Statistics and Computing (2024) 34:125
https://doi.org/10.1007/s11222-024-10435-3

ORIGINAL PAPER

A modified EM-type algorithm to estimate semi-parametric mixtures of non-parametric regressions

Sphiwe B. Skhosana[1] · Salomon M. Millard[1] · Frans H. J. Kanfer[1]

Received: 22 November 2023 / Accepted: 8 May 2024

Abstract
Semi-parametric Gaussian mixtures of non-parametric regressions (SPGMNRs) are a flexible extension of Gaussian mixtures of linear regressions (GMLRs). The model assumes that the component regression functions (CRFs) are non-parametric functions of the covariate(s) whereas the component mixing proportions and variances are constants. Unfortunately, the model cannot be reliably estimated using traditional methods. A local-likelihood approach for estimating the CRFs requires that we maximize a set of local-likelihood functions. Using the Expectation-Maximization (EM) algorithm to separately maximize each local-likelihood function may lead to label-switching. This is because the posterior probabilities calculated at the local E-step are not guaranteed to be aligned. The consequence of this label-switching is wiggly and non-smooth estimates of the CRFs. In this paper, we propose a unified approach to address label-switching and obtain sensible estimates. The proposed approach has two stages. In the first stage, we propose a model-based approach to address the label-switching problem. We first note that each local-likelihood function is a likelihood function of a Gaussian mixture model (GMM). Next, we reformulate the SPGMNRs model as a mixture of these GMMs. Lastly, using a modified version of the Expectation Conditional Maximization (ECM) algorithm, we estimate the mixture of GMMs. In addition, using the mixing weights of the local GMMs, we can automatically choose the local points where local-likelihood estimation takes place. In the second stage, we propose one-step backfitting estimates of the parametric and non-parametric terms. The effectiveness of the proposed approach is demonstrated on simulated data and real data analysis.

Keywords EM algorithm · Local-likelihood · Mixture models · Gaussian mixtures of regressions · Local-polynomial regression

1 Introduction

Finite mixture models have become a useful tool for studying any variable, say y, that takes its values from a population that is made up of a number of *a priori* known, say K, sub-populations mixed randomly in proportion to their relative sizes $\pi_1, \pi_2, \ldots, \pi_K$. In this case, each sub-population, known as a component, is usually distributed by a parametric distribution having a density function $f(\cdot|\theta_k)$, for $k = 1, 2, \ldots, K$. The component parameters θ_k, often vector-valued, and the relative sizes (weights), positive and summing to unity, are distinct across the components.

The most frequently used mixture model for a univariate variable y arises when each component density is assumed to be normal, henceforth Gaussian. In this case, the parameter vector $\boldsymbol{\theta}_k = (\mu_k, \sigma_k^2)$. The mixture density function of y is a convex combination of the Gaussian component densities

$$f(y) = \pi_1 f(y|\mu_1, \sigma_1^2) + \cdots + \pi_K f(y|\mu_K, \sigma_K^2) = \sum_{k=1}^{K} \pi_k \mathcal{N}\{y|\mu_k, \sigma_k^2\} \quad (1)$$

where $\mathcal{N}\{\cdot|\mu, \sigma^2\} = f(\cdot|\mu, \sigma^2)$ denotes a Gaussian density with mean μ and variance σ^2. The weights π_k are also known

Salomon M. Millard and Frans H. J. Kanfer contributed equally to this work.

✉ Sphiwe B. Skhosana
spiwe.skhosana@up.ac.za

Salomon M. Millard
sollie.millard@up.ac.za

Frans H. J. Kanfer
frans.kanfer@up.ac.za

[1] Department of Statistics, University of Pretoria, Pretoria 0028, South Africa

Published online: 29 May 2024

as mixing proportions or probabilities. Model (1) is known as a Gaussian mixture model (GMM). For the theory and application of GMMs and mixture models, in general, see Titterington et al. (1985); McLachlan and Peel (2000) and more recently (Fruhwirth-Schnatter et al. 2019).

Suppose the variable y depends on a set of D covariates $\mathbf{x} = (x_1, x_2, \ldots, x_D)$ and we are interested in studying this dependence. In this case, each component, known as a regression component, is typically a linear regression model of y on $\mathbf{x}$ having a Gaussian error distribution. The resulting model is a Gaussian mixture of linear regressions (GMLRs) given as

$$f(y|\mathbf{X} = \mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}\{y|m_k(\mathbf{x}), \sigma_k^2\} \tag{2}$$

where $m_k(\mathbf{x}) = \mathbf{x}^\intercal \boldsymbol{\beta}_k$ is the regression function of the k^{th} regression component, $\mathbf{x} = (x_0, x_1, x_2, \ldots, x_D)$, with $x_0 = 1$, and $\boldsymbol{\beta}_k = (\beta_0, \beta_1, \ldots, \beta_K)$ is the regression parameter vector of the k^{th} regression component.

GMLRs were first introduced by Quandt (1972) as switching regression models. The models have received widespread adoption in areas such as economics (Quandt and Ramsey 1978), marketing (DeSarbo and Cron 1988), machine learning (Jacobs et al. 1991), environmental economics (Hurn et al. 2003), medicine (Schlattmann 2009), among many other fields. See Chapter 8 of (Frühwirth-Schnatter 2006) for more details on the theory of GMLRs, in particular, and mixtures of regression models, in general.

The linearity assumption imposed on model (2), through the component regression functions (CRFs), is quite restrictive. The main reason for this assumption is that an additive covariate effect makes for ease of interpretation (Hastie and Tibshirani 1990). Efforts to relax this assumption, partly or completely while retaining the desirable additive covariate effect, have emerged in the literature. The proposed models assume that some of the covariates are linearly related to the response variable y while the relationship between y and the other variables is characterised by additive non-parametric univariate functions. Let $\mathbf{x} = (\mathbf{x}, \mathbf{t})$, the general form of this class of models is

$$f(y|\mathbf{X} = \mathbf{x}, \mathbf{T} = \mathbf{t}) = \sum_{k=1}^{K} \pi_k \mathcal{N}\{y|m_k(\mathbf{x}, \mathbf{t}), \sigma_k^2\}, \tag{3}$$

where $m_k(\mathbf{x}, \mathbf{t}) = \mathbf{x}\boldsymbol{\beta}_k + \sum_{r=1}^{D_2} g_k(t_r)$.

In model (3), the covariates $\mathbf{x} \in \mathbb{R}^{D_1}$ are assumed to enter the model linearly (hence, parametric) characterised by the regression parameters $\boldsymbol{\beta_k}$, for $k = 1, 2, \ldots, K$. On the other hand, the covariates $\mathbf{t} \in \mathbb{R}^{D_2}$ are assumed to be characterised by smooth unknown (hence, non-parametric) additive univariate functions $g_k(t_r)$ of the covariates $t_r, r = 1, 2, \ldots, D_2$, respectively. Thus, the CRFs are semi-parametric functions.

Model (3) was first introduced and studied by (Zhang (2020)) as a finite semi-parametric Gaussian mixture of partially linear additive models (SPGMPLAMs). For identifiability, $\mathbb{E}\{g_k(t_r)\} = 0$, for $t = 1, 2, \ldots, D_2$. Moreover, without loss of generality, we assume that the covariates $t_r : r = 1, 2, \ldots, D_2$ take values on the compact interval $[a, b]$, where $b > a$.

If $K = 1$, model (3) reduces to an additive partial linear model (APML) (Opsomer 1999). If each $g_k(t_r)$, for $t_r, r = 1, 2, \ldots, D_2$, is a linear function of the corresponding covariate, then model (3) is the same as model (2). Thus, model (3) is a natural extension of an APLM and a GMLRs model.

Model (3) encompasses many Gaussian mixtures of regressions some of which were introduced recently. The following list is in no way exhaustive:

1. If $D_2 = 0$ and $D_1 = 1$, model (3) reduces to the semi-parametric Gaussian mixture of non-parametric regressions model (SPGMNRs) introduced by Xiang and Yao (2018).
2. If $D_2 = 1$, model (3) reduces to the semi-parametric Gaussian mixture of partially linear models (SPGMPLMs) introduced by Wu (2016).
3. If $D_1 = 0$, model (3) reduces to the semi-parametric Gaussian mixture of additive regressions model (SPGMARs) introduced by Zhang (2017).

From a statistical inference point of view, the advantage of model (3) is that it combines the flexibility of a non-parametric model and the simplicity, in particular interpretability, of a parametric model. However, in practice, due to the presence of both parametric and non-parametric terms, model (3) poses an estimation and computational challenge. First, a likelihood approach for estimating the non-parametric functions requires that we maximize a set of locally defined likelihood functions. Using the Expectation-Maximization (EM) algorithm to separately maximize each local likelihood function may lead to label switching (Huang 2012 and Huang and Li (2013)). This problem is illustrated in Sect. 3. Second, note that efficient parametric estimation requires all the observed data whereas non-parametric estimation uses data in the neighbourhood of a local point. Thus, how can we construct an estimation procedure that is appropriate for estimating both the parametric and non-parametric term?

In this paper, we propose a unified approach to address all of these challenges. The proposed approach has two stages. In the first stage, we propose a model-based approach to address the label-switching problem. Briefly, we first note that each local likelihood function is a likelihood function of a GMM (1). Next, we rewrite model (3) as a mixture of these

GMMs. Lastly, using a modified Expectation-Conditional-Maximization (ECM) algorithm, we estimate the mixture of GMMs thus simultaneously estimating the component non-parametric functions. We refer to this approach as a model-based approach. More details are given in Sect. 4. In the second stage, we propose one-step backfitting estimates of the parametric and non-parametric terms.

To aid the reader's comprehension of the novelty of the proposed ideas, in this paper we will develop the proposed estimation approach for a simple special case of the general model (3), the SPGMNRs given by

$$f(y|X=x)=\sum_{k=1}^{K}\pi_k\mathcal{N}\{y|m_k(x),\sigma_k^2\}, \tag{4}$$

An extension of the method proposed in this paper to the general model (3) can be found in Appendix A. Throughout the paper, we assume that the number of components K is known. In practice, K is unknown and its optimal value is obtained using a data-driven approach such as the information criteria (see Huang and Li (2013)).

The rest of the paper is organized as follows: Sect. 2 presents the traditional (naive) local likelihood approach used to estimate model (4). Section 3 discusses the label-switching problem encountered when estimating the non-parametric term. Section 4 presents the proposed estimation strategy to estimate model (4) and address label-switching. Section 5.2 and 6 presents a simulation study and two real data applications to demonstrate the performance of the proposed approach, respectively. Section 7 concludes the paper and then provides direction for future research.

2 Estimation

Consider a random sample $\{(x_i, y_i) : i = 1, 2, \ldots, n\}$ of size n obtained from model (4). The corresponding log-likelihood function is given as

$$\ell(\boldsymbol{\theta})=\sum_{i=1}^{n}\log\left[\sum_{k=1}^{K}\pi_k\mathcal{N}\{y_i|m_k(x_i),\sigma_k^2\}\right] \tag{5}$$

where $\boldsymbol{\theta}=(\boldsymbol{\pi},\mathbf{m},\boldsymbol{\sigma}^2)=(\pi_1,\ldots,\pi_K;\mathbf{m}_1,\ldots,\mathbf{m}_K;\sigma_1^2,\ldots,\sigma_K^2)$, with $\mathbf{m}_k=(m_k(x_1),\ldots,m_k(x_n))$, for $k=1,2,\ldots,K$, is the vector of all the model parameters.

In order to estimate model (4), we must estimate $\boldsymbol{\theta}$, this is done using a likelihood approach. Direct maximization of the log-likelihood function (5) with respect to $\boldsymbol{\theta}$ poses a challenge due to the presence of both a parametric term $(\boldsymbol{\pi},\boldsymbol{\sigma}^2)$ (henceforth, global parameters) and a non-parametric term $\mathbf{m}$. It is straightforward to maximize (5) with respect to either $\boldsymbol{\pi}$ or $\boldsymbol{\sigma}^2$, however this is not the case for $\mathbf{m}$. Maximizing (5) with respect to $\mathbf{m}$ without any constraints or restrictions on the component regression functions $\mathbf{m}$ would result in estimates that are practically useless due to overfitting (Tibshirani and Hastie 1987). To overcome this problem, we make use of the local-likelihood estimation (LLE) (Tibshirani and Hastie 1987). LLE is an extension of local-polynomial kernel estimation (see Fan and Gijbels (1996)) for likelihood-based models (see Tibshirani and Hastie (1987) for more details).

2.1 Local-polynomial likelihood (LPL) estimator

The local polynomial likelihood (LPL) estimation procedure proceeds as follows. Let $\mathcal{U}=\{u_1,u_2,\ldots,u_N\}$ be a set of N local points on the domain of the covariate x. Assume that at each $u\in\mathcal{U}$, each component regression function $m_k(x)$, for $k=1,2,\ldots,K$, has a $(p+1)^{th}$ derivative. By Taylor expansion, a p^{th} degree polynomial function can be used to locally approximate each component regression function $m_k(x)$, for $k=1,2,\ldots,K$, in the neighbourhood of u, as

$$\begin{aligned}m_k(x)&\approx m_k^{(0)}(u)[x-u]^0+\cdots+\frac{m_k^{(p)}(u)}{p!}[x-u]^p\\&=\sum_{j=0}^{p}m_{kj}(u)[x-u]^j\end{aligned} \tag{6}$$

where $m_k^{(r)}(u)$ denotes the r^{th} derivative of $m_k(u)$ at local point u and $m_{kj}(u)=\frac{m_k^{(j)}(u)}{j!}$ for $k=1,2,\ldots,K$.

Let $\mathbf{m}(u)=(\mathbf{m}_1(u),\ldots,\mathbf{m}_K(u))$, with $\mathbf{m}_k(u)=(m_{k0}(u),m_{k1}(u),\ldots,m_{kp}(u))$, be the vector of all local parameters at local point u. The estimate of $\mathbf{m}_k(u)$, denoted $\hat{\mathbf{m}}_k(u)$, for $k=1,2,\ldots,K$ and $u\in\mathcal{U}$, is obtained by maximizing the following weighted (local) log-likelihood function

$$\begin{aligned}\ell[\mathbf{m}(u)]=\sum_{i=1}^{n}\log\left[\sum_{k=1}^{K}\pi_k\mathcal{N}\left\{y_i|m_k(u),\sigma_k^2\right\}\right]\\\times K_h(x_i-u)\end{aligned} \tag{7}$$

where $m_k(u)=\sum_{j=0}^{p}m_{kj}(u)[x_i-u]^j$, $K_h(x)=K(x/h)/h$ and $K(x)$ is a kernel function used to assign weights to the data points in the neighbourhood of a given local point u and $h>0$ is the bandwidth used to specify the size of the neighbourhood.

From (6), we obtain the estimator of $m_k(u)$, denoted by $\hat{m}_k(u)$, for $k=1,2,\ldots,K$, as

$$\hat{m}_k(u)=\hat{m}_{k0}(u) \tag{8}$$

$\hat{m}_{k0}(u)$ can be referred to as a local polynomial likelihood (LPL) estimator. To estimate $m_k(u)$, for all $u\in\mathcal{U}$, we repeat the above maximization. To obtain the estimated CRFs

$\hat{m}_k(x_i)$, for $i = 1, 2, \ldots, n$ and $k = 1, 2, \ldots, K$, we interpolate over $\hat{m}_k(u_t)$, for $t = 1, 2, \ldots, N$ and $k = 1, 2, \ldots, K$.

