Journal of Statistical Modelling: Theory and Applications
Vol. 6, No. 1, 2025, pp. 133-150
Yazd University Press 2025

JSMTA

Research Paper

Robust mixture of experts modeling using symmetric α-stable distributions

Shaho Zarei*
Department of Statistics, Faculty of Science, University of Kurdistan, Sanandaj, Iran

Received: April 29, 2025/ Revised: August 31, 2025/ Accepted: December 23, 2025

Abstract: The mixture of experts framework is widely utilized in statistics and machine learning to address data heterogeneity in tasks such as regression, classification, and clustering. In clustering continuous data, the mixture of experts typically employs experts that follow a Gaussian distribution. However, outliers can adversely affect clustering outcomes. To address this issue, various methods have been proposed in the literature. In this paper, we introduce a novel approach that models the experts using the symmetric α-stable distribution. This flexible distribution effectively accommodates different types of outliers (especially extreme outliers) and skewness, while also encompassing Gaussian experts as a special case when $\alpha = 2$. The maximum likelihood estimates of the model parameters (excluding α) are obtained using an expectation-maximization approach, while α is estimated using Monte Carlo integration and interpolation. The effectiveness of this approach is demonstrated through analyses of both real and simulated data.

Keywords: Mixture of experts; Model-based clustering; Robust modeling; Stable distribution; Varying mixing proportions.
Mathematics Subject Classification (2010): 60E07, 62H30.

1 Introduction

The mixture of experts (MoE) model, introduced by Jacobs et al. (1991), has garnered significant attention in both statistics and machine learning. This model is characterized by a fully conditional mixture framework, where both the mixing proportions-known as gating functions-and the component densities-referred to as experts-are conditioned on specific input covariates. The MoE has been extensively studied in both

*Corresponding author: sh.zarei@uok.ac.ir

its simple and hierarchical forms, as discussed in Jordan and Jacobs (1994) and detailed in Section 5.12 of McLachlan and Peel (2000). Applications of MoE span various domains, including regression, model-based clustering, and discriminant analysis. A comprehensive review of MoE models can be found in the work of Yuksel et al. (2012).

In the context of continuous data, specifically within non-linear regression and model-based clustering, MoE typically employs normal experts, known as the normal mixture of experts (NMoE). However, it is well-documented that the normal distribution is sensitive to outliers, rendering NMoE unsuitable for datasets that exhibit noise. Additionally, when dealing with datasets containing groups of observations characterized by heavy tails, the use of normal experts may be inappropriate and can adversely affect the overall fit of the MoE model.

There are two categories of atypical observations: mild and gross (Ritter, 2014). Mild outliers are points that deviate from the distribution within a cluster, but they would fit well if the distribution inside the cluster as a whole had heavy(er) tails or some (more) skewness. Gross (extreme) outliers, on the other hand, are points that are far from any of the elements (Farcomeni and Punzo, 2020). Chamroukhi (2016) used the t-distribution for modeling mild outlier data in the mixture of experts model. His model, which we denote as TMoE, performs better than the NMoE model in the presence of outlier data.

In this paper, we propose an enhanced and robust MoE model that addresses these limitations by incorporating symmetric α-stable (SαS) distributions as experts. The adoption of SαSs allows for better handling of heavy-tailed and atypical (particularly gross) data, thereby improving the model's robustness and applicability in real-world scenarios. Our approach aims to extend the capabilities of traditional MoE frameworks while maintaining interpretability and computational efficiency.

The rest of the paper is organized as follows. The α-stable distribution is introduced in Section 2. The symmetric α-stable mixture of experts (SαSMoE) and an EM-type algorithm to obtain maximum likelihood estimates of the model parameters are outlined in Section 3. Some simulation studies, described in Section 4, are designed to compare the SαSMoE to some existing methods, and they demonstrate the effectiveness of the proposed model. In Section 5, some applications to real data are illustrated. The paper concludes with the discussion in Section 6.

2 The α-stable distribution

Stable (or α-stable) distributions are a diverse family of probability distributions known for their ability to exhibit skewness and heavy tails. This class of distributions possesses many interesting mathematical properties. Initially introduced by Paul Lévy (Lévy, 1925) in his study of sums of independent and identically distributed random variables, these distributions received little practical attention until Benoît Mandelbrot employed them in (Mandelbrot, 1961). In that work, he also proposed a simple algorithm for estimating their parameters. Mandelbrot labeled these distributions as "stable Paretian distributions," with a particular emphasis on those exhibiting maximal skewness in the positive direction, specifically for $1 < \alpha < 2$, which he referred to as "Pareto-Lévy distributions." He considered these distributions to provide more accurate representations of stock and commodity prices than normal distributions (Mandelbrot, 1963b).

