mathbf{m} = (m_{11}, \ldots, m_{1k}, \ldots, m_{M1}, \ldots, m_{Mk})$.

To maximize $Q^*(\mathbf{m})$, we adopt a 3-optimal method, a common algorithm in the local search methodology. Denote

$$\delta_{uj}^+ = Q^*(\mathbf{m}+\mathbf{e}_{uj}) - Q^*(\mathbf{m})$$

and

$$\delta_{uj}^- = Q^*(\mathbf{m}-\mathbf{e}_{uj}) - Q^*(\mathbf{m}),$$

where $\mathbf{e}_{uj}$ is an $M\times k$-length vector with the $((u-1)\times k+j)$th entry equal to 1 and others 0. The shape parameters are adjusted by $\mathbf{m}^{[l]} = \mathbf{m}^{[l-1]} + \Delta\mathbf{m}$, where $\mathbf{m}^{[0]} = \mathbf{m}^{(t)}$, $\Delta\mathbf{m} = (\Delta m_{11}, \ldots, \Delta m_{1k}, \ldots, \Delta m_{M1}, \ldots, \Delta m_{Mk})$ with

$$\Delta m_{uj} = \begin{cases} 1, & \max\{\delta_{uj}^+, \delta_{uj}^-\} > 0,\ \delta_{uj}^+ > \delta_{uj}^-, \\ -1, & \max\{\delta_{uj}^+, \delta_{uj}^-\} > 0,\ \delta_{uj}^- > \delta_{uj}^+,\ m_{uj} > 1, \quad u = 1, \ldots, M, j = 1, \ldots, k. \\ 0, & \text{others}, \end{cases} \tag{2.26}$$

This process for searching new shape parameters repeats until the value of the parameters do not change anymore.

Finally, suppose the final estimates to be $\widehat{\Phi} = \{\widehat{\beta}_u, \widehat{m}_{uj}, \widehat{\theta}, u = 1, 2, \ldots, M, j = 1, \ldots, k\}$. Then the estimates of the original mixing weights are

$$\widehat{\alpha}_u = c\frac{\widehat{\beta}_u}{\prod_{j=1}^{k}[F(t_j^r|\widehat{m}_{uj}, \widehat{\theta}) - F(t_j^l|\widehat{m}_{uj}, \widehat{\theta})]}, u = 1, 2, \ldots, M, \tag{2.27}$$

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

2.4. Estimation of tuning parameter

Two quantities $l(\Phi|\mathbf{z})$ and $\frac{1}{\theta^5}\boldsymbol{\beta}'\boldsymbol{P}^*\boldsymbol{\beta}$ in (2.16) may be very different in scale. If $l(\Phi|\mathbf{z})$ is too large to dominate $\frac{\lambda}{\theta^5}\boldsymbol{\beta}'\boldsymbol{P}^*\boldsymbol{\beta}$, the effect of the roughness penalty is not significant. Reversely, the fitted model may be too smooth to produce a good fit to data. To estimate, we first calculate the values of $l(\Phi|\mathbf{z})$ and $\frac{1}{\theta^5}\boldsymbol{\beta}'\boldsymbol{P}^*\boldsymbol{\beta}$ using the non-penalized GECM algorithm and use them to obtain the initial range for the tuning parameter. As suggested in [14], the value of $|l(\Phi|\mathbf{z})|$ should not be more than 100 times of the value of $\frac{\lambda}{\theta^5}\boldsymbol{\beta}'\boldsymbol{P}^*\boldsymbol{\beta}$, which we will use to set a lower bound for the tuning parameter. Similarly, we set an upper bound for the tuning parameter so that $\frac{\lambda}{\theta^5}\boldsymbol{\beta}'\boldsymbol{P}^*\boldsymbol{\beta}$ is at most 100 times of the value of $|l(\Phi|\mathbf{z})|$. We then adopt the golden section search as described below (also see [16]) to find the optimal tuning parameter within the bounds. To make the range of the tuning parameter more manageable, we define a new tuning parameter ν using the transformation $\nu = \log_{10}\lambda$.

The cross-validation or CV method is used as a criterion to determine the tuning parameter. The basic idea behind cross-validation is to split the data into two groups: a *training set* to be fitted to the model and the remaining of the data the *validation set*, in order to see how well the model fits to the data that are not used to estimate the model. We adopt the 10-fold cross-validation approach. For a given data set D, we randomly partition it into 10 equally sized groups. 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.

Let the training set be denoted by D_T and the validation set by D_V. Assume the estimated parameters obtained from the training set D_T with a pre-estimated tuning parameter are $\widehat{\Phi} = \{\widehat{\alpha}_u, \widehat{m}_{uj}, \widehat{\theta}, u = 1, \ldots, M, j = 1, \ldots, k\}$. For any data

point $\mathbf{x}_v$ in the validation set, we introduce the score function

$$CV(\mathbf{x}_v; \nu) = \ln\left\{\sum_{u=1}^{M}\widehat{\alpha}_u\prod_{j=1}^{k}\widetilde{f}(x_{vj}|\widehat{m}_{uj},\widehat{\theta})\right\} - \ln\left\{\sum_{u=1}^{M}\widehat{\alpha}_u\prod_{j=1}^{k}(F(t_j^r|\widehat{m}_{uj},\widehat{\theta}) - F(t_j^l|\widehat{m}_{uj},\widehat{\theta}))\right\}, \tag{2.28}$$

where $\widetilde{f}(x_{vj}|\widehat{m}_{uj},\widehat{\theta})$ has the same form as in (2.11). The score function on the validation set D_V is now defined as

$$CV(D_V; \nu) = \sum_{\mathbf{x}_v\in D_V} CV(\mathbf{x}_v; \nu). \tag{2.29}$$

The rationale for using such a score function to measure the adequacy of the fitness of a statistical model can be found in [17].

Next, 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. The 10 results from the folds are averaged to produce a single estimation for the score function that we denote it by $CV(D; \nu)$.

The procedure of the golden section search is then used to find the optimal tuning parameter as follows.

(i) Suppose that (a, b) is the current range for the tuning parameter and the scores at the endpoints are $CV(D; a)$ and $CV(D; b)$. Let $c = a + (1-\varphi)(b-a)$, $d = a + \varphi(b-a)$, $\varphi = \frac{\sqrt{5}-1}{2}$. Calculate $CV(D; c)$ and $CV(D; d)$.

(ii) If $CV(D; c) > CV(D; d)$, then the range for the tuning parameter is changed to (a, d); otherwise, the range for the tuning parameter is changed to (c, b).