2.2 Local-likelihood fitting algorithm

To maximize the likelihood function for any mixture model, the standard algorithm is the Expectation-Maximization (EM) algorithm (Dempster et al. 1977). Recall that we have both global and local parameters. As already mentioned, estimation of the latter uses only the data in a neighbourhood of some local point whereas efficient estimation of the former requires the use of all the observed data. Thus, to satisfy these competing interests, the estimation procedure must be implemented in two stages. In the first-stage, we locally maximize (7) with respect to $\mathbf{m}(u)$, $\boldsymbol{\pi}(u)$ and $\boldsymbol{\sigma}^2(u)$, for $u \in \mathcal{U}$. Let $\hat{m}_k(u) = \hat{m}_{k0}(u)$, for $k = 1, 2, \ldots, K$, be the resulting local parameter estimates obtained from maximizing (7) at local point u. Obtain $\hat{m}_k(x_i)$, for $i = 1, 2, \ldots, n$ and $k = 1, 2, \ldots, K$ by linear interpolation. In the second-stage, given $\hat{m}_k(x_i)$, for $i = 1, 2, \ldots, n$ and $k = 1, 2, \ldots, K$, globally estimate $\boldsymbol{\pi}$ and $\boldsymbol{\sigma}^2$ by maximizing

$$\ell(\boldsymbol{\pi}, \boldsymbol{\sigma}^2) = \sum_{i=1}^{n} \log\Big[\sum_{k=1}^{n} \pi_k \mathcal{N}\{y_i|\hat{m}_k(x_i), \sigma_k^2\}\Big] \tag{9}$$

with respect to $\boldsymbol{\pi}$ and $\boldsymbol{\sigma}^2$. Let $\hat{\mathbf{m}}$ and $(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\sigma}}^2)$ be the resulting estimates from the first-stage and second-stage, respectively. These estimators are the so-called *one-step estimators* of the local and global parameters, see Carroll et al. (1997). The one-step algorithm is an intermediate step of the one-step backfitting algorithm of Xiang and Yao (2018) for fitting model (4). The one-step procedure is summarized in Algorithm 1.

Algorithm 1 One-step algorithm for the SPGMNRs model (4)

Ensure: 1: Maximize (7), for each $u \in \mathcal{U}$, in turn, with respect to $\mathbf{m}(u)$, $\boldsymbol{\pi}(u)$ and $\boldsymbol{\sigma}^2(u)$ and then obtain $\hat{\mathbf{m}}$ by interpolation.

Ensure: 2: Given $\hat{\mathbf{m}}$, maximize (9) with respect to $\boldsymbol{\pi}$ and $\boldsymbol{\sigma}^2$ to obtain the respective estimates $\hat{\boldsymbol{\pi}}$ and $\hat{\boldsymbol{\sigma}}^2$.

Notice that, in Stage 1 of Algorithm 1, we must define N local likelihood functions, where N is the number of local points (see (7)). Thereafter, the natural way to proceed is to apply the EM algorithm to each local likelihood function in turn. This is demonstrated below.

For each $u \in \mathcal{U}$, the EM algorithm to maximize (7) proceeds as follows. Define a $K-$dimensional latent variable $\mathbf{z}_i = (z_{i1}, z_{i2}, \ldots, z_{iK})^\intercal$ for $i = 1, 2, \ldots, n$. The k^{th} element of this variable is set equal to 1 if observation (x_i, y_i) belongs to the k^{th} component and the rest of the elements are set equal to 0. Let $\{(x_i, y_i, \mathbf{z}_i) : i = 1, 2, \ldots, n\}$ be the complete data. Then the corresponding complete-data log likelihood is

$$\ell^c\{\boldsymbol{\theta}(u)\} = \sum_{i=1}^{n}\sum_{k=1}^{K} z_{ik}\big[\log\pi_k(u) + \log\mathcal{N}\{y_i|m_k(u), \sigma_k^2(u)\}\big]K_h(x_i - u) \tag{10}$$

where $\boldsymbol{\theta}(u) = (\boldsymbol{\pi}(u), \mathbf{m}(u), \boldsymbol{\sigma}^2(u))$, with $\boldsymbol{\pi}(u) = (\pi_1(u), \ldots, \pi_K(u))$, $\mathbf{m}(u) = (m_1(u), \ldots, m_K(u))$ and $\boldsymbol{\sigma}^2(u) = (\sigma_1^2(u), \ldots, \sigma_K^2(u))$, is a vector of the local parameters at local point u. At the E-step, we calculate the expected value of $\ell^c(\boldsymbol{\theta}(u))$ with respect to the conditional distribution of $\mathbf{z}$, denoted $Q\{\boldsymbol{\theta}(u)|\boldsymbol{\theta}^{(r)}(u)\}$. This corresponds to calculating the latent variable z_{ik}, for $i = 1, 2, \ldots, n$ and $k = 1, 2, \ldots, K$, using its conditional expectation $\mathbb{E}(z_{ik}|x_i, y_i, \boldsymbol{\theta}^{(r)}(u))$ as

$$\gamma_{ik}^{(r+1)}(u) = \frac{\pi_k^{(r)}(u)\mathcal{N}\{y_i|m_k^{(r)}(u), \sigma_k^{2(r)}(u)\}}{\sum_{\ell=1}^{K}\pi_\ell^{(r)}(u)\mathcal{N}\{y_i|m_\ell^{(r)}(u), \sigma_\ell^{2(r)}(u)\}} \tag{11}$$

for $i = 1, 2, \ldots, n$ and $k = 1, 2, \ldots, K$.

$\gamma_{ik}^{(r+1)}(u)$ is referred to the responsibility of the k^{th} component for the i^{th} observation (see Bishop (2006) and Hastie et al. (2009)). It gives the probability that the i^{th} observation belongs to the k^{th} component.

From (11), it follows that $Q\{\boldsymbol{\theta}(u)|\boldsymbol{\theta}^{(r)}(u)\}$ is

$$Q\{\boldsymbol{\theta}(u)|\boldsymbol{\theta}^{(r)}(u)\} = \sum_{i=1}^{n}\sum_{k=1}^{K}\gamma_{ik}^{(r)}(u)\big[\log\pi_k(u) + \log\mathcal{N}\{y_i|m_k(u), \sigma_k^2(u)\}\big]K_h(x_i - u_t) \tag{12}$$

At the M-step, we maximize $Q(\boldsymbol{\theta}(u)|\boldsymbol{\theta}^{(r)}(u))$ to update $\boldsymbol{\theta}(u)$. For instance, to update $m_k^{(r)}(u)$, for $u \in \mathcal{U}$ and $k = 1, 2, \ldots, K$, let $(\hat{m}_{k0}^{(r)}(u), \hat{m}_{k1}^{(r)}(u), \ldots, \hat{m}_{kp}^{(r)}(u))$ be the maximizers of

$$\sum_{i=1}^{n}\gamma_{ik}^{(r+1)}(u)\log\mathcal{N}\Big\{y_i|\sum_{j=0}^{p}m_{kj}(u)[x_i - u]^j, \sigma_k^2(u)\Big\} \times K_h(x_i - u) \tag{13}$$

Then $m_k^{(r+1)}(u) = \hat{m}_{k0}^{(r+1)}(u)$, for $k = 1, 2, \ldots, K$. An expression for $\hat{m}_{k0}^{(r+1)}(u)$ using matrix notation is useful and can be easily obtained.

Let

$$\mathbf{X} = \begin{bmatrix} 1 & (x_1 - u) & \dots & (x_1 - u)^p \\ 1 & (x_2 - u) & \dots & (x_2 - u)^p \\ \vdots & \vdots & \ddots & \vdots \\ 1 & (x_n - u) & \dots & (x_n - u)^p \end{bmatrix}$$

be the design matrix at local point u and set $\mathbf{y} = (y_1, y_2, \dots, y_n)^\intercal$ and $\mathbf{m}_k(u) = (m_{k0}(u), m_{k1}(u), \dots, m_{kp}(u))^\intercal$. Moreover, let

$$\mathbf{W}_k = \text{diag}\{\gamma_{1k}^{(r+1)}(u)K_h(x_1 - u), \dots, \gamma_{nk}^{(r+1)}(u)K_h(x_n - u)\} \quad (14)$$

be the $n \times n$ diagonal matrix of the weights at local point u. The maximum likelihood criterion can be written as

$$\max_{\mathbf{m}_k(u)} -(\mathbf{y} - \mathbf{X}^\intercal \mathbf{m}_k(u))^\intercal \mathbf{W}_k (\mathbf{y} - \mathbf{X}^\intercal \mathbf{m}_k(u)) \quad (15)$$

Solving (15), gives the following expression for $\hat{m}_{k0}^{(r+1)}(u)$ (and consequently $m_k^{(r+1)}(u)$)

$$m_k^{(r+1)}(u) \equiv \hat{m}_{k0}^{(r+1)}(u) = \mathbf{e}^\intercal \mathbf{A}_k^{-1} \mathbf{B}_k \mathbf{y} \quad (16)$$

where $\mathbf{A}_k = (\mathbf{X}^\intercal \mathbf{W}_k \mathbf{X})$, $\mathbf{B}_k = \mathbf{X}^\intercal \mathbf{W}_k$ and $\mathbf{e}$ is a $(p+1)-$dimensional vector where the first entry is 1 and the other entries are set to zero. The local estimators of the other local parameters $(\boldsymbol{\pi}(u), \boldsymbol{\sigma}^2(u))$ can be obtained in a similar fashion. However, note that for $p > 0$, LPL estimator of $\boldsymbol{\pi}(u)$ does not have a closed form expression. Thus, we estimate this local parameter using the LCE. Furthermore, with the assumption that the regression components are homoscedastic, an LCE can be used to estimate $\boldsymbol{\sigma}^2(u)$ and the additional improvement from using an LPL estimator with $p > 0$ will be negligible.

The above EM algorithm proceeds by repeatedly iterating between the E-Step and M-Step until convergence.

3 Label-switching problem

In this Section, we give a description of the label-switching problem encountered when using Algorithm 1 and we review previous work proposed to address the problem.

3.1 A brief description of the label-switching problem

In order to obtain $\hat{m}_k(x_i)$, for $x_i \notin \mathcal{U}$ and $k = 1, 2, \dots, K$, we interpolate over $\hat{m}_k(u)$, for $u \in \mathcal{U}$. Let $\hat{\mathbf{m}}_k = (\hat{m}_k(x_1), \hat{m}_k(x_2), \dots, \hat{m}_k(x_n))$, for $k = 1, 2, \dots, K$, be the resulting component non-parametric functions. The latter may be non-smooth, exhibiting irregular and non-uniform behaviour. This is because, for each local point $u \in \mathcal{U}$, the M-step is based on a unique set of local responsibilities $\{\gamma_{ik}(u) : i = 1, 2, \dots, n; k = 1, 2, \dots, K\}$. These sets of local responsibilities are not guaranteed to be aligned across the local points. In the event of a misalignment, the labels attached to the mixture components may switch from one local grid point to the next. The practical consequence of this label-switching is estimates of the non-parametric functions that are characterised by discontinuities near the points where the switch took place. Figure 1 illustrates this label-switching phenomenon. The figure shows a simple example of a $K = 2$ component mixture of non-parametric regressions where the regression function of one component is consistently above that of the other component (given by the solid black curves). Consider maximising the local-likelihood functions at three local points $u = -1, 0$ and 1 using Algorithm 1. There are $(2!)^3 = 8$ possible configurations of the component regression functions when we join the local parameter estimates at the three local points. Figure 1 shows two of these configurations, the true configuration given by the solid curves and another configuration given by the dotted curves where the labels of the local parameter estimates at local point -1 and 1 have switched. Note that only 2 of these configurations will result in the correct CRFs. Thus, there is 0.75 probability that Algorithm 1 will result in label-switching. This probability is approximately 1 for $K > 2$.

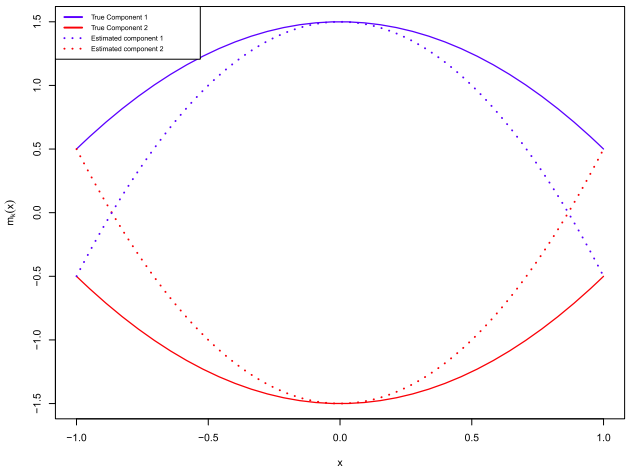


Fig. 1 Label switching problem: **a** A $K = 2$ component case showing the true component regression functions (solid curves). The dotted curves are the fitted component regression functions at three local grid points -1, 0 and 1 which shows that there was a switch at grid points -1 and 1

Thus, Algorithm 1 does not work. Henceforth, we refer to Algorithm 1 as the naive EM algorithm.

3.2 Previous work addressing label-switching

This form of label-switching problem was first mentioned by Huang (2012) and subsequently (Huang and Li 2013). To address the problem, the authors proposed a modified EM-type algorithm that simultaneously maximizes the complete-data local-likelihood functions (12) using the same (common) responsibilities $\gamma_{ik}^{(r)} = \gamma_{ik}^{(r)}(u)$ for all $u \in \mathcal{U}$. In other words, the responsibilities are independent of the local points. This algorithm has been applied by many authors to estimate models of the form (3). It is used in the estimation procedure (PL-EM) of Wu (2016) for estimating SPGM-PLMs. It is an intermediate part of the one-step backfitting (LEM) algorithm of Xiang and Yao (2018) for estimating model (4) and the spline-backfitted kernel (SBK) EM algorithm of Zhang (2017) for estimating SPGMARs.

In particular, the LEM algorithm is a modified version of Algorithm 1 where in Stage 1, the responsibilities (11) at the E-step are replaced by

$$\gamma_{ik}^{(r+1)} = \frac{\pi_k^{(r)}(x_i)\mathcal{N}\{y_i|m_k^{(r)}(x_i), \sigma_k^{2(r)}(x_i)\}}{\sum_{\ell=1}^{K}\pi_\ell^{(r)}(x_i)\mathcal{N}\{y_i|m_\ell^{(r)}(x_i), \sigma_\ell^{2(r)}(x_i)\}} \tag{17}$$

In other words, the responsibilities are independent of the local grid points. This implies that the LEM algorithm and the other above-mentioned EM-type algorithms do not directly maximize the observed local log-likelihood functions but the complete-data local log-likelihood functions. Thus, the calculation of the common responsibilities does not take into account the local information. In a previous work (Skhosana et al. 2022), the authors of the current paper proposed a novel EM-type algorithm that obtains the common responsibilities $\{\gamma_{ik} : i = 1, 2, \ldots, n; k = 1, 2, \ldots, K\}$ from the local responsibilities $\{\gamma_{ik}(u) : i = 1, 2, \ldots, n; k = 1, 2, \ldots, K; u \in \mathcal{U}\}$ thus incorporating the local information. As with the LEM algorithm, the proposed EM-type algorithm is a modified version of Algorithm 1. Briefly, the algorithm replaces the responsibilities (1) with common responsibilities selected as the local responsibilities that correspond to the smoothest estimates of the component non-parametric functions. See the paper for more details. The algorithm was later extended to estimate SPGMPLMs (Skhosana et al. 2023). In the next section, we propose an alternative estimation strategy to address label-switching.

4 The proposed approach

In this section, we propose to address label-switching by reformulating model (4) as a mixture of GMMs. Estimating the mixture of GMMs is, in effect, equivalent to simultaneously estimating all the parameters of each local GMM and hence the component non-parametric functions. Note that, in contrast to existing estimation strategies, this implies that the proposed estimation strategy estimates all the local parameters by maximizing only one likelihood function. Nevertheless, the strategies follow the same principle, simultaneous maximization (estimation) of the local likelihood functions (parameters).