The univariate α-stable distribution is characterized by four parameters: the index of stability (shape) $\alpha \in (0, 2]$, skewness $\eta \in [-1, 1]$, scale $\gamma > 0$, and location $\mu \in \mathbb{R}$. The notation $Y \sim S(\alpha, \eta, \gamma, \mu)$ is commonly used to denote that the random variable Y follows a stable distribution with the aforementioned parameters, and we denote the density function of Y by $S(y; \alpha, \eta, \gamma, \mu)$.

Stable distributions lack closed-form density functions, necessitating numerical methods and characteristic functions for property analysis. Various parameterizations exist for stable distributions; however, they converge when using the SαS distribution. We employ one such parameterization, where the characteristic function of a stable random variable $S(\alpha, \eta, \gamma, \mu)$ is defined as follows

$$\phi(w) = \begin{cases} \exp(-|w\gamma|^{\alpha}[1 - i\text{sign}(w)\eta \tan(\frac{\pi\alpha}{2})] + i\mu w) & \alpha \neq 1 \\ \exp(-|w\gamma|[1 + i\frac{2}{\pi}\text{sign}(w)\eta \log(|w|)] + i\mu w) & \alpha = 1, \end{cases} \tag{1}$$

where sign$(\cdot)$ is the sign function, and $i = \sqrt{-1}$. Although, in general, there is no analytical form for the probability density function (PDF) of the α-stable distribution, the PDF of the α-stable distribution is obtained by taking the inverse Fourier transform of the characteristic function, which can be calculated by evaluating the following integral (Salas-Gonzalez et al., 2009):

$$S(y; \alpha, \eta, \gamma, \mu) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \phi(w) \exp(iwy) dw.$$

As important special cases, positive stable distributions have $\eta = 1$ and $\alpha < 1$, while symmetric α-stable distributions centered around μ have $\eta = 0$. As an example, a box and density function plots of a random variable $S(1.25, 0, 1, 0)$ is shown in Figure 1, which indicates outlying and heavy-tailed data in this distribution. For more information on stable distributions, one can refer to Nolan (2009) and Samorodnitsky and Taqqu (1994).

The SαS distribution generalizes the normal distribution, with tail weight adjusted by the parameter α. This makes SαS suitable for modeling normal data and accommodating various outlier types, which are crucial in robust statistics. The α-stable distribution is applied in finance for asset returns and volatility modeling, as well as in signal processing and image analysis for non-Gaussian noise. However, it poses analytical challenges and often requires numerical methods for estimation and inference.

The SαS distribution is the most significant subclass of α-stable distributions. From (1), it can be observed that the Gaussian distribution with mean μ and variance γ^2 is represented as $S(2, 0, \frac{\gamma}{\sqrt{2}}, \mu)$. The S$\alpha$S distributions are obtained by multiplying a Gaussian distribution by the square root of a positive α-stable distribution. This relationship can be formally defined as follows.

Definition 2.1. *(**Scale mixtures of normals of SαS**) Suppose U is a zero-mean Gaussian random variable with variance σ^2, and let $P \sim S\left(\frac{\alpha}{2}, 1, \left(\cos\left(\frac{\pi\alpha}{4}\right)\right)^{\frac{2}{\alpha}}, 0\right)$ be a positive stable random variable, independent of U. Then, the random variable $Y = \mu + \sqrt{P}U$ follows the distribution $S(\alpha, 0, \frac{\sigma}{\sqrt{2}}, \mu)$, where μ is a constant.*

The variance of the α-stable distribution diverges to infinity when $\alpha < 2$ (Nolan, 2009). This property allows models based on the α-stable distribution to exhibit increased robustness in the presence of gross outliers in the data. Moreover, the SαS

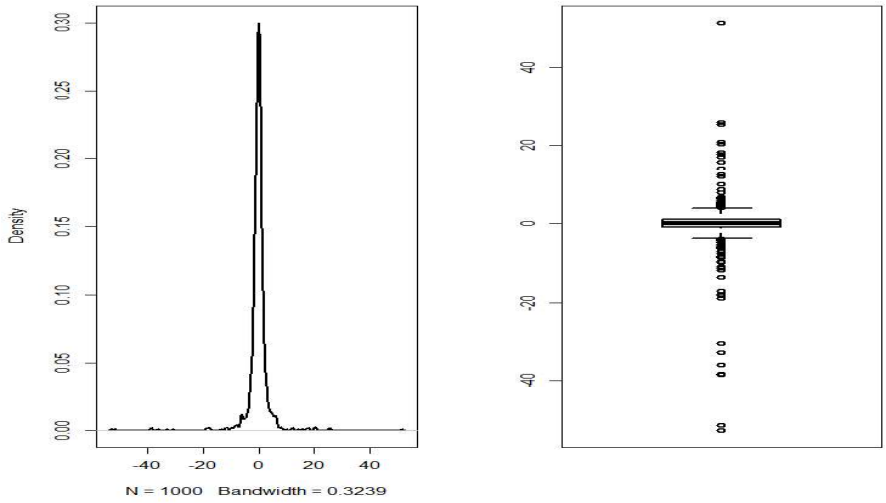

Figure 1: Density plot and box plot of S(1.25,0,1,0).

