Semi-parametric mixtures of quantile regressions

By

Divan Gerhard Gouws

u21727709

Supervisors: Prof SM Millard and Prof FHJ Kanfer

Submitted in partial fulfillment of the requirements for the degree
MSc in Advanced Data Analytics

in the Faculty of Natural and Agricultural Sciences

University of Pretoria

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

15 February 2023

Abstract

Mixtures of quantile regressions are explored through the lens of a kernel density based EM-type algorithm and a newly proposed CEM-type algorithm. This allows the simultaneous clustering and modeling of conditional quantiles without the need to assume symmetric or identical error distributions for any of the components. We conduct simulation studies and apply both algorithms to real life datasets. The first has already been investigated by fitting the EM-type algorithm and we show that the CEM-type algorithm produces similar results. The second is a homoscedastic dataset which has been explored through the lens of univariate quantile regression. We begin by modeling the mixtures of the conditional medians as a robust alternative to mixtures of conditional means. Mixtures of other conditional quantiles are modeled as well to get a more complete understanding of the conditional distribution. This, however, proves to be a challenging task for datasets which are not easily separable and may lead to unsatisfactory results, especially when considering low quantiles or high quantiles such as 0.1 or 0.9 respectively. The theory of the EM-type algorithm is provided in detail and the proposed CEM-type algorithm is shown to provide a substantial improvement in the model convergence speed, but often at the cost of increased bias in the parameter estimates. We conclude with a discussion of some of the limitations and areas for future research.

Declaration

I, *Divan Gerhard Gouws*, declare that this dissertation, which I hereby submit for the degree MSc Advanced Data Analytics at the University of Pretoria, is my own work and has not previously been submitted by me for a degree at this or any other tertiary institution.

Divan Gerhard Gouws

Prof SM Millard and Prof FHJ Kanfer

15 February 2023

Date

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

Mixtures of quantile regressions is a statistical method that simultaneously divides the dataset into different components based on some unobserved source of heterogeneity and estimates a quantile regression model per component. It may be used in lieu of mixtures of linear regressions to estimate conditional quantiles rather than conditional means when the variance is not constant across the domain of the predictor variables or the error distributions for each component cannot be assumed to be symmetric. Means are known to be sensitive to outliers, while medians and other quantiles are more robust, and this property is central to motivating the use of quantiles as an alternative to the more familiar mixtures of linear regressions approach.

The most pertinent research that forms the basis of this work is that of Wu and Yao (2016). When modeling conditional quantiles, the residuals no longer follow a symmetric distribution. They overcome the limitations of modeling symmetric error densities in mixtures of regressions by augmenting the traditional Expectation-Maximisation (EM) algorithm to incorporate kernel density estimators which adapt to the dataset. This extension to the EM algorithm is referred to as an Expectation-Maximisation type (EM-type) algorithm and it enables the unification of quantile regression and mixtures of linear regressions into a single modeling framework that can perform model-based clustering, is robust to outliers and can simultaneously estimate conditional quantiles for multiple subcomponents in the data. Accurate modeling of the error distributions is necessary to correctly compute the responsibilities of each mixture component for each data point in the E-step of the EM-type algorithm.

1.1 Contribution

This mini dissertation aims to explore and elucidate the theory in the research of Wu and Yao (2016). This is followed by proposing the addition of a classification step between the E and M steps of their Expectation-Maximisation type (EM-type) algorithm, with the aim of reducing the computation speed. Faria and Soromenho (2010) showed in simulation studies that the traditional Classification EM (CEM) algorithm always converges in fewer iterations than the traditional EM algorithm when fitting mixtures of linear regressions models. We use simulation studies to compare the Classification Expectation-Maximisation type (CEM-type) algorithm to the EM-type algorithm for fitting mixtures of quantile regressions, both in terms of speed and the bias and variance of parameter estimates, to ascertain if similar results hold. These simulation studies are adapted from those in Wu and Yao (2016) to understand the viability of this new approach in datasets which intersect or follow highly skewed error distributions.

Finally, both algorithms are fitted to two datasets. The first is the tone perception experiment, which was analysed by Wu and Yao (2016) by fitting the EM-type algorithm. Their results are reproduced and used as a benchmark for comparing the stability of the proposed CEM-type algorithm. The second is the aforementioned dataset of the daily maximum temperatures in Melbourne, Australia, analysed by Koenker and Hallock (2001). The focus on this dataset is to determine if quantile mixture regression produces superior results to that of standard quantile regression, however both the EM-type and CEM-type algorithm are employed to this end.

1.2 Notation

A mixture of quantile regressions for a univariate dependent variable y may be written as

$$y_i(\tau) = \begin{cases} \mathbf{x}_i^T \boldsymbol{\beta}_1(\tau) + \epsilon_{i1}(\tau) & \text{with probability } \pi_1 \\ \mathbf{x}_i^T \boldsymbol{\beta}_2(\tau) + \epsilon_{i2}(\tau) & \text{with probability } \pi_2 \\ \vdots & \\ \mathbf{x}_i^T \boldsymbol{\beta}_M(\tau) + \epsilon_{iM}(\tau) & \text{with probability } \pi_M \end{cases}$$

where

τ	is the quantile such that $0 \leq \tau \leq 1$ where $\tau = 0.5$ is the median
i	is the observation index, with $i = 1, \ldots, N$
k	is the component index, with $k = 1, \ldots, M$
y_i	is the ith observation of the dependent variable.
$\mathbf{x}_i^T$	is the transpose of the p-dimensional vector of independent variables, which includes the intercept term.
$\boldsymbol{\beta}_k(\tau)$	is the p-dimensional vector of regression coefficients for the τth quantile and the kth component
$\epsilon_{ik}(\tau)$	are error terms for the τth quantile assumed to be independent of $\mathbf{x}_i^T$
π_k	are mixing probabilities that satisfy $0 < \pi_k < 1$ and $\sum_{k=1}^{M} \pi_k = 1$

Furthermore, we will later use

(b)	normally given as a superscript to denote the bth iteration of the algorithm
γ_{ik}	for the responsibilities
$\boldsymbol{\theta}$	as the set of all model parameters, that is $\boldsymbol{\theta} = (\boldsymbol{\pi}, \boldsymbol{\beta})$ where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_M)$ and $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_M)$

2 Literature review

We first describe quantile regression and mixtures of linear regressions in detail with a focus on applications. These preliminaries will help to fully contextualize the mixtures of quantile regressions models. We also include a survey of the challenges in model estimation and briefly explore alternative models which may be used in lieu of quantile regression.

2.1 Quantile regression

Quantile regression was introduced in the seminal book of the same name by Koenker and Hallock (2001), who explored the estimation of conditional quantile functions. They noted that the assumption of normality was crucial to estimating conditional means, and that the median methods for linear regressions developed by Edgeworth (1888) does not rely on this assumption and may produce estimates with lower variances. Using this as a basis, they went on to develop the theory of quantile regression and applied it to certain datasets. One of the datasets explored in the book, the daily maximum temperatures in Melbourne, Australia, will be re-examined in this work through the lens of mixtures of quantile regressions.

2.1.1 Applications

Yu et al. (2003) present some applications of quantile regression modelling. In medicine, medical reference charts, also known as centile charts, are used to identify abnormal patients during preliminary medical diagnosis when some measurement of a patient's health such as weight is highly dependent on a covariate like age. This is done by modelling a symmetric set of quantiles of the relationship between the dependent and independent variable, as outlined in Cole and Green (1992). Patients are compared to these quantiles and would be considered unusual if they are observed to lie in the tail of the distribution, that is, in either a very large or very small quantile.

A similar approach is used in survival analysis when studying the effects of covariates on individuals with high, medium and low risk profiles, further detailed in Koenker and Geling (2001). A possible approach is to associate the quantiles of the survival times to the corresponding risk profiles. For example, $\tau = 0.1$ for high-risk, $\tau = 0.5$ for medium-risk and $\tau = 0.9$ for low-risk individuals, since longer survival times generally correspond to lower risk individuals. Univariate regression models may then be fitted for each of these quantiles where the response variable is survival time and the covariate is the factor under investigation, such as age. Yang (1999) illustrates how ordinary mean regression is not appropriate to model survival times in the presence of censoring, and how quantile regression could be used instead.

Beyond the health sciences, there are applications in finance, such as Value at Risk (VAR) modelling. Banks are required by regulation to report on their market risk measures, and VAR models are the most common way to measure this (Lauridsen, 2000). Usually, $\tau = 0.05$ is

chosen and it desired to find the value y which satisfies $P(Y < y) = \tau = 0.05$ where the random variable Y is the financial return. Simply put, there is only a 5% probability of making a greater loss than y. As Y may depend on market factors, investment compositions or macroeconomic variables, the conditional returns are modeled and the sensitivity of the bank's investments to different scenarios or combinations of financial assets held by the institution is evaluated.

Broader statistical applications include identifying heteroscedasticity in cases where ordinary linear regression would otherwise be used. That is, quantile regression may be used to determine if the constant variance assumption of linear regression is met by fitting multiple quantile regression functions and checking whether they are parallel, as is done by Koenker and Hallock (2001). According to the authors, the constant variance assumption has been violated if the quantile regression functions are not parallel. Cho et al. (2008) followed a similar approach to detect outliers. They developed the R package OutlierD, which fits constant, linear, non-linear and non-parametric quantile regression models for $\tau = 0.25$ and $\tau = 0.75$ and calculates the interquartile range over the entire domain of the independent variables. Values more than 1.5 times the interquartile range greater than the upper quartile or smaller than the lower quartile are considered outliers.

2.1.2 Contemporary research

Contemporary research has focused on extending the methodology of quantile regression to more complex data sets and situations. For example, Wang, Wu and Li (2012) and Belloni and Chernozhukov (2011) proposed a penalized quantile regression approach for modelling high-dimensional data, where the number of predictors is greater than the number of observations. This showed an improvement in prediction accuracy. Another example is time-series data. Farcomeni (2012) uses Hidden Markov quantile regression models to handle time-dependent heterogeneity. This is closer to mixture modeling and forms part of the class of switching regression models.

The fitting process also involves solving an optimization problem rather than simply computing a matrix inverse and thus quantile regression models can be computationally intensive, which can make them slow or impractical for large datasets. Zheng (2011) has addressed this by focusing on developing efficient algorithms for fitting quantile regression models. They introduced a method for fitting mixtures of quantile regression functions using a gradient-based optimization algorithm. This approach has been shown to be faster and more stable than other optimization methods.

One alternative to quantile regression is the use of generalized additive models (GAMs), which allow for the inclusion of non-linear and smooth functions in the regression model. GAMs can be fit using a variety of algorithms, including maximum likelihood and Bayesian methods, and can provide more flexible and robust estimates of the relationships in the data compared to quantile regressions.

Several machine learning inspired approaches also appear in the literature. Meinshausen and Ridgeway (2006) has proposed the use of quantile regression forests, which combine the flexibility of quantile regression with the power of random forests. Quantile regression forests can be fit using a variety of algorithms, including gradient boosting and decision trees, and can provide more accurate and robust estimates of the relationships in the data compared to mixtures of quantile regressions. Vaysse and Lagacherie (2017) applies this to digital soil mapping.

Another example of this is the neural network quantile regression approach described by Taylor (2000) as well as Zhang et al. (2018) who apply it to modelling financial returns and electricity load forecasting.

2.2 Mixtures of linear regressions

Mixture modelling is an unsupervised learning technique that assigns a variable y to one of M components for which the label is not known (Hastie et al., 2009). These M mixture components may be interpreted as clusters and once the model has been fitted, posterior cluster probabilities may be used to create a fuzzy partition of the observations for soft clustering (Chamroukhi, 2016). Hard clustering may also be performed by assigning each observation to the component with the greatest posterior probability.

Some of the earliest work in mixture modelling in the presence of covariates was presented by DeSarbo and Cron (1988). Their work extends the maximum likelihood theory for the ordinary linear model to simultaneously cluster and fit separate linear models to each cluster using the Expectation-Maximisation algorithm. This involves assigning the data points to clusters and this enabled the modelling of a dataset with underlying heterogeneity by permitting a different conditional expectation and conditional variance of y given a set of covariates for each component. That is, fitting multiple regression models simultaneously for each clustered component in the dataset. This classical mixture of regressions model was sensitive to noisy data, outliers and leverage values. Due to this, several attempts have been made to improve the robustness of the model, such as using the t-distribution (Wei, 2012) or the Laplace distribution (Song et al., 2014) to model error densities with heavier tails or trimming outliers (García-Escudero et al., 2010). The greatest departure from the assumption of normality in mixture regression was explored by Hunter and Young (2012) who described a semi-parametric mixture of regressions model. These approaches all assume a symmetric error distribution, but there are cases where modeling conditional medians or other quantiles is more appropriate (Wu and Yao, 2016).

2.2.1 Applications

Applications of mixture modelling are found across many domains. In healthcare, Wu and Sampson (2009) used the classical mixtures of linear regressions model in schizophrenia research to identify patients with the disease. Wedel and Kamakura (2000) use the framework to present a case study in market segmentation. McFarlane et al. (2021) use mixtures of Gaussian regressions to monitor the air quality in Ghana. Generally speaking, any application of ordinary linear regression with some unobserved characteristic may be an opportunity to apply the method. There are certain difficulties and limitations associated with mixtures of linear regressions however, which are detailed in the following subsections.

2.2.2 Model selection and parameter estimation

Mixture of regressions models can be difficult to estimate and interpret, especially when there are many components or when the components have complex distributions. The first challenge is the selection of M, the number of components. A common approach is to fit multiple models for different values of M and choose the one that minimizes some information criterion, such as the Akaike Information Criterion (AIC) or the Bayesian Information Criterion (BIC). These criteria cannot be computed for semi-parametric models and the number of components is usually assumed to be known or chosen by the modeler (Hunter and Young, 2012).

Another key challenge in mixture modelling is assessing the stability of the model by estimating the variance of the parameter estimates. This is because the data in each component may have different variances. One approach to measuring the consistency of parameter estimates in mixtures of regressions is to use the bootstrap resampling method. Generally, this involves repeatedly resampling the data to create multliple synthetic datasets, refitting the model, collating the parameter estimates and considering the distribution of these parameter estimates. There are various resampling strategies such as the case bootstrap or model bootstrap methods in Wu and Yao (2016), which are explained in more detail in the next section. Another approach is to use cross-validation, which involves dividing the data into a number of folds and fitting the mixture model to each fold, using the remaining folds as the validation data as shown by Faria and Soromenho (2010). The consistency of the parameter estimates can then be assessed by comparing the estimates obtained from each fold.

3 Background Theory

This section details the mathematical results and background theory necessary to understand the mixtures of quantile regressions. Mathematical details are provided where other works omit certain steps or to unify the notation used in disparate works. We begin by investigating the identifiability of the semi-parametric error densities used in mixtures of quantile regressions. This is necessary to understand the modifications made to the standard EM algorithm. As part of this, we expand on the update steps for the $\boldsymbol{\beta}$ parameters and the kernel density estimate and briefly discuss some initialization strategies. The CEM algorithm is then introduced in detail. Finally, we describe the three methods used by Wu and Yao (2016) to estimate the variance of the parameter estimates.