At the end of Sect. 4.1, we show that the proposed approach encompasses, as a special case, an estimation strategy similar to the one proposed in (Skhosana et al. (2022)).

4.1 The mixture of GMMs

As discussed in Sect. 3, label-switching takes place when estimating the local parameters by separately maximizing each local-likelihood function (7). In the following, we propose an estimation strategy that can

1. simultaneously estimate the local parameters in order to address label switching; and
2. select the optimal set of local grid points.

Towards that end, we reformulate the model (4) by introducing a second source of missing information. We assume that the parameters π_k and σ_k^2 are also non-parametric functions of x and let $\mathcal{U} = \{u_1, u_2, \ldots, u_N\}$ be a set of N local points in the domain of the covariate x. It follows that, at each local point u_t, for $t = 1, 2, \ldots, N$, model (4) is a GMM (1)

$$f_{u_t}(y) = \sum_{k=1}^{K}\pi_k(u_t)\mathcal{N}\left\{y|m_k(u_t), \sigma_k^2(u_t)\right\} \tag{18}$$

where $\pi_k = \pi_k(u_t)$, $\mu_k = m_k(u_t)$ and $\sigma_k^2 = \sigma_k^2(u_t)$.

One of these local GMMs can be viewed as a distribution of the response variable y. Since we do not observe the identity of this local GMM, y follows a mixture of these local GMMs

$$\begin{aligned} f(y) &= \sum_{t=1}^{N}\lambda_t f_{u_t}(y) \\ &= \sum_{t=1}^{N}\lambda_t\left[\sum_{k=1}^{K}\pi_k(u_t)\mathcal{N}\left\{y|m_k(u_t), \sigma_k^2(u_t)\right\}\right] \\ &= \sum_{t=1}^{N}\sum_{k=1}^{K}\lambda_t\pi_k(u_t)\mathcal{N}\left\{y|m_k(u_t), \sigma_k^2(u_t)\right\} \end{aligned} \tag{19}$$

where $\lambda_t > 0$ (satisfying $\sum_{t=1}^{N}\lambda_t = 1$) is the mixing proportion, probability or weight. As a mixing proportion, λ_t can be viewed as the relative number of data points that were generated by the t^{th} local GMM. As a mixing probability, λ_t can be interpreted as the probability that a given data point,

say y_i, was generated by the t^{th} local GMM. Thus, the larger the value of λ_t, the more data will be associated with the t^{th} local GMM. Alternatively, the larger the value of λ_t, the more likely that a given data point was generated by the local model $f_{u_t}(y)$. As a mixing weight, λ_t can be viewed as specifying the relative importance of the t^{th} local GMM. The larger the weight, the more significant the local model is to the the overall model. Stated differently, a local model with a small weight ($\lambda_t \approx 0$) is indicative of a sparse local region with few to no data points in the neighbourhood of the local point.. This in turn implies that the local model has little to no information about the data and consequently about the overall model. Thus, the use of the corresponding local point is of little value to the overall fit of the model.

From the previous discussion, the benefits of the weights $(\lambda_1, \lambda_2, \ldots, \lambda_N)$ become apparent. They can be used in various innovative ways as we discuss below.

To estimate model (19), we first need to specify the set of local grid points $\mathcal{U}$. We can follow convention and use the observed covariate values or a set of equally-spaced values from the domain of the covariate. Alternatively, we can use the weights as follows: we begin by setting $\mathcal{U}$ as all the observed covariate values. Next, we modify the EM algorithm by introducing a step between the E- and M- step that determines all the weights that are below a certain threshold, say λ_0, that measures relative importance. Recall that the weights correspond with the local grid points. Thus, all the local grid points whose corresponding weights are below λ_0 are removed and the algorithm continues with the remaining local grid points. We repeat the steps of this modified EM algorithm until convergence. The advantage of this approach is that it finds both the number, N, and location of the grid points.

Another benefit of the weights is in suggesting an alternative approach to address label-switching. As mentioned before, estimating model (19) is equivalent to simultaneously estimating all the local parameters thus addressing label-switching. Moreover, the estimation can be done using the classical EM algorithm or the modified EM algorithm described above. An alternative strategy to addressing label-switching is to estimate all the local GMMs and choose the one with the largest weight and use its resulting local responsibilities as the common responsibilities used to maximise all the local-likelihood functions. In this manner, this proposed alternative approach is, in principle, similar to the approach proposed in Skhosana et al. (2022).

Note that since the set of local points $\mathcal{U}$ is determined by the range $\mathcal{X}$ of the covariate x, model (19) represents a reformulation of model (4). Moreover, due to the mixture of mixtures structure (19), the new model is a hierarchical. To highlight this hierarchy, model (4) can be written as

$$f(y|\mathbf{X}=\mathbf{x}) = \sum_{t=1}^{N} \lambda_t \sum_{k=1}^{K} \pi_{t,k} \mathcal{N}\left\{y|m_{t,k}, \sigma^2_{t,k}\right\}, \tag{20}$$

where $\pi_{t,k} = \pi_k(u_t)$, $m_{t,k} = m_k(u_t)$ and $\sigma^2_{t,k} = \sigma^2_k(u_t)$.

4.2 Estimation procedure

In this section, we propose an estimation procedure for model (20). Consider a random sample $\{(x_i, y_i) : i = 1, 2, \ldots, n\}$ from model (20). The corresponding log-likelihood function is

$$\ell_0(\boldsymbol{\lambda}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \log\left[\sum_{t=1}^{N}\sum_{k=1}^{K} \lambda_t \pi_{t,k} \mathcal{N}\left\{y|m_{t,k}, \sigma^2_{t,k}\right\}\right] \tag{21}$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_N)$ and $\boldsymbol{\theta} = (\boldsymbol{\theta}(u_1), \ldots, \boldsymbol{\theta}(u_N))$ with $\boldsymbol{\theta}(u_t) = (\boldsymbol{\pi}_{t\cdot}, \mathbf{m}_{t\cdot}, \boldsymbol{\sigma}^2_{t\cdot})$, $\boldsymbol{\pi}_{t\cdot} = (\pi_{t,1}, \ldots, \pi_{t,K})$, $\mathbf{m}_{t\cdot} = (m_{t,1}, \ldots, m_{t,K})$ and $\boldsymbol{\sigma}^2_{t\cdot} = (\sigma^2_{t,1}, \ldots, \sigma^2_{t,K})$, for $t = 1, 2, \ldots, N$.

We propose a modified Expectation Conditional Maximization (ECM-) type (Meng and Rubin 1993) to maximize (21). The ECM is a modified version of the classical EM algorithm where the M-step is split into simpler M-steps also known as conditional M (CM-) steps. Note that we now have two latent variables. The first latent variable serves as an indicator variable for the identity of the local model that generated a given data point. For each data point, we define this latent variable as $\mathbf{v}_i = (v_{i1}, v_{i2}, \ldots, v_{iN})$ where $v_{it} = 1$ if the i^{th} data point belongs or was generated by the t^{th} local model and 0 otherwise. The second latent variable, $\mathbf{z}_{it}$, serves as an indicator variable for the identity of the Gaussian component, from the t^{th} local model, that generated a given data point. Thus, $\mathbf{z}_{it} = (z_{it1}, z_{it2}, \ldots, z_{itK})$, where $z_{itk} = 1$ if the i^{th} data point was generated by the k^{th} component from the t^{th} local mixture model. Given the completed-data $\{(x_i, y_i, \mathbf{z}_{it}, \mathbf{v}_i) : i = 1, 2, \ldots, n; t = 1, 2, \ldots, N\}$, the corresponding (complete-data) log-likelihood is

$$\ell^c_0(\boldsymbol{\lambda}, \boldsymbol{\theta}) = \ell^{1c}_0(\boldsymbol{\lambda}) + \ell^{2c}_0(\boldsymbol{\pi}) + \ell^{3c}_0(\boldsymbol{\theta}), \tag{22}$$

where

$$\ell^{1c}_0(\boldsymbol{\lambda}) = \sum_{t=1}^{N}\sum_{i=1}^{n} v_{it} \log\lambda_t,$$

$$\ell^{2c}_0(\boldsymbol{\pi}) = \sum_{t=1}^{N}\sum_{i=1}^{n}\sum_{k=1}^{K} v_{it} z_{itk} \log\pi_{t,k},$$

$$\ell^{3c}_0(\boldsymbol{\theta}_2) = \sum_{t=1}^{N}\sum_{i=1}^{n}\sum_{k=1}^{K} v_{it} z_{itk} \log\mathcal{N}\left\{y_i|m_{t,k}, \sigma^2_{t,k}\right\},$$

with $\boldsymbol{\pi} = (\boldsymbol{\pi}_{t\cdot})_{1\le t\le N}$ and $\boldsymbol{\theta} = (\mathbf{m}_{t\cdot}, \boldsymbol{\sigma}^2_{t\cdot})_{1\le t\le N}$.

Let $\mathcal{T} = \{t|\lambda_t > \lambda_0\}$ be the set of all indices of the local models where the weights λ_t's are greater than some constant $0 < \lambda_0 < 1$. The constant λ_0 is a threshold that specifies a level beyond which a local point can be considered to be significant in the sense discussed above. The threshold λ_0 is a free parameter (hyperparameter) that can be chosen subjectively or objectively based on the data. More details will be given below.

At the $(r+1)^{th}$ iteration of the E-step, we calculate the conditional expected value of $\ell_0^{1c}(\boldsymbol{\lambda})$, $\ell_0^{2c}(\boldsymbol{\pi})$ and $\ell_0^{3c}(\boldsymbol{\theta})$, denoted by $Q(\boldsymbol{\lambda}|\boldsymbol{\lambda}^{(r)})$, $Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(r)})$ and $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(r)})$, respectively, with respect to the conditional distribution of $\mathbf{v}$ and $\mathbf{z}$. This corresponds to estimating the latent variables v_{it} and z_{itk} for $i = 1, 2, \ldots, n$, $k = 1, 2, \ldots, K$ and $t \in \mathcal{T}^{(r)}$ using $\mathbb{E}[v_{it}|x_i, y_i, \lambda_t^{(r)}, \boldsymbol{\pi}_{t\cdot}^{(r)}, \boldsymbol{\theta}^{(r)}(u_t)]$ and $\mathbb{E}[z_{itk}|x_i, y_i, \mathbf{v}_i, \boldsymbol{\pi}_{t\cdot}^{(r)}, \boldsymbol{\theta}^{(r)}(u_t)]$, respectively. Using Bayes' theorem, the latter are calculated as

$$P(v_{it}=1|y_i,x_i) = \frac{P(v_{it}=1)P(y_i|v_{it}=1,x_i)}{P(y_i|x_i)}$$
$$\hat{v}_{it}^{(r+1)} = \frac{\lambda_t^{(r)}\sum_{k=1}^K \pi_{t,k}^{(r)}\mathcal{N}\left\{y_i|m_{t,k}^{(r)},\sigma_{t,k}^{2(r)}\right\}}{\sum_{\ell\in\mathcal{T}^{(r)}}\lambda_\ell^{(r)}\sum_{k=1}^K \pi_{\ell,k}^{(r)}\mathcal{N}\left\{y_i|m_{\ell,k}^{(r)},\sigma_{\ell,k}^{2(r)}\right\}} \quad (23)$$

and

$$P(z_{itk}=1|\mathbf{v}_i,y_i,x_i) = \frac{P(z_{itk}=1|\mathbf{v}_i)P(y_i|z_{itk}=1,\mathbf{v}_i,x_i)}{P(y_i|\mathbf{v}_i,x_i)}$$
$$\hat{z}_{itk}^{(r+1)} = \frac{\pi_{t,k}^{(r)}\mathcal{N}\left\{y_i|m_{t,k}^{(r)},\sigma_{t,k}^{2(r)}\right\}}{\sum_{\ell=1}^K \pi_{t,\ell}^{(r)}\mathcal{N}\left\{y_i|m_{t,\ell}^{(r)},\sigma_{t,\ell}^{2(r)}\right\}} \quad (24)$$

Note that (24) is similar to (11), with the difference being that we now have to take into account the value of $\mathbf{v}_i$, for $i = 1, 2, \ldots, n$. Expression (23) $\hat{v}_{it}^{(r+1)}$ can be interpreted as the probability that the i^{th} data point was generated by the t^{th} local model. In other words, it represents the responsibility of the t^{th} local model for the i^{th} data point. Given that the i^{th} data point belongs to the t^{th} local model, $\hat{z}_{itk}^{(r+1)}$ has the same interpretation as $\gamma_{ik}(u_t)$.

After replacing v_{it} with $\hat{v}_{it}$ and z_{itk} with $\hat{z}_{itk}$ in (22), we obtain $Q(\boldsymbol{\lambda}|\boldsymbol{\lambda}^{(r)})$, $Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(r)})$ and $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(r)})$.

At the first CM-step, on the $(r+1)^{th}$ iteration, we update $\lambda^{(r)}$, by maximizing $Q(\boldsymbol{\lambda}|\boldsymbol{\lambda}^{(r)})$, given $\mathcal{T}^{(r)}$, to obtain

$$\hat{\lambda}_t^{(r+1)} = \frac{\sum_{i=1}^n \hat{v}_{it}^{(r+1)}}{n} \qquad \text{for } t \in \mathcal{T}^{(r)} \quad (25)$$

To update $\mathcal{T}^{(r)}$, let

$$\mathcal{T}^{(r+1)} = \{t|\hat{\lambda}_t^{(r+1)} > \lambda_0\}. \quad (26)$$

At the second CM-step, we update $\boldsymbol{\pi}^{(r)}$ and $\boldsymbol{\theta}^{(r)}$ by maximizing $Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(r)})$ and $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(r)})$, respectively. Note that if we maximize, say $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(r)})$, with respect to $m_{t,k}$, for $t \in \mathcal{T}^{(r)}$, the resulting estimated function $m_k(x_i)$, for $i = 1, 2, \ldots, n$, may exhibit wild oscillations. This is because, at each local point u_t, the contribution of all the covariate values $\{x_1, x_2, \ldots, x_n\}$ to the likelihood function is equal. Thus, the local parameter estimate, say $\hat{m}_{t,k}$, will be sensitive to values of the covariate that are not within its neighbourhood. This might possibly lead to a biased estimate.

To remedy this, we propose to maximize kernel weighted versions of these complete-data log-likelihood functions

$$Q^w(\boldsymbol{\pi}|\boldsymbol{\pi}^{(r)}) = \sum_{t\in\mathcal{T}^{(r+1)}}\sum_{i=1}^n\sum_{k=1}^K \hat{v}_{it}^{(r+1)}\hat{z}_{itk}^{(r+1)} \times K_h(x_i-u_t)\log\pi_{t,k} \quad (27)$$

$$Q^w(\boldsymbol{\theta}|\boldsymbol{\theta}^{(r)}) = \sum_{t\in\mathcal{T}^{(r+1)}}\sum_{i=1}^n\sum_{k=1}^K \hat{v}_{it}^{(r+1)}\hat{z}_{itk}^{(r+1)} \times K_h(x_i-u_t)\log\mathcal{N}\left\{y_i|m_{t,k},\sigma_{t,k}^2\right\} \quad (28)$$

where the kernel function $K_h(x_i - u_t)$ is used to provide a weight to x_i relative to the local point u_t. Note that if we choose $K_h(\cdot)$ as the uniform kernel function, the above problem persists. Thus, $Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(r)})$ and $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(r)})$ are implicitly kernel weighted, where the kernel function is uniform.