(iii) Repeat Steps (i)–(ii) until the length of the range is less than a predefined threshold $\delta > 0$.

(iv) The tuning parameter is estimated as the midpoint of the final range.

The above penalized GECM algorithm optimizes the parameters for the Erlang mixtures with fixed M. In order to reach a satisfactory fitting result and to avoid overfitting, we apply a forward selection approach to select the least possible number of components. That is, the penalized GECM algorithm is carried out for a 2-component Erlang mixture first and is then increased by one each time. The search stops when the average of the score function does not increase anymore.

3. Parameter initialization

As an iterative algorithm an EM algorithm highly depends on initial values. In this section we extend the method of clusterized method of moments (CMM) along with K-means algorithm proposed in [11] to multivariate case.

For the corresponding random complete data $(\mathbf{X}_1, \mathbf{Z}_1), (\mathbf{X}_2, \mathbf{Z}_2), \ldots, (\mathbf{X}_n, \mathbf{Z}_n)$, under the assumptions presented in Section 2, we have the following results, for $v = 1, \ldots, n, j = 1, \ldots, k, u = 1, \ldots, M$,

$$E[Z_{vu}] = P(Z_{vu} = 1) = \alpha_u, \tag{3.1}$$

$$E[X_{vj}|Z_{vu} = 1] = \theta m_{uj} \stackrel{\triangle}{=} \mu_{uj}, \tag{3.2}$$

$$E[X_{vj}Z_{vu}] = \alpha_u\mu_{uj}, \tag{3.3}$$

$$E[X_{vj}] = \theta\sum_{u=1}^{M}\alpha_u m_{uj} = \sum_{u=1}^{M}\alpha_u\mu_{uj}, \tag{3.4}$$

$$E[X_{vj}^2] = \theta^2\sum_{u=1}^{M}\alpha_u(m_{uj} + m_{uj}^2) = \theta E[X_{vj}] + \sum_{u=1}^{M}\alpha_u\mu_{uj}^2, \tag{3.5}$$

where the formula $\theta m_{uj} \stackrel{\triangle}{=} \mu_{uj}$ in (3.2) means θm_{uj} is denoted by μ_{uj}.

For convenience, we re-parametrize Φ as $\Phi' = \{\alpha_u, \mu_{uj}, \theta, u = 1, \ldots, M, j = 1, \ldots, k\}$, where $\mu_{uj} = \theta m_{uj}, u = 1, \ldots, M, j = 1, \ldots, k$. In this case, the shape parameters are estimated by $m_{uj} = \lceil \mu_{uj}/\theta \rceil$, where $\lceil x \rceil$ is the ceiling function of x. The initialization of the parameters is now summarized as follows.

(1) Apply the K-means clustering method to group the data into M groups so that Group $u, u = 1, \ldots, M$, represents data from the uth component distribution of the mixture. The K-means algorithm works especially well for multivariate data clustering. The detailed procedure can be seen in [18] and [11] and we omit it here.

(2) According to (3.1), the mixing weights are estimated by

$$\widehat{\alpha}_u = \frac{\sum_{v=1}^{n} z_{vu}}{n} = \frac{n_u}{n}, u = 1, 2, \ldots, M, \tag{3.6}$$

where $n_u = \sum_{v=1}^{n} z_{vu}$ represents the number of the points clustered into the uth group.

Table 1
The parameters of the 5-component trivariate Erlang mixture.

u	α_u	$\mathbf{m}_u$	θ
1	0.2	(5, 20, 4)	0.01
2	0.2	(10, 30, 5)	
3	0.3	(30, 40, 5)	
4	0.1	(30, 70, 6)	
5	0.2	(80, 70, 6)	

(3) According to (3.3), the mean parameters are estimated by

$$\widehat{\mu}_{uj} = \frac{\sum_{v=1}^{n} x_{vj} z_{vu}}{n_u}, u = 1, 2, \ldots, M, j = 1, \ldots, k. \tag{3.7}$$

(4) We adopt the weighted least square estimation to estimate the common scale parameter as

$$\begin{aligned} \widehat{\theta}^{w} &= \arg\min \left\{ \sum_{j=1}^{k} w_j \left(\overline{x_j^2} - \theta \overline{x}_j - \sum_{u=1}^{M} \widehat{\alpha}_u \widehat{\mu}_{uj}^2 \right)^2 \right\} \\ &= \sum_{j=1}^{k} w_j \left(\overline{x_j^2} - \sum_{u=1}^{M} \widehat{\alpha}_u \widehat{\mu}_{uj}^2 \right) \overline{x}_j / \sum_{j=1}^{k} w_j \overline{x}_j^2, \end{aligned} \tag{3.8}$$

where $\overline{x}_j = \frac{1}{n}\sum_{v=1}^{n} x_{vj}$, $\overline{x_j^2} = \frac{1}{n}\sum_{v=1}^{n} x_{vj}^2$, $j = 1, \ldots, k$, w_j is the weight corresponding to jth dimension. Here we choose $w_j = \frac{1}{\overline{x}_j}$ (the reason we choose this weight will be explained later) and the scale parameter is estimated by

$$\widehat{\theta} = \sum_{j=1}^{k} \left(\overline{x_j^2} - \sum_{u=1}^{M} \widehat{\alpha}_u \widehat{\mu}_{uj}^2 \right) / \sum_{j=1}^{k} \overline{x}_j. \tag{3.9}$$

If $\hat{\theta}$ is too large, many shape parameters may be initialized to be 1. Therefore, we further adjust the estimate with

$$\hat{\theta}^* = \min\{\hat{\theta}, \min\{\widehat{\mu}_{uj}\}, u = 1, \ldots, M, j = 1, \ldots, k\}. \tag{3.10}$$

(5) The initial shape parameters are estimated by

$$\hat{m}_{uj} = \lceil \widehat{\mu}_{uj} / \hat{\theta}^* \rceil, u = 1, 2, \ldots, M, j = 1, 2, \ldots, k. \tag{3.11}$$

Given the initial estimates above, we have the following equation

$$\sum_{j=1}^{k} \sum_{v=1}^{n} (x_{vj} - \overline{x}_j)^2 = \sum_{j=1}^{k} \sum_{u=1}^{M} \widehat{\alpha}_u (\widehat{\mu}_{uj} - \overline{x}_j)^2 + \widehat{\theta} \sum_{j=1}^{k} \overline{x}_j, \tag{3.12}$$

which explains why we choose the K-means algorithm to deal with this clustering issue. This equation holds if and only if the weight $w_j = \frac{1}{\overline{x}_j}, j = 1, \ldots, k$ and hence such choice of weights is rational.