3.1 Identifiability of semi-parametric error densities

Wu and Yao (2016) give the mixture distribution of y as

$$f(y|\mathbf{x}, \boldsymbol{\theta}, \mathbf{G}) = \sum_{k=1}^{M} \pi_k g_k(y - \mathbf{x}^T \boldsymbol{\beta}_k(\tau)).$$

The error densities are assumed to follow the density function $\epsilon_{ik}(\tau) \sim g_k(\cdot)$. The EM-type algorithm will estimate the parametric part of the model given by
$\boldsymbol{\theta} = (\pi_1, \boldsymbol{\beta}_1(\tau), \ldots, \pi_M, \boldsymbol{\beta}_M(\tau))$ as well as the vector of non-parametric density functions $\mathbf{G} = (g_1, \ldots, g_M)$. The error density functions $g_k(\cdot)$ are not assumed to be symmetric and are fully unspecified. The only restriction imposed on the error density functions is that their τth quantiles are equal to zero (Wu and Yao, 2016).

The identifiability of this model is based on the results in Wang, Yao and Hunter (2012), which prove the identifiability of the error densities $g_k(\cdot)$ without requiring symmetry or identical error densities. Wu and Yao (2016) expand on this result and show that it holds for the mixture of τth quantile regressions. In their work, they show that the mixtures of τth quantile regressions is identifiable for $\mathbf{G} = (g_1, \ldots, g_M)$ and $\boldsymbol{\theta} = (\pi_1, \boldsymbol{\beta}_1(\tau), \ldots, \pi_M, \boldsymbol{\beta}_M(\tau))$ up to the same permutation on $\mathbf{G}$ and $\boldsymbol{\theta}$ if $\tilde{\boldsymbol{\beta}}_k = (\beta_{1k}, \ldots, \beta_{pk})$. They also provide the necessary conditions and a proof in the Appendix of their work, which will not be repeated here.

The EM-type algorithm to fit this model is described in the following subsection. Convergence of the algorithm will be achieved when successive iterations produce small changes in the estimated parameter values.

3.2 Expectation-maximisation type algorithm: E-step

The E-step involves computing the responsibilities γ_{ik}. The lack of parametric assumptions imposed on the error densities $g_k(\cdot)$ result in the absence of a likelihood function that may be maximised in order to estimate the model parameters. Consequently, a plug-in estimate for the likelihood based on the $g_k(\cdot)$ will be used instead. These densities are updated with each iteration of the algorithm and are not unique, thus it is impossible to maximize a unique objective function on these densities. Instead, convergence will be achieved once successive iterations produce small changes in the estimated parameter values rather than the plug-in estimate for the likelihood function.

3.2.1 Updating the responsibilities

The responsibilities γ_{ik} may be found by conditioning on the residuals $e_{ik} = y_i - \mathbf{x}_i^T \boldsymbol{\beta_k}$, which depend only on observed data and parameters.

$$
\begin{aligned}
\gamma_{ik} &= E(z_{ik}) && (1)\\
&= P(z_{ik} = 1 | e_{ik}) \\
&= \frac{P(z_{ik} = 1, e_{ik})}{P(e_{ik})} && (2)\\
&= \frac{P(e_{ik} | z_{ik} = 1) P(z_{ik} = 1)}{P(e_{ik})} \\
&= \frac{P(e_{ik} | z_{ik} = 1) P(z_{ik} = 1)}{\sum_{k=1}^{M} P(e_{ik} | z_{ik} = 1) P(z_{ik} = 1)} && (3)\\
&= \frac{\pi_k P(e_{ik} | z_{ik} = 1)}{\sum_{k=1}^{M} \pi_k P(e_{ik} | z_{ik} = 1)} \\
&= \frac{\pi_k \hat{g}_k(e_{ik})}{\sum_{k=1}^{M} \pi_k \hat{g}_k(e_{ik})} && (4)
\end{aligned}
$$

where we have (1) by definition, (2) by the law of conditional probability and (3) by the total law of probability. The last line (4) uses $P(e_{ik}|z_{ik} = 1) = g_k(e_{ik})$ to rewrite the conditional probabilities in terms of the estimated densities $\hat{g}_k$.

3.3 Expectation-maximisation type algorithm: M-step

The M-step involves the computation and maximisation of a **Q**-function with respect to each of its constituent parameters to determine the effective sample size. The **Q**-function is constructed from the plug-in estimate of the conditional complete-data log-likelihood and maximized by the method of Lagrange. Next, the $\boldsymbol{\beta}$ estimates are updated by minimizing a loss function and finally, the error kernel density estimate is updated by performing a constrained optimization method, inspired by Hall and Presnell (1999).

3.3.1 Updating the π parameters

The π parameters may be found by optimisation of an objective function. First, the estimated nonparametric version of the complete-data log-likelihood is the following

$$\begin{aligned}\hat{\ell}_{\text{np}}(\boldsymbol{\theta}|\mathbf{y},\mathbf{z}) &= \sum_{i=1}^{N}\sum_{k=1}^{M} z_{ik}\log\{\pi_k \hat{g}_k(e_{ik})\} \\ &= \sum_{i=1}^{N}\sum_{k=1}^{M} z_{ik}\{\log\pi_k + \log\hat{g}_k(e_{ik})\}.\end{aligned}$$

Taking the expectation of the complete-data log-likelihood with respect to Z and using equation (1) gives the following definition of the **Q**-function

$$\begin{aligned}\mathbf{Q}(\boldsymbol{\theta}) &= E_{\mathbf{Z}}\hat{\ell}_{\text{np}}(\boldsymbol{\theta}|\mathbf{y},\mathbf{z}) \\ &= \sum_{i=1}^{N}\sum_{k=1}^{M} \gamma_{ik}\{\log\pi_k + \log\hat{g}_k(e_{ik})\}.\end{aligned}$$

Using the constraint $\sum_{k=1}^{M}\pi_k = 1$, the problem may be restated in terms of Lagrange multipliers as

$$\mathbf{Q}_L(\boldsymbol{\theta}) = \sum_{i=1}^{N}\sum_{k=1}^{M} \gamma_{ik}\{\log\pi_k + \log\hat{g}_k(e_{ik})\} + \lambda\left(\sum_{k=1}^{M}\pi_k - 1\right)$$

with the following derivatives set equal to zero

$$\begin{aligned}\frac{\partial}{\partial\pi_k}\mathbf{Q}_L(\boldsymbol{\theta}) &= \sum_{i=1}^{N}\frac{\gamma_{ik}}{\pi_k} + \lambda = 0 \\ \frac{\partial}{\partial\lambda}\mathbf{Q}_L(\boldsymbol{\theta}) &= \sum_{i=1}^{N}\pi_k - 1 = 0.\end{aligned}$$

The derivative of $\mathbf{Q}_L$ may be multiplied by π_k and summed over $k = 1, \ldots, M$

$$\begin{aligned}\sum_{k=1}^{M}\sum_{i=1}^{N}\gamma_{ik} + \lambda\sum_{k=1}^{M}\pi_k &= 0\\ N + \lambda &= 0\\ \lambda &= -N.\end{aligned}$$

Substituting the value for λ into the derivative of $\mathbf{Q}_L$ yields

$$\begin{aligned}\frac{\partial}{\partial\pi_k}\mathbf{Q}_L(\boldsymbol{\theta}) &= 0\\ \sum_{i=1}^{N}\frac{\gamma_{ik}}{\pi_k} - N &= 0\\ \hat{\pi}_k &= \frac{\sum_{i=1}^{N}\gamma_{ik}}{N}.\end{aligned}$$

3.3.2 Updating the β parameters

The $\hat{\boldsymbol{\beta}}_k^{(b+1)}$ parameters are updated by fitting a weighted quantile regression model as initially propsed by Koenker and Hallock (2001). The following function is minimised

$$\hat{\boldsymbol{\beta}}_k^{(b+1)}(\tau) = \arg\min_{\boldsymbol{\beta}_k}\sum_{i=1}^{N}\gamma_{ik}^{(b+1)}\rho_\tau(y_i - \mathbf{x}_i^T\boldsymbol{\beta}_k)$$

where $\rho_\tau(y_i - \mathbf{x}_i^T\boldsymbol{\beta}_k)$ is the pinball loss function with input parameter u defined as

$$\rho_\tau(u) = \begin{cases}(1-\tau)u & \text{if } u \leq 0\\ \tau u & \text{if } u > 0.\end{cases}$$

This function differs for each value of τ. Considering $\tau = 0.9$ for example, the loss function would apply a greater penalty to positive errors as shown below in Figure 1.

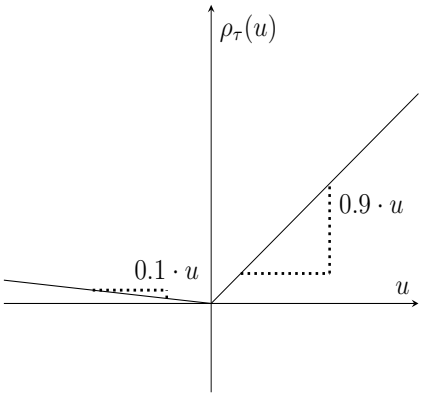

Figure 1: A plot of the pinball loss function for $\tau = 0.9$ with input parameter u.

3.3.3 Updating the error kernel density estimate

Bowman and Azzalini (1997) provide the basic definition for a kernel estimator

$$\hat{f}(x) = \frac{1}{nh} \sum_{i=1}^{N} K((x - x_i)/h)$$

where K is the kernel function which satisfies the conditions of a probability density function and is usually chosen to be symmetric and h is the smoothing parameter.

This definition is an equally weighted average over all observations. Hall and Presnell (1999) introduce a weighted average to extend this definition to include constraints for linear estimators of the form

$$\hat{f}(x|p) = \sum_{i=1}^{N} p_i K_i(x, h) \tag{5}$$

where $K_i(x, h) = h^{-1}K\{(x - x_i)/h\}$ with $j = 0, \ldots, r$ constraints of the form

$$\sum_{i=1}^{N} p_i T_j(K_i) = t_j.$$

The first constraint is always imposed and is defined as setting $T_0(K_i) = 1$ and $t_0 = 1$ to ensure that the weights add up to 1

$$\sum_{i=1}^{N} p_i = 1. \tag{6}$$

The p_i weights are chosen to minimize the distance to the uniform weights $p_{\text{unif}} = (n^{-1}, \ldots, n^{-1})$. More precisely, we use the Kullback-Leibler distance to measure the distance between the p_i weights and the uniform weights and employ the method of Lagrange to find the set of p_i weights that minimize the distance

$$D_0(p) = -\sum_{i=1}^{N} \log(np_i)$$

subject to the constraints

$$\sum_{i=1}^{N} p_i T_j(K_i) = t_j \text{ for } j = 0, \ldots, r.$$

The Lagrangian function is

$$\mathcal{L}(p_i, c_0, c_1, \ldots, c_r) = -\sum_{i=1}^{N} \log(np_i) + c_0 \left(\sum_{i=1}^{N} p_i T_0(K_i) - t_0 \right) + c_1 \left(\sum_{i=1}^{N} p_i T_1(K_i) - t_1 \right) + \cdots + c_r \left(\sum_{i=1}^{N} p_i T_r(K_i) - t_r \right).$$

This is simplified by using $T_0(K_i) = 1$ and $t_0 = 1$ as before

$$\mathcal{L}(p_i, c_0, c_1, \ldots, c_r) = -\sum_{i=1}^{N} \log(np_i) + c_0 \left(\sum_{i=1}^{N} p_i - 1 \right) + c_1 \left(\sum_{i=1}^{N} p_i T_1(K_i) - t_1 \right) + \cdots + c_r \left(\sum_{i=1}^{N} p_i T_r(K_i) - t_r \right).$$

Further, taking partial derivatives and setting them equal to zero yields

$$\begin{aligned}
\frac{\partial}{\partial p_i}\mathcal{L}(p_i, c_0, c_1, \ldots, c_r) &= -\frac{1}{p_i} + c_0 + c_1 T_1(K_i) + c_r T_r(K_i) = 0 \qquad (7)\\
\frac{\partial}{\partial c_0}\mathcal{L}(p_i, c_0, c_1, \ldots, c_r) &= \sum_{i=1}^{N} p_i - 1 = 0 \implies \sum_{i=1}^{N} p_i = 1\\
\frac{\partial}{\partial c_1}\mathcal{L}(p_i, c_0, c_1, \ldots, c_r) &= \sum_{i=1}^{N} p_i T_1(K_i) - t_1 = 0 \implies \sum_{i=1}^{N} p_i T_1(K_i) = t_1\\
&\vdots\\
\frac{\partial}{\partial c_r}\mathcal{L}(p_i, c_0, c_1, \ldots, c_r) &= \sum_{i=1}^{N} p_i T_r(K_i) - t_r = 0 \implies \sum_{i=1}^{N} p_i T_r(K_i) = t_r.
\end{aligned}$$

Equation (7) expresses a relationship between p_i and the $(r+1)$ unknowns $c_0, c_1, \ldots, c_r$. This may be further simplified by multiplying (7) by p_i and summming over $i = 1, \ldots, N$

$$\begin{aligned}
c_0 + c_1 T_1(K_i) + \cdots + c_r T_r(K_i) &= \frac{1}{p_i}\\
c_0 p_i + c_1 p_i T_1(K_i) + \cdots + c_r p_i T_r(K_i) &= 1\\
c_0 \sum_{i=1}^{N} p_i + c_1 \sum_{i=1}^{N} p_i T_1(K_i) + \cdots + c_r \sum_{i=1}^{N} p_i T_r(K_i) &= \sum_{i=1}^{N} 1\\
c_0 + c_1 t_1 + \cdots + c_r t_r &= N\\
N - (c_1 t_1 + \cdots + c_r t_r) &= c_0. \qquad (8)
\end{aligned}$$

Substituting the value for c_0 from (8) into (7) yields

$$\begin{aligned}
\frac{1}{p_i} &= c_0 + c_1 T_1(K_i) + c_r T_r(K_i) \\
\frac{1}{p_i} &= \{N - (c_1 t_1 + \cdots + c_r t_r)\} + c_1 T_1(K_i) + c_r T_r(K_i) \\
\frac{1}{p_i} &= N - \sum_{j=1}^{r} c_j [T_j(K_i) - t_j] \\
p_i &= \left\{ N - \sum_{j=1}^{r} c_j [T_j(K_i) - t_j] \right\}^{-1}.
\end{aligned}$$

This has reduced (7) to an equation in r unknowns. Generally, Hall and Presnell (1999) states that these constants may be found through a Newton-Raphson type of algorithm. Most importantly, a solution exists for the constants such that the weights both satisfy the constraints and minimize the distance between p_i and the uniform weights.