distribution satisfies the generalized central limit theorem, which states that the only possible non-trivial limit of normalized sums of independent identically distributed random variables is stable. Specifically, a normalized sum of independent and identically distributed random variables converges in distribution to an α-stable distribution (with SαS being a special case of this family). This feature is particularly important in various applications, such as financial modeling (Kring et al., 2009; Zarei et al., 2019).

According to Definition 2.1, the SαS distribution possesses the scale mixtures of normals (SMiN) property. SMiN indicates a symmetric stable distribution in a conditionally Gaussian form. This characteristic enables the direct application of standard procedures based on the Gaussian distribution in statistical inference involving models that include SαS terms. The desirable properties of the SαS distribution, as stated, are our motivation for using this distribution in the mixture of experts model.

3 Symmetric α-stable mixture of experts

MoE is basically an extension of finite mixture regression models (Goldfeld, and Quandt, 1973). These models have been widely utilized across various fields, including business, marketing, and social sciences, to investigate the relationships among data from numerous unidentified latent homogeneous groups. For additional details and references on this topic, please consult Yao et al. (2014) and Bai et al. (2012).

Assume that Z is a latent class variable. Given $Z = g$, the relationship between the response y and the $(p+1)$-dimensional predictor $\boldsymbol{x}$ (where $\boldsymbol{x}$ comprises both predictors and the constant 1) is modeled as follows

$$y = \mu(\boldsymbol{x}; \boldsymbol{\beta}_g) + \epsilon_g, \quad g = 1, \ldots, G,$$

where G represents the number of components in mixture models, and $\epsilon_g \sim N(0, \sigma_g^2)$,

which indicates a normal distribution with mean 0 and variance σ_g^2. Additionally, $\boldsymbol{\beta}_g = (\beta_{g0}, \beta_{g1}, \ldots, \beta_{gp})^T$. Let $P(Z = g) = \pi_g$ for $g = 1, \ldots, G$, and assume that Z is independent of $\boldsymbol{x}$. Then, the conditional density of Y given $\boldsymbol{x}$, without observing Z, is expressed as

$$f(y \mid \boldsymbol{x}, \boldsymbol{\theta}) = \sum_{g=1}^{G} \pi_g \phi\left(y; \mu(\boldsymbol{x}; \boldsymbol{\beta}_g), \sigma_g^2\right),$$

where $\phi(\cdot)$ is the density function of the Gaussian distribution with mean $\mu(\boldsymbol{x}; \boldsymbol{\beta}_g)$ and variance σ_g^2, and $\boldsymbol{\theta} = \left(\pi_1, \boldsymbol{\beta}_1, \sigma_1^2, \ldots, \pi_G, \boldsymbol{\beta}_G, \sigma_G^2\right)^T$.

In the MoE framework, the mixing proportions are modeled as a function of certain covariates $\mathbf{r}$ (or formally, a concomitant variable). These are typically modeled using a logistic or softmax function, which may be the same as $\mathbf{x}$ (Chamroukhi, 2016).

Motivated by the considerations in Section 2, we aim to accommodate data with atypical observations by considering the expert distributions as $S(\alpha, \eta = 0, \gamma, \mu)$. The proposed G-component SαSMoE is defined as follows

$$f(y \mid \mathbf{r}, \boldsymbol{x}; \boldsymbol{\Psi}) = \sum_{g=1}^{G} \pi(\boldsymbol{r}; \boldsymbol{\delta}_g) S\left(y; \alpha_g, \eta_g = 0, \gamma_g, \mu(\boldsymbol{x}; \boldsymbol{\beta}_g)\right), \tag{2}$$

where $\boldsymbol{\Psi} = \left(\boldsymbol{\delta}_1^T, \ldots, \boldsymbol{\delta}_{G-1}^T, \boldsymbol{\Psi}_1^T, \ldots, \boldsymbol{\Psi}_G^T\right)^T$, and $\boldsymbol{\Psi}_g = \left(\alpha_g, \gamma_g, \boldsymbol{\beta}_g^T\right)^T$ is the parameter vector for the gth expert component, which follows an symmetric α-stable distribution. The location parameter is defined as $\mu(\mathbf{x}, \boldsymbol{\beta}_g) = \mathbf{x}^T \boldsymbol{\beta}_g = \boldsymbol{\beta}_g^T \mathbf{x}$, where $\boldsymbol{\beta}_g \in \mathbb{R}^p$.