4. Simulation studies

This section provides simulation studies to illustrate the versatility of the multivariate Erlang mixture and efficiency of the proposed algorithm. In the first study, we generate data from a multivariate Erlang mixture and compare the estimated parameters with the true values. In Study 2, we generate data from a bivariate log-normal distribution and use some statistical tools to test the goodness-of-fit of the fitted model.

4.1. Fitting data from a multivariate Erlang mixture

We generate data from a multivariate Erlang mixture and use the algorithm to fit the data to check whether we can recover the original model. For simulation data, we consider a 5-component trivariate Erlang mixture with the parameters given in Table 1.

2000 points are generated from the mixture. The GECM algorithm with the 10-fold cross-validation and the golden section search selects a 5-component Erlang mixture to fit the data. Table 2 shows the steps of golden section search for the tuning parameter. The initial range for the tuning parameter is $(-2.57, 1.43)$ and the scores at the two endpoints are 787.24 and 594.44, respectively. The search stops when the length of the interval is less than 0.1. According to the results in Table 2, the tuning parameter is $\nu = -2.22$.

To test the efficiency of the roughness penalty, we compare the results obtained by both the penalized GECM method and the one without penalty. Table 3 shows the estimated parameters with tuning parameter $\nu = -2.22$ and those without penalty which corresponds to $\nu = -\infty$. One can see that the ultimate estimates for the parameters (especially for the shape parameters and the scale parameter) are much closer to the true values when the penalty is applied.

Table 2
Golden section search for the tuning parameter.

Step	c	d	$CV(D; c)$	$CV(D; d)$	Range
1	−1.04	−0.10	793.63	749.62	(−2.57, −0.10)
2	−1.63	−1.04	804.35	793.63	(−2.57, −1.04)
3	−1.99	−1.63	811.60	804.35	(−2.57, −1.63)
4	−2.21	−1.99	813.14	811.60	(−2.57, −1.99)
5	−2.35	−2.21	811.53	813.14	(−2.35, −1.99)
6	−2.21	−2.12	813.14	812.85	(−2.35, −2.13)
7	−2.26	−2.21	813.05	813.14	(−2.26, −2.13)
8	−2.21	−2.18	813.14	813.03	(−2.26, −2.18)

Table 3
Comparison of the parameters of the fitted trivariate Erlang mixtures.

$\nu = -2.22$				$\nu = -\infty$			
u	α_u	$\mathbf{m}_u$	θ	u	α_u	$\mathbf{m}_u$	θ
1	0.1951	(5, 20, 4)	0.0010	1	0.1926	(6, 22, 4)	0.0093
2	0.2043	(10, 30, 5)		2	0.1999	(11, 33, 5)	
3	0.3026	(30, 40, 5)		3	0.3149	(32, 43, 5)	
4	0.1041	(30, 69, 6)		4	0.1026	(33, 75, 6)	
5	0.1939	(79, 69, 6)		5	0.1900	(87, 75, 7)	

Table 4
The parameters of the fitted trivariate Erlang mixture for truncated data.

u	α_u	$\mathbf{m}_u$	θ
1	0.1975	(6, 23, 5)	0.0091
2	0.1720	(11, 35, 5)	
3	0.3148	(33, 44, 6)	
4	0.1171	(33, 77, 7)	
5	0.1986	(87, 77, 7)	

Table 5
The parameters of the fitted Erlang mixture for the log-normal data.

u	α_u	$\mathbf{m}_u$	θ
1	0.3752	(136, 266)	0.0077
2	0.0894	(164, 343)	
3	0.5354	(149, 301)	

We now take truncation into consideration and assume that $\mathbf{t}^l = (0.02, 0.02, 0.02)$ and $\mathbf{t}^r = (1, 1, 1)$. After removing the points outside the truncation range, 1852 data points remain. The tuning parameter in this case is $\nu = -2.04$. Table 4 presents the estimated parameters. One may see that the estimated parameters are still close to the true values.

4.2. Fitting data from a multivariate log-normal distribution

In this subsection, we use a bivariate Erlang mixture to fit data generate from a bivariate lognormal distribution. Suppose that a random vector $\mathbf{X} = (X_1, X_2)$ comes from a bivariate normal distribution with mean vector $\boldsymbol{\mu} = (0.1, 0.8)$ and covariance matrix $\boldsymbol{\Sigma} = \begin{pmatrix} 0.01 & 0.005 \\ 0.005 & 0.01 \end{pmatrix}$. Let $\mathbf{Y} = e^{\mathbf{X}}$, then the joint distribution of $\mathbf{Y}$ is a bivariate lognormal distribution. We generate 2000 points from the distribution and use a bivariate Erlang mixture to fit the simulated data. The estimated parameters are given in Table 5. The estimated tuning parameter in this case is $\nu = 0.046$. Fig. 1 shows the surface of the fitted joint density function.

Again, to illustrate the performance of the roughness penalty, we compare the fitted model with and the fitted model without penalty. In Fig. 2, the true density, fitted density of the marginals with and without penalty are presented. The results clearly show that the proposed algorithm is better.

To further study the efficiency of the GECM algorithm, we may also use the L^2 distance between the true density f and the fitted density $\hat{f}$ as a closeness measure:

$$D^2(f, \hat{f}) = \underbrace{\int_0^\infty \cdots \int_0^\infty}_{k} [f(\mathbf{x}) - \hat{f}(\mathbf{x})]^2 d\mathbf{x}. \tag{4.1}$$

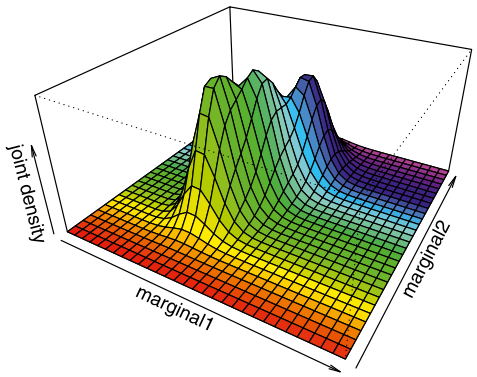

Fig. 1. Surface of the fitted density function.