Using this framework, a linear constraint may be imposed on quantiles to adjust the bias of $\hat{f}$, such that the τth quantile of the distribution may equal the sample quantile $\hat{\xi}_\tau$. This is achieved by setting $T_1(K_i) = \int_{-\infty}^{\hat{\xi}_\tau} K_i(y)dy$ and $t_1 = \tau$, yielding the following

$$\begin{aligned}
\sum_{i=1}^{N} p_i T_1(K_i) &= t_1 \\
\sum_{i=1}^{N} p_i \int_{-\infty}^{\hat{\xi}_q} K_i(y)dy &= \tau. \qquad (9)
\end{aligned}$$

Wu and Yao (2016) writes the constrained density estimator in (5) for component k as

$$\hat{g}_k^{(b+1)}(t) = h_k^{-1} \sum_{\ell=1}^{2} \sum_{i=1}^{N} w_{\ell k}^{(b+1)} \gamma_{ik}^{(b+1)} K\{(t - e_{ik}^{(b+1)})/h_k\} I_\ell(e_{ik}^{(b+1)})$$

with both the required constraint for weights to add to 1 in (6) as well as the quantile constraint in (9) as the following pair of simultaneous equations in $w_{\ell k}^{(b+1)}$

$$\sum_{\ell=1}^{2} \sum_{i=1}^{N} w_{\ell k}^{(b+1)} \gamma_{ik}^{(b+1)} I_\ell(e_{ik}^{(b+1)}) = 1 \qquad (10)$$

$$\sum_{\ell=1}^{2} \sum_{i=1}^{N} w_{\ell k}^{(b+1)} \gamma_{ik}^{(b+1)} v_{ik}^{(b+1)} I_\ell(e_{ik}^{(b+1)}) = \tau \qquad (11)$$

where $I_1(e_{ik}^{(b+1)}) = I(e_{ik}^{(b+1)} \leq 0)$ and $I_2(e_{ik}^{(b+1)}) = I(e_{ik}^{(b+1)} > 0)$.

It is clear that (10) is of the same form as (6) by noting that either $w_{1k}^{(b+1)} = 0$ if $I_1(e_{ik})^{(b+1)} = 0$ or $w_{2k}^{(b+1)} = 0$ if $I_2(e_{ik})^{(b+1)}$ for each $\gamma_{ik}^{(b+1)}$, reducing the $2N$ terms in equation (6) to N terms. These N terms serve as weights that sum to 1. For example, if all the residuals are positive, this becomes

$$\sum_{i=1}^{N} w_{2k}^{(b+1)} \gamma_{ik}^{(b+1)} = 1$$

with $w_{2k}^{(b+1)} \gamma_{ik}^{(b+1)}$ effectively fulfilling the same role as p_i in (6).

To verify that (11) has the same form as (9), we must show that $v_{ik}^{(b+1)} = \int_{-\infty}^{\hat{\xi}_q} K_i(y)dy$. We may write the definition for $v_{ik}^{(b+1)}$ provided by Wu and Yao (2016) out in full

$$v_{ik}^{(b+1)} = h^{-1} \int_{-\infty}^{0} K\{(t - e_{ik}^{(b+1)})/h\}dt.$$

We observe that this is of the same form as $K_i(x) = h^{-1}K\{(x - x_i)/h\}$ used by Hall and Presnell (1999) with x_i replaced by $e_{ik}^{(b+1)}$ and $\hat{\xi}_q$ replaced by zero. Thus, this constraint is over the residuals rather than the observed x-values and we wish to set the τth quantile of the distribution equal to 0 rather than the sample quantile $\hat{\xi}_q$.

Furthermore, the kernel function K simplifies to the below equation when using a normal kernel

$$\begin{aligned} v_{ik}^{(b+1)} &= \frac{1}{\sqrt{2\pi}h} \int_{-\infty}^{0} e^{-\frac{1}{2}\left(\frac{t - e_{ik}^{(b+1)}}{h}\right)^2} dt \\ &= P(T \leq 0), \quad \text{where } T \sim N(e_{ik}^{(b+1)}, h). \end{aligned}$$

The weights (10) and (11) may also be stated as solutions to the matrix equation

$$\begin{bmatrix} \sum_{i=1}^{N} \gamma_{ik}^{(b+1)} I_1(e_{ik}^{(b+1)}) & \sum_{i=1}^{N} \gamma_{ik}^{(b+1)} I_2(e_{ik}^{(b+1)}) \\ \sum_{i=1}^{N} \gamma_{ik}^{(b+1)} v_{ik}^{(b+1)} I_1(e_{ik}^{(b+1)}) & \sum_{i=1}^{N} \gamma_{ik}^{(b+1)} v_{ik}^{(b+1)} I_2(e_{ik}^{(b+1)}) \end{bmatrix} \begin{bmatrix} w_{1k}^{(b+1)} \\ w_{2k}^{(b+1)} \end{bmatrix} = \begin{bmatrix} 1 \\ \tau \end{bmatrix}$$

$$\begin{bmatrix} w_{1k}^{(b+1)} \\ w_{2k}^{(b+1)} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{N} \gamma_{ik}^{(b+1)} I_1(e_{ik}^{(b+1)}) & \sum_{i=1}^{N} \gamma_{ik}^{(b+1)} I_2(e_{ik}^{(b+1)}) \\ \sum_{i=1}^{N} \gamma_{ik}^{(b+1)} v_{ik}^{(b+1)} I_1(e_{ik}^{(b+1)}) & \sum_{i=1}^{N} \gamma_{ik}^{(b+1)} v_{ik}^{(b+1)} I_2(e_{ik}^{(b+1)}) \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ \tau \end{bmatrix}$$

which is a 2×2 linear system with a unique solution for $w_{1k}^{(b+1)}$ and $w_{2k}^{(b+1)}$ provided that

$$\sum_{i=1}^{N} \gamma_{ik}^{(b+1)} I_1(e_{ik}^{(b+1)}) \cdot \sum_{i=1}^{N} \gamma_{ik}^{(b+1)} v_{ik}^{(b+1)} I_2(e_{ik}^{(b+1)}) \neq \sum_{i=1}^{N} \gamma_{ik}^{(b+1)} I_2(e_{ik}^{(b+1)}) \cdot \sum_{i=1}^{N} \gamma_{ik}^{(b+1)} v_{ik}^{(b+1)} I_1(e_{ik}^{(b+1)}).$$

3.4 Hyperparameter selection and initialization

Hyperparameters are values that do not form part of the model directly, but are chosen by the user to control the fitted model. In this case, we need to choose the number of components in the model, the degree of smoothing for the kernel density estimator, as well as the starting values for the responsibilities.

3.4.1 The number of components M

The model is semi-parametric and convergence is not achieved by maximizing a likelihood function. The lack of a likelihood function means that we cannot use AIC or BIC criteria nor likelihood ratio tests to determine the number of components M. Wu and Yao (2016) simply assumes that the number of components M is known in advance, and the authors do not discuss model selection any further. We use this working assumption, as model selection is not the focal point of this body of work.

3.4.2 The smoothing parameter h

The bandwidth or smoothing parameter h controls the variance of the kernel function and impacts the estimated error distribution of the EM-type algorithm. We need to first establish whether it is reasonable to assume that error distributions for individual components have equal variances or not.

If we assume equal variance, Silverman (2018) suggests that we use $h = 1.06\sigma N^{-1/5}$ where σ is the standard deviation of the error distribution and N is the total number of observations as before. Practically, we may measure σ as the standard deviation of residuals weighted by the responsibilities. For unequal variances, we have $h_j = 1.06\sigma_j N_j^{-1/5}$ for each component $j = 1, \ldots, M$ where σ_j is the standard deviation of the error distribution for component j and N_j is the effective sample size for component j. The effective sample size is the sum of responsibilities for component j.

3.4.3 Initial responsibilities

One approach to selecting initial classification is clustering-based initialization such as using the k-means algorithm to assign observations to clusters, but this may not be successful when the classes are not highly separable. Random initialization or assigning uniform $1/M$ probabilities to each class may also be considered as initial values, though these strategies perform poorly according to Wu and Yao (2016)

Another approach would be to fit multiple quantile regression lines to the entire dataset and use the proximity of the observations to each fitted line as a classification criterion. Either by classifying each observation as belonging to the nearest quantile, or by performing a similar multinomial trial based on a reasonable score of the distances. For example, an observation is x absolute units away from line A and another observation is $y > x$ absolute units away from line B, its probability of belonging to component A should be greater than that for component B.

We may also use the output of a mixture of linear regressions model as initial values. The responsibilities may be used directly or hard clustering may be performed on the responsibilities, either by choosing the class with the maximum classification probability or classifying each observation according to the outcome of a multinomial trial based on the responsibilities.

3.4.4 Pseudocode for the EM-type algorithms

If it is inappropriate to assume equal variances of the error densities, we use the following algorithm

Algorithm 1 EM-type algorithm using a Normal kernel, assuming unequal variances for error densities

1: Hyperparameters: choose values for the number of components M, the quantile to be estimated τ and the tolerance for convergence

2: Initial values: find initial values for the initial responsibilities $\gamma_{ik}^{(0)}$

3: **for** $b = 1, 2, \ldots,$ maximum iterations **do**

4: Update

$$\hat{\pi}_k^{(b)} \leftarrow \frac{\sum_{i=1}^N \gamma_{ik}^{(b-1)}}{N}$$

5: Fit a weighted quantile regression model for each component. That is, update

$$\hat{\boldsymbol{\beta}}_k^{(b)}(\tau) \leftarrow \arg\min_{\boldsymbol{\beta}_k} \sum_{i=1}^N \gamma_{ik}^{(b-1)} \rho_\tau(y_i - \mathbf{x}_i^T \boldsymbol{\beta}_k), \quad \text{for } k = 1, \ldots, M$$

6: Set $h_k^{(b)} \leftarrow 1.06 \sigma_k N_k^{-1/5}$ separately for each component $k = 1, \ldots, M$

7: Calculate

$$v_{ik}^{(b)} \leftarrow P(T \leq 0), \quad \text{where } T \sim N(e_{ik}^{(b)}, h_k^{(b)})$$

8: Calculate

$$\begin{bmatrix} w_{1k}^{(b)} \\ w_{2k}^{(b)} \end{bmatrix} \leftarrow \begin{bmatrix} \sum_{i=1}^N \gamma_{ik}^{(b-1)} I_1(e_{ik}^{(b)}) & \sum_{i=1}^N \gamma_{ik}^{(b-1)} I_2(e_{ik}^{(b)}) \\ \sum_{i=1}^N \gamma_{ik}^{(b-1)} v_{ik}^{(b)} I_1(e_{ik}^{(b)}) & \sum_{i=1}^N \gamma_{ik}^{(b-1)} v_{ik}^{(b)} I_2(e_{ik}^{(b)}) \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ \tau \end{bmatrix}$$

9: Update the error densities using

$$\hat{g}_k^{(b)}(t) \leftarrow h_k^{-1} \sum_{\ell=1}^2 \sum_{i=1}^N w_{\ell k}^{(b)} \gamma_{ik}^{(b-1)} K\{(t - e_{ik}^{(b)})/h_k\} I_\ell(e_{ik}^{(b)})$$

10: Update the responsibilities

$$\gamma_{ik}^{(b)} \leftarrow \frac{\pi_k^{(b)} \hat{g}_k^{(b)}(e_{ik}^{(b)})}{\sum_{k=1}^M \pi_k^{(b)} \hat{g}_k^{(b)}(e_{ik}^{(b)})}$$

11: **Break** out of the loop if the algorithm converged, i.e. if

$$\text{if } \sum_{k=1}^M \left[|\pi_k^{(b)} - \pi_k^{(b-1)}| + |\boldsymbol{\beta}_k^{(b)} - \boldsymbol{\beta}_k^{(b-1)}| \right] < \text{tolerance}$$

12: **end for**

If the equal variances assumption is sensible, we use a pooled estimate of the error density instead. Wu and Yao (2016) suggests that this pooled estimate may improve the speed of convergence, however it may induce bias in the responsibilities if used inappropriately. The pooled estimate is found by amending steps 8 and 9 of Algorithm 1, described in full below.

Algorithm 2 EM-type algorithm using a Normal kernel, assuming equal variances for error densities

1: Follow initial steps 1 and 2 of Algorithm 1
2: **for** $b = 1, 2, \ldots,$ maximum iterations **do**
3: Follow steps 4 to 7 in the for loop of Algorithm 1
4:

$$\begin{bmatrix} w_1^{(b)} \\ w_2^{(b)} \end{bmatrix} \leftarrow \begin{bmatrix} \sum_{k=1}^{M}\sum_{i=1}^{N}\gamma_{ik}^{(b-1)} I_1(e_{ik}^{(b)}) & \sum_{k=1}^{M}\sum_{i=1}^{N}\gamma_{ik}^{(b-1)} I_2(e_{ik}^{(b)}) \\ \sum_{k=1}^{M}\sum_{i=1}^{N}\gamma_{ik}^{(b-1)} v_{ik}^{(b)} I_1(e_{ik}^{(b)}) & \sum_{k=1}^{M}\sum_{i=1}^{N}\gamma_{ik}^{(b-1)} v_{ik}^{(b)} I_2(e_{ik}^{(b)}) \end{bmatrix}^{-1} \begin{bmatrix} 1 \\ \tau \end{bmatrix}$$

5: Update the pooled error density using

$$\hat{g}^{(b)}(t) \leftarrow h^{-1} \sum_{\ell=1}^{2}\sum_{k=1}^{M}\sum_{i=1}^{N} w_\ell^{(b)} \gamma_{ik}^{(b-1)} K\{(t - e_{ik}^{(b)})/h\} I_\ell(e_{ik}^{(b)})$$

6: Follow steps 10 to 11 in the for loop of Algorithm 1
7: **end for**

3.5 Classification EM-type algorithm

A classification step (C-step) may be inserted between the E and M-steps of the EM algorithm to speed up convergence. Celeux and Govaert (1992) introduce this idea as the Classification EM (CEM) algorithm. This step classifies each observed data point to the component where their classification probability γ_{ik} in the M-step is the greatest.

Formally, the observed data are partitioned $\mathbf{\Gamma} = (\mathbf{\Gamma}_1, \ldots, \mathbf{\Gamma}_M)$ by assigning observations according to

$$\mathbf{\Gamma}_k = \{(x_i, y_i) : \gamma_{ik} = \arg\max_h \gamma_{ih}\}.$$

If any of these partitions contain one or fewer observations, the partitions are regarded as empty and discarded. Subsequent iterations of the algorithm thus continue with $K-1$ partitions. Both the E and the M steps are carried out unchanged.

According to Faria and Soromenho (2010), the CEM algorithm always converges in a finite number of iterations unlike the standard EM and generally converges faster when classes are highly separable and the initial values are close to the truth.

To see if the same holds for our CEM-type algorithm, we will measure the number of iterations to convergence, as well as time and compare the performance of the model with and without the

additional classification step in both the simulation study and the applications section. We will ensure that we examine both highly separable cases and cases where the components overlap to understand the conditions under which this model may perform worse than the ordinary EM-type model in terms of speed or accuracy.

The augmented algorithm is as follows:

Algorithm 3 CEM-type algorithm

Follow steps 1 and 2 of Algorithm 1 or the corresponding steps of Algorithm 2
for $b = 1, 2, \ldots,$ maximum iterations **do**
  Follow steps 4 to 10 in the for loop of Algorithm 1 or the corresponding steps of Algorithm 2
  for $k = 1, \ldots, M$ **do**
    Initialize variable component sum $\Sigma \leftarrow 0$
    for $i = 1, \ldots, N$ **do**
      if $k = \arg\max_h \gamma_{ih}$ **then**
        Set $\gamma_{ik}^{(b)} \leftarrow 1$
        Increment component sum $\Sigma \leftarrow \Sigma + 1$
      else
        Set $\gamma_{ik}^{(b)} \leftarrow 0$
      end if
    end for
    if $\Sigma = 0$ **then**
      Drop component
      $M \leftarrow M - 1$
    end if
  end for
  Follow step 11 of of Algorithm 1 or the corresponding step of Algorithm 2
end for

3.6 Parameter variance estimation

Neither the EM-type nor CEM-type algorithms allow the direct estimation of the variance-covariance matrix of the parameters. Wu and Yao (2016) overcame this by employing three different resampling methods to estimate the variances of the parameter estimates. The last of these, the stochastic EM-type (SEM-type) algorithm, was developed in their research and compared to established resampling methods such as case and model bootstrapping.