Maximizing (27) with respect to $\pi_{t,k}$, we get

$$\pi_{t,k}^{(r+1)} = \frac{\sum_{i=1}^n \hat{v}_{it}^{(r+1)}\hat{z}_{itk}^{(r+1)}K_h(x_i-u_t)}{\sum_{i=1}^n \hat{v}_{it}^{(r+1)}K_h(x_i-u_t)} \quad (29)$$

Maximizing (28) with respect to $m_{t,k}$ and $\sigma_{t,k}^2$ we get

$$m_{t,k}^{(r+1)} = \frac{\sum_{i=1}^n w_{itk}^{(r+1)}y_i}{\sum_{i=1}^n w_{itk}^{(r+1)}} \quad (30)$$

$$\sigma_{t,k}^{2(r+1)} = \frac{\sum_{i=1}^n w_{itk}^{(r+1)}(y_i - m_{t,k}^{(r+1)})^2}{\sum_{i=1}^n w_{itk}^{(r+1)}} \quad (31)$$

where $w_{itk}^{(r+1)} = \hat{v}_{it}^{(r+1)}\hat{z}_{itk}^{(r+1)}K_h(t_i-u_t)$.

We repeat the above E- and CM-steps until convergence.

The derivations of (29), (30) and (31) are given in Appendix B.

Let $r = R$ be the iteration index at convergence. To obtain $\hat{m}_k(x_i)$ for $i = 1, 2, \ldots, n$ and $k = 1, 2, \ldots, K$, we linearly interpolate over $m_{t,k}^{(R)}$, for $k = 1, 2, \ldots, K$ and $t \in \mathcal{T}^{(R)}$. The

first-stage estimates of the other non-parametric functions can be obtained in a similar manner.

We refer to the above algorithm as the model-based EM-type (henceforth, MB-EM) algorithm. Model-based because of its hierarchical mixture of mixtures structure as well as its ability to select the local grid points in a principled manner by making use of a probability distribution (model) and EM because it is a modified version of the classical EM algorithm.

Note the following properties of the MB-EM algorithm:

Choice of λ_0: Based on empirical evidence in Sect. 5.2, we showed that the algorithm is not sensitive to the choice of the parameter λ_0;

Ascent property: An important and attractive property of the classical EM algorithm is the ascent property. That is, at each iteration $\ell_0^{(r+1)}(\boldsymbol{\lambda}, \boldsymbol{\beta}) \geq \ell_0^{(r)}(\boldsymbol{\lambda}, \boldsymbol{\beta})$. Empirical evidence shows that the MB-EM algorithm also has this property;

Convergence: The convergence of the algorithm can be evaluated in either one of the following ways: (1) Stop the algorithm when the increase in the likelihood from one iteration to the next is below some small pre-specified threshold. (2) Stop the algorithm when the change in the estimated parameters from one iteration to the next is smaller than some small value. For instance, $||\boldsymbol{\lambda}^{(r+1)} - \boldsymbol{\lambda}^{(r)}||_1 < 10^{-5}$ or $||\boldsymbol{\lambda}^{(r+1)} - \boldsymbol{\lambda}^{(r)}||_2 < 10^{-5}$, where $||\cdot||_1$ and $||\cdot||_2$ denotes the L_1 and L_2 norm, respectively, on $\mathbb{R}^{N^{(r+1)}}$. The superscript $N^{(r+1)}$ is used to denote the number of local grid points at the $(r+1)^{th}$ iteration.

Algorithm complexity: At each iteration of the MB-EM algorithm, the overall time complexity of the E-step is $O(n \times N^{(r+1)} \times K)$. In comparison, the time complexity of the NaiveEM algorithm and the LEM algorithm is $O(n \times N \times K)$ and $O(n \times K)$, respectively. It is known that the slow convergence of the classical EM algorithm is largely as a result of the E-step computations (see Chapter 2 of Fruhwirth-Schnatter et al. (2019)). This implies that the LEM algorithm should be computationally faster than the proposed algorithm.

Note that the overall time complexity of the proposed CM-steps is $O(n \times N^{(r+1)} \times K)$ and that of the M-step of both the NaiveEM algorithm and LEM algorithm is $O(n \times N \times K)$. However, as shown in the simulations, the computational advantage of the LEM comes at the cost of inaccurate estimation.

Let $(\hat{\boldsymbol{\pi}}, \hat{\mathbf{m}}, \hat{\boldsymbol{\sigma}}^2)$ be the estimates of the parametric and non-parametric terms $(\boldsymbol{\pi}, \boldsymbol{\sigma})$ and $\mathbf{m}$, respectively, obtained from estimating model (20). Note that when defining the mixture of GMMs (20), we assumed that the global parameters $(\boldsymbol{\pi}, \boldsymbol{\sigma}^2)$ were local. However, to obtain efficient estimates of the global parameters, we must use all of the data during estimation. Thus, in an effort to improve the estimates $(\hat{\boldsymbol{\pi}}, \hat{\boldsymbol{\sigma}}^2)$, given $\hat{\mathbf{m}}$, we propose updated estimates $\tilde{\boldsymbol{\pi}}$ and $\tilde{\boldsymbol{\sigma}}^2$ obtained by maximizing the global log-likelihood function

$$\ell_1(\boldsymbol{\pi}, \boldsymbol{\sigma}^2) = \sum_{i=1}^{n} \log\Big[\sum_{k=1}^{K} \pi_k \mathcal{N}\{y_i | \hat{m}_k(x_i), \sigma_k^2\}\Big] \tag{32}$$

Given the global parameter estimates $\tilde{\boldsymbol{\pi}}$ and $\tilde{\boldsymbol{\sigma}}^2$, we can improve the local estimate $\hat{\mathbf{m}}$. To achieve this, we propose the estimate $\tilde{\mathbf{m}}$ obtained by maximizing the local log-likelihood function

$$\ell_2[\mathbf{m}(u_t)] = \sum_{i=1}^{n} \log\Big\{\sum_{k=1}^{K} \tilde{\pi}_k \mathcal{N}\{y_i | m_k(u_t), \tilde{\sigma}_k^2\}\Big\} \times K_h(x_i - u_t), \tag{33}$$

over all grid points $u_t, t = 1, 2, \ldots, N$.

Note that the global parameter estimates $\tilde{\boldsymbol{\pi}}$ and $\tilde{\boldsymbol{\sigma}}^2$ are well labelled. This implies that the local log-likelihood functions (33) can be maximized separately without being concerned about label switching.

In summary, the proposed estimation procedure proceeds in two stages. In the first stage, we obtain the estimates $(\hat{\boldsymbol{\pi}}, \hat{\mathbf{m}}, \hat{\boldsymbol{\sigma}}^2)$. Thereafter, in the second stage, we obtain the estimates $(\tilde{\boldsymbol{\pi}}, \tilde{\boldsymbol{\sigma}}^2, \tilde{\mathbf{m}})$.

We refer to the second-stage estimates $(\tilde{\boldsymbol{\pi}}, \tilde{\boldsymbol{\sigma}}^2, \tilde{\mathbf{m}})$ as the one-step backfitting estimate.

4.3 One-step backfitting algorithm

In this section, we propose a one-step backfitting algorithm to obtain the one-step backfitting estimates $\tilde{\boldsymbol{\theta}}$. The MB-EM algorithm is an intermediate part of this algorithm.

Stage 0: Initializing the algorithm

Obtain appropriate initial estimates of the global parameters and the non-parametric functions, denoted $(\mathbf{m}^{(0)}, \boldsymbol{\sigma}^{2(0)})$ and $\boldsymbol{\pi}^{(0)}$, respectively, by making use of, say mixture of regression splines (see Xiang and Yao (2018)). Moreover, let $\mathcal{U}$ be the set of N grid points, $\mathcal{T}^{(0)} = \{1, 2, \ldots, N\}$ be the initial set of indices and specify λ_0. In our preliminary numerical experiments, we found that the model estimates are not sensitive to the specified value of λ_0 provided that the value is not chosen too large. In this case, the algorithm may fail because it is using few to no local points. In the extreme case, $\mathcal{T}$ will be empty. Thus, we recommend using the parameter value $\lambda_0 = 1 \times 10^{-5}$.

Stage 1: MB-EM algorithm to maximize ℓ_0

Let $\boldsymbol{\lambda}^{(r)}$, $\boldsymbol{\theta}_1^{(r)}$ and $\boldsymbol{\theta}_2^{(r)}$ be the parameter estimates obtained at the r^{th} iteration.

E-Step: At the $(r+1)^{th}$ iteration, calculate $Q(\boldsymbol{\lambda}|\boldsymbol{\lambda}^{(r)})$, $Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(r)})$ and $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(r)})$ by first estimating $\mathbf{v}_i$ and $\mathbf{z}_i$, for $i = 1, 2, \ldots, n$, using (23) and (24), respectively.

CM-Step 1: Maximize $Q(\boldsymbol{\lambda}|\boldsymbol{\lambda}^{(r)})$ to obtain $\boldsymbol{\lambda}^{(r+1)}$ and $\mathcal{T}^{(r+1)}$ using (25) and (26), respectively.

CM-Step 2: Given $\mathcal{T}^{(r+1)}$, maximize $Q(\boldsymbol{\pi}|\boldsymbol{\pi}^{(r)})$ and $Q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(r)})$ to obtain $\boldsymbol{\pi}^{(r+1)}$ and $\boldsymbol{\theta}^{(r+1)}$ using (29), (30) and (31), respectively.

Repeat the above E- and CM-steps until convergence.

Stage 2(a): EM algorithm to maximize ℓ_1

Given $\hat{\mathbf{m}}$ obtained from Stage 1, we obtain the global estimates $\tilde{\boldsymbol{\pi}}$ and $\tilde{\boldsymbol{\sigma}}^2$ of the global parameters $\boldsymbol{\pi}$ and $\boldsymbol{\sigma}^2$, respectively, by maximizing ℓ_1 in (32) using the usual EM algorithm.

E-Step: At the $(r+1)^{th}$ iteration, calculate the expected value of the latent variable as

$$\gamma_{ik}^{(r+1)} = \frac{\pi_k^{(r)} \mathcal{N}\{y_i|\hat{m}_k(x_i), \sigma_k^{2(r)}\}}{\sum_{\ell=1}^{K} \pi_\ell^{(r)} \mathcal{N}\{y_i|\hat{m}_\ell(x_i), \sigma_\ell^{2(r)}\}} \tag{34}$$

M-Step: We obtain the global parameter estimates $\boldsymbol{\pi}^{(r+1)}$ and $\boldsymbol{\sigma}^{2(r+1)}$ using the following equations

$$\pi_k^{(r+1)} = \frac{\sum_{i=1}^{n} \gamma_{ik}^{(r+1)}}{n} \tag{35}$$

$$\sigma_k^{2(r+1)} = \frac{\sum_{i=1}^{n} \gamma_{ik}^{(r+1)} \left(y_i - \hat{m}_k(x_i)\right)^2}{\sum_{i=1}^{n} \gamma_{ik}^{(r+1)}} \tag{36}$$

Repeat the above E- and M-step until convergence

Stage 2(b): EM algorithm to maximize ℓ_2

Given $\tilde{\boldsymbol{\pi}}$ and $\tilde{\boldsymbol{\sigma}}^2$ obtained from Stage 2(a), we propose an improved estimate of the component non-parametric functions, denoted by $\tilde{\mathbf{m}}$, obtained by maximizing each local log-likelihood function in (33) using the usual EM algorithm.

E-Step: At the $(r+1)^{th}$ iteration, calculate the expected value of the latent variable as

$$\gamma_{ik}^{(r+1)}(u_t) = \frac{\tilde{\pi}_k \mathcal{N}\{y_i|m_k^{(r)}(u_t), \tilde{\sigma}_k^2\}}{\sum_{\ell=1}^{K} \tilde{\pi}_\ell \mathcal{N}\{y_i|m_\ell^{(r)}(u_t), \tilde{\sigma}_\ell^2\}} \tag{37}$$

M-Step: We obtain $m_k^{(r+1)}(u_t)$, for $t = 1, 2, \ldots, N$, using (16).

Repeat the above E- and M-step until convergence.

At convergence of the EM algorithm of Stage 2(b), we obtain $\tilde{\mathbf{m}} = (\tilde{\mathbf{m}}_1, \tilde{\mathbf{m}}_2, \ldots, \tilde{\mathbf{m}}_K)$, where $\tilde{\mathbf{m}}_k = (\tilde{m}_k(x_1), \ldots, \tilde{m}_k(x_n))$ by linear interpolation over $m_k^{(R)}(u_t)$ for $t = 1, 2, \ldots, N$ and $k = 1, 2, \ldots, K$.

We refer to the estimates $\tilde{\boldsymbol{\pi}}$, $\tilde{\boldsymbol{\sigma}}^2$ and $\tilde{\mathbf{m}}$ as the one-step backfitting estimates. To further improve the one-step backfitting estimates, we can repeat Stage 2(a) and 2(b) of the algorithm until convergence.

Remark 1 Note that label-switching is not a concern when obtaining the non-parametric estimates $\tilde{\mathbf{m}}$. This is because the global parameter estimates $\tilde{\boldsymbol{\pi}}$ and $\tilde{\boldsymbol{\sigma}}^2$ are the same across all the local points in Stage 2(b).

5 Simulations

In this section, we perform numerical experiments to demonstrate the performance of the proposed method. The purpose of these experiments is two fold. First, we want to demonstrate the effectiveness of the proposed method towards addressing label-switching. Second, we want to evaluate the accuracy of the proposed one-step backfitting estimators. Moreover, we want to demonstrate the practical suitability of the fitted model based on these estimators. For the rest of the chapter, we refer to the proposed model-based one-step backfitting algorithm, simply as the MB-EM algorithm. All numerical experiments are performed using the R programming language (R Core Team 2023).

5.1 Choosing the bandwidth, *h*

Among other things, local polynomial fitting requires the bandwidth, h. In practice, this component is usually chosen using a data-driven approach such as cross-validation (CV). In this paper, we propose a generalized CV (GCV) approach (see Craven and Wahba (1979)) for bandwidth selection. The GCV approach is less computationally intensive compared to the ordinary multi-fold CV approach, it alleviates the tendency of the ordinary CV approach to undersmooth (Hastie et al. 2009) and, more importantly, it allows us to express the CV error as a function of the complexity (number of parameters) of the estimator. This is important when comparing two different local polynomial estimators as will be shown in section 5.2.

Let $\hat{\mathbf{y}}_k = (\hat{y}_{1k}, \ldots, \hat{y}_{nk})^\intercal$ be the vector of fitted values, where $\hat{y}_{ik} = \tilde{m}_k(x_i)$ is the one-step backfitting estimate of $m_k(x_i)$. Using (16), it can be shown that $\tilde{m}_k(x_i) = \mathbf{s}(x_i)\mathbf{y}$, where $\mathbf{s}(x_i) = \mathbf{e}^\intercal \mathbf{A}_k^{-1} \mathbf{B}_k$ after replacing u by x_i. Then

$$\hat{\mathbf{y}}_k = \mathbf{S}_{hk}\mathbf{y} \qquad \text{for } k = 1, 2, \ldots, K \tag{38}$$

where $\mathbf{S}_{hk} = (\mathbf{s}(x_1), \mathbf{s}(x_2), \ldots, \mathbf{s}(x_n))^\intercal$ is known as the smoother matrix, see Buja et al. (1989) for more details. The

first subscript shows that the smoother matrix depends on the bandwidth h, among others. We propose the following GCV error

$$\begin{aligned}\text{GCV}(h) &= \sum_{k=1}^{K} \frac{(\mathbf{y}-\hat{\mathbf{y}}_k)^\intercal \mathbf{W}_k(\mathbf{y}-\hat{\mathbf{y}}_k)/n_k}{(1-\text{df}_k/n_k)^2} \\ &= \sum_{k=1}^{K} \frac{\text{ASE}_k}{(1-\text{df}_k/n)^2} \end{aligned} \tag{39}$$

where $\text{ASE}_k=(\mathbf{y}-\hat{\mathbf{y}}_k)^\intercal \mathbf{W}_k(\mathbf{y}-\hat{\mathbf{y}}_k)/n_k$, with $n_k = \sum_{i=1}^{n} \hat{\gamma}_{ik}$, is the average squared error (ASE) of the fitted k^{th} CRF, $\mathbf{W}_k = \text{diag}(\hat{\gamma}_{1k}, \hat{\gamma}_{2k}, \ldots, \hat{\gamma}_{nk})$ is the diagonal matrix of the responsibilities of the k^{th} component obtained based on the one-step backfitting estimates $\tilde{\boldsymbol{\theta}} = (\tilde{\boldsymbol{\pi}}, \tilde{\boldsymbol{\sigma}}^2, \tilde{\mathbf{m}})$ and

$$\text{df}_k = \text{trace}(\mathbf{S}_{hk}) = \sum_{i=1}^{n} s_{ii} \tag{40}$$

where s_{ii}, for $i = 1, 2, \ldots, n$, are the diagonal entries of the smoother matrix $\mathbf{S}_{hk}$. Expression (40) denotes the degrees of freedom of the k^{th} component. The latter quantifies the complexity of the fitted CRF as it gives the effective number of parameters used to estimate the k^{th} CRF, see Buja et al. (1989) for more details. This concept is very useful for comparing local polynomial estimates of different degrees. We will demonstrate this in our simulation study.