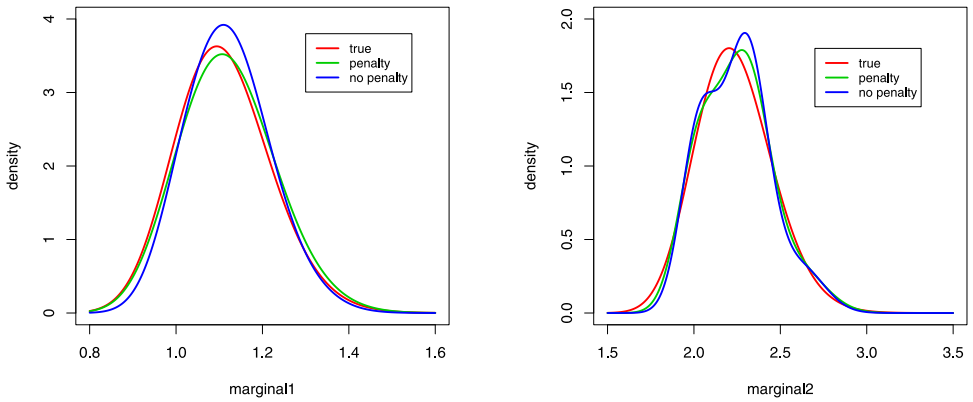

Fig. 2. Densities of the marginals (left panel: the first marginal; right panel: the second marginal).

Table 6
Distance between the true density and the fitted density.

Methods	Penalized GECM	Non-penalized GECM
Components	3	3
Cross-validation value	203.63	186.36
Distance	11.01	11.18
Covariance	0.0119	0.0103

We now calculate the distances between the true density and the fitted densities obtained by the non-penalized GECM algorithm and the penalized GECM algorithm, respectively. Table 6 compares the results from penalized algorithm and non-penalized algorithm that indicate that both the fitted models have the same number of components, but the fitted density obtained by the penalized GECM algorithm is closer to the true density. A closer covariance value to the true value 0.0125 and a larger cross-validation value also indicate that the penalized GECM algorithm can fit the validation data better.

The GECM algorithm is also applicable for randomly censored data as illustrated below. Along with the above 2000 uncensored data points $(x_{vj}), v = 1, \ldots, 2000, j = 1, 2$ we now generate 2000 points from a univariate log-normal distribution with parameters $\mu = 0.005$ and $\sigma = 0.01$ and denote them as $y_1, \ldots, y_{2000}$. For $v = 1, \ldots, 2000$, the data point x_{v1} is set to be left censored if $x_{v1} < y_v$ with censoring point $c^l_{v1} = y_v$. Similarly, we generate 2000 points, $z_1, \ldots, z_{2000}$, from another log-normal distribution with parameters $\mu = 1$ and $\sigma = 0.01$. For $v = 1, \ldots, 2000$, the data point x_{v2} is set to be right censored if $x_{v2} > z_v$ with censoring point $c^r_{v2} = z_v$. Thus, we have 2000 points with 360 left censored data points and 166 right censored data points. We fit the multivariate Erlang mixture with the roughness penalty to the simulated data. The tuning parameter is $\nu = 0.15$ in this case. A 3-component multivariate Erlang mixture is obtained and the estimated parameters are given in Table 7.

We may also consider the fitness to the distribution of aggregated data: $S_2 = X_1 + X_2$ for the reason that a good fit often indicates that the fitted model may well capture the dependence structure of the data. It follows from Section 2 that S_2 has a univariate Erlang mixture with the same mixing weights and scale parameter and the shape parameters being the sum of shape parameters of the corresponding components. Fig. 3 presents the fitting results for the aggregate

Table 7
Parameter values of the fitted Erlang mixture for censored lognormal data.

u	α_u	$\mathbf{m}_u$	θ
1	0.0927	(154, 319)	0.0081
2	0.5553	(141, 286)	
3	0.3520	(129, 253)	

Table 8
Goodness-of-fit tests for the Erlang mixture.

Test	Statistic	p-value	Not rejected at 5% significant value
K-S	0.0217	0.3054	Yes
A-D	1.5422	0.1667	Yes
Cv-M	0.2777	0.2564	Yes

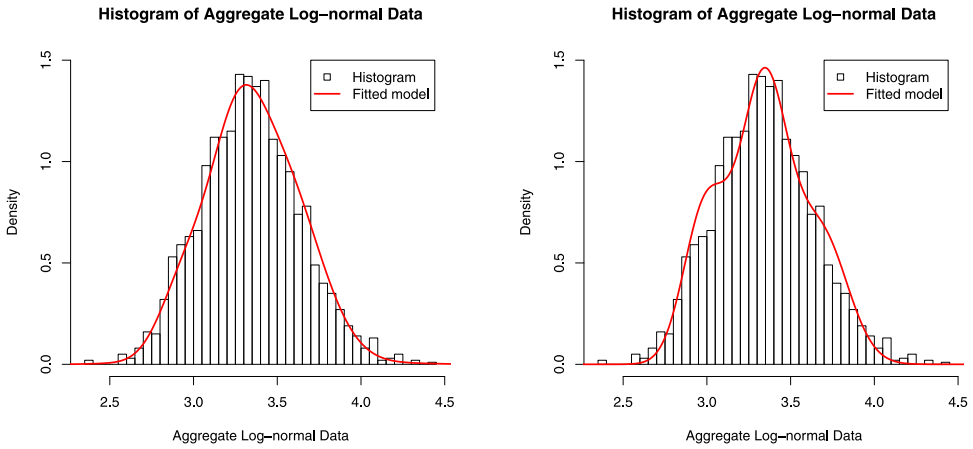

Fig. 3. Histograms of the aggregate data (left panel: with roughness penalty with tuning parameter $\nu = 0.15$; right panel: without roughness penalty).

data. The plot in the left panel is the fitted density by using the roughness penalty and the right panel the fitted density without the penalty. It is easy to see that the density becomes much smoother when the penalty is applied.

We may further examine the fitness quantitatively by performing several statistical tests such as the Kolmogorov–Smirnov (K–S) test, the Anderson–Darling (A–D) test and the Cramer–von Mises (Cv–M) test. Table 8 summarizes the results of these three common goodness-of-fit tests. All the tests indicate a good fit to the simulated data.

5. Applications

In this section, we apply the proposed algorithm to fit the multivariate Erlang mixture to several real data sets.

5.1. Old faithful geyser data

The data represent the waiting time between two consecutive eruptions and the duration of the eruption for the Old Faithful geyser in Yellowstone National Park of the United States. The dataset contains 299 observations. It is also studied in [9,17].

Using the 10-fold cross-validation, we set the tuning parameter to be $\nu = 3.92$ for this data set. A 7-component bivariate Erlang mixture is selected with the parameters given in Table 9.