3.6.1 Case bootstrapping

The first variance estimation strategy, referred to by Wu and Yao (2016) as case bootstrapping, involves resampling with replacement with equal probability from the dataset and refitting the algorithm to the resampled dataset. This procedure is performed repeatedly, specifically 500 times by the authors, giving different parameter estimates for each dataset. The mean and

variance may be calculated from these parameter estimates, the latter being the quantity which describes the stability of the algorithm.

The simulation results in Wu and Yao (2016) show that case boostrapping may overestimate the true variances. Further, for regression parameters, Wu (1986) shows that case bootstrapping to estimate parameter variances produces biased results and is less consistent than residual bootstrapping. This is highly dependent on the patterns present in the data, however. This method will be illustrated in the simulation study in the following section, as well as the applications in the subsequent section.

3.6.2 Model bootstrapping

The model bootstrapping approach involves creating a synthetic dataset by sampling from the estimated model. New y values are estimated from the model and the $\mathbf{x}$-values are kept constant, no direct resampling is done on the original dataset. The detailed steps are:

Algorithm 4 Model bootstrapping

1: Estimate the model on the original dataset. That is, find

$$\hat{f}(y|\mathbf{x}, \hat{\boldsymbol{\theta}}, \hat{\mathbf{G}}) = \sum_{k=i}^{K} \hat{\pi}_k g_k(y - \mathbf{x}^T \hat{\boldsymbol{\beta}}_k(\tau))$$

2: The latent variables $\mathbf{Z}$ are drawn from a multinomial distribution where $\mathbf{Z} \sim MN(\hat{\pi}_1, \ldots, \hat{\pi}_K, 1)$.

3: The residuals are drawn from the estimated error densities given the realised value of the latent variable for that observation i.e. $e_{ik}^{(b)} \sim \hat{g}_k(t)|\mathbf{Z}_i$

4: Use these residual values to estimate new y-values from $\hat{y_i} = \mathbf{x}_i^T \hat{\boldsymbol{\beta}}_k(\tau) + e_{ik}^{(b)}$ where k is such that $Z_{ik} = 1$. In other words, the component where the value 1 was drawn for the latent variable for that observation.

3.6.3 Stochastic EM

Wu and Yao (2016) describe a stochastic version of the EM-type algorithm in which they insert a stochastic step between the E-step and M-step to estimate the variance-covariance matrix of the parameter estimates. The authors show that it is generally faster and produces a lower MSE than both case and model bootstrapping.

This stochastic variation of the algorithm unfortunately cannot be used in conjuction with the CEM-type algorithm. They are different and incompatible approaches to modeling the belonging of observations to clusters. The stochastic version of the algorithm would reduce to the CEM-type algorithm as a special case if it is performed after the classification step, as component memberships will be sampled with probability one.

4 Simulation study

Three synthetic datasets are described and simulated. First, 500 iterations of both the case and model bootstrapping methods are used to measure the consistency of the EM-type and CEM-type algorithms. We show consistency with the results of the first simulation study in Wu and Yao (2016), next we investigate a dataset with an overlapping region and lastly a 3-component dataset with highly skewed error densities that are not identically distributed.

4.1 Two component single variable data with i.i.d. errors

The generalised description of the data generation process in the first simulation study in Wu and Yao (2016) is a mixture model given by

$$Y = \begin{cases} \beta_{10} + \beta_{11} x_1 + \epsilon & \text{if } Z = 1 \\ \beta_{20} + \beta_{21} x_2 + \epsilon & \text{if } Z = 2 \end{cases}$$

with $P(Z = 1) = \pi_1$ and $P(Z = 2) = 1 - \pi_1$.

The error densities are independent and identically distributed for both components, and are generated from a separate Normal mixture model such that the τth quantile is approximately centered around zero. There is no closed form expression for the quantile function of a mixture of Normal distributions, so the claim that the τth quantile is approximately centered around zero may be observed from the fact that 99.7% of observations are within three standard deviations of the mean and consequently, $\tau \cdot 100\%$ of the observations are below zero. This mixture model is given by

$$\epsilon \sim \tau \cdot N(-1, 1^2) + (1 - \tau) \cdot N(1, 2^2).$$

The simulation in Wu and Yao (2016) has $\beta_{10} = 10, \beta_{11} = -10, \beta_{20} = -10$ and $\beta_{21} = 10$. The two components are simulated with equal probability such that $\pi_1 = 0.5$ and the error densities are estimated with $\tau = 0.5$. The covariates x_1 and x_2 are both drawn from a $U(0, 1)$ distribution.

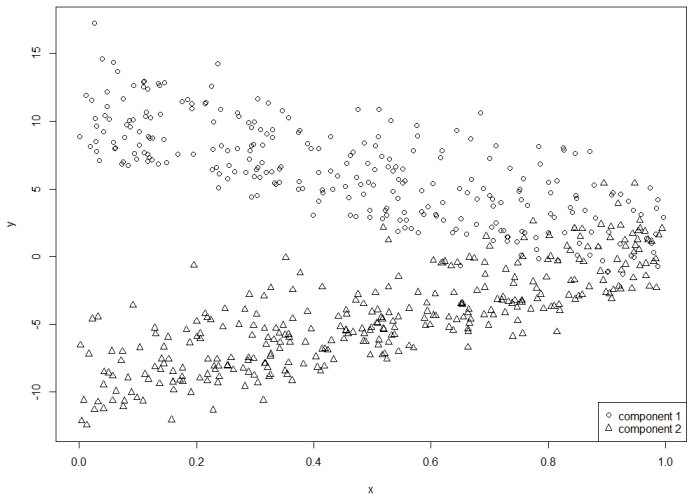

Figure 2: Simulation of a two component Normally distributed dataset for $N = 600$. The components are indicated by the different shapes shown in the legend and they intersect from $x > 0.7$.

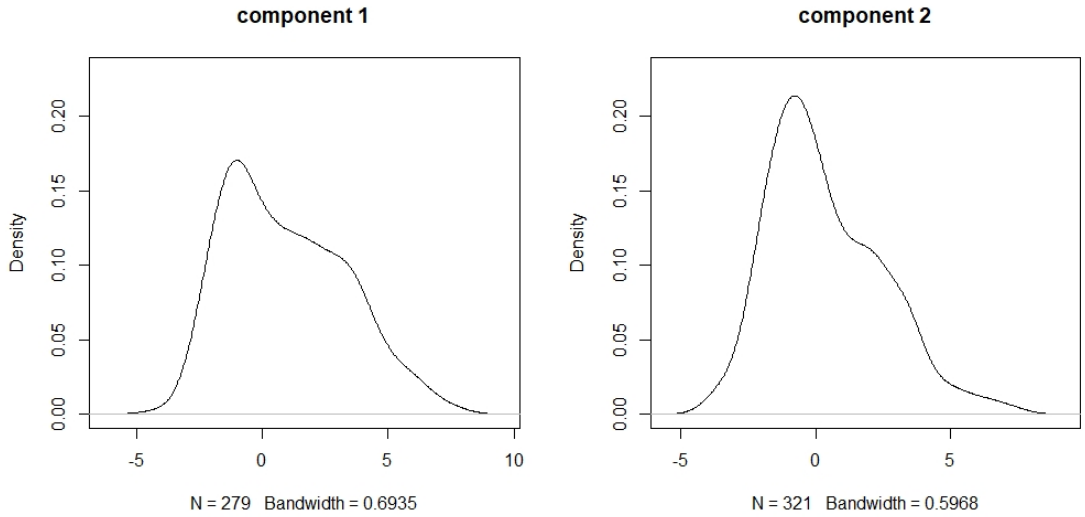

Figure 3: KDEs of error terms for both components of the simulated data in Figure 2. These KDEs are estimated on the residuals found by fitting the known regression equations separately to each component, and are indicative of the shapes to expect when fitting the EM-type and CEM-type algorithms.

Figure 2 shows a plot of such a generated dataset for $N = 600$ and Figure 3 shows the KDEs of the generated error terms. We observe that both error densities are skewed to the right with modes to the left of zero. They are not exactly the same due to randomness inherent in the data generation process. The data plot displays the same skewness, as both components seem to be increasingly sparse for larger values of y.

A typical result of both the EM-type and CEM-type algorithms without assuming equal variances is given in Figure 4. Both algorithms are fitted to the data without assuming equal variances. The observations are classified as observation i belonging to component k if $\gamma_{ik} > 0.5$

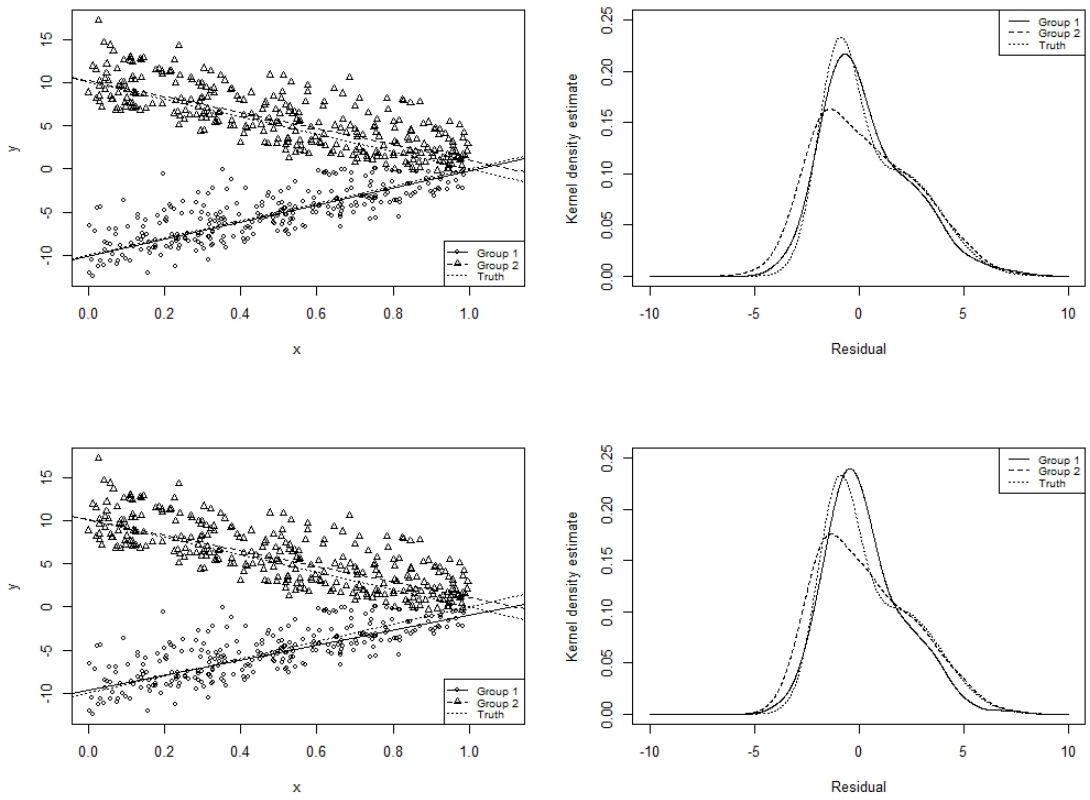

Figure 4: Comparison of EM-type and CEM-type algorithms for simulated data in Figure 2. The fitted regression lines and estimated error densities are given for the EM-type algorithm in the top row and the CEM-type algorithm in the bottom row respectively.

We note only minor differences between the two algorithms from visual inspection. The estimated error densities have the same form and deviate from the true error density due to the data generation process. Neither algorithm shows a remarkable deviation from the regression line either and the biases appear to be within the same order of magnitude.

We use different seed values to generate 500 different datasets for $N = 100$, a further 500 for $N = 300$ and a final 500 for $N = 600$ which is a grand total of 1 500 datasets. For each one of these datasets, we perform 500 case and 500 model bootstrap iterations for all combinations of the EM-type and the CEM-type algorithms assuming both equal and unequal component variances, that is, 4 different models in total. We then use the bootstrapped statistics to calculate the mean, variance and absolute bias of the parameter estimates under these various conditions, as well as the mean and variance of the iterations to convergence and time taken in seconds, to compare the efficiency of the models. The results are provided on the following pages.

Table 1: Case bootstrapping with equal variance on 500 iterations. The parameter estimates, their variance and absolute bias, as well as the speed of convergence are measured for different sample sizes. The results of the CEM-type and EM-type algorithms are given pairwise in rows, and the superior results are in bold.

n	Estimate	Alg	π_1	β_{10}	β_{11}	β_{20}	β_{21}	N iter	Seconds
100	Mean	CEM	0.516	9.223	-9.667	-10.048	10.255	42.020	0.535
		EM	0.517	9.205	-9.521	-9.996	10.116	**13.590**	**0.333**
	Var	CEM	0.002	**0.167**	1.436	0.292	1.371	1967.111	0.309
		EM	0.002	0.191	**1.409**	**0.190**	**0.823**	**61.759**	**0.036**
	|Bias|	CEM	**0.016**	**0.777**	**0.333**	0.048	0.255		
		EM	0.017	0.795	0.479	**0.004**	**0.116**		
300	Mean	CEM	0.541	10.328	-9.502	-9.192	8.383	**4.570**	**0.671**
		EM	0.559	10.306	-9.723	-9.336	8.964	6.920	0.987
	Var	CEM	0.001	0.244	0.991	0.278	0.603	394.187	5.962
		EM	0.001	**0.198**	**0.802**	**0.262**	**0.566**	**3.468**	**0.055**
	|Bias|	CEM	**0.041**	0.328	0.498	0.808	1.167		
		EM	0.059	**0.306**	**0.277**	**0.664**	**1.036**		
600	Mean	CEM	0.437	10.431	-10.137	-9.794	9.022	**3.190**	**1.758**
		EM	0.465	10.604	-10.657	-10.085	10.173	6.540	3.077
	Var	CEM	< 0.001	**0.165**	**0.408**	**0.081**	**0.152**	**1.428**	**0.263**
		EM	< 0.001	0.245	0.560	0.099	0.417	4.150	0.703
	|Bias|	CEM	0.063	**0.431**	**0.137**	0.206	0.978		
		EM	**0.035**	0.604	0.657	**0.085**	**0.173**		

Case bootstrapping with the equal variance assumption shows that the EM-type algorithm is faster with smaller variance for $N = 100$ and the absolute bias is better some of the time. The speed gains from the CEM-type algorithm become more pronounced as the sample size N increases to 300 and 600, however it generally produces more biased results. The CEM-type algorithm also shows less variability at $N = 600$ across the parameter estimates and the efficiency metrics, suggesting a more stable algorithm.