5.2 Simulation studies

For each of our numerical experiments, we generate 500 data sets of sizes $n = 250$, 500, 1000 and 2000. We make use of $N = 100$ local points chosen uniformly on the domain of x. In all our simulations, the covariate x is generated from a uniform distribution on the interval (0, 1). We make use of the Gaussian kernel function.

To initialize the proposed method, we make use of the mixture of regression splines (MRS) (Xiang and Yao 2018). To estimate the MRS, we make use of the `bs` and `ns` functions from the R package `splines`. The knots are chosen as the quartiles of x.

To evaluate the performance of the proposed method, we make use of the following measures:

Root average squared error (RASE):

$$\text{RASE}^2(\mathbf{y}) = \sum_{k=1}^{K} \text{ASE}_k \tag{41}$$

$$\text{RASE}^2(\mathbf{m}) = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\left[\tilde{m}_k(x_i) - m_k(x_i)\right]^2 \tag{42}$$

$$\text{RASE}^2(f_{\boldsymbol{\theta}}) = \frac{1}{n}\sum_{i=1}^{n}\left[\hat{f}_{\tilde{\boldsymbol{\theta}}}(y_i|x_i) - f_{\boldsymbol{\theta}}(y_i|x_i)\right]^2 \tag{43}$$

Adjusted Rand Index (ARI) is used to evaluate the clustering ability of the fitted model (ARI; Hubert and Arabie (1985)).

Kolmogorov-Smirnov (KS) statistic is used to assess the goodness of the fit $\hat{F}_{\tilde{\boldsymbol{\theta}}}$ as

$$\text{KS} = \max_i |F_{\boldsymbol{\theta}}(y_i|x_i) - \hat{F}_{\tilde{\boldsymbol{\theta}}}(y_i|x_i)|, \tag{44}$$

for $i = 1, 2, \ldots, n$.

Finally, to evaluate the accuracy of the estimated parameters $(\tilde{\boldsymbol{\pi}}, \tilde{\boldsymbol{\sigma}}^2)$, we make use of the ASE.

where $f_{\boldsymbol{\theta}}$ and $F_{\boldsymbol{\theta}}$ are the true conditional probability distribution (4) and the corresponding cumulative conditional probability distribution, respectively, and $\hat{f}_{\tilde{\boldsymbol{\theta}}}$ and $\hat{F}_{\tilde{\boldsymbol{\theta}}}$ are the respective estimates.

Evaluating the performance of the proposed method towards addressing label-switching We first demonstrate that the proposed method is less sensitive to label-switching and produces reliable model estimates. First, we consider data generated from a $K = 2$ component SPGMNRs given in Table 1.

The CRFs, $m_k(x)'$s, in Table 1 are given in Fig. 2a. We fit model (4) for $K = 2$ on the generated data using the LCEs obtained via the naive EM algorithm (*naiveEM*), the proposed MB-EM algorithm and the local EM algorithm of Xiang and Yao (2018). The bandwidths were chosen as 0.05, 0.045, 0.04 and 0.035 for the sample sizes $n = 250$, 500, 1000 and 2000, respectively.

Fig. 3 shows examples of the fitted CRFs for typical samples of sizes $n = 250$, 500, 1000 and 2000. These fitted CRFs were each chosen from the fitted models, among the 500 replicates, with the largest likelihood value based on the results of the naiveEM. As can be seen from the figure, the estimates based on the naiveEM (right-column) are wiggly and non-smooth whereas those based on both the proposed MB-EM (center) and the LEM (right-column) algorithm appear to be stable. For a full picture of the performance of the proposed method compared with both the naiveEM and LEM algorithm, Table 2 gives the average and standard devia-

Table 1 Data generating model

k	1	2
π_k	0.65	0.35
$m_k(x)$	$1-\cos(2\pi x)$	$\exp(2x)$
σ_k^2	0.09	0.16

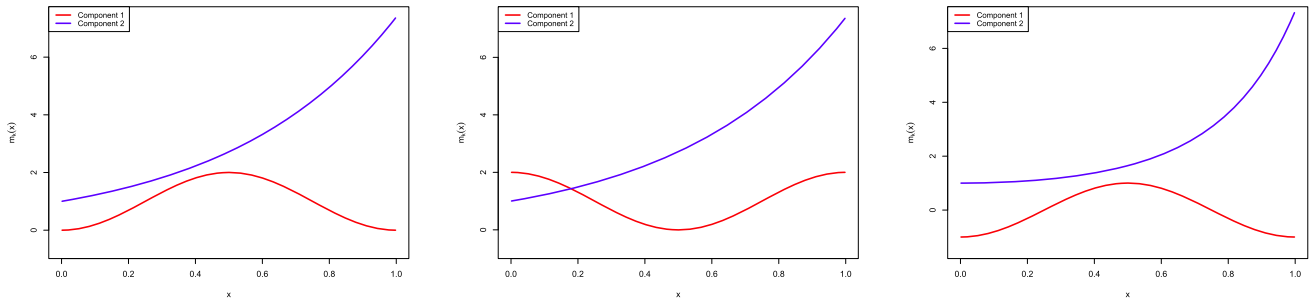


Fig. 2 CRFs for the model in (**a**) Table 1, (**b**) Table 3 and (**c**) Table 5

Table 2 Average (and standard deviations) of the performance measures over the 500 replications using data generated from the model in Table 1

n		RASE($f_{\boldsymbol{\theta}}$)	KS	ASE$_{\pi}$	ASE$_{\sigma_1^2}$	ASE$_{\sigma_2^2}$	RASE(**m**)	ARI
250	naiveEM	0.142 (0.026)	0.020 (0.009)	0.082 (0.038)	0.003 (0.004)	0.005 (0.002)	0.280 (0.075)	0.676 (0.062)
	MB-EM	0.108 (0.022)	0.022 (0.01)	0.082 (0.043)	0.004 (0.004)	0.005 (0.002)	0.186 (0.043)	0.699 (0.060)
	LEM	0.114 (0.028)	0.023 (0.011)	0.074 (0.042)	0.003 (0.004)	0.005 (0.002)	0.185 (0.043)	0.686 (0.106)
500	naiveEM	0.112 (0.019)	0.016 (0.006)	0.085 (0.028)	0.004 (0.003)	0.005 (0.002)	0.239 (0.053)	0.686 (0.043)
	MB-EM	0.081 (0.016)	0.016 (0.007)	0.081 (0.040)	0.005 (0.004)	0.004 (0.002)	0.140 (0.032)	0.703 (0.045)
	LEM	0.097 (0.043)	0.021 (0.015)	0.075 (0.039)	0.004 (0.003)	0.004 (0.002)	0.145 (0.033)	0.641 (0.178)
1000	naiveEM	0.090 (0.014)	0.013 (0.004)	0.086 (0.025)	0.004 (0.002)	0.005 (0.001)	0.217 (0.038)	0.693 (0.030)
	MB-EM	0.061 (0.011)	0.012 (0.005)	0.078 (0.041)	0.005 (0.003)	0.004 (0.002)	0.107 (0.021)	0.709 (0.028)
	LEM	0.064 (0.020)	0.012 (0.007)	0.076 (0.036)	0.004 (0.003)	0.004 (0.002)	0.112 (0.023)	0.701 (0.067)
2000	naiveEM	0.075 (0.010)	0.012 (0.003)	0.087 (0.019)	0.005 (0.002)	0.005 (0.001)	0.201 (0.029)	0.694 (0.020)
	MB-EM	0.047 (0.008)	0.009 (0.004)	0.081 (0.038)	0.005 (0.003)	0.004 (0.002)	0.084 (0.016)	0.707 (0.020)
	LEM	0.047 (0.008)	0.009 (0.003)	0.079 (0.031)	0.004 (0.002)	0.004 (0.002)	0.090 (0.016)	0.708 (0.020)

Table 3 Data generating model

k	1	2
π_k	0.65	0.35
$m_k(x)$	$1+\cos(2x\pi)$	$\exp(2x)$
σ_k^2	0.09	0.16

tions of the performance measures over all the 500 replicates. The results from Table 2 show that the proposed MB-EM algorithm significantly outperforms the naiveEM. Moreover, for small sample sizes, MB-EM generally gives stable (small standard deviations) and slightly better estimates compared with the LEM algorithm.

Next, we consider data generated from the model given in Table 3. The CRFs in Table 3 are plotted in Fig. 2b.

We fitted model (4) for $K = 2$ on the data using the LCEs obtained via the naiveEM, the MB-EM and the LEM. The results are given in Table 4. The results from Table 4 show that MB-EM performs slightly better than both the naiveEM and the LEM algorithm. To further emphasise this last point, Fig. 4 shows examples of the fitted CRFs for typical samples of sizes $n = 250$, 500, 1000 and 2000 chosen as before. As can be seen from the figure, the fitted CRFs based on MB-EM appear to be stable and, more importantly, in line with the true CRFs. In contrast, the fitted CRFs based on the naiveEM exhibit wild oscillations and hence are unstable whereas the estimates based on the LEM, although stable, may not be in line with the true CRFs. Thus, given the instability of the naiveEM, the latter is not useful in practice. Moreover, the estimates based on the LEM cannot be relied upon as they may lead to wrong conclusions. This serves as a further motivation for the proposed method.

Local-constant estimator vs. Local-linear estimator Next, we compare the LCEs and LLEs obtained using the proposed MB-EM. The data for this experiment is generated from the model in Table 5. A plot of the CRFs is given in Fig. 2c. It is known that the first and second derivatives of the regression function is a multiplicative and additive term, respectively, in the theoretical bias of a LCE of the regression function (see Fan (1992)). Thus, the CRF for component 2 was chosen so that its first and second derivatives are large. Since we are interested in the performance of the estimators in estimating the CRFs, we only report the RASE(**m**).

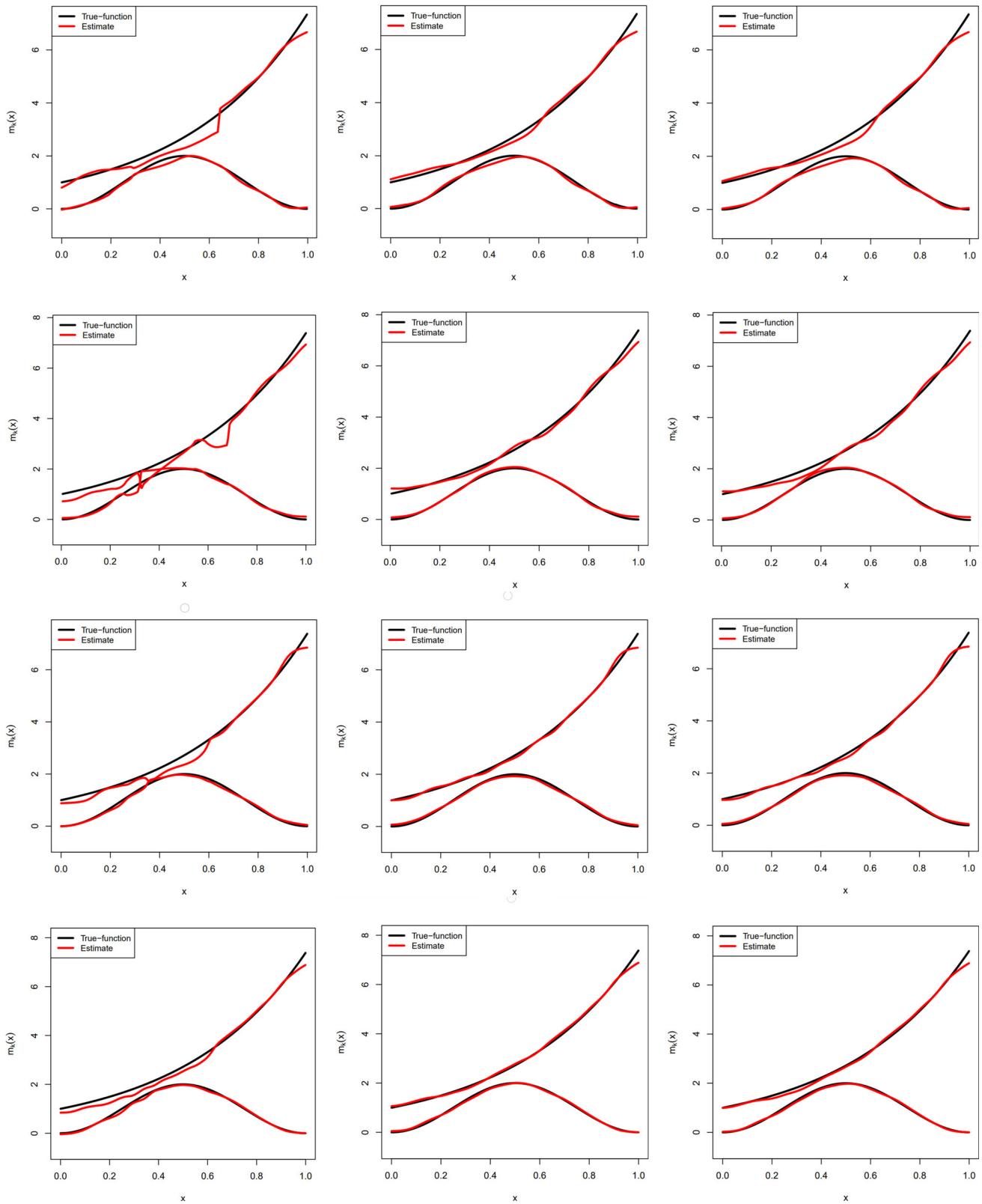


Fig. 3 True (black curves) and fitted (red curves) CRFs obtained via the NaiveEM algorithm (**left-column**), the MB-EM algorithm (**center**) and LEM algorithm (**righ-column**) for samples of sizes $n = 250$ (**first-row**), 500 (**second-row**), 1000 (**third-row**) and 2000 (**fourth-row**) generated from the model in Table 1. These CRFs were chosen from the fitted models with the largest likelihood value based on the naiveEM