As mentioned earlier, one advantage of this proposed algorithm is to avoid overfitting. We compare the number of components of fitted models by 3 different methods: (a) the penalized GECM algorithm in this paper; (b) the non-penalized GECM algorithm with the number of component determined by the leave-one-out cross-validation method; (c) the EM algorithm presented in [9] with BIC being used to determine the number of components. The results in Table 10 show that the proposed algorithm leads to the smallest number of components.

Fig. 4 displays the contour plots from the fitted models. Clearly, there are fewer peaks when the roughness penalty is applied, which indicates a smoother density surface.

If we are interested in the distribution of the duration time of a complete cycle, i.e., the time from the ending of one eruption to the ending of the next eruption, it is exactly the sum of the waiting time and the duration of the eruption. The

Table 9
Parameter estimates of the fitted 7-components Erlang mixture.

u	α_u	$\mathbf{m}_u$	θ
1	0.0444	(28, 528)	0.1301
2	0.1574	(16, 414)	
3	0.0883	(35, 684)	
4	0.1037	(17, 469)	
5	0.3061	(34, 628)	
6	0.0925	(16, 368)	
7	0.2076	(33, 586)	

Table 10
The number of components using 3 different methods.

Method	Penalized	one-out CV	BIC
Number	7	15	11

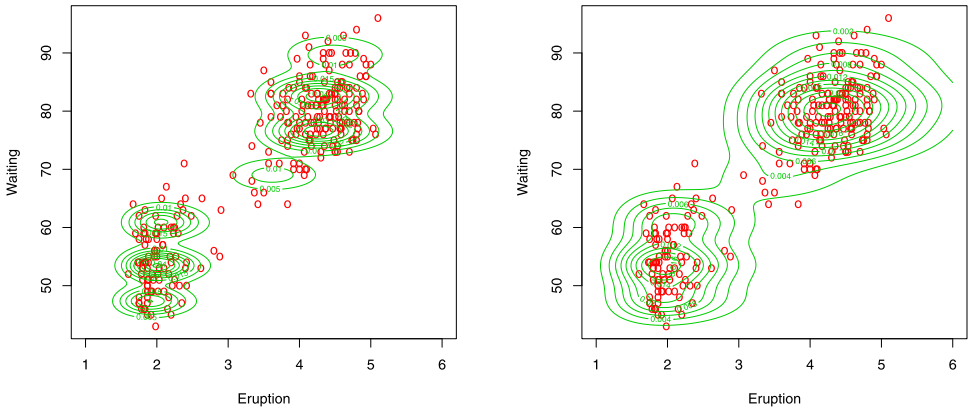

Fig. 4. Contour plots of the fitted models (left panel: without roughness penalty; right panel: with roughness penalty).

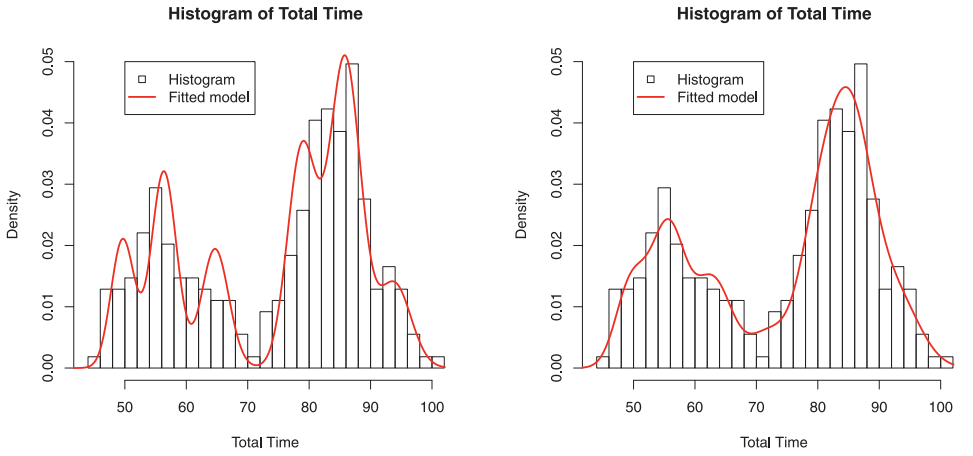

Fig. 5. Histograms and fitted densities of the duration time of a complete cycle (left panel: without roughness penalty; right panel: with roughness penalty).

distribution can explicitly be obtained from the fitted multivariate Erlang mixture. Fig. 5 shows the densities of the fitted aggregate distribution. It is obvious that the curve becomes much smoother when applying the penalty. Fig. 6 shows the PP-plot and QQ-plot of the fitted aggregate distribution when considering the roughness penalty.

The three goodness-of-fit tests are again used to examine the performance. The results are given in Table 11, which reconfirms good fitness.

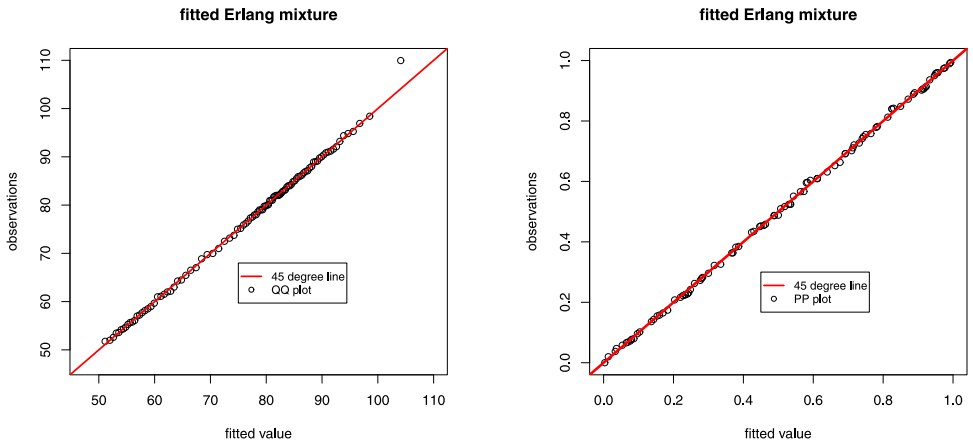

Fig. 6. QQ and PP Plots for the total duration time with the roughness penalty.

Table 11
Goodness-of-fit tests for the Erlang mixture with the roughness penalty.