Table 2: Model bootstrapping with equal variance on 500 synthetic datasets. The parameter estimates, their variance and absolute bias, as well as the speed of convergence are measured for different sample sizes. The results of the CEM-type and EM-type algorithms are given pairwise in rows, and the superior results are in bold.

n	Estimate	Alg	π_1	β_{10}	β_{11}	β_{20}	β_{21}	N iter	Seconds
100	Mean	CEM	0.512	9.295	-10.044	-9.899	10.081	23.880	0.329
		EM	0.507	9.230	-9.908	-9.973	10.339	**11.870**	**0.295**
	Var	CEM	**0.001**	0.879	2.679	0.649	2.019	724.006	0.108
		EM	0.002	**0.558**	**1.857**	**0.475**	**1.481**	**170.296**	**0.089**
	|Bias|	CEM	0.012	**0.705**	**0.044**	0.101	**0.081**		
		EM	**0.007**	0.770	0.092	**0.027**	0.339		
300	Mean	CEM	0.519	10.193	-9.274	-8.912	7.660	**2.860**	**0.462**
		EM	0.539	10.554	-10.294	-9.466	9.096	6.820	0.967
	Var	CEM	0.001	**0.158**	**0.634**	**0.155**	**0.609**	**0.788**	**0.013**
		EM	0.001	0.242	0.855	0.180	0.680	3.078	0.052
	|Bias|	CEM	**0.019**	**0.193**	0.726	1.088	2.340		
		EM	0.039	0.554	**0.294**	**0.534**	**0.904**		
600	Mean	CEM	0.419	10.263	-9.711	-9.581	8.461	**3.170**	**1.704**
		EM	0.471	10.562	-10.834	-10.042	10.252	6.680	3.125
	Var	CEM	< 0.001	0.080	**0.287**	**0.082**	**0.228**	**1.031**	**0.180**
		EM	< 0.001	**0.079**	0.351	0.114	0.532	4.624	0.775
	|Bias|	CEM	0.081	**0.263**	**0.289**	0.419	1.539		
		EM	**0.029**	0.562	0.834	**0.042**	**0.252**		

Model bootstrapping with the equal variance assumption also shows the EM-type algorithm generally being faster and with a smaller variance and sometimes absolute bias at the smaller sample size $N = 100$. In contrast to before, the CEM-type algorithm immediately becomes faster and more stable from $N = 300$ and this holds for $N = 600$ as well. We do, however, see the CEM-type algorithm produce an extremely biased estimate for β_{21} at $N = 300$ and to a lesser extent at $N = 600$. Larger sample sizes reduce sampling error and consequently, lead to lower variances as above. On the contrary, they do not necessarily affect bias systematically and it may increase or decrease with sample size in an unpredictable manner.

Table 3: Case bootstrapping with unequal variance on 500 iterations. The parameter estimates, their variance and absolute bias, as well as the speed of convergence are measured for different sample sizes. The results of the CEM-type and EM-type algorithms are given pairwise in rows, and the superior results are in bold.

n	Estimate	Alg	π_1	β_{10}	β_{11}	β_{20}	β_{21}	N iter	Seconds
100	Mean	CEM	0.526	8.940	-8.799	-10.007	10.033	**3.080**	**0.052**
		EM	0.516	9.223	-9.667	-10.048	10.255	42.020	0.535
	Var	CEM	0.002	0.310	3.406	0.338	1.671	**1.165**	**< 0.001**
		EM	0.002	**0.167**	**1.436**	**0.292**	**1.371**	1967.111	0.309
	|Bias|	CEM	0.026	1.060	1.201	**0.007**	**0.033**		
		EM	**0.016**	**0.777**	**0.333**	0.048	0.255		
300	Mean	CEM	0.542	10.164	-9.311	-9.114	8.348	**2.630**	**0.218**
		EM	0.547	10.420	-10.000	-9.415	9.105	17.250	1.090
	Var	CEM	0.001	**0.171**	**0.530**	0.323	0.697	**0.680**	**0.004**
		EM	0.001	0.267	1.024	**0.270**	**0.675**	61.078	0.225
	|Bias|	CEM	**0.042**	**0.164**	0.689	0.886	1.652		
		EM	0.047	0.420	**< 0.001**	**0.585**	**0.895**		
600	Mean	CEM	0.420	10.074	-9.420	-9.658	8.615	**3.190**	**0.781**
		EM	0.467	10.619	-10.735	-10.130	10.298	9.250	1.958
	Var	CEM	< 0.001	**0.163**	**0.412**	**0.087**	**0.168**	**1.133**	**0.050**
		EM	< 0.001	0.175	0.422	0.107	0.406	12.593	0.508
	|Bias|	CEM	0.080	**0.074**	**0.580**	0.342	1.385		
		EM	**0.033**	0.619	0.735	**0.130**	**0.298**		

Case bootstrapping under the unequal variance assumption interestingly displays faster convergence than under the equal variance assumption, despite the data having been generated with equal error densities. This is contrary to the view expressed by Wu and Yao (2016), who stated that the equal densities assumption may improve the model efficiency. The authors also noted that the equal error densities may lead to more biased estimates if the error densities assumption is unreasonable. In the above results, absolute biases seem comparable between models fitted with both equal and unequal variances assumptions. This is expected, given that the data were generated with equal error densities.

Table 4: Model bootstrapping with unequal variance on 500 synthetic datasets. The parameter estimates, their variance and absolute bias, as well as the speed of convergence are measured for different sample sizes. The results of the CEM-type and EM-type algorithms are given pairwise in rows, and the superior results are in bold.

n	Estimate	Alg	π_1	β_{10}	β_{11}	β_{20}	β_{21}	N iter	Seconds
100	Mean	CEM	0.528	8.436	-7.611	-9.696	9.223	**2.750**	**0.042**
		EM	0.512	9.294	-10.044	-9.899	10.081	23.880	0.329
	Var	CEM	0.003	**0.540**	**1.798**	**0.359**	**1.096**	**0.856**	**< 0.001**
		EM	**0.001**	0.879	2.679	0.649	2.019	724.006	0.108
	|Bias|	CEM	0.028	1.564	2.389	0.304	0.777		
		EM	**0.012**	**0.706**	**0.044**	**0.101**	**0.081**		
300	Mean	CEM	0.518	10.159	-9.325	-8.960	7.725	**2.750**	**0.211**
		EM	0.531	10.530	-10.429	-9.553	9.224	15.310	0.956
	Var	CEM	0.001	**0.215**	**0.938**	0.167	0.578	**0.856**	**0.004**
		EM	**< 0.001**	0.300	1.121	**0.129**	**0.559**	82.337	0.280
	|Bias|	CEM	**0.018**	**0.159**	0.675	1.040	2.275		
		EM	0.031	0.530	**0.429**	**0.447**	**0.776**		
600	Mean	CEM	0.420	10.074	-9.420	-9.658	8.615	**3.190**	**0.781**
		EM	0.474	10.488	-10.656	-10.051	10.221	11.000	2.414
	Var	CEM	< 0.001	0.105	0.372	**0.089**	**0.267**	**1.145**	**0.038**
		EM	< 0.001	**0.099**	**0.350**	0.116	0.454	21.798	0.976
	|Bias|	CEM	0.080	**0.074**	**0.580**	0.342	1.385		
		EM	**0.026**	0.488	0.656	**0.051**	**0.221**		

Model bootstrapping without assuming equal variances also yields faster results for the CEM-type algorithm across all sample sizes. Variance is comparable across sample sizes and absolute bias is again larger for the CEM-type algorithm. Overall, the simulation study has shown that the CEM-type algorithm usually performs faster, especially at larger sample sizes, but at the cost of increased absolute bias. The variance and thus stability of both models is never conclusively better or worse, so neither model can be said to be more stable overall or at specific sample sizes. The parameter estimates of the same simulations in Wu and Yao (2016) are given for their stochastic algorithm rather than the resampling methods employed here. Overall, these results are comparable to theirs and exhibit the same behaviours such as bias increasing between $N = 300$ and $N = 600$ for π_1 when assuming equal variances.

4.2 Two component single variable dataset with intersection and i.i.d. errors

The discussion in Wu and Yao (2016) shows that biased estimates can occur when the clusters have imbalanced intersections. This usually happens when the error densities are asymmetric. We investigate this in detail by simulating a dataset of the same form as the dataset in Section 4.1, but with the parameter values $\beta_{10} = -5, \beta_{11} = 10, \beta_{20} = 10$ and $\beta_{21} = -10$ instead. Furthermore, the covariates x_1 and x_2 are drawn from $U(0, 1)$ and $U(0.5, 1.5)$ distributions respectively. As in Section 4.1, the two components are simulated with equal probability such that $\pi_1 = 0.5$. The data are plotted in Figure 5 as follows

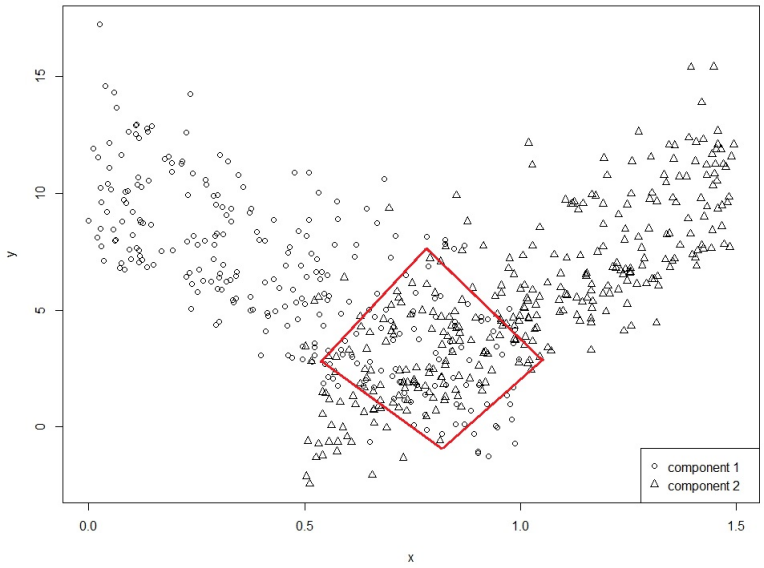

Figure 5: Simulation of two component Normally distributed dataset with intersection enclosed by the red box for $N = 600$. The components are indicated by the shapes shown in the legend and observations belonging to both components are found within the intersection in equal quantities.

In Figure 5, the red shape encloses the region where both components intersect. Both components have asymmetric error densities, but the region contains an equal amount of the high density part of both components. Thus, the imbalance in this intersection is counter-balanced which should reduce bias. If the region contained an overlap of the high density part of one component and the low density part of another, the concentration of data points will influence the fit of the latter component. The details of this are given in the considerations section of Wu and Yao (2016) along with graphical examples and this will not be repeated here. To investigate the impact that this has on the modeling process, we fit both the EM-type and CEM-type algorithm to understand to what extent the parameter estimates are biased.

Figure 6 shows the fitted EM-type algorithm in the top row and the fitted CEM-type algorithm in the bottom row, both assuming unequal variances.

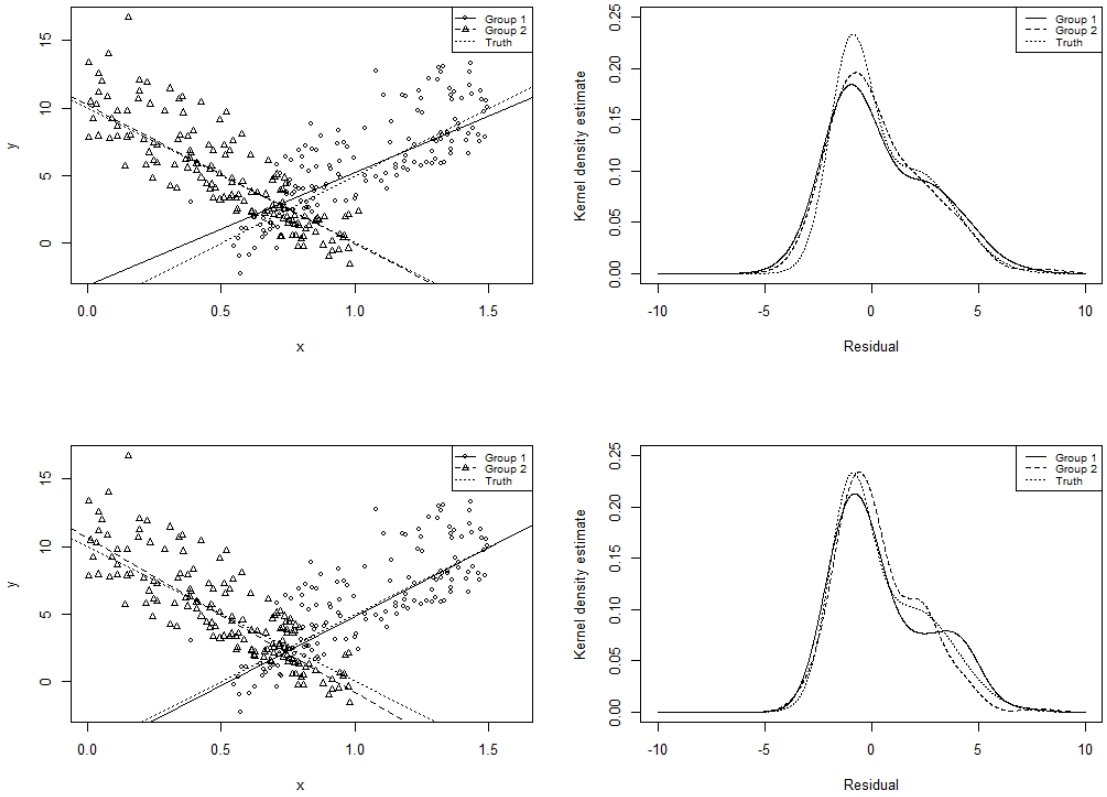

Figure 6: Comparison of EM-type and CEM-type algorithms for simulated data in Figure 5. The fitted regression lines and estimated error densities are given for the EM-type algorithm in the top row and the CEM-type algorithm in the bottom row respectively.

The algorithms indicate comparable biased estimates, albeit for different components, and minor deviations from the true error distribution. Next, we conduct a numerical simulation to study how consistent these results are. This time, we only use a sample size of $N = 300$, but we generate 500 datasets as before and perform a further 500 case and 500 model bootstrapping iterations on each of these datasets, all without assuming equal variances. We calculate the same metrics as before including variance, absolute bias and efficiency metrics. The results are provided on the following page.

Table 5: Summary of parameter estimates using 500 simulations and $N = 300$ without assuming equal variances. The parameter estimates, their variance and absolute bias, as well as the speed of convergence are measured for both case and model bootstrapping. The results of the CEM-type and EM-type algorithms are given pairwise in rows, and the superior results are in bold.