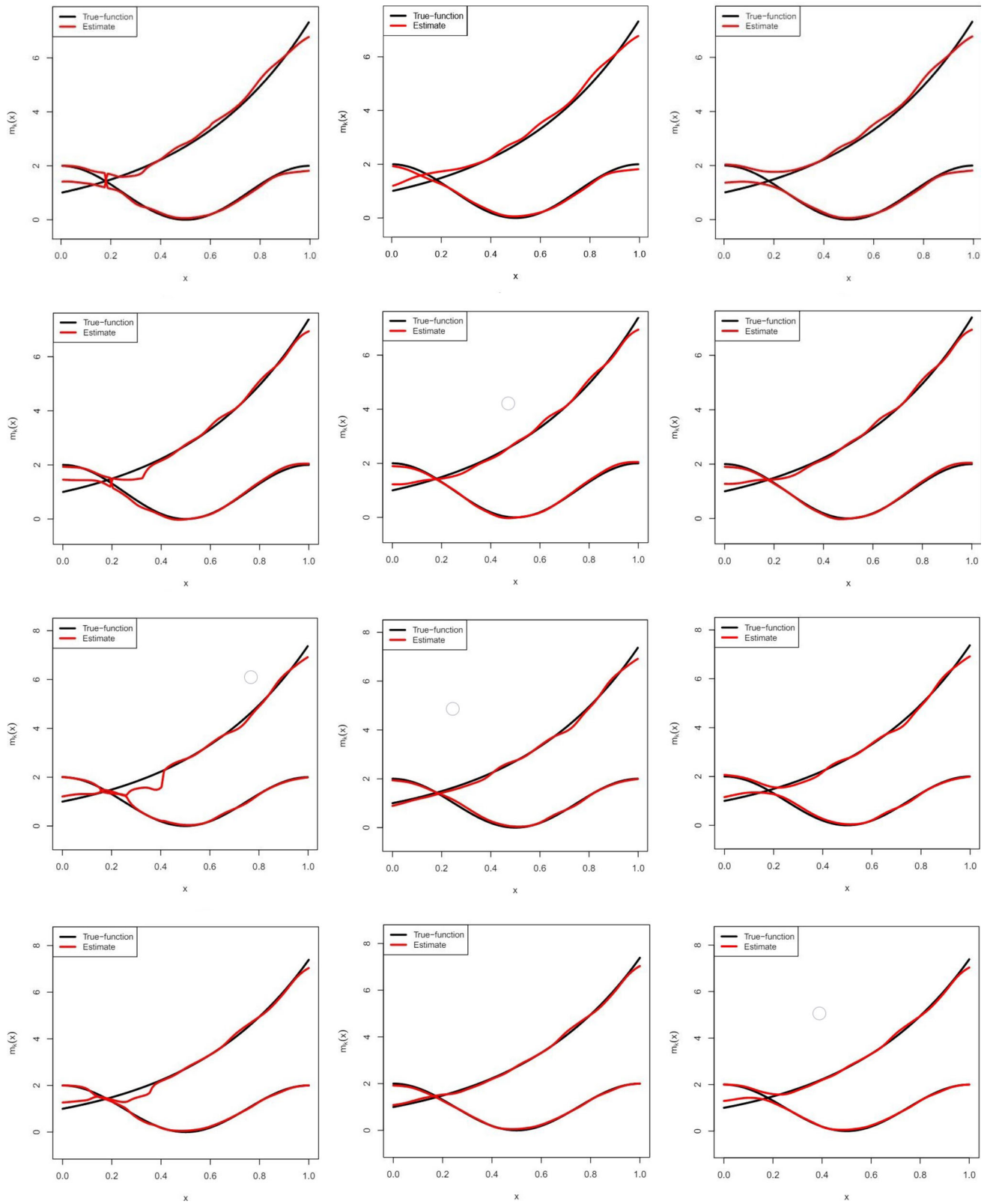


Fig. 4 True (black curves) and fitted (red curves) CRFs obtained via the NaiveEM algorithm (**left-column**), the MB-EM algorithm (**center**) and LEM algorithm (**righ-column**) for samples of sizes $n = 250$ (**first-row**), 500 (**second-row**), 1000 (**third-row**) and 2000 (**fourth-row**) generated from the model in Table 3. These CRFs were chosen from the fitted models with the largest likelihood value based on the NaiveEM

Table 4 Average (and standard deviations) of the performance measures over the 500 replications using data generated from the model in Table 3

n		RASE($f_{\boldsymbol{\theta}}$)	KS	ASE$_{\pi}$	ASE$_{\sigma_1^2}$	ASE$_{\sigma_2^2}$	RASE(**m**)	ARI
250	naiveEM	0.116 (0.021)	0.016 (0.007)	0.076 (0.032)	0.004 (0.002)	0.004 (0.002)	0.237 (0.074)	0.766 (0.058)
	MB-EM	0.107 (0.024)	0.018 (0.008)	0.074 (0.045)	0.003 (0.004)	0.005 (0.002)	0.180 (0.039)	0.763 (0.070)
	LEM	0.129 (0.046)	0.023 (0.014)	0.070 (0.038)	0.004 (0.002)	0.004 (0.002)	0.193 (0.059)	0.614 (0.193)
500	naiveEM	0.094 (0.016)	0.012 (0.005)	0.078 (0.031)	0.004 (0.002)	0.004 (0.002)	0.195 (0.047)	0.768 (0.040)
	MB-EM	0.079 (0.017)	0.014 (0.006)	0.074 (0.041)	0.004 (0.003)	0.004 (0.002)	0.139 (0.028)	0.775 (0.046)
	LEM	0.133 (0.028)	0.018 (0.006)	0.072 (0.036)	0.004 (0.002)	0.004 (0.002)	0.154 (0.035)	0.527 (0.106)
1000	naiveEM	0.077 (0.013)	0.009 (0.003)	0.082 (0.026)	0.004 (0.001)	0.004 (0.001)	0.168 (0.036)	0.774 (0.026)
	MB-EM	0.058 (0.011)	0.010 (0.005)	0.076 (0.039)	0.004 (0.003)	0.004 (0.002)	0.105 (0.019)	0.784 (0.028)
	LEM	0.130 (0.016)	0.015 (0.004)	0.077 (0.032)	0.004 (0.002)	0.004 (0.002)	0.133 (0.026)	0.498 (0.051)
2000	naiveEM	0.064 (0.011)	0.007 (0.003)	0.082 (0.026)	0.004 (0.001)	0.004 (0.001)	0.148 (0.026)	0.777 (0.018)
	MB-EM	0.044 (0.008)	0.007 (0.003)	0.078 (0.036)	0.004 (0.002)	0.004 (0.002)	0.082 (0.014)	0.786 (0.018)
	LEM	0.124 (0.010)	0.013 (0.003)	0.081 (0.028)	0.004 (0.001)	0.004 (0.001)	0.118 (0.021)	0.496 (0.028)

Table 5 Data generating model

k	1	2
π_k	0.65	0.35
$m_k(x)$	$1-\cos(2x\pi)$	$\exp(2x^2)$
σ_k^2	0.09	0.16

Following (Buja et al. 1989), we obtain the bandwidths such that the two estimators have the same total degrees of freedom (tdf), $\sum_{k=1}^{K} \text{df}_k$, where df_k is given by (40). This is done so that we can be able to compare the results based on the LCE and LLE (see Buja et al. (1989) for more details). Table 6 gives the average and standard deviations of the RASE, over all the 500 replicates, using the LCEs and LLEs obtained via the proposed MB-EM. As can be seen from the table, LLEs perform better than the LCEs for estimating the CRFs. This is not unexpected. As alluded to above, if the true non-parametric function has a large first and second derivative, then the LCEs will be subject to bias (see Fan (1992)).

Evaluating the sensitivity of the proposed MB-EM algorithm on the value of the parameter λ_0 Next, we evaluate the sensitivity of the proposed MB-EM algorithm on the value of the parameter λ_0. Before presenting any empirical results, intuitively, the value of λ_0 should not be too large because it might lead to the choice of an inadequately small (or zero!) number of local points. In the extreme case the algorithm will fail. On the other hand, if λ_0 is chosen too small, the algorithm may not be able to select the optimal set of local points. The resulting local neighbourhood will include all the initial local points.

We evaluate the sensitivity of the fitted model on the value of λ_0 using data generated from the models in Tables 1 and 3. For a sample of size $n = 500$, Table 7 gives the results of the MB-EM algorithm for a range of values of λ_0. The value $1 \times 10^{-5} = 0.00001$.

As can be seen from the table, for values of λ_0 at most 1×10^{-4}, the performance of the algorithm is virtually the same. However, when λ_0 is chosen greater than 1×10^{-4}, the performance deteriorates. In terms of choosing the number of local points where the estimation takes place, for $\lambda_0 = 1 \times 10^{-2}$, the algorithm tends to choose $2-10$ local points thus resulting in an inadequate fit. On the other hand, for $\lambda_0 = 1 \times 10^{-8}$, the algorithm tends to choose $95-100$ local points. This results are consistent with our above intuition. Thus, any value of λ_0 that is not too small (to prevent a large non-local neighbourhood) and not too large (to prevent empty neighbourhoods) will suffice. Clearly, the latter scenario results in the most undesirable outcome. In our simulations and applications, we chose our value of λ_0 to be sufficiently small.

Note that the above results still hold if we increase the sample size to say $n = 1000$. The results can be provided upon request from the authors.

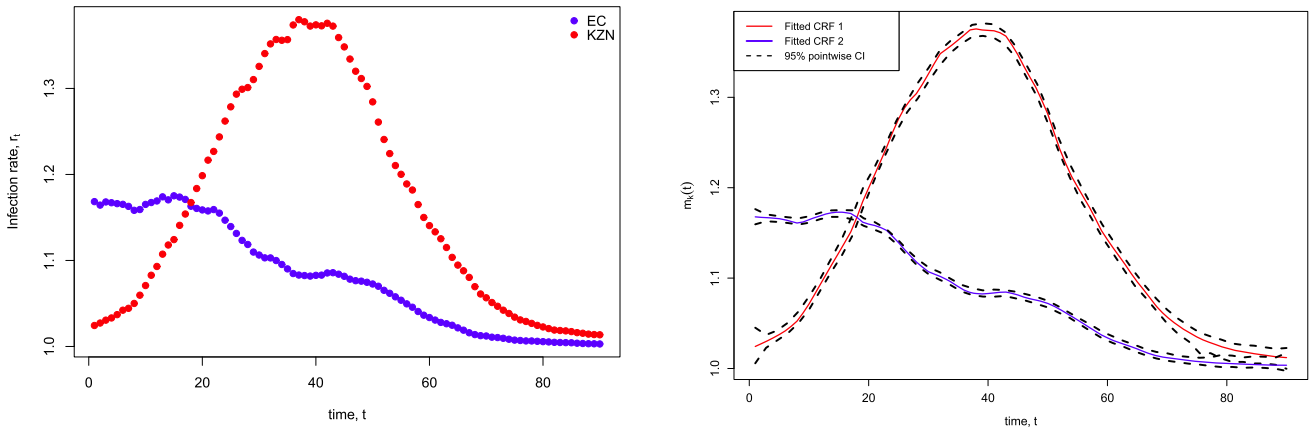
Evaluating the computational time Finally, we evaluate the computational time when practically implementing the proposed algorithm compared with the LEM algorithm. The simulations were conducted on a computer with 2 Skylake CPUs each with 24-cores at 2.6 GHz frequency and a 512 GB RAM. Table 8 gives the average time (in minutes) it takes to run the MB-EM algorithm and the LEM algorithm for samples of sizes $n = 250, 500, 1000$ and 2000 using data generated from the models in Tables 1 and 3. The results show that the LEM algorithm is computationally faster than the proposed MB-EM algorithm. However, we believe that the practical performance of the MB-EM algorithm, in terms of producing accurate estimates as clearly shown in Table 4 and Fig. 4, justifies the computational cost of the algorithm.

Table 6 Average (and standard deviations) of the RASE(**m**) over the 500 replications based on the LCEs and LLEs obtained using the MB-EM algorithm

Estimator	n			
	250	500	1000	2000
LCE	0.210 (0.054)	0.165 (0.029)	0.132 (0.022)	0.109 (0.016)
LLE	0.157 (0.045)	0.113 (0.025)	0.088 (0.020)	0.067 (0.014)

Table 7 Evaluating the sensitivity of the MB-EM algorithm on the value of λ_0: average (and standard deviations) of the performance measures over the 500 replications for samples of size $n = 500$

Model	λ_0	RASE($f_{\boldsymbol{\theta}}$)	RASE(**m**)	KS	ASE_{π}	$\text{ASE}_{\sigma_1^2}$	$\text{ASE}_{\sigma_2^2}$	ARI
Table 1	1×10^{-8}	0.081 (0.015)	0.016 (0.007)	0.082 (0.041)	0.005 (0.004)	0.004 (0.002)	0.141 (0.030)	0.703 (0.043)
	1×10^{-6}	0.081 (0.015)	0.016 (0.007)	0.083 (0.040)	0.005 (0.004)	0.004 (0.002)	0.141 (0.030)	0.703 (0.044)
	1×10^{-5}	0.081 (0.015)	0.016 (0.007)	0.085 (0.040)	0.005 (0.004)	0.004 (0.002)	0.141 (0.030)	0.703 (0.043)
	1×10^{-4}	0.081 (0.015)	0.016 (0.007)	0.081 (0.042)	0.005 (0.004)	0.004 (0.002)	0.141 (0.030)	0.702 (0.044)
	1×10^{-2}	0.142 (0.067)	0.017 (0.008)	0.090 (0.056)	0.365 (1.613)	0.035 (0.122)	0.166 (0.049)	0.594 (0.147)
Table 3	1×10^{-8}	0.078 (0.018)	0.013 (0.006)	0.076 (0.041)	0.004 (0.003)	0.004 (0.002)	0.138 (0.028)	0.778 (0.043)
	1×10^{-6}	0.078 (0.018)	0.013 (0.006)	0.076 (0.040)	0.004 (0.003)	0.004 (0.002)	0.138 (0.028)	0.778 (0.043)
	1×10^{-5}	0.078 (0.017)	0.013 (0.006)	0.076 (0.041)	0.004 (0.003)	0.004 (0.002)	0.138 (0.028)	0.778 (0.043)
	1×10^{-4}	0.078 (0.017)	0.013 (0.006)	0.075 (0.041)	0.004 (0.004)	0.004 (0.002)	0.138 (0.028)	0.778 (0.043)
	1×10^{-2}	0.139 (0.063)	0.015 (0.007)	0.070 (0.043)	0.184 (0.690)	0.020 (0.088)	0.153 (0.045)	0.697 (0.116)



(a) (b)

Fig. 5 SA Covid-19 data: **a** Scatter plot of the data. **b** Fitted CRFs for the SA Covid data using the LLE estimator via the MB-EM algorithm. Also included are the 95% pointwise bootstrap confidence intervals

6 Applications

In this section, we demonstrate the practical usefulness of the proposed method on real data. For real data analysis,

1. we present results based on the proposed MB-EM algorithm and compare them with the results based on the LEM algorithm;
2. we initialize each fitting algorithm by making use of the fitted model based on the local constant estimator;
3. we use the GCV criterion to to select the bandwidth for the local constant estimator. We then choose a bandwidth for the local linear estimator such that the total degrees of freedom (tdf) of the two estimators are the same. As before, this renders the fit based on the two estimators comparable;
4. we measure the goodness-of-fit using the RASE and Bayesian information criterion (BIC)

$$\text{BIC} = -2\ell + \text{df} \times \log(n) \tag{45}$$

Table 8 Average computation time (in minutes) of the MB-EM and LEM algorithm over 200 replications based on the LCEs

Model	Algorithm	n			
		250	500	1000	2000
Table 1	MB-EM	0.401	0.614	1.341	6.145
	LEM	0.021	0.058	0.321	3.591
Table 3	MB-EM	0.391	0.624	1.394	6.327
	LEM	0.022	0.055	0.326	3.648

where df $=$ tdf $+ 2K - 1$ is the overall model degrees of freedom and ℓ is the maximum log-likelihood value. Moreover, we assess the predictive ability of the fitted model using the mean squared prediction error (MPSE). Following (Xiang and Yao 2018), we calculate the MSPE via a Monte Carlo cross validation (MCCV) procedure. The MCCV procedure randomly partitions the data into a training set with size $n(1-r)$ and a test set with size nr, where r is the proportion of data in the test set. The model is estimated using the data in the training set and then validated using data in the test set. The procedure is repeated T times and we take the average of the MSPEs. We use $r = 0.1$ and $T = 200$; and

5. lastly, we use a conditional bootstrap approach to calculate the pointwise 95% confidence intervals of the fitted CRFs and the 95% confidence intervals of the component mixing proportions and variances. That is, for a given value of x, we sample the corresponding value of the response, denoted by y^*, from the fitted SPGMNRs model $\sum_{k=1}^{K} \hat{\pi}_k \mathcal{N}\{y|\hat{m}_k(x), \hat{\sigma}_k^2\}$. We repeat this sampling process n times to get a bootstrap sample $\mathcal{S} = \{(x_i, y_i^*) : i = 1, 2, \ldots, n\}$. We generate B bootstrap samples $\mathcal{S}^{(1)}, \mathcal{S}^{(2)}, \ldots, \mathcal{S}^{(B)}$ in the above manner. We fit the SPGMNRs model (4) on each of these bootstrap samples, thus generating a sampling distribution of $\hat{\pi}_k, \hat{\sigma}_k^2$ and $\hat{m}_k(x)$. To compute the 95% confidence intervals, we take the 2.5^{th} and 97.5^{th} percentiles of the sampling distributions as the lower and upper limits, respectively, of the interval. We set $B = 200$.