Test	Statistic	p-value	Not rejected at 5% significant value
K-S	0.0269	0.9892	Yes
A-D	0.1386	0.9993	Yes
Cv-M	0.0187	0.9981	Yes

Table 12
Parameter estimates with the 4-components mixture fitted to the mastitis data.

u	α_u	$\mathbf{m}_u$	θ
1	0.5794	(2, 2, 2, 2)	34.60
2	0.1651	(4, 6, 5, 5)	
3	0.1618	(10, 12, 13, 10)	
4	0.0937	(12, 10, 5, 5)	

5.2. Mastitis data

Mastitis is economically one of the most important diseases in the dairy sector since it leads to reduced milk yield and milk quality. In this application, we consider infectious disease data from a mastitis study. This dataset contains 100 records and it has also been used in [19,20] and [9]. The objective of this application is to study the infection times of individual cow udder quarters with a bacterium. The infection status is assessed from the time of parturition until the end of the lactation period. A cow was assumed to be infection-free at the parturition time. Two types of covariates are often considered. That is, the infection times at the udder quarter level and at the cow level. In this study, we consider the infection times at the udder quarter level. One quarter might be infected while the other three quarters remain infection-free, hence it generates a 4-dimensional dataset and the dependence among the infection times of the four udder quarters of a cow must be modeled. The infection times are not known exactly since a daily checkup would not be feasible and this generates interval-censored data with lower bound the last time at which it was infection-free and upper bound the first time at which it was infected. The infection observations are right censored if no infection occurred before the end of the lactation period or if the cow is lost to follow-up during the study, for example due to culling. Borrow the notation in [9], the udder quarters are denoted as RL (rear left), FL (front left), RR (rear right) and FR (front right).

We use a 4-variate Erlang mixture to fit the data and a 4-component Erlang mixture is obtained. The tuning parameter is $\nu = 12.54$ for this dataset and the parameters are given in Table 12. Fig. 7 shows the empirical and fitted survival curves of each udder quarter. The green curves in Fig. 7 are the Kaplan–Meier survival curves with interval censoring and right censoring taken into account, along with 95% confidence intervals. Fig. 8 shows the contour plots of the fitted model among the udder quarters. The interval censored data points are depicted by the midpoints and the right censored points are depicted by the corresponding right censoring points in the scatter plots.

As a measure of the infectivity of the agent causing the disease, we are interested in the correlation between udder infection times. Due to the fact that the bivariate marginals again belong to the Erlang mixture class, we have closed-form expressions for Kendall's τ and Spearman's ρ.

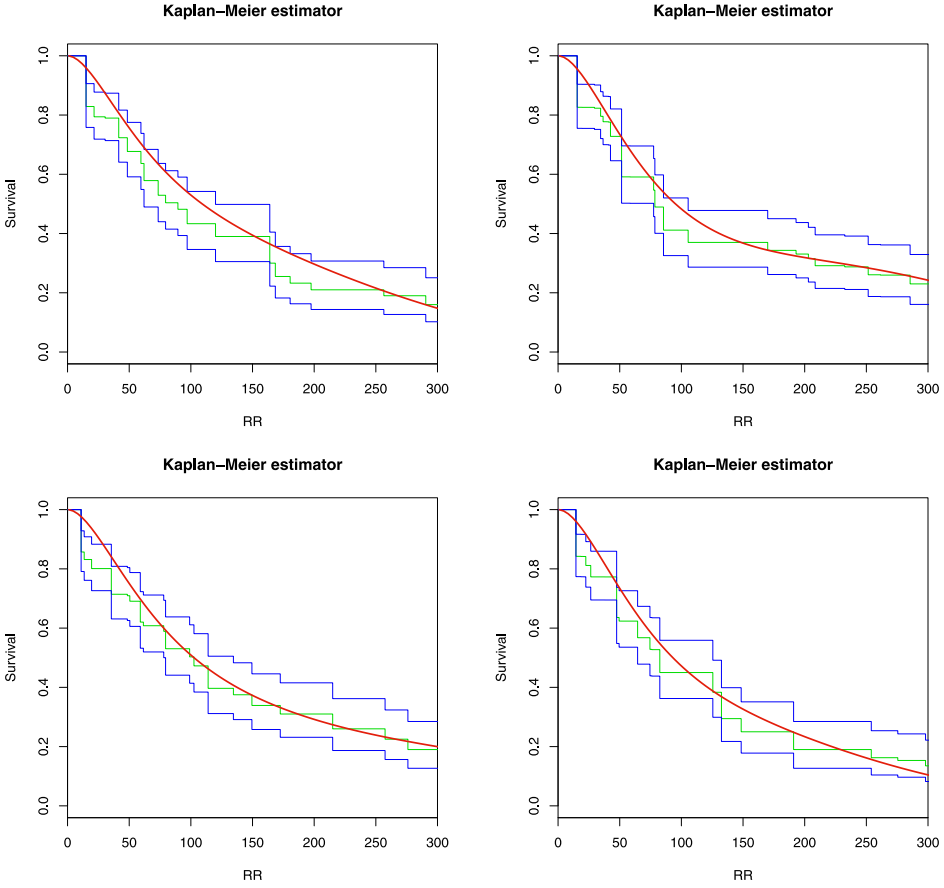

Fig. 7. Empirical and fitted survival curves of each quarter.

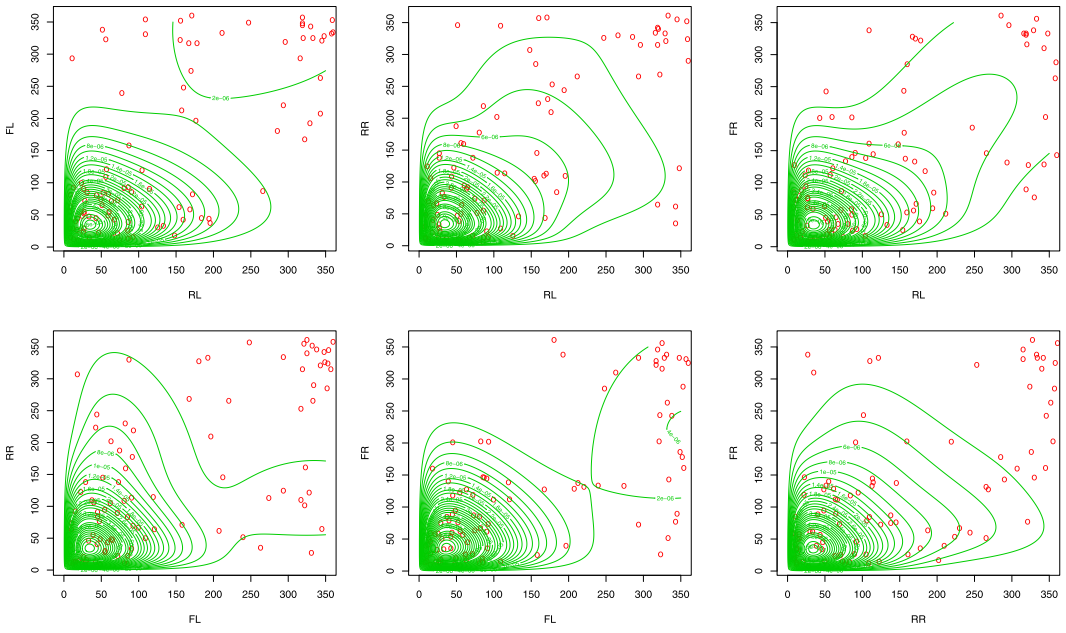

Fig. 8. Contour plots of the fitted model.