Method	Estimate	Alg	π_1	β_{10}	β_{11}	β_{20}	β_{21}	N iter	Seconds
Case	Mean	CEM	0.513	-4.399	9.407	10.505	-11.173	**6.332**	**0.397**
		EM	0.531	-2.743	8.052	10.39	-10.574	12.368	0.776
	Var	CEM	0.001	4.173	**2.531**	**0.266**	**1.238**	233.573	0.702
		EM	$<$ **0.001**	**4.152**	2.830	0.373	1.338	**71.648**	**0.236**
	|Bias|	CEM	**0.013**	**0.601**	**0.593**	0.505	1.173		
		EM	0.031	2.257	1.948	**0.390**	**0.574**		
Model	Mean	CEM	0.471	-5.611	10.378	10.603	-11.591	**4.852**	**0.314**
		EM	0.513	-3.166	8.373	10.127	-10.176	11.226	0.706
	Var	CEM	0.001	**0.382**	**0.407**	**0.258**	**0.261**	**3.601**	**0.012**
		EM	$<$ **0.001**	0.420	0.519	0.354	0.456	28.616	0.096
	|Bias|	CEM	0.029	**0.611**	**0.378**	0.603	1.591		
		EM	**0.013**	1.834	1.627	**0.127**	**0.176**		

As expected, the CEM-type algorithm is more efficient, taking less than half the time and number of iterations to converge on average. Addtionally, the CEM-type algorithm has a lower variance and is more stable, but the differences appear to be minor at this sample size. Similar to Figure 6, we see the CEM-type algorithm produce a more biased estimate for one component, while the EM-type algorithm produces a more biased estimate for the other. The equal parts of both components in the imbalanced intersection has canceled out some of the bias. Overall, the CEM-type algorithm converged faster without greatly affecting the bias, compared to the EM-type algorithm.

4.3 Three component multivariable dataset with non-i.i.d. errors

The final simulation study will illustrate that the algorithms are applicable to datasets with more than two components and that the error distributions of these components need not be identical. We generate data according to the following set of equations.

$$Y = \begin{cases} \beta_{10} + \beta_{11}x_1 + \beta_{12}x_1 + \epsilon & \text{if } Z = 1 \\ \beta_{20} + \beta_{21}x_2 + \beta_{22}x_2 + \epsilon & \text{if } Z = 2 \\ \beta_{30} + \beta_{31}x_2 + \beta_{32}x_2 + \epsilon & \text{if } Z = 3 \end{cases}$$

with $P(Z = 1) = \pi_1$, $P(Z = 2) = \pi_2$ and $P(Z = 3) = 1 - (\pi_1 + \pi_2)$.

We use zero intercepts, that is $\beta_{10} = \beta_{20} = \beta_{30} = 0$, as well as the coefficients $\beta_{11} = \beta_{12} = -10$, $\beta_{21} = \beta_{22} = 0$ and $\beta_{31} = \beta_{32} = 10$. The mixing probabilities used are $\pi_1 = \pi_2 = \pi_3 = 1/3$ and the covariates x_1 and x_2 are drawn from a $U(0, 1)$ distribution. The error densities are generated from Weibull distributions. The error terms follow a Weibull distribution with shape parameter κ and scale parameter λ, that is, $\epsilon_k \sim \text{Weib}(\kappa, \lambda)$. To ensure that we center the appropriate quantile around zero, we note that this distribution has the quantile function

$$G^{-1}(\tau, \kappa, \lambda) = \lambda[-\ln(1 - \tau)]^{1/\kappa}, \quad \tau \in [0, 1).$$

The error densities are thus given by

$$\begin{aligned} \epsilon_1 &\sim \text{Weib}(1.5, 0.50) - G^{-1}(\tau, 1.5, 0.50) \\ \epsilon_2 &\sim \text{Weib}(1.5, 4.00) - G^{-1}(\tau, 1.5, 4.00) \\ \epsilon_3 &\sim \text{Weib}(1.5, 0.25) - G^{-1}(\tau, 1.5, 0.25). \end{aligned}$$

The Weibull distribution with shape parameter $\kappa = 1.5$ is skewed to the right. The scale parameter does not alter the skewness of the distribution. Such a distribution with scale parameter $\lambda = 1$ for example, takes the following form as in Figure 7.

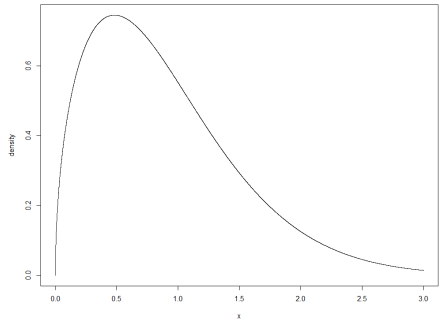

Figure 7: Weibull density with $\kappa = 1.5$ and $\lambda = 1$

The algorithms are fitted without assuming equal variances since the data was generated with different error densities for each component. Similarly to before, we generate 500 synthetic datasets of size $N = 300$ and perform both 500 case and 500 model bootstrap iterations and calculate the same metrics.

Table 6: Case bootstrap assuming equal variances for the Weibull data using 500 iterations of size $N = 300$. The parameter estimates, their variance and absolute bias, as well as the speed of convergence are measured for different sample sizes. The results of the CEM-type and EM-type algorithms are given pairwise in columns, and the superior results are in bold.

	Mean		Var		|Bias|	
	CEM	EM	CEM	EM	CEM	EM
π_1	0.306	0.303	0.001	0.001	**0.027**	0.030
π_2	0.334	0.337	0.001	0.001	**0.001**	0.004
π_3	0.360	0.360	0.001	0.001	0.027	0.027
β_{10}	-0.023	-0.016	**0.004**	0.005	0.023	**0.016**
β_{11}	-9.983	-9.984	**0.011**	0.013	0.017	**0.016**
β_{12}	-9.772	-9.777	0.008	**0.007**	0.228	**0.223**
β_{20}	0.207	0.184	**1.184**	1.241	0.207	**0.184**
β_{21}	-0.449	-0.432	1.301	**1.259**	0.449	**0.432**
β_{22}	0.145	0.203	**1.576**	1.683	**0.145**	0.203
β_{30}	-0.038	-0.035	0.002	0.002	0.038	**0.035**
β_{31}	10.089	10.086	0.003	0.003	0.089	**0.086**
β_{32}	9.981	9.978	0.003	0.003	**0.019**	0.022
Iterations	**1.986**	10.182	**0.198**	107.48		
Seconds	**0.242**	0.864	**0.006**	0.662		

For case bootstrapping, the CEM-type algorithm is approximately 4 times faster. The stability of both models is comparable with neither one completely dominating the other. On the other hand, the increase in bias is clearly the price paid for the improvement in speed. The CEM-type model does not provide a worse fit in all cases though and the mixing probabilities are especially closer to the truth, so it may yield more promising results if the model is to be used for classification purposes, though this is not further explored.

Table 7: Model bootstrap assuming equal component variances for the Weibull data using 500 synthetic datasets of size $N = 300$. The parameter estimates, their variance and absolute bias, as well as the speed of convergence are measured for different sample sizes. The results of the CEM-type and EM-type algorithms are given pairwise in columns, and the superior results are in bold.

	Mean		Var		|Bias|	
	CEM	EM	CEM	EM	CEM	EM
π_1	0.307	0.303	0.001	0.001	**0.026**	0.030
π_2	0.329	0.339	0.001	0.001	**0.004**	0.006
π_3	0.363	0.358	0.001	0.001	0.030	**0.025**
β_{10}	-0.026	-0.015	0.010	0.010	0.026	**0.015**
β_{11}	-9.960	-9.970	0.017	**0.016**	0.040	**0.030**
β_{12}	-9.789	-9.802	**0.015**	0.016	0.211	**0.198**
β_{20}	-0.032	-0.154	0.728	0.840	**0.032**	0.154
β_{21}	-0.613	-0.449	**1.186**	1.269	0.613	**0.449**
β_{22}	0.592	0.674	**1.186**	1.245	**0.592**	0.674
β_{30}	-0.038	-0.036	**0.002**	0.003	0.038	**0.036**
β_{31}	10.088	10.087	0.004	0.004	0.088	**0.087**
β_{32}	9.980	9.972	0.004	0.004	**0.020**	0.028
Iterations	**1.902**	8.008	**0.177**	48.529		
Seconds	**0.239**	0.684	**0.011**	0.296		

Model bootstrapping further corroborates the results from case bootstrapping. The efficiency of the CEM-type algorithm is again 4 times faster with variance very similar across both algorithms. The CEM-type algorithm once again produces biased estimates for the most part and appears to identify the mixing probabilities more accurately. We have shown that both algorithms succeed at modeling data where the components do not have identical error distributions, however, the bias for component 2 is clearly larger than that for the other components. Morover, the magnitude of the bias of the β estimates for a component, appears to be positively correlated with the scale parameter λ of the error distribution of that component, though this is merely an observation and not sufficient evidence to make a conclusive claim about a larger tendency of the relationship between estimator bias and the underlying densities.

5 Applications

5.1 Tone perception

The first application is a tone perception experiment originally analysed by Cohen (1984). In this experiment, the extent to which a tuning ratio affects the perception of a fundamental tone was investigated and compared to two musical perception theories. This dataset is available in the fpc R package (Hennig, 2020).

The study consisted of 150 repetitions recorded from a single musician. The independent variable is the actual stretching ratio, which is the ratio of difference in frequencies between a fundamental tone and the electronically generated overtone added to it. The response variable is the perception of this ratio. This dataset contains two homogeneous components. One corresponds to the correct tuning and the other to the tuning of the first overtone. This dataset has the property that the two classes are separable over the majority of the domain of the independent variable, but intersect where the first overtone is equal to the correct tuning.

Some mixture of regressions models have historically been fitted to the dataset. Doğru and Arslan (2017) use their own adaptation of an EM-type algorithm to fit the skew t mixture regression model which does not rely on the assumption of normality, but still employs a likelihood function and consequently, the authors use the AIC and BIC criteria for model selection. Their focus is on studying the robustness of the EM-type algorithm with a skew t error distribution. To this end, they first fit their model to the original dataset, then add outlying data points at $(0, 5)$ and refit the model to measure the changes in parameter estimates in the presence of these additional high leverage data points. The results shown before augmenting the dataset are in line with the parameter estimates obtained by Wu and Yao (2016), however the addition of the high leverage points drastically affect the results. The work done by Song et al. (2014) goes some way to reducing the impact of leverage points in the x-direction by assuming that the errors follow a Laplace distribution and modifying the EM-type algorithm to trim high leverage points.

More pertinent to this work, Wu and Yao (2016) fit a mixture of median regressions with $\tau = 0.5$ to the dataset and identify the two components with $\hat{\pi}_1 = 0.373$ and median regression lines with coefficients $\hat{\beta}_{10} = 0.003, \hat{\beta}_{11} = 0.999$ for the first component and $\hat{\beta}_{20} = 1.950, \hat{\beta}_{21} = 0.030$ for the second. This analysis displays the use of the mixtures of quantile regressions model as an exploratory tool. The error densities do not appear highly skewed however, so the intersection of data points is not expected to be imbalanced in the sense that it would introduce bias in the parameter estimates. We fit both the CEM-type algorithm and the EM-type algorithm of Wu and Yao (2016) and compared the results. The Shapiro-Wilk test for Normality is also conducted on the residuals of the fitted models to determine whether the Normality assumption would be appropriate and consequently, if mixtures of mean regressions could have been used instead.

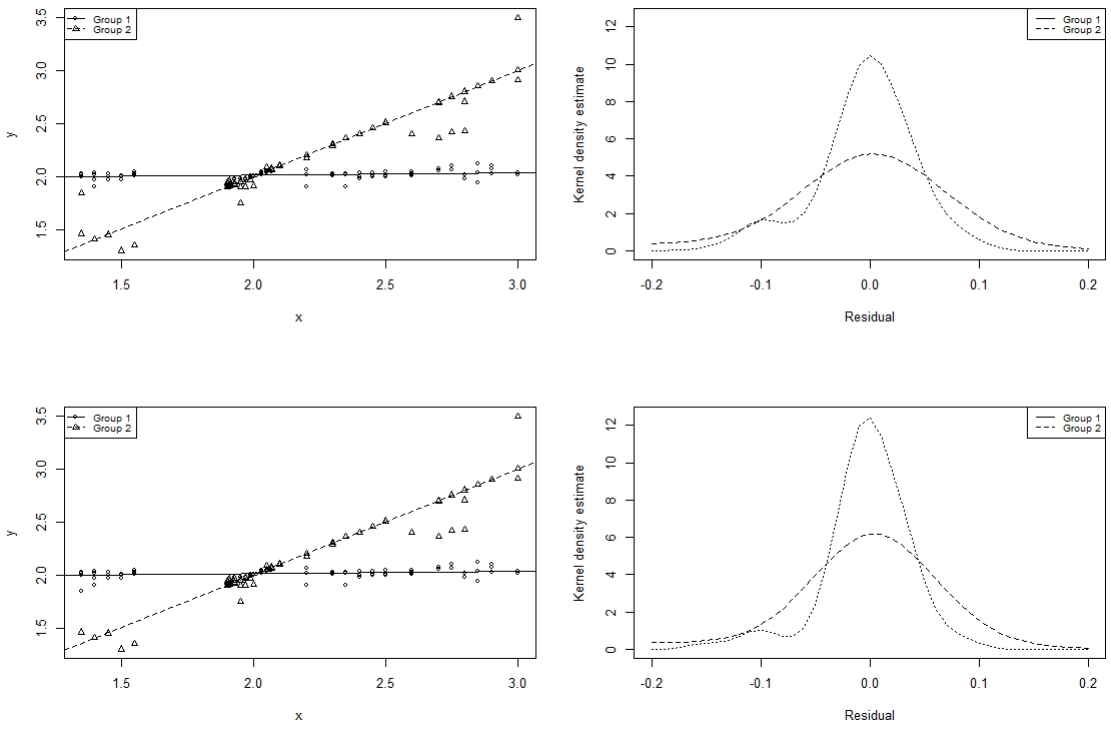

Figure 8: Tone data with fitted EM-type and CEM-type algorithms in the top and bottom rows respectively. Classifications for the observations are given by the shapes shown in the legends. The corresponding error densities are given in the column on the right.

The results of the EM-type algorithm are displayed on the top row and the CEM-type on the bottom row. Both algorithms indeed produce very similar results. The only differences are the classification of some observations, such as the one near coordinate $(1.4, 1.9)$ for example, which seems to be correctly classified only by the CEM-type algorithm in this case. These results are consistent with the findings in Wu and Yao (2016), except for the observations near $(2.6, 2.4)$ that were classified differently than both methods employed here.

Furthermore, the estimated error densities show the same structure as in Wu and Yao (2016) where they cross to the left of the median, but a different structure to the right of the median, which explains why some observations were classified differently. The peaks of the error densities are closer to one another in Wu and Yao (2016) as well. The impact of these differences depend on whether the model will be used for classification or regression. It is not known what initialization strategy was used by Wu and Yao (2016) and this is one potential source of variation.