Similar to the approach outlined in Chamroukhi (2016), we assume that the mixing proportions are given by

$$\pi(\boldsymbol{r}; \boldsymbol{\delta}_g) = \mathbb{P}(Z = g \mid \boldsymbol{r}; \boldsymbol{\delta}_g) = \frac{\exp\left(\boldsymbol{\delta}_g^T \boldsymbol{r}\right)}{\sum_{\ell=1}^{G} \exp\left(\boldsymbol{\delta}_\ell^T \boldsymbol{r}\right)}, \tag{3}$$

where $\mathbf{r} \in \mathbb{R}^q$ is a covariate vector, $\boldsymbol{\delta}_g$ is the q-dimensional coefficient vector associated with $\mathbf{r}$, and $\boldsymbol{\delta} = \left(\boldsymbol{\delta}_1^T, \ldots, \boldsymbol{\delta}_{G-1}^T\right)^T$ is the parameter vector of the gating network. Notably, $\boldsymbol{\delta}_G$ is set to the zero vector to ensure that $\sum_{g=1}^{G} \pi_g(\mathbf{r}; \boldsymbol{\delta}_g) = 1$. Thus, the S$\alpha$SMoE model constitutes a fully conditional mixture model where both the mixing proportions (the gating functions) and the component densities (the experts) are conditional on predictors (respectively denoted here by $\boldsymbol{r}$ and $\boldsymbol{x}$).

3.1 Parameter estimation in SαSMoE

To estimate the SαSMoE parameters, we extend the methodology explained by Zarei and Mohammdpour (2020) into a new EM algorithm based on different sampling schemes. The core idea behind the expectation-maximization (EM) algorithm is to find a latent (hidden) variable whose probability density function depends on the parameter of interest (in this case, α) such that maximizing the latent variable's distribution is more tractable than maximizing the main variable's distribution (Roche, 2011). To hit this end, let y_i, $i = 1, \ldots, n$ be a sample from population Y, where n is the sample size.

According to Definition 2.1, we can express the distribution of SαS conditionally as a Gaussian distribution. Consequently, statistical inference based on Gaussian distribution is applicable. However, in this case we have to deal with the positive α-stable distribution. As previously mentioned, there is no closed-form expression for this distribution. Therefore, we treat this random variable as a latent random variable.

Let $y_1, \ldots, y_n, p_1, \ldots, p_n$ and $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ be the complete data corresponding to (2) where y_i and $p_i, i = 1, \ldots, n$, are observed and missing data, respectively. Furthermore, $\boldsymbol{z}_1, \ldots, \boldsymbol{z}_n$ are the G-dimensional component labels in which $z_{ig} = 1$ if ith observation comes from gth component ($z_{ig} = 0$, otherwise). According to Definition 2.1, Y_i for $i = 1, \ldots, n$ is related to a positive stable random variable and a Gaussian random variable as

$$Y_i \stackrel{d}{=} \mu(\boldsymbol{x}_i, \boldsymbol{\beta}) + \sqrt{P_i} U_i,$$

where P_i is a positive α-stable random variable with the tail index α ($\alpha < 1$), U_i is a zero-mean Gaussian random variable with variance σ^2 and $\mu(\boldsymbol{x}_i, \boldsymbol{\beta})$ is the location parameter for Y_i and $\stackrel{d}{=}$ denotes equality in distribution. Thus, we have

$$Y_i \mid P_i = p_i, Z_{ig} = 1 \sim N\left(\mu(\boldsymbol{x}_i, \boldsymbol{\beta}_g), p_i \sigma_g^2\right), \tag{4}$$

$$P_i \mid Z_{ig} = 1 \sim S\left(\frac{\alpha_g}{2}, 1, \left(\cos\left(\frac{\pi\alpha_g}{4}\right)\right)^{\frac{2}{\alpha_g}}, 0\right), \tag{5}$$

for $g = 1, \ldots, G$ and $i = 1, \ldots, n$. Because of the conditional structure of the complete data distributions (4) and (5), the complete data log-likelihood function is

$$\begin{aligned} \log L_c(\boldsymbol{\Psi}) &= \log \prod_{i=1}^{n} f(y_i, p_i, \boldsymbol{z}_i; \boldsymbol{\Psi}) = \sum_{i=1}^{n} \sum_{g=1}^{G} z_{ig} \Big[\log(\mathbb{P}(Z_i = g \mid \boldsymbol{r}_i; \boldsymbol{\delta}_g)) \\ &\quad + \log(f_{P_i|Z_{ig}}(p_i \mid Z_{ig} = 1)) + \log(f_{Y_i|P_i,Z_{ig}}(y_i \mid p_i, Z_{ig} = 1, \boldsymbol{x}_i)) \Big]. \end{aligned}$$