Table 13
Estimates of correlation measures and standard error.

		FL	FR	RL	RR
FL	τ	1			
	ρ	1			
FR	τ	0.4339(0.0103)	1		
	ρ	0.6383(0.0138)	1		
RL	τ	0.4134(0.0125)	0.3752(0.0143)	1	
	ρ	0.5578(0.0170)	0.6012(0.0195)	1	
RR	τ	0.3980(0.0108)	0.4285(0.0103)	0.3979(0.0139)	1
	ρ	0.5266(0.0145)	0.6247(0.0139)	0.5770(0.0188)	1

Table 14
Parameter estimates with 8-component Erlang mixture fitted to travel reviews data.

u	α_u	$\mathbf{m}_u$	θ
1	0.0772	(14, 30, 23, 8, 18, 34, 45, 40, 23, 36)	0.0711
2	0.1601	(13, 17, 16, 6, 9, 21, 45, 40, 20, 42)	
3	0.2247	(11, 19, 5, 7, 15, 27, 44, 41, 23, 41)	
4	0.1608	(11, 18, 32, 8, 15, 29, 44, 40, 22, 36)	
5	0.0264	(22, 6, 7, 7, 8, 18, 45, 41, 20, 39)	
6	0.1943	(12, 19, 4, 7, 7, 19, 46, 40, 22, 42)	
7	0.1396	(13, 16, 18, 8, 18, 33, 45, 41, 22, 38)	
8	0.0171	(14, 28, 5, 30, 15, 28, 47, 39, 21, 35)	

The bootstrap method is used to estimate the uncertainty of the estimated relationship measures. Based on re-sample the observed data for B times, we generate a collection of Kendall's τ: $\tau(\Phi_1), \ldots, \tau(\Phi_B)$, the standard error for $\tau(\Phi)$ is

$$SE(\tau(\Phi)) = \sqrt{\frac{\frac{1}{B-1}\sum_{b=1}^{B}\tau(\Phi_b)^2 - \overline{\tau}^2(\Phi)}{B}}, \tag{5.1}$$

where $\overline{\tau}(\Phi) = \frac{1}{B}\sum_{b=1}^{B}\tau(\Phi_b)$. The estimation and standard error for Spearman's ρ can be obtained as well.

We generate $B = 1000$ bootstrap samples by re-sampling from the original dataset. The estimates and standard errors for Kendall's τ and Spearman's ρ are showed in Table 13. The standard errors are less than the results in [9] indicate that more accurate estimates have been obtained.

5.3. Travel reviews data

In this subsection, we considered a social media dataset from tourism domain for the analysis and captured the results. The dataset corresponds to user interest information accrued from reviews, feedbacks on different types of point of interests and ratings on attractions. This data set is populated by crawling TripAdvisor.com. Reviews on destinations in 10 categories mentioned across East Asia are considered. Each traveler rating is mapped as Excellent (4), Very Good (3), Average (2), Poor (1), and Terrible (0) and average rating is used against each category per user. This dataset is also studied in [21].

The dataset contains 980 user records with 10 feedback attributes inferred from numerous reviews. The 10 mentioned categories across East Asia are art galleries, dance clubs, juice bars, restaurants, museums, resorts, parks/picnics spots, beaches, theaters and religious institutions. First we consider only the 978 records for which the rating to each component has a non-zero value (the average rating on dance clubs is zero for User 309 and User 517). We fit a 10-variate Erlang mixture to the data and an 8-component multivariate mixture is selected. The tuning parameter is $\nu = 0.052$ for this dataset. In Table 14, parameter estimates of the fitted distribution are given.

Again we consider the correlation between the category ratings and Table 15 shows the Kendall's τ values, where the upper triangular entries are empirical estimates and the lower triangular entries are estimates from the fitted model. From the results, we can see that some ratings between the categories show positive dependence. For example, the Kendall's τ between Category 5 and Category 6 based on the fitted model is 0.4936, which means one who gives a high evaluation on museums is more likely to give a high evaluation on resorts. While the ratings on some pairs show negative dependence, for example the ratings on resorts and religious institutions. Some other pairs may show weak dependence.

We now consider the conditional distribution of rating on one category given the value of the rating on another category. Denote the ratings by $(X_1, \ldots, X_{10})$, we look at the conditional distribution $F_{X_i}(x_i|X_j = x_j)$, $1 \le i, j \le 10, i \ne j$. It is easy to see that the conditional distributions again belong to Erlang mixtures and the parameters are easily obtained from the fitted model. Hence, we can estimate some interesting quantities such as the conditional expectation $E_{X_i}(X_i|X_j = x_j)$ and the conditional quantiles for a given rating on category j. Some visual results are presented in Fig. 9. In Fig. 9(a) we present the curves of expected ratings on the resorts and the 25, 75 and 95 quantiles given the ratings on the museums. From the results in Table 15, we can see that the ratings between the resorts and the museums show positive dependence which

Table 15
Kendall's τ between the categories.

CATG	1	2	3	4	5	6	7	8	9	10
1	1	−0.1296	0.0526	0.0101	−0.0441	0.1097	0.0040	0.0022	−0.0043	0.0179
2	−0.1172	1	0.0011	0.0334	0.0792	0.0642	0.0564	−0.1059	0.0530	−0.0181
3	0.0384	0.0096	1	0.1430	0.2205	0.2501	0.5751	−0.1014	−0.0417	−0.2977
4	0.0156	0.0278	0.1635	1	0.1350	0.2881	0.3646	−0.3760	0.0769	−0.4131
5	−0.0374	0.0993	0.2081	0.0674	1	0.4301	0.2006	−0.0080	0.0191	−0.1771
6	0.1129	0.0963	0.2412	0.2686	0.4936	1	0.3495	−0.0060	0.0843	−0.3091
7	0.0017	0.0375	0.2490	0.3140	0.2906	0.2898	1	−0.0708	0.0837	−0.6113
8	0.0079	−0.1069	−0.1075	−0.3070	−0.0032	−0.0066	−0.0321	1	0.1161	0.0100
9	−0.0038	0.0316	−0.0727	0.0143	0.0228	0.0114	0.4889	0.0519	1	0.0989
10	0.0225	−0.0298	−0.1727	−0.5082	−0.1267	−0.1296	−0.5060	0.0959	0.0163	1