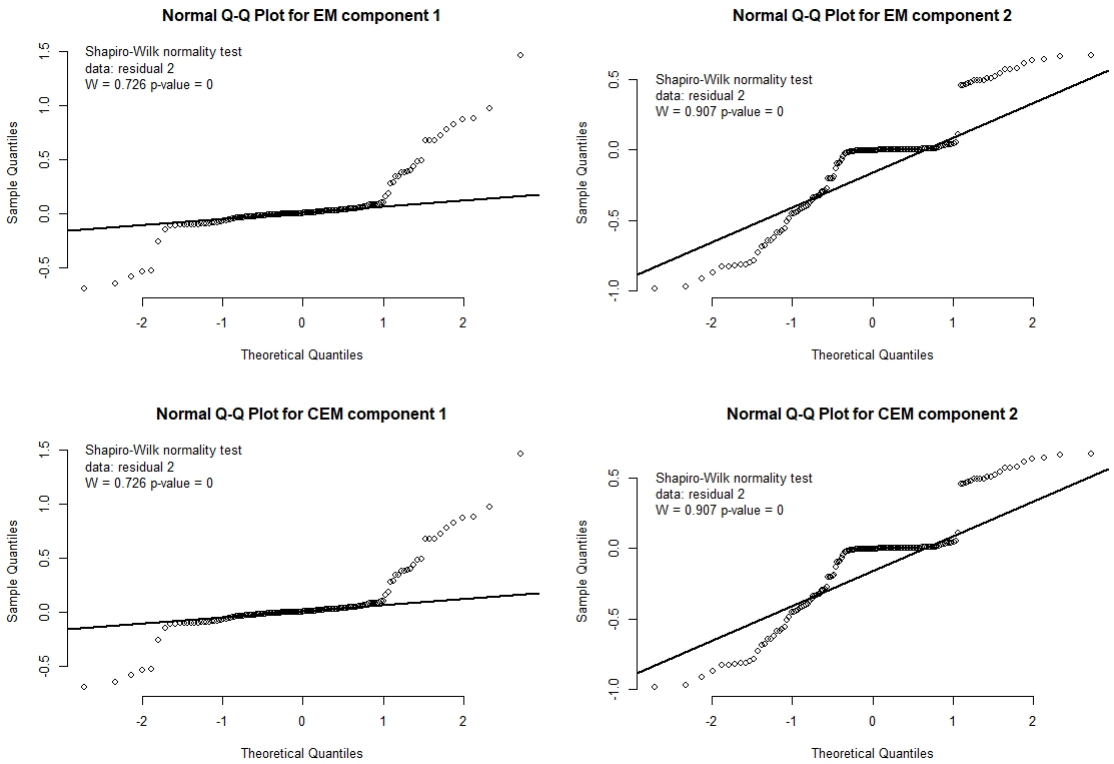

Figure 9: Normal Q-Q plots for the Tone data residuals are given for the EM-type algorithm in the top row and the CEM-type algorithm in the bottom row. Each plot is annoted with the Shapiro-Wilk test results.

The Q-Q plots for both algorithms appear identical. This is due to the closeness of the estimated betas below. For both algorithms, there is systematic deviation from Normality. This suggests that the Normality assumption is violated and a mixture of mean regressions model is expected to provide an inappropriate fit. This Q-Q plot should, however, be treated as a guideline rather than a formal test according to Wu and Yao (2016).

The parameter estimates are given in the following table and differ only for component 2 in the order of 10^{-3}. There is thus not a major difference when choosing whether to employ the EM-type or CEM-type algorithm and given the small size of the dataset, speed is not a deciding factor either.

Table 8: Tone data parameter estimates

	CEM	EM
π_1	0.587	0.578
β_{10}	1.964	1.964
β_{11}	0.023	0.023
β_{20}	0.003	0.005
β_{21}	0.999	0.998

5.2 Melbourne Daily Maximum Temperatures

This dataset contains the daily maximum temperatures in Melbourne, Australia over the period 1981 - 1990. As part of their exploratory analysis, Hyndman et al. (1996) noted that each day's maximum temperature is dependent on the maximum temperature the day before. Days when the temperature is under 30°C are usually followed by hotter days, while days with a maximum temperature greater than 30°C tend to be followed by cooler days.

The authors plotted the daily maximum temperatures as a response variable against the previous day's maximum temperature as a covariate and showed that this lagged dataset suggests the existence of the aforementioned two components towards the right side of the plot. The plot is shown below in Figure 10

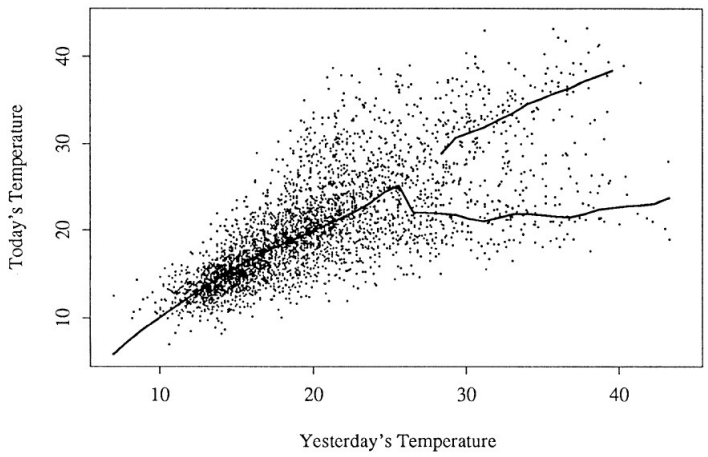

Figure 10: The lagged scatterplot in Hyndman et al. (1996) of each day's maximum temperature against the previous day's maximum temperature. The lines are model regression functions fitted by the authors.

These components aren't easily separable, however, and do somewhat resemble a heteroscedastic dataset with an increasing variance over the domain of the dependent variable. Hyndman et al. (1996) go on to fit a modal regression model using a kernel density estimate to analyse the conditional density of the lagged dataset and find graphically that the estimated conditional mean and variance display the same forking behaviour.

Koenker and Hallock (2001) take a different approach and fit 12 quantile regression functions to the data. These curves are used to construct conditional densities of today's maximum temperature at certain values of the previous day's maximum temperature. That is, given a certain x-value, the concentration of the density curves in the y-direction approximates the conditional density function. This corroborates the forking seen in the data where, conditional densities at lower x-values tend to be unimodal and contrary to that, conditional densities at higher x-values tend to be more bimodal.

This dataset has not been analysed with a mixture of quantile regressions. One would expect to indentify two distinct components in the data with significant overlap towards the left-hand side of the x-domain. Both the EM-type and CEM-type models are fitted with $\tau = 0.5$, $K = 2$ and without assuming equal variances for the error densities of the components. The results are plotted in Figure 11 below with the EM-type algorithm's output on the top row and the CEM-type algorithm's output on the bottom row. The corresponding error density estimates are also plotted.

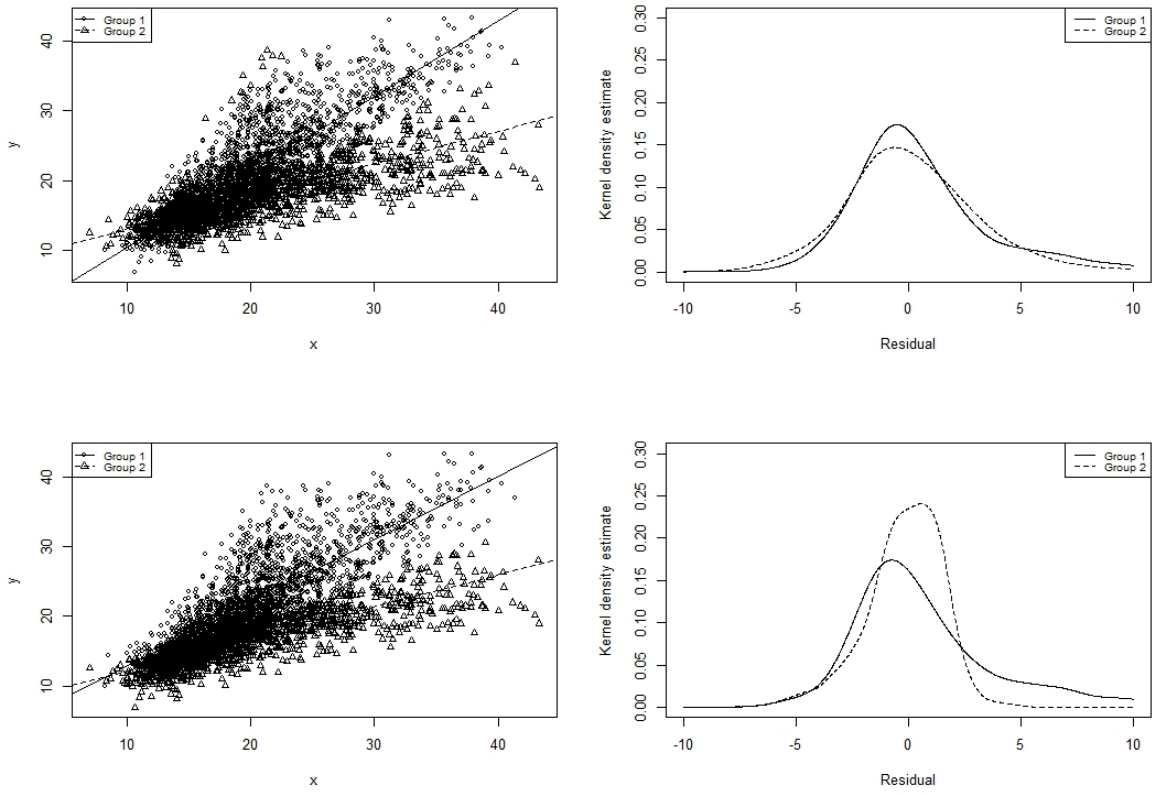

Figure 11: Temperature data with fitted EM-type and CEM-type in the top and bottom rows respectively for $\tau = 0.5$. The classification of observations is indicated by the shapes in the legends and the estimated densities are plotted on the right.

The estimated regression lines manage to identify the same two components found in the work of Hyndman et al. (1996) with some differences in the estimated regression lines, as well as differences in the classification of observations. The CEM-type algorithm produces a skewed error density estimate, compared to the EM-type algorithm which produces symmetric error density estimates for both components. Even though the fitted regression lines for the CEM-type algorithm appear reasonable, this suggests that the observations are less evenly allocated under this scheme, leading to a bias in the regression line for the component.

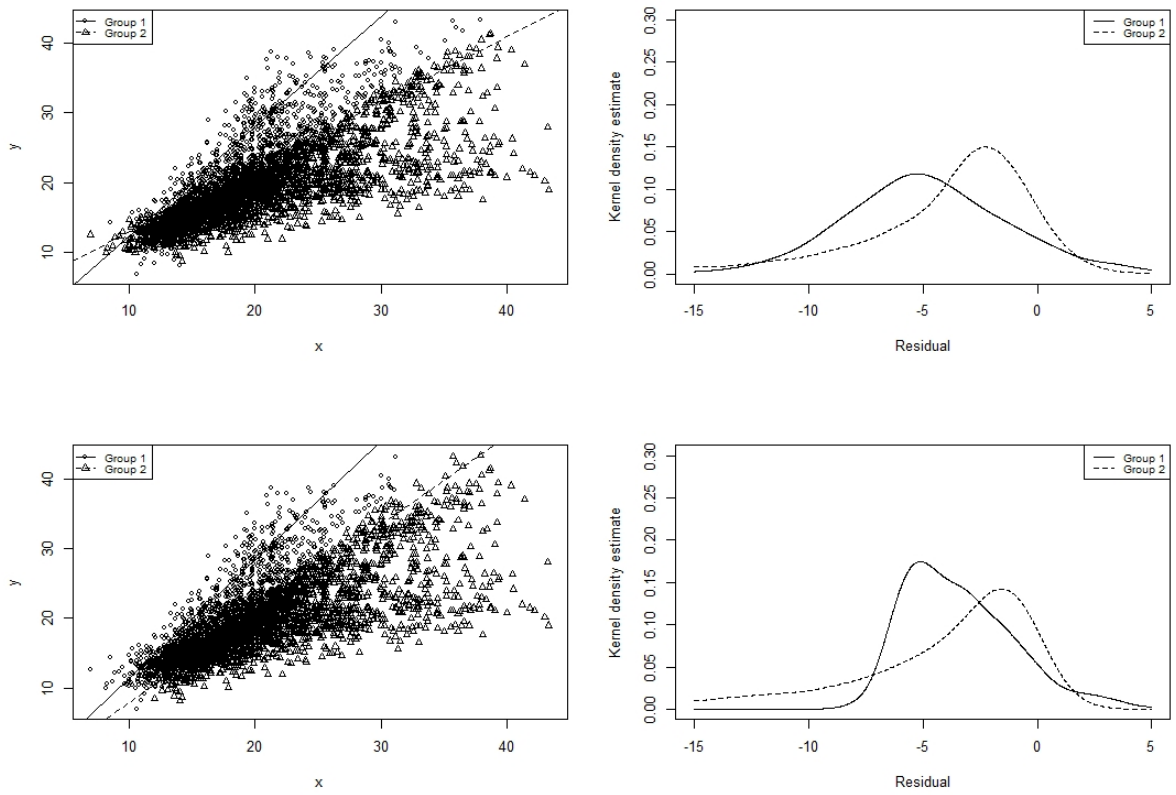

Figure 12: Temperature data with fitted EM-type and CEM-type in the top and bottom rows respectively for $\tau = 0.9$. The classification of observations is indicated by the shapes in the legends and the estimated densities are plotted on the right.

Rerunning the same comparison for $\tau = 0.9$, we find that the two algorithms agree on the classification of 74.541% of the observations. Both models classify many more points as belonging to the same component and allocate only the minority of observations to the other. This agrees with the error density estimates where the component that most of the observations was classfied as belonging to, has an error density highly skewed to the left, which is the region in which the bulk of these observations lie.

Usually, we would expect the classification of observations to be very similar to those in Figure 11 and the regression lines within these classifications to estimate the 90th quartile, but both algorithms have failed to capture this, due to the fact that the data are not easily separable. Additionally, the CEM-type algorithm again behaves worse than the EM-type algorithm in terms of bias, which is readily observed from the error density estimate.

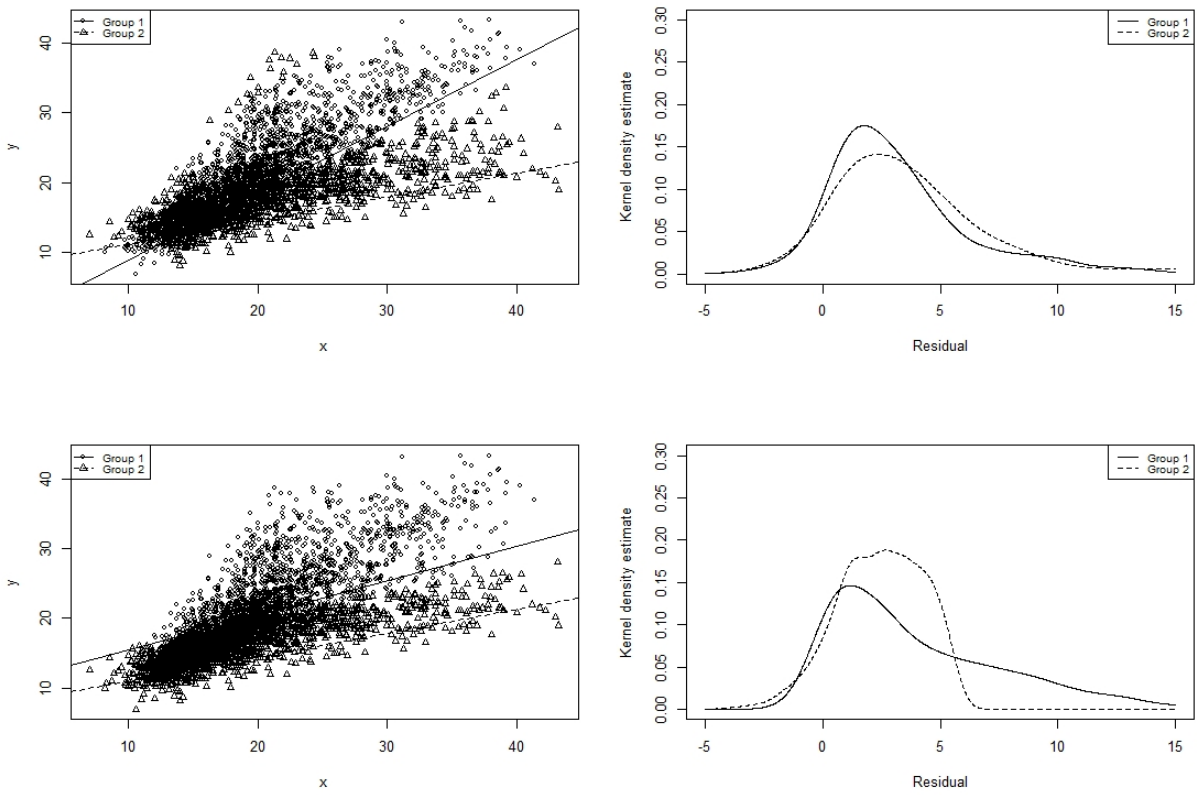

Figure 13: Temperature data with fitted EM-type and CEM-type in the top and bottom rows respectively for $\tau = 0.1$. The classification of observations is indicated by the shapes in the legends and the estimated densities are plotted on the right.