6.1 South African Covid-19 data

For our first application, we consider the Covid-19 infection rates (r_t) over time (t) in two South African provinces, Kwa-Zulu Natal (KZN) and the Eastern Cape (EC), for the period December 2020 to 15 February 2021. This data set was previously used by Millard and Kanfer (2022) where a description can be found. The data was collected from the Data Science for Social Impact COVID-19 data repository.

Figure 5a gives a scatter plot of the data along with the identity of the province that generated each data point. The purpose of this application is to demonstrate the effectiveness of the proposed method in addressing label-switching and identify each data point with the province that generated it. Thus, we take province as a latent variable. It is clear from Fig. 5a that the relationship between the infection rate, r_t, and time, t, is non-linear in each province. Thus, we fit a $K = 2$ component SPGMNRs to the data.

The GCV criterion gave a bandwidth of 1.0249 for the local constant estimator which corresponds with a tdf of about 71. The bandwidth for the local linear estimator with about the same tdf is 1.0468.

Table 9 gives the results of the fitted model obtained using the MB-EM algorithm and LEM algorithm. Since we know the actual component (province) where each data point belongs to, we also measure the clustering ability of the fitted models using the ARI. For this data, the local constant estimated model is slightly better than the local linear estimate, with a small BIC and RASE. However, the predictive ability of the two estimates is virtually the same. The results based on the proposed MB-EM and the LEM algorithm are virtually the same for this data set.

Figure 5b shows the fitted component regression functions (CRFs) using the proposed MB-EM algorithm. We can see that the proposed method was able to detect the "latent" structure.

6.2 African CO2 data

For our next analysis on real data, we consider the relationship between carbon dioxide (CO_2) emissions, a measure of environmental degradation, and gross domestic product (GDP), a measure of the monetary value produced by a country in a given period. Figure 6a shows a scatter plot of CO_2 per capita (in metric tons) on GDP per capita (in US$) for a group of 51 African countries in 2014. The countries includes, among others, South Africa (ZAF), Botswana (BWA) and Zimbabwe (ZWE). The data were obtained from the World Bank's World development indicators database (accessed on 10 April 2023). A quick visual inspection of Fig. 6a reveals two clusters (groups) of countries based on the relationship between CO_2 and GDP. Moreover, this relationship is not linear in either of the two groups. A mixture of non-parametric regression analysis is apt for this data. Such an analysis can assist us in answering questions such as

- What development path is adopted by each group of countries? Especially, the low GDP countries.
- Which countries, if any, are pursuing economic growth at a high cost to the environment?
- Is a linear relationship between CO2 and GDP appropriate for each group of countries?
- Are there more than two groups of countries?

Table 9 SA Covid-19 data: The fitted model using the local constant estimator (LCE) and local linear estimator (LLE) via the MB-EM algorithm and the LEM algorithm

	MB-EM		LEM	
	LCE	LLE	LCE	LLE
RASE($\times 10$)	0.0237	0.0241	0.0238	0.00241
BIC	-1204.1	-1198	-1212.4	-1207.7
ARI	1	1	1	1
MSPE	0.0002 (0.0002)	0.0001 (0.0001)	0.0002 (0.0002)	0.0001 (0.0001)

Table 10 BIC values obtained for the SPGMNRs fitted using the MB-EM algorithm and the GMLRs model fitted using the EM algorithm. The SPGMNRs and GMLRs with $K = 1$ corresponds with the non-parametric regression model and simple linear regression model, respectively

K	Model	
	SPGMNRs	GMLRs
1	70.888	100.420
2	-15.046	9.793
3	10.598	25.520
4	16.355	32.574
5	26.168	56.975

After standardizing the variables, we fit a $K = 2$ component SPGMNRs model to the data on Fig. 6a in an attempt to answer some of the questions above. The GCV criterion chose a bandwidth of 0.1725 for the LCE which corresponds to a tdf of about 14. To obtain about the same tdf, the bandwidth of the LLE was chosen to be 0.2343. To confirm that there are indeed two groups and the regression relationships are non-linear, we also fitted the SPGMNRs and GMLRs models with $K = 1, 3, 4$ and 5 components and compared them based on the BIC. The SPGMNRs and the GMLRs for $K = 1$ are essentially the non-parametric regression and linear regression models, respectively. These models were fitted using the R functions: `locfit` (Loader 2023) and `glm`, respectively.

The results (Table 10) show that the $K = 2$ component SPGMNRs model is appropriate for this data having the smallest value of the BIC. Thus, we have confirmed that there are indeed two groups of countries. We therefore proceed with the fitted $K = 2$ component SPGMNRs model.

Table 11 gives the results from the fitted model. It can be seen that the model based on the local linear estimator is the best as it attains the best overall model goodness-of-fit and good performance on out-of-sample prediction. Moreover, the overall performance of the proposed MB-EM algorithm is slightly better than that the other LEM for this data set.

Based on the proposed local linear one-step backfitting estimators via the MB-EM algorithm, the mixing proportions and and variances, along with their 95% bootstrap confidence intervals, were obtained as 0.4775 (0.2425 - 0.5054), 0.5225 (0.4946 - 0.7576), 0.0106 (0.0047 - 0.0343) and 0.0053 (0.0010 - 0.0148), respectively. Figure 6c and 6d gives the fitted CRFs obtained using the proposed LLEs via the MB-EM algorithm. Included in Fig. 6 are the 95% pointwise bootstrap confidence intervals. The estimated CRFs based on the LEM are similar and hence they are excluded.

The estimated CRF in Fig. 6c reveals an interesting phenomenon. CO_2 emissions increase up until a certain level of GDP. Thereafter, beyond this level, they exhibit a slow down in further increases of CO_2 emissions. This is consistent with the well-known environmental Kuznets curve (EKC) hypothesis in environmental economics (see Dinda (2004)). The EKC says that, at the development phase, the value of a country's economy increases at a high cost to the environment due to high carbon emissions from the industrialization process. Beyond a certain level of growth, this effect is reversed and economic growth leads to lower carbon emissions. This phenomenon hypothesizes a non-linear negative parabolic-like relationship between CO2 and GDP. Assuming that all countries follow the same EKC, for a cross-section of countries, the estimated EKC's in Fig. 6 show countries at different stages of development (Dinda 2004). Using model-based clustering (see McNicholas (2016)), we can use the fitted model to assign each country to a given group. The results are given in Fig. 6b. We find that the developmental path given by the curve in Fig. 6c is made up by countries such as Namibia, Swaziland and Botswana. Countries in which the energy mix is becoming less dominated by fossil fuels. Whereas the developmental path given by the curve in Fig. 6d is made up by countries such as South Africa, Morocco and Egypt. Countries in which the energy mix is still heavily dominated by fossil fuels.

7 Conclusion

This paper was concerned with addressing the label-switching problem encountered when estimating semi-parametric Gaussian mixtures of non-parametric regressions (SPGMNRs) using local likelihood methods. Applying the EM algorithm to maximize each local likelihood function separately does not guarantee that the component labels on the local parameter estimates will be aligned. We proposed a two-stage approach to: (1) address label-switching and (2) obtain good estimates of the parametric and non-parametric terms of the model. In the first-stage, we use a model-based approach to, in effect, simultaneously maximize the local-likelihood

Table 11 CO2 data: The fitted model using the LCE and LLE via the MB-EM algorithm and LEM algorithm

	MB-EM		LEM	
	LCE	LLE	LCE	LLE
RASE	0.1264	0.1233	0.1213	0.1072
BIC	−34.4426	−39.2662	−16.9129	−28.6801
MSPE	0.1143 (0.1414)	0.0954 (0.1256)	0.1390 (0.16274)	0.1003 (0.1356)

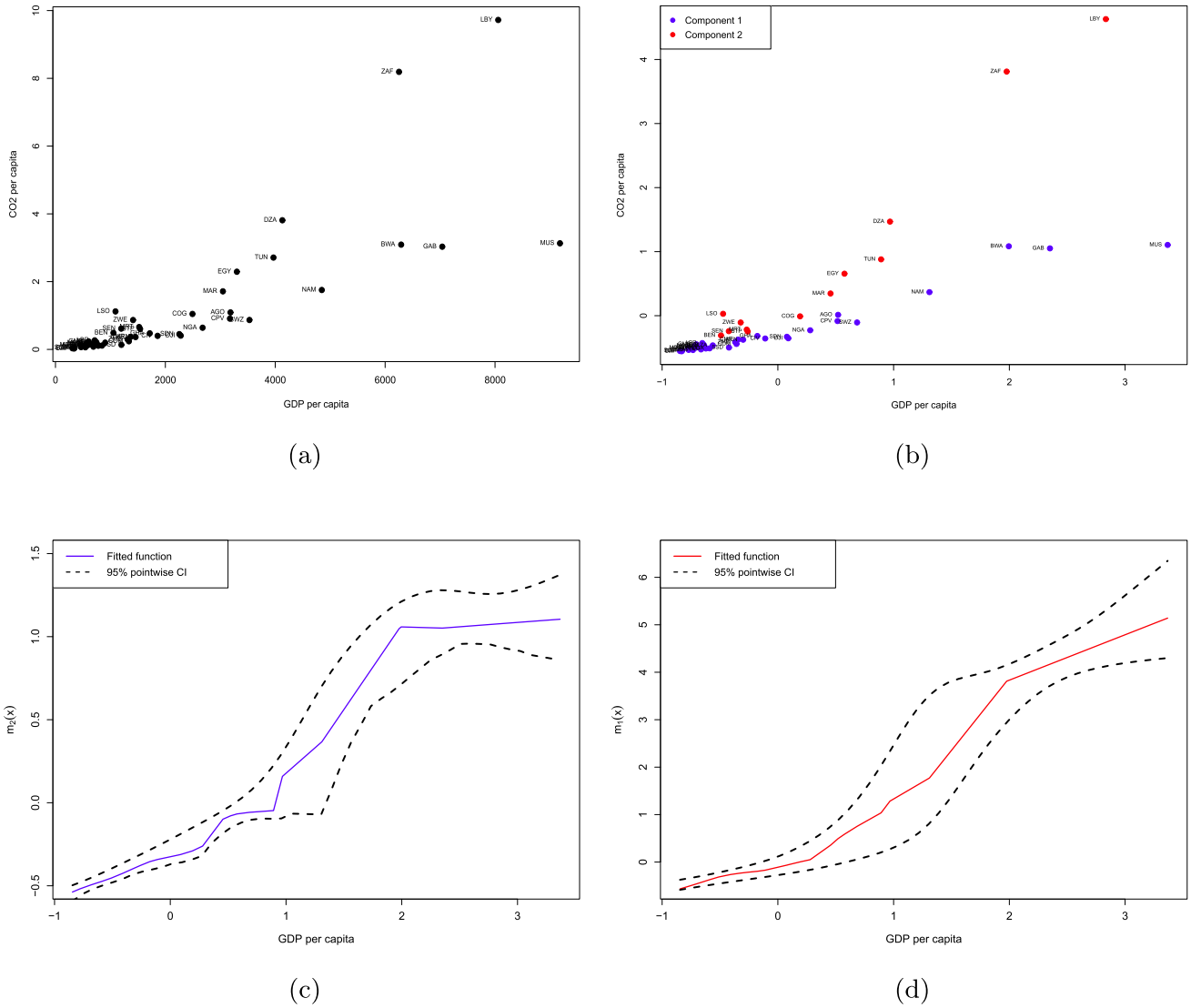


Fig. 6 **a** Scatter plot of the CO2 data. Fitted $K = 2$ component SPGMNRs model obtained using the LLE via the MB-EM algorithm. **b** hard clustered data based on the fitted model. **c**–**d** Fitted CRF for component 1 and component 2, respectively

functions thus addressing label-switching. Within the model-based framework, we use a modified ECM algorithm, to automatically choose the number and location of the local points. In the second-stage, to improve the first-stage estimates, we propose one-step backfitting estimates of the parametric and non-parametric terms. We demonstrated the effectiveness and usefulness of the proposed method on simulated and two real datasets.

The proposed approach incorporates a tuning parameter (threshold) λ_0 which was shown by simulation, under different scenarios, to have less influence on the results if chosen to be neither too small or too large. This is similar to the bias-variance trade-off when choosing the optimal bandwidth. As with the bandwidth, we can use a data-driven approach to choose the optimal λ_0. This will be a subject for future studies. Given the success of the proposed method when only the

CRFs are semi- or non-parametric, it could be interesting, and of practical use for future studies, to investigate the effectiveness of the approach when, in addition to the CRFs, the mixing proportions and/or variances are also non-parametric.

Appendix A

In this appendix, we show how the proposed estimation strategy can be extended to estimate the general model (3).

Let $\bar{\pi}_k$, $\bar{\boldsymbol{\beta}}_k$, $\bar{\sigma}_k^2$ and $\bar{g}_k(t_r)$, for $r = 1, 2, \ldots, D_2$ and $k = 1, 2, \ldots, K$, be the pilot or initial estimates of π_k, $\boldsymbol{\beta}_k$, σ_k^2 and $g_k(t_r)$, for $r = 1, 2, \ldots, D_2$ and $k = 1, 2, \ldots, K$, respectively. To estimate $g_k(t_q)$, for $k = 1, 2, \ldots, K$, using the model-based approach, define the pseudo response variable $y_q = y - \sum_{k=1}^{K} \bar{\pi}_k \big[\mathbf{x}^\intercal \bar{\boldsymbol{\beta}}_k + \sum_{r \neq q} \bar{g}_k(t_r)\big]$ corresponding to the covariate t_q. Then, model (3) reduces to

$$f(y_q|T_q = t_q) = \sum_{k=1}^{K} \pi_k \mathcal{N}\{y_q | g_k(t_q), \sigma_k^2\}, \tag{A1}$$

Model (A1) is the SPGMNRs (4). The estimation of model (A1) can be done similar to that of model (4) as discussed in section 4.

Let $\hat{g}_k(t_q)$, for $k = 1, 2, \ldots, K$,- be the estimates of $g_k(t_q)$, for $k = 1, 2, \ldots, K$, obtained from fitting model (A1).

To obtain the estimates $\hat{g}_k(t_r)$, for $r \neq q$ and $k = 1, 2, \ldots, K$, for the other non-parametric additive functions, we repeat the above procedure.

Finally, let $\hat{g}_k(t_r)$, for $r = 1, 2, \ldots, D_2$ and $k = 1, 2, \ldots, K$, be the model-based estimates of the non-parametric additive functions.