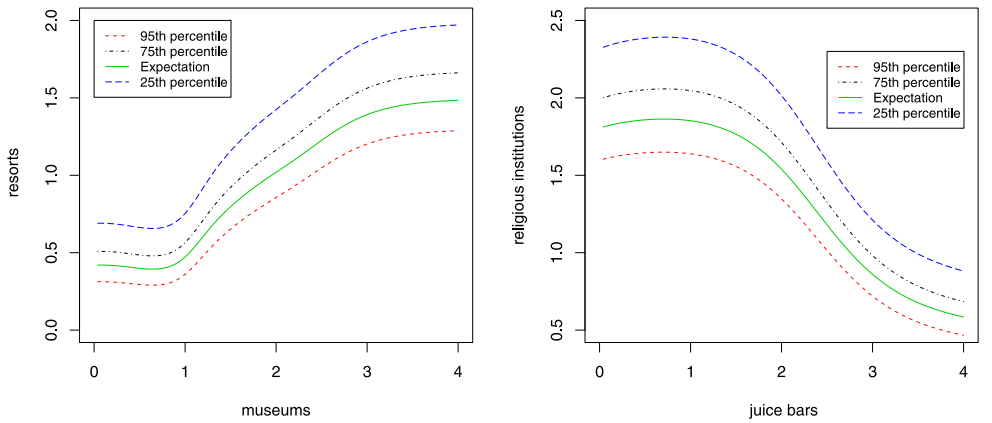

Fig. 9. Expectation and quantile curves: (a) resorts vs museum; (b) juice bars vs religious institutions.

is consistent with the visualization in Fig. 9(a). Similarly, the corresponding curves between the juice bars and religious institutions are presented in Fig. 9(b).

6. Conclusion

In this paper, we propose a GECM algorithm to estimate the parameters of multivariate Erlang mixtures by extending the GEM-CMM algorithm in [11]. The objective is to overcome the non-smoothness problem: the fitted curve is not smooth when a traditional EM algorithm is used. We propose a roughness penalty based on the integral of second derivative to deal with this issue. A common cross-validation is used to estimate the tuning parameter and we adopt a golden section search to find the optimal tuning parameter. The algorithm proposed in this paper also results in fewer components in Erlang mixtures so that we can further minimize the overfitting issue when compared with the use of the AIC or BIC criterion. The simulation studies and real data applications demonstrate the efficiency and effectiveness of the algorithm when fitting the multivariate Erlang mixture model to data.

Acknowledgments

This research was partly supported by grants from the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Natural Science Foundation of China (No. 11471272).

References

[1] J. Chen, P. Li, Y. Fu, Inference on the order of a normal mixture, J. Amer. Statist. Assoc. 107 (499) (2012) 1096–1105.
[2] H. Kasahara, K. Shimotsu, Testing the number of components in normal mixture regression models, J. Amer. Statist. Assoc. 110 (512) (2015) 1632–1645.
[3] S.C. Lee, X.S. Lin, Modeling and evaluating insurance losses via mixtures of Erlang distributions, N. Am. Actuar. J. 14 (1) (2010) 107–130.
[4] A. Mazza, A. Punzo, Mixtures of multivariate contaminated normal regression models, Statist. Papers (2017) 1–36.
[5] S.C. Lee, X.S. Lin, Modeling dependent risks with multivariate Erlang mixtures, Astin Bull. 42 (1) (2012) 153–180.
[6] H. Cossette, .M.P. Côté, E. Marceau, K. Moutanabbir, Multivariate distribution defined with Farlie–Gumbel–Morgenstern copula and mixed Erlang marginals: aggregation and capital allocation, Insurance Math. Econom. 52 (3) (2013) 560–572.

[7] G.E. Willmot, J.K. Woo, On some properties of a class of multivariate Erlang mixtures with insurance applications, Astin Bull. 45 (1) (2015) 151–173.
[8] E. Hashorva, G. Ratovomirija, On Sarmanov mixed Erlang risks in insurance applications, Astin Bull. 45 (1) (2015) 175–205.
[9] R. Verbelen, K. Antonio, G. Claeskens, Multivariate mixtures of Erlangs for density estimation under censoring, Lifetime Data Anal. 22 (3) (2016) 429–455.
[10] C. Yin, X.S. Lin, Efficient estimation of Erlang mixtures using iSCAD penalty with insurance application, Astin Bull. 46 (3) (2016) 779–799.
[11] W. Gui, R. Huang, X.S. Lin, Fitting the Erlang mixture model to data via a GEM-CMM algorithm, J. Comput. Appl. Math. 343 (2018) 189–205.
[12] C. Yin, X.S. Lin, R. Huang, H. Yuan, On the consistency of penalized MLEs for Erlang mixtures, Statist. Probab. Lett. 145 (2019) 12–20.
[13] S. Fung, A. Badescu, X.S. Lin, A Class of Mixture of Experts Models for General Insurance: Theoretical Developments, 2019, Available at SSRN 3315741.
[14] J.O. Ramsay, Functional data analysis, Encycl. Statist. Sci. (2004).
[15] G.H. Givens, J.A. Hoeting, Computational Statistics, Wiley, Hoboken, NJ, 2013.
[16] M. Avriel, D.J. Wilde, Optimally proof for the symmetric fibonacci search technique, Fibonacci Q. J. (1966).
[17] B.W. Silverman, Density Estimation for Statistics and Data Analysis, CRC Press, 1986.
[18] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
[19] K. Goethals, B. Ampe, D. Berkvens, H. Laevens, P. Janssen, L. Duchateau, Modeling interval-censored, clustered cow udder quarter infection times through the shared gamma frailty model, J. Agricul. Biol. Environ. Statist. 14 (1) (2009) 1–14.
[20] B. Ampe, K. Goethals, H. Laevens, L. Duchateau, Investigating clustering in interval-censored udder quarter infection times in dairy cows using a gamma frailty model, Prevent. Veter. Med. 106 (3–4) (2012) 251–257.
[21] S. Renjith, A. Sreekumar, M. Jathavedan, Evaluation of partitioning clustering algorithms for processing social media data in tourism domain, EEE Recent Adv. Intell. Comput. Syst. (RAICS) 12 (2018) 7–131.