Thus, from (4) and (5), the complete-data log-likelihood of $\boldsymbol{\psi}$ is given by

$$\begin{aligned} l_c(\boldsymbol{\psi}) &= C + \sum_{g=1}^{G} \sum_{i=1}^{n} z_{ig} \log\left(\pi(\boldsymbol{r}_i; \boldsymbol{\delta}_g)\right) + \sum_{g=1}^{G} \sum_{i=1}^{n} z_{ig} \log\left(f_P\left(p_i \mid \alpha_g\right)\right) \\ &\quad - \frac{1}{2} \sum_{g=1}^{G} \sum_{i=1}^{n} z_{ig} \log\left(\sigma_g^2\right) - \frac{1}{2} \sum_{g=1}^{G} \sum_{i=1}^{n} z_{ig} \left(y_i - \mu(\boldsymbol{x}_i, \boldsymbol{\beta}_g)\right)^2 \left(p_i \sigma_g^2\right)^{-1}, \end{aligned}$$

where C is a constant and free from $\alpha_g, \sigma_g^2, \boldsymbol{\beta}_g$ and $\boldsymbol{\delta}_g$, and for $g = 1, \ldots, G$, $\sigma_g^2 = 2\gamma_g^2$. Furthermore, $f_P(.)$ is the probability density function of positive α-stable random variable P.

3.2 E-step

The E-step in the $(t+1)$th iteration requires calculating

$$Q\left(\boldsymbol{\psi} \mid \boldsymbol{\psi}^{(t)}\right) = E_{\boldsymbol{\psi}^{(t)}}\left(l_c(\boldsymbol{\psi}) \mid y_1, \boldsymbol{x}_1, \boldsymbol{r}_1, \ldots, y_n, \boldsymbol{x}_n, \boldsymbol{r}_n\right)$$

$$
\begin{aligned}
&= C + \sum_{g=1}^{G}\sum_{i=1}^{n} e_{zig}^{(t)} \log\left(\pi(\boldsymbol{r}_i;\boldsymbol{\delta}_g)\right) \\
&\quad + \sum_{g=1}^{G}\sum_{i=1}^{n} e_{zig}^{(t)} E_{\boldsymbol{\psi}^{(t)}}\left(\log\left(f_P\left(p_i \mid \alpha_g\right)\right) \mid y_i, \boldsymbol{x}_i, \boldsymbol{r}_i\right) \\
&\quad - \frac{1}{2}\sum_{g=1}^{G}\sum_{i=1}^{n} e_{zig}^{(t)} \log(\sigma_g^2) - \frac{1}{2}\sum_{g=1}^{G}\sum_{i=1}^{n} e_{zig}^{(t)} e_{pig}^{(t)} \left(y_i - \mu(\boldsymbol{x}_i, \boldsymbol{\beta}_g)\right)^2 \sigma_g^{-2}. \quad (6)
\end{aligned}
$$

To this end and since $\boldsymbol{x}_i$ and $\boldsymbol{r}_i$ for $i = 1, \ldots, n$ are not random vectors, we should calculate $e_{zig}^{(t)} = E_{\boldsymbol{\psi}^{(t)}}\left(Z_{ig} \mid y_i\right)$ and $e_{pig}^{(t)} = E_{\boldsymbol{\psi}^{(t)}}\left(P_i^{-1} \mid y_i\right)$ for $g = 1, \ldots, G, i = 1, \ldots, n$ and $\boldsymbol{\psi}^{(t)} = \left(\pi(\boldsymbol{r}_i;\boldsymbol{\delta}_g)^{(t)}, \alpha_g^{(t)}, \sigma_g^{2(t)}, \mu_g^{(t)}\right)$. It is easily shown that

$$
e_{zig}^{(t)} = \frac{\pi(\boldsymbol{r}_i;\boldsymbol{\delta}_g^{(t)}) S(y_i; \alpha_g^{(t)}, 0, \gamma_g^{(t)}, \mu(\boldsymbol{x}_i, \boldsymbol{\beta}_g^{(t)}))}{\sum_{l=1}^{G} \pi(\boldsymbol{r}_i;\boldsymbol{\delta}_g^{(t)}) S(y_i; \alpha_g^{(t)}, 0, \gamma_g^{(t)}, \mu(\boldsymbol{x}_i, \boldsymbol{\beta}_g^{(t)}))}.
$$

Since there is no closed-form for stable densities, we calculated $e_{zig}^{(t)}$ and $e_{pig}^{(t)}$ numerically using functions in the `STABLE` package in R software available at `http://www.robustanalysis.com` and Monte Carlo integration (for computing $e_{pig}^{(t)}$ see Appendix).