Finally, running the same comparison for $\tau = 0.1$, we find that the two algorithms agree on the classification of 66.840% of the observations. This displays the inverse of the behaviour shown in Figure 12. Observations are again allocated unequally, but in the opposite direction. The error density estimate for component 2 of the CEM-type algorithm is especially concerning, with corresponding regression lines that appear almost parallel. Overall, the CEM-type model has produced undesirable results in this instance, as both fitted regression lines are within the bottom half of the data range, and it can be seen that a much larger portion of the observations were classified as belonging to the top component.

An interesting observation concerning the EM-type algorithm is that points further from line 2 than line 1 in Figure 13 are classified as belonging to line 2, but this may be seen in the error density estimate where the tails for the components cross over, and it implies that the probability of observing extremely large residuals for component 2 is greater than the probability of observing smaller residuals for component 1 - there is a crossover point as shown in Figure 14 below. For example, the density function for component 1 evaluated at 15 is 0.0018 while the density function for component 2 evaluated at 20 is 0.0024.

This behaviour is undesirable, and the model would not provide accurate classifications. It should not substantially affect the use of the model for regression purposes over the fact that the regression lines are biased as previously discussed. Though we have only used the guidelines in Silverman (2018) to determine the smoothing parameter h, this issue may be alleviated by changing the value of h, though this is not explored in further detail here.

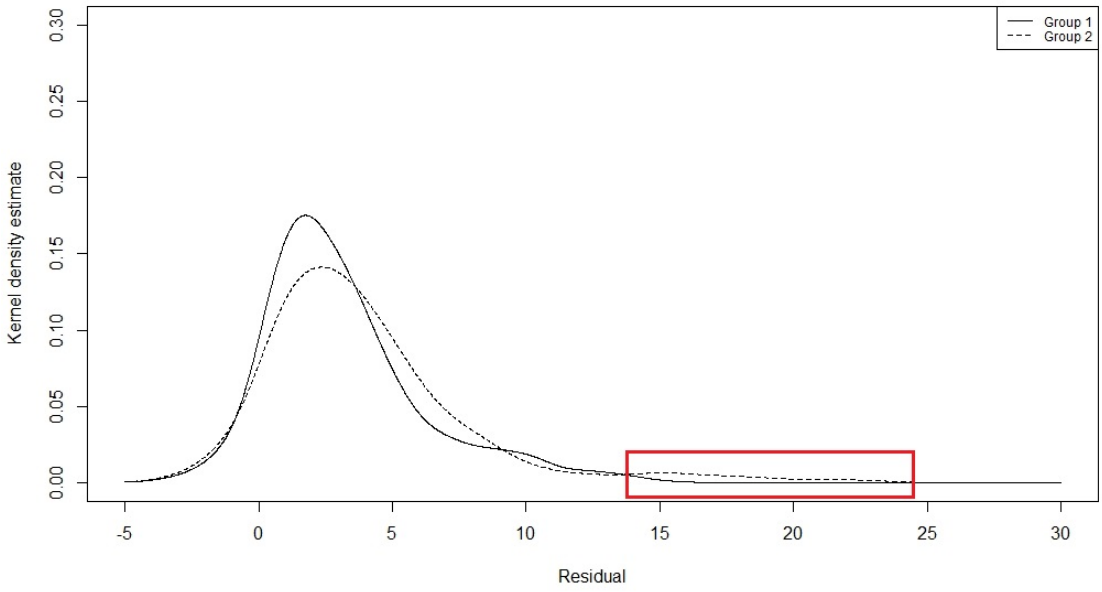

Figure 14: An enlarged plot of the error density estimates fitted by the EM-type algorithm to the Melbourne data. The thick tail in the error density estimate of component 2 is where large residuals may be incorrectly classified as belonging to component 2, despite being nearer to component 1.

When modeling data that is not easily separable within the mixtures of quantile regressions framework, care must be taken when choosing the value of τ. It is also not advised to use the CEM-type algorithm as it produces biased results and this is one clear example of where the algorithm fails to replace the standard EM-type algorithm. The increased bias induced by the CEM-type algorithm compared to EM-type algorithm in this example is an extreme case of the larger bias of CEM-type algorithm as shown numerically in Faria and Soromenho (2010).

6 Conclusion

This work explored mixtures of quantile regressions and in particular, the semi-parametric EM-type algorithm used to fit this model. We explored the theory given in Wu and Yao (2016) and detailed the steps underlying the EM-type algorithm. This involved showing the connection of the constrained kernel density estimator used to model the component error distributions to the work in Hall and Presnell (1999) with unified notation and verifying that the Lagrangian optimisation does indeed minimize the distance between the constrained weights in the KDE and the uniform weights $1/n$. The final EM-type algorithm was concisely described using the Gaussian kernel as a special case, along with matrix equations and probability functions rather than integrals for easy numerical implementation. An exposition of the resampling schemes used to estimate parameter bias and variances was also provided. These were mentioned in Wu and Yao (2016), but they focused on the stochastic EM-type algorithm to evaluate the stability of the parameter estimates instead.

Built upon this foundation, a classification step inspired by the work of Faria and Soromenho (2010) was added between the E and M steps of the algorithm in Wu and Yao (2016). This had not been done for mixtures of quantile regressions before, and the subsequent simulation studies and applications focused on evaluating the speed and accuracy of this algorithm, and comparing it to the EM-type algorithm. It was expected to always converge in a finite number of iterations and converge faster than the EM-type algorithm, especially at larger sample sizes, at the cost of increased bias, in the same way as in Faria and Soromenho (2010). The results of all three simulation studies confirmed that the CEM-type algorithm may be successfully employed on datasets with non-symmetric error densities and intersecting data regions, always converged in a finite number of iterations, and was decidedly faster for larger sample sizes, such as $N = 600$. The bias and variance of parameter estimates were not consistently worse for the CEM-type model and appeared to better estimate mixing probabilities at all, suggesting that it may be a superior classification model in certain circumstances.

Both algorithms yielded similar results for the Tone data study. This was a highly separable case with a sample size of 150 observations. It was shown that the error densities are not Normally distributed and the mixtures of quantile regressions model is useful for exploratory data analysis, as well as classification and regression on this dataset. On the contrary, the CEM-type algorithm proved weaker than the EM-type algorithm when the data were not easily separable. This was explored in the lagged Melbourne daily maximum temperature dataset. It may however be argued that the EM-type algorithm also failed to provide satisfactory results in this case, especially for $\tau = 0.1$ or $\tau = 0.9$, though this was more pronounced for the CEM-type algorithm which classified a majority of the observations to a single component.

We conclude that the CEM-type algorithm provides superior performance to the EM-type algorithm for larger sample sizes where speed of convergence is essential, as well as for highly separable datasets. The EM-type algorithm on the other hand is preferred for smaller or less separable datasets, especially if classification is the aim of the analysis. The semi-parametric mixtures of quantile regressions modeling framework is a useful tool in data analysis, not only for classification, but exploratory data analysis, robust regression and modeling heterogeneity in the presence of covariates. It may be applied in similar contexts as mixtures of mean regressions, but provides additional information of the conditional distribution of each component by describing the conditional quantiles thereof.

This mini dissertation treated the number of components M as a quantity known in advance. Further work could explore model selection techniques for semi-parametric mixtures of quantile regressions. The Normal kernel was used in both algorithms with bandwidth h as given by Silverman (2018). The analysis could be expanded upon by investigating the effect of using different kernel functions or varying the smoothing parameter. Lastly, the problem of imbalanced intersections described in the considerations section of Wu and Yao (2016) remains open. This mini dissertation observed imbalanced intersections through the lens of the CEM-type algorithm and it neither alleviates nor exacerbates the problem.

References

Belloni, A. and Chernozhukov, V. (2011), '1-penalized quantile regression in high-dimensional sparse models', *The Annals of Statistics* **39**(1), 82–130.

Bowman, A. W. and Azzalini, A. (1997), *Applied smoothing techniques for data analysis: the kernel approach with S-Plus illustrations*, Vol. 18, OUP Oxford.

Celeux, G. and Govaert, G. (1992), 'A classification em algorithm for clustering and two stochastic versions', *Computational statistics & Data analysis* **14**(3), 315–332.

Chamroukhi, F. (2016), 'Unsupervised learning of regression mixture models with unknown number of components', *Journal of Statistical Computation and Simulation* **86**(12), 2308–2334.

Cho, H., Kim, Y.-j., Jung, H. J., Lee, S.-W. and Lee, J. W. (2008), 'Outlierd: an r package for outlier detection using quantile regression on mass spectrometry data', *Bioinformatics* **24**(6), 882–884.

Cohen, E. A. (1984), 'Some effects of inharmonic partials on interval perception', *Music Perception* **1**(3), 323–349.

Cole, T. J. and Green, P. J. (1992), 'Smoothing reference centile curves: the lms method and penalized likelihood', *Statistics in medicine* **11**(10), 1305–1319.

DeSarbo, W. S. and Cron, W. L. (1988), 'A maximum likelihood methodology for clusterwise linear regression', *Journal of classification* **5**(2), 249–282.

Doğru, F. Z. and Arslan, O. (2017), 'Robust mixture regression based on the skew t distribution', *Revista Colombiana de Estadística* **40**(1), 45–64.

Edgeworth, F. Y. (1888), 'Xxii. on a new method of reducing observations relating to several quantities', *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* **25**(154), 184–191.

Farcomeni, A. (2012), 'Quantile regression for longitudinal data based on latent markov subject-specific parameters', *Statistics and Computing* **22**(1), 141–152.

Faria, S. and Soromenho, G. (2010), 'Fitting mixtures of linear regressions', *Journal of Statistical Computation and Simulation* **80**(2), 201–225.

García-Escudero, L. A., Gordaliza, A., Mayo-Íscar, A. and San Martín, R. (2010), 'Robust clusterwise linear regression through trimming', *Computational Statistics & Data Analysis* **54**(12), 3057–3069.

Hall, P. and Presnell, B. (1999), 'Density estimation under constraints', *Journal of Computational and Graphical Statistics* **8**(2), 259–277.

Hastie, T., Tibshirani, R., Friedman, J. H. and Friedman, J. H. (2009), *The elements of statistical learning: data mining, inference, and prediction*, Vol. 2, Springer.

Hennig, C. (2020), *fpc: Flexible Procedures for Clustering*. R package version 2.2-9.
URL: *https://CRAN.R-project.org/package=fpc*

Hunter, D. R. and Young, D. S. (2012), 'Semiparametric mixtures of regressions', *Journal of Nonparametric Statistics* **24**(1), 19–38.

Hyndman, R. J., Bashtannyk, D. M. and Grunwald, G. K. (1996), 'Estimating and visualizing conditional densities', *Journal of Computational and Graphical Statistics* **5**(4), 315–336.

Koenker, R. and Geling, O. (2001), 'Reappraising medfly longevity: a quantile regression survival analysis', *Journal of the American Statistical Association* **96**(454), 458–468.

Koenker, R. and Hallock, K. F. (2001), 'Quantile regression', *Journal of economic perspectives* **15**(4), 143–156.

Lauridsen, S. (2000), 'Estimation of value at risk by extreme value methods', *Extremes* **3**(2), 107–144.

McFarlane, C., Raheja, G., Malings, C., Appoh, E. K., Hughes, A. F. and Westervelt, D. M. (2021), 'Application of gaussian mixture regression for the correction of low cost pm2. 5 monitoring data in accra, ghana', *ACS Earth and Space Chemistry* **5**(9), 2268–2279.

Meinshausen, N. and Ridgeway, G. (2006), 'Quantile regression forests.', *Journal of machine learning research* **7**(6).

Silverman, B. W. (2018), *Density estimation for statistics and data analysis*, Routledge.

Song, W., Yao, W. and Xing, Y. (2014), 'Robust mixture regression model fitting by laplace distribution', *Computational Statistics & Data Analysis* **71**, 128–137.

Taylor, J. W. (2000), 'A quantile regression neural network approach to estimating the conditional density of multiperiod returns', *Journal of Forecasting* **19**(4), 299–311.

Vaysse, K. and Lagacherie, P. (2017), 'Using quantile regression forest to estimate uncertainty of digital soil mapping products', *Geoderma* **291**, 55–64.

Wang, L., Wu, Y. and Li, R. (2012), 'Quantile regression for analyzing heterogeneity in ultra-high dimension', *Journal of the American Statistical Association* **107**(497), 214–222.

Wang, S., Yao, W. and Hunter, D. (2012), 'Mixture of linear regression models with unknown error density. url: http://www-personal. ksu. edu/˜ wxyao/material/submitted/mixlinnonerr. pdf'.

Wedel, M. and Kamakura, W. A. (2000), Mixture regression models, *in* 'Market segmentation', Springer, pp. 101–124.

Wei, Y. (2012), 'Robust mixture regression models using t-distribution'.

Wu, C.-F. J. (1986), 'Jackknife, bootstrap and other resampling methods in regression analysis', *the Annals of Statistics* **14**(4), 1261–1295.

Wu, Q. and Sampson, A. R. (2009), 'Mixture modeling with applications in schizophrenia research', *Computational statistics & data analysis* **53**(7), 2563–2572.

Wu, Q. and Yao, W. (2016), 'Mixtures of quantile regressions', *Computational Statistics & Data Analysis* **93**, 162–176.

Yang, S. (1999), 'Censored median regression using weighted empirical survival and hazard functions', *Journal of the American Statistical Association* **94**(445), 137–145.

Yu, K., Lu, Z. and Stander, J. (2003), 'Quantile regression: applications and current research areas', *Journal of the Royal Statistical Society: Series D (The Statistician)* **52**(3), 331–350.

Zhang, W., Quan, H. and Srinivasan, D. (2018), 'An improved quantile regression neural network for probabilistic load forecasting', *IEEE Transactions on Smart Grid* **10**(4), 4425–4434.

Zheng, S. (2011), 'Gradient descent algorithms for quantile regression with smooth approximation', *International Journal of Machine Learning and Cybernetics* **2**(3), 191–207.