Given the estimates $\hat{g}_k(t_r)$, for $r = 1, 2, \ldots, D_2$ and $k = 1, 2, \ldots, K$, we can improve the estimates $\bar{\pi}_k$, $\bar{\boldsymbol{\beta}}_k$ and $\bar{\sigma}_k^2$ by maximizing the log-lilekihood function

$$\ell_1(\boldsymbol{\pi}, \boldsymbol{\beta}, \boldsymbol{\sigma}^2) = \sum_{i=1}^{n} \log\Big[\sum_{k=1}^{K} \pi_k \mathcal{N}\{y_i | \mathbf{x}^\intercal \boldsymbol{\beta}_k + \sum_{r=1}^{D_2} \hat{g}_k(t_{ir}), \sigma_k^2\}\Big] \tag{A2}$$

Let $\tilde{\pi}_k$, $\tilde{\boldsymbol{\beta}}_k$ and $\tilde{\sigma}_k^2$ be the global parameter estimates obtained from maximizing (A2).

Given $\tilde{\pi}_k$, $\tilde{\boldsymbol{\beta}}_k$ and $\tilde{\sigma}_k^2$ and $\hat{g}_k(t_r)$, for $r \neq q$ and $k = 1, 2, \ldots, K$, we can improve the estimates of the non-parametric additive functions $\hat{g}_k(t_q)$, for $k = 1, 2, \ldots, K$, by maximizing the local-likelihood function

$$\ell_2\{\boldsymbol{g}(u_t)\} = \sum_{i=1}^{n} \log\Big[\sum_{k=1}^{K} \tilde{\pi}_k \mathcal{N}\{y_i | \mathbf{x}^\intercal \tilde{\boldsymbol{\beta}}_k + \sum_{r \neq q} \hat{g}_k(t_{ir}) + g_k(u_t), \tilde{\sigma}_k^2\}\Big] K_h(t_q - u_t) \tag{A3}$$

for $u \in \mathcal{U}$, where $\mathcal{U}$ is the set of all the local grid points in the domain of the covariate t_q.

Let $\tilde{g}_k(t_q)$, for $k = 1, 2, \ldots, K$, be the new estimates of $g_k(t_q)$, for $k = 1, 2, \ldots, K$ and set $\hat{g}_k(t_q) = \tilde{g}_k(t_q)$, for $k = 1, 2, \ldots, K$. We repeat the above procedure to obtain the estimates $\tilde{g}_k(t_r)$, for $r \neq q$ and $k = 1, 2, \ldots, K$.

Finally, let $\tilde{\pi}_k, \tilde{\boldsymbol{\beta}}_k, \tilde{\sigma}_k^2$ and $\tilde{g}_k(t_r)$, for $r = 1, 2, \ldots, D_2$ and $k = 1, 2, \ldots, K$, be the one-step backfitting model-based EM estimates.

The above estimating strategy for estimating model (3) is a three-stage estimation procedure. In the first-stage, we obtain the pilot or initial estimates of the parameters and non-parametric additive functions. This can be done using B-splines as in Zhang (2020). In the second-stage, we use the proposed model-based approach to estimate the non-parametric additive functions. Finally, in the third-stage, we re-estimate the parameters and then the non-parametric additive functions.

Appendix B Derivations

B.1 Derivation of $\pi_{t,k}^{(r+1)}$

Note that $\sum_{k=1}^{K} \pi_{t,k} = 1$, for $t \in \mathcal{T}^{(r+1)}$. Thus, the maximization of $Q^w(\boldsymbol{\theta}|\boldsymbol{\theta}^{(r)})$ with respect to $\pi_{t,k}$ is subject to the above constraint.

Let η be the Lagrange multiplier, the Lagrangian function is given as

$$Q_\eta^w(\boldsymbol{\theta}|\boldsymbol{\theta}^{(r)}) = Q^w(\boldsymbol{\theta}|\boldsymbol{\theta}^{(r)}) + \eta\Big[\sum_{t \in \mathcal{T}} \big(\sum_{k=1}^{K} \pi_{t,k} - 1\big)\Big] \tag{B1}$$

Maximizing (B1) with respect to $\pi_{t,k}$ gives

$$\begin{aligned} \frac{\partial Q_\eta^w(\boldsymbol{\theta}|\boldsymbol{\theta}^{(r)})}{\partial \pi_{t,k}} &= \frac{\sum_{i=1}^{n} \hat{v}_{it}^{(r+1)} \hat{z}_{itk}^{(r+1)} K_h(x_i - u_t)}{\pi_{t,k}} + \eta \\ 0 &\overset{set}{=} \frac{\sum_{i=1}^{n} \hat{v}_{it}^{(r+1)} \hat{z}_{itk}^{(r+1)} K_h(x_i - u_t)}{\pi_{t,k}^{(r+1)}} + \eta \\ -\eta \pi_{t,k}^{(r+1)} &= \sum_{i=1}^{n} \hat{v}_{it}^{(r+1)} \hat{z}_{itk}^{(r+1)} K_h(x_i - u_t) \end{aligned} \tag{B2}$$

Summing both sides of (B2) over $k = 1, 2, \ldots, K$, we obtain

$$-\eta \sum_{k=1}^{K} \pi_{t,k}^{(r+1)} = \sum_{i=1}^{n} \hat{v}_{it}^{(r+1)} K_h(x_i - u_t) \sum_{k=1}^{K} \hat{z}_{itk}^{(r+1)}$$
$$-\eta = \sum_{i=1}^{n} \hat{v}_{it}^{(r+1)} K_h(x_i - u_t) \tag{B3}$$

Note that $\sum_{k=1}^{K} \pi_{t,k} = 1$ and $\sum_{k=1}^{K} \hat{z}_{itk} = 1$, for $t \in \mathcal{T}^{(r+1)}$ and $i = 1, 2, \ldots, n$.

Substituting (B3) into (B2) followed by a bit of algebra gives

$$\pi_{t,k}^{(r+1)} = \frac{\sum_{i=1}^{n} \hat{v}_{it}^{(r+1)} \hat{z}_{itk}^{(r+1)} K_h(x_i - u_t)}{\sum_{i=1}^{n} \hat{v}_{it}^{(r+1)} K_h(x_i - u_t)} \tag{B4}$$

B.2 Derivation of $m_{t,k}^{(r+1)}$ and $\sigma_{t,k}^{2(r+1)}$

Let $w_{itk}^{(r+1)} = \hat{v}_{it}^{(r+1)} \hat{z}_{itk}^{(r+1)} K_h(x_i - u_t)$, then $Q^w(\boldsymbol{\theta}|\boldsymbol{\theta}^{(r)})$ can be expressed as

$$Q^w(\boldsymbol{\theta}|\boldsymbol{\theta}^{(r)}) = -\sum_{t \in \mathcal{T}^{(r+1)}} \sum_{i=1}^{n} \sum_{k=1}^{K} w_{itk}^{(r+1)} \left[\frac{1}{2} \log(2\pi\sigma_{t,k}^2) + \frac{1}{2\sigma_{t,k}^2} (y_i - m_{t,k})^2 \right] \tag{B5}$$

Maximizing $Q^w(\boldsymbol{\theta}|\boldsymbol{\theta}^{(r)})$ with respect to $m_{t,k}$ gives

$$\frac{\partial Q_\eta^w(\boldsymbol{\theta}|\boldsymbol{\theta}^{(r)})}{\partial m_{t,k}} = \frac{\sum_{i=1}^{n} w_{itk}^{(r+1)} (y_i - m_{t,k})}{\sigma_{t,k}^2}$$
$$0 \overset{set}{=} \frac{\sum_{i=1}^{n} w_{itk}^{(r+1)} (y_i - m_{t,k}^{(r+1)})}{\sigma_{t,k}^2}$$
$$m_{t,k}^{(r+1)} \sum_{i=1}^{n} w_{itk}^{(r+1)} = \sum_{i=1}^{n} w_{itk}^{(r+1)} y_i$$
$$m_{t,k}^{(r+1)} = \frac{\sum_{i=1}^{n} w_{itk}^{(r+1)} y_i}{\sum_{i=1}^{n} w_{itk}^{(r+1)}} \tag{B6}$$

Maximizing $Q^w(\boldsymbol{\theta}|\boldsymbol{\theta}^{(r)})$ with respect to $\sigma_{t,k}^2$ gives

$$\frac{\partial Q_\eta^w(\boldsymbol{\theta}|\boldsymbol{\theta}^{(r)})}{\partial \sigma_{t,k}^2} = -\sum_{i=1}^{n} w_{itk}^{(r+1)} \left[\frac{1}{2\sigma_{t,k}^2} - \frac{1}{2(\sigma_{t,k}^2)^2} \times (y_i - m_{t,k}^{(r+1)})^2 \right]$$
$$0 \overset{set}{=} -\sum_{i=1}^{n} w_{itk}^{(r+1)} \left[\frac{1}{2\sigma_{t,k}^{2(r+1)}} - \frac{1}{2(\sigma_{t,k}^{2(r+1)})^2} \times (y_i - m_{t,k}^{(r+1)})^2 \right]$$
$$\frac{\sum_{i=1}^{n} w_{itk}^{(r+1)}}{2\sigma_{t,k}^{2(r+1)}} = \frac{\sum_{i=1}^{n} w_{itk}^{(r+1)} (y_i - m_{t,k}^{(r+1)})^2}{2(\sigma_{t,k}^{2(r+1)})^2}$$
$$\sigma_{t,k}^{2(r+1)} = \frac{\sum_{i=1}^{n} w_{itk}^{(r+1)} (y_i - m_{t,k}^{(r+1)})^2}{\sum_{i=1}^{n} w_{itk}^{(r+1)}} \tag{B7}$$

Acknowledgements The authors would like to thank Dr. Jannie Pretorius from the Center for the Advancement of Scholarship at the University of Pretoria for providing the computing platform used to do perform the numerical data analysis presented in this research.

Author Contributions Conceptualization: [Sphiwe B. Skhosana; Salomon M. Millard and Frans H. J. Kanfer], Methodology: [Sphiwe B. Skhosana; Salomon M. Millard and Frans H. J. Kanfer], Formal analysis and investigation: [Sphiwe B. Skhosana], Writing - original draft preparation: [Sphiwe B. Skhosana]; Writing - review and editing: [Sphiwe B. Skhosana; Salomon M. Millard and Frans H. J. Kanfer], Supervision: [Salomon M. Millard and Frans H. J. Kanfer]. All authors have read and approved the final manuscript.

Funding Open access funding provided by University of Pretoria. Partial financial support was received from STATOMET at the University of Pretoria and the New Generation of Academics programme (nGAP), Department of Higher Education, South Africa.

Availability of data and materials All the data and software used in this research can be accessed through the link: Data and Software

Declarations

Conflict of interest The authors have no Conflict of interest to declare that are relevant to the content of this article.

References

Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)

Buja, A., Hastie, T., Tibshirani, R.: (1989) Linear smoothers and additive models. Ann. Stat. pp. 453–510

Carroll, R.J., Fan, J., Gijbels, I., et al.: Generalized partially linear single-index models. J. Am. Stat. Assoc. **10**(1080/01621459), 10474001 (1997)

Craven, P., Wahba, G.: Smoothing noisy data with spline functions: estimating the correct degree of smoothing by the method of generalized cross-validation. Numer. Math. **31**(4), 377–403 (1979)
Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Stat. Soc.: Ser. B Methodol. **39**(1), 1–22 (1977)
DeSarbo, W.S., Cron, W.L.: A maximum likelihood methodology for clusterwise linear regression. J. Classif. **5**, 249–282 (1988)
Dinda, S.: Environmental Kuznets curve hypothesis: a survey. Ecol. Econ. **49**(4), 431–455 (2004)
Fan, J.: Design-adaptive nonparametric regression. J. Am. Stat. Assoc. **87**(420), 998–1004 (1992)
Fan, J., Gijbels, I.: Local Polynomial Modelling and its Applications. CRC Press, New York (1996)
Frühwirth-Schnatter, S.: Finite Mixture and Markov Switching Models. Springer Series in Statistics, Springer, New York (2006)
Fruhwirth-Schnatter, S., Celeux, G., Robert, C.P.: Handbook of Mixture Analysis. CRC Press, New York (2019)
Hastie, T., Tibshirani, R., Friedman, J.H.: The Elements Of Statistical Learning: Data Mining, Inference, and Prediction. Springer, New York (2009)
Hastie, T.J., Tibshirani, R.J.: Generalized Additive Models. Taylor & Francis, New York (1990)
Huang, M., Yao, W.: Mixture of regression models with varying mixing proportions: a semiparametric approach. J. Am. Stat. Assoc. **10**(1080/01621459), 682541 (2012)
Huang, M., Li, R., Wang, S.: Nonparametric mixture of regression models. J. Am. Stat. Assoc. **10**(1080/01621459), 772897 (2013)
Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**, 193–218 (1985)
Hurn, M., Justel, A., Robert, C.P.: Estimating mixtures of regressions. J. Comput. Graph. Stat. (2003). https://doi.org/10.1198/1061860031329
Jacobs, R.A., Jordan, M.I., Nowlan, S.J., et al.: Adaptive mixtures of local experts. Neural Comput. **3**(1), 79–87 (1991)
Loader, C.: (2023) locfit: local regression, likelihood and density estimation. https://CRAN.R-project.org/package=locfit, r package version 1.5-9.8
McLachlan, G., Peel, D.: Finite Mixture Models. Wiley Series in Probability and Statistics, Toronto (2000)
McNicholas, P.D.: Model-based clustering. J. Classif. **33**, 331–373 (2016)
Meng, X., Rubin, D.B.: Maximum likelihood estimation via the ECM algorithm: a general framework. Biometrika **80**(2), 267–278 (1993)
Millard, S.M., Kanfer, F.H.J.: Mixtures of semi-parametric generalised linear models. Symmetry (2022). https://doi.org/10.3390/sym14020409
Opsomer, J.D., Ruppert, D.: A root-n consistent backfitting estimator for semiparametric additive modeling. J. Comput. Graph. Stat. **10**(1080/10618600), 10474845 (1999)
Quandt, R.E.: A new approach to estimating switching regressions. J. Am. Stat. Assoc. **67**(338), 306–310 (1972)
Quandt, R.E., Ramsey, J.B.: Estimating mixtures of normal distributions and switching regressions. J. Am. Stat. Assoc. **73**(364), 730–738 (1978)
R Core Team (2023) R: a language and environment for statistical computing. R foundation for statistical computing, Vienna, Austria, https://www.R-project.org/
Schlattmann, P.: Medical Applications of Finite Mixture Models. Springer, Berlin (2009)
Skhosana, S.B., Kanfer, F.H.J., Millard, S.M.: Fitting non-parametric mixture of regressions: introducing an EM-type algorithm to address the label-switching problem. Symmetry (2022). https://doi.org/10.3390/sym14051058
Skhosana, S.B., Millard, S.M., Kanfer, F.H.J.: A novel EM-type algorithm to estimate semi-parametric mixtures of partially linear models. Mathematics (2023). https://doi.org/10.3390/math11051087
Tibshirani, R., Hastie, T.: Local likelihood estimation. J. Am. Stat. Assoc. **82**(398), 559–567 (1987)
Titterington, D.M., Smith, A.F.M., Makov, U.E.: Statistical Analysis of Finite Mixture Distributions. Wiley, New York (1985)
Wu, X., Liu, T.: Estimation and testing for semiparametric mixtures of partially linear models. Commun. Stat. Theory Method. **10**(1080/03610926), 1189569 (2016)
Xiang, S., Yao, W.: Semiparametric mixtures of nonparametric regressions. Ann. Inst. Statistical Math. (2018). https://doi.org/10.1007/s10463-016-0584-7
Zhang, Y., Pan, W.: (2022) Estimation and inference for mixture of partially linear additive models. Commun. Stat. Theory Method. **10**(1080/03610926), 1777305 (2020)
Zhang, Y.: Zheng Q (2018) Semiparametric mixture of additive regression models. Communications in Statistics-Theory and Methods **10**(1080/03610926), 1310243 (2017)