3.3 M-step

In the M-step, we maximize the expected complete-data log-likelihood obtained from the E-step to update our parameter estimates. Each M-step of our algorithm, on the same iteration, has three parts.

3.3.1 First part: Updating gating weights

In the first part, we focus on updating the gating weights. Since there is no analytical solution for updating the gating network parameters, the update for the component weight parameters is obtained via a numerical optimization step. Similar to Chamroukhi (2016), this optimization is performed using the Iteratively Reweighted Least Squares (IRLS) algorithm. We aim to update the parameter vector $\boldsymbol{\delta}_g$, which appears in the following objective function

$$
Q(\boldsymbol{\delta}_g; \boldsymbol{\Psi}^{(t)}) = \sum_{g=1}^{G}\sum_{i=1}^{n} e_{zig}^{(t)} \log\left(\pi(\boldsymbol{r}_i;\boldsymbol{\delta}_g)\right).
$$

Alternatively, based on equation (3), the objective function can also be expressed as

$$
Q(\boldsymbol{\delta}_g; \boldsymbol{\Psi}^{(t)}) = \sum_{i=1}^{n}\sum_{g=1}^{G} e_{zig}^{(t)} \left[\boldsymbol{r}_i^T \boldsymbol{\delta}_g - \log\left\{\sum_{\ell=1}^{G} \exp\left(\boldsymbol{r}_i^T \boldsymbol{\delta}_\ell\right)\right\}\right].
$$

The IRLS algorithm is employed to maximize $Q(\boldsymbol{\delta}_g; \boldsymbol{\Psi}^{(t)})$ with respect to the vector $\boldsymbol{\delta}_g$ for each component $g = 1, \ldots, G-1$ where $\boldsymbol{\delta}_G = 0$. The IRLS algorithm is a Newton–Raphson method that iteratively updates the estimate of $\boldsymbol{\delta}_g$. Starting with an initial

vector $\boldsymbol{\delta}_g^{(0)}$, the update at the $(t+1)$-th iteration is given by

$$\boldsymbol{\delta}_g^{(t+1)} = \boldsymbol{\delta}_g^{(t)} - \left[\frac{\partial^2 Q(\boldsymbol{\delta}_g; \boldsymbol{\Psi}^{(t)})}{\partial \boldsymbol{\delta}_g \partial \boldsymbol{\delta}_g^T}\Big|_{\boldsymbol{\delta}_g=\boldsymbol{\delta}_g^{(t)}}\right]^{-1} \frac{\partial Q(\boldsymbol{\delta}_g; \boldsymbol{\Psi}^{(t)})}{\partial \boldsymbol{\delta}_g}\Big|_{\boldsymbol{\delta}_g=\boldsymbol{\delta}_g^{(t)}},$$

where $\frac{\partial^2 Q(\boldsymbol{\delta}_g;\boldsymbol{\Psi}^{(t)})}{\partial \boldsymbol{\delta}_g \partial \boldsymbol{\delta}_g^T}$ and $\frac{\partial Q(\boldsymbol{\delta}_g;\boldsymbol{\Psi}^{(t)})}{\partial \boldsymbol{\delta}_g}$ are the Hessian matrix and the gradient vector of $Q(\boldsymbol{\delta}_g; \boldsymbol{\Psi}^{(t)})$ with respect to $\boldsymbol{\delta}_g$, respectively. The gradient vector of $Q(\boldsymbol{\delta}_g; \boldsymbol{\Psi}^{(t)})$ is given by

$$\frac{\partial Q(\boldsymbol{\delta}_g; \boldsymbol{\Psi}^{(t)})}{\partial \boldsymbol{\delta}_g} = \sum_{i=1}^{n} e_{zig}^{(t)} \left[\boldsymbol{r}_i - \frac{\exp(\boldsymbol{r}_i^T \boldsymbol{\delta}_g)}{\sum_{\ell=1}^{G} \exp(\boldsymbol{r}_i^T \boldsymbol{\delta}_\ell)} \boldsymbol{r}_i\right].$$

Furthermore, the Hessian matrix of $Q(\boldsymbol{\delta}_g; \boldsymbol{\Psi}^{(t)})$ is given by

$$\frac{\partial^2 Q(\boldsymbol{\delta}_g; \boldsymbol{\Psi}^{(t)})}{\partial \boldsymbol{\delta}_g \partial \boldsymbol{\delta}_g^T} = -\sum_{i=1}^{n} e_{zig}^{(t)} \left[\frac{\exp(\boldsymbol{r}_i^T \boldsymbol{\delta}_g)}{\sum_{\ell=1}^{G} \exp(\boldsymbol{r}_i^T \boldsymbol{\delta}_\ell)} \boldsymbol{r}_i \boldsymbol{r}_i^T - \frac{(\exp(\boldsymbol{r}_i^T \boldsymbol{\delta}_g))^2 \boldsymbol{r}_i \boldsymbol{r}_i^T}{\left(\sum_{\ell=1}^{G} \exp(\boldsymbol{r}_i^T \boldsymbol{\delta}_\ell)\right)^2}\right].$$

3.3.2 Second part: Estimation of $\boldsymbol{\beta}_g$ and σ_g^2

In the second part, with taking the derivative of $Q\left(\boldsymbol{\psi} \mid \boldsymbol{\psi}^{(t)}\right)$ with respect to $\boldsymbol{\beta}_g$ and σ_g^2, these parameters are updated for gth component in $(t+1)$th iteration as follows

$$\begin{aligned}
\boldsymbol{\beta}_g^{(t+1)} &= \frac{\sum_{i=1}^{n} e_{zig}^{(t)} e_{pig}^{(t)} y_i \boldsymbol{x}_i}{\sum_{i=1}^{n} e_{zig}^{(t)} e_{pig}^{(t)} \boldsymbol{x}_i \boldsymbol{x}_i^T}, \\
\sigma_g^{2(t+1)} &= \frac{\sum_{i=1}^{n} e_{zig}^{(t)} e_{pig}^{(t)} (y_i - \boldsymbol{\beta}_g^{(t+1)T} \boldsymbol{x}_i)^2}{\sum_{i=1}^{n} e_{zig}^{(t)}}.
\end{aligned}$$

3.3.3 Third part: Estimation of α_g

To estimate α_g for $g = 1, \cdots, G$, we consider (6) and aim to maximize the function

$$\ell_{\alpha_g}(\theta) = \sum_{i=1}^{n} e_{zig}^{(t)} E_{\boldsymbol{\psi}^{(t)}} \left(\log\left(f_P\left(p_i \mid \alpha_g\right)\right) \mid y_i, \boldsymbol{x}_i, \boldsymbol{r}_i\right), \tag{7}$$

with respect to α_g. Since α_g is a parameter of a random variable with positive stable distributions that has no analytical form of PDF, the stochastic EM (SEM; Celeux and Diebolt (1985); Roche (2011); Zarei and Mohammdpour (2020)) algorithm is typically used for updating α_g, for $g = 1, \ldots, G$. In SEM, the auxiliary function $\ell_{\alpha_g}(\theta)$ is approximated by the conditional distribution of the unobserved variable, given the observed variables (Roche, 2011). A random sample is then generated from this conditional distribution, and the parameter value that maximizes the marginal density function of the unobserved variable serves as an estimate for the parameter of interest. However, direct sampling from the conditional distribution is not feasible, and rejection sampling is often employed. Consequently, this method can be computationally expensive and

time-consuming. To address these challenges, we propose a simple and efficient method based on interpolation.

Similar to the calculations performed for the approximation of e_{pig} (see Appendix), we have

$$E\left(\log\left(f(p_i;\alpha_g)\right)|y_i,\boldsymbol{x}_i,\boldsymbol{r}_i\right)\approx\frac{\sum_{b=1}^{B}\log(f(p_{bi}^{mc};\alpha_g))p_{bi}^{mc(\frac{-1}{2})}\exp\left\{\frac{-(y_i-\mu(\boldsymbol{x}_i,\boldsymbol{\beta}_g))^2}{2p_{bi}^{mc}\sigma_g^2}\right\}}{\sum_{b=1}^{B}p_{bi}^{mc(\frac{-1}{2})}\exp\left\{\frac{-(y_i-\mu(\boldsymbol{x}_i,\boldsymbol{\beta}_g))^2}{2p_{bi}^{mc}\sigma_g^2}\right\}}.$$

Therefore, (7) can be approximated as

$$\ell_{\alpha_g}(\theta)\approx\sum_{i=1}^{n}e_{zig}^{(t)}\left[\frac{\sum_{b=1}^{B}\log(f(p_{bi}^{mc};\alpha_g))p_{bi}^{mc(\frac{-1}{2})}\exp\left\{\frac{-(y_i-\mu(\boldsymbol{x}_i,\boldsymbol{\beta}_g))^2}{2p_{bi}^{mc}\sigma_g^2}\right\}}{\sum_{b=1}^{B}p_{bi}^{mc(\frac{-1}{2})}\exp\left\{\frac{-(y_i-\mu(\boldsymbol{x}_i,\boldsymbol{\beta}_g))^2}{2p_{bi}^{mc}\sigma_g^2}\right\}}\right].$$

The proposed method operates as follows: In the t-th iteration, we first generate a sample of size B from the positive α-stable distribution for each i, denoted as $p_{1i}^{mc},\cdots,p_{Bi}^{mc}$. Then, we initialize α_g to a value, for example, 0.5. By substituting the updated values of the other unknown parameters, we calculate the value of the approximated log-likelihood for $\alpha_g=0.5$. Next, we incrementally increase the value of α_g (e.g., by 0.01) and recalculate the sum. The value of α_g that yields the largest value of the sum is taken as the estimated value for α_g. In other words, after generating the Monte Carlo samples, we employ a trial-and-error approximation method to estimate α_g, where smaller step sizes yield more precise results. It is worth noting that generating random numbers from the positive alpha-stable distribution, calculating the values of the density function at specific points, and performing other computations related to α-stable distributions can be efficiently done in the R software using the `STABLE` package.

4 Model evaluation of $S\alpha SMoE$

In this section, to evaluate the performance of the proposed robust model represented by SαSMoE and to compare with NMoE and TMoE models, we do some simulations.

4.1 Determining the number of mixture components

In practice, since true label values are unknown, well-known indicators such as the Bayesian information criterion (BIC) (Schwarz, 1978) can be used to select the appropriate model and the number of components, which is defined as

$$BIC=2\log(L(\hat{\boldsymbol{\theta}}))-m\log(n),$$

where $\hat{\boldsymbol{\theta}}$ is the maximum likelihood estimate of $\boldsymbol{\theta}$, $\log(L(\hat{\boldsymbol{\theta}}))$ is the logarithm of the maximum value of the observed likelihood function, and m is the total number of free parameters in the model. In the proposed model, $m=q(G-1)+G+G+pG$, where the first term is related to the mixture weights $\boldsymbol{\delta}_g$, and the second term to the parameters of $Y|\boldsymbol{x}$, i.e., α_g,γ_g and $\boldsymbol{\beta}_g$, for $g=1,\ldots,G$.

4.2 Stopping rule and initialization

Since the data may contain outliers, we use the results from the k-median clustering (Jain and Dubes, 1988) method with $k = G$ to determine initial values. This means that after partitioning the data into G groups based on k-median, we estimate the values of α_g, γ_g, and $\boldsymbol{\beta}_g$ for $g = 1, \ldots, G$ using the $STABLE$ package, which will be used as initial values. In addition, we use results obtained from clustering with NMoE for estimating initial values of $\boldsymbol{\delta}_g$.

As a general rule, the algorithm stops when the relative change in the log-likelihood of the observed data, i.e.,

$$\frac{\log L(\psi^{(t+1)}) - \log L(\psi^{(t)})}{|\log L(\psi^{(t)})|},$$

reaches a specified threshold (for example, $\epsilon = 10^{-4}$). We refer to this stage as the burning time. It should be noted that since the values of α_g are estimated through a stochastic EM algorithm, the algorithm may not converge uniformly and may exhibit slight oscillatory behavior. Therefore for a more robust estimation of the parameters, the algorithm was executed for an additional ten runs post-convergence. We then computed the mean value across these runs to serve as the final parameter estimate.

4.3 Identifiability of the SαSMoE model

The identifiability of mixture models is a known challenge. Our model, similar to many other mixture distributions, isn't inherently identifiable due to the permutational invariance of its components. This means we can reorder the component labels without altering the likelihood function. Jiang and Tanner (1999) established that ordered, initialized, and irreducible MoEs are identifiable. Therefore, to address the identifiability and the label switching problem, we impose identifiability constraints on our model's parameters. In our simulations, we tackled this by forcing the stability indices to be in ascending order. During each iteration, we assigned the parameter estimates with the smallest stability index to the first cluster, the second-smallest to the second cluster, and so on. In practice, if the estimated α values are nearly equal, the algorithm can be rerun. Following the strategy of Salas-Gonzalez et al. (2009), we then impose an increasing order on the location parameters to ensure proper identifiability of the components.

4.4 Simulation 1

Our generating setting is an extension of Experiment 1 in Chamroukhi (2016) and similar to Simulation 1 in Zarei (2024). Each simulated sample consisted of n observations with increasing values of the sample size n: 200 and 400. The simulated data are generated from a two component mixture of linear experts, that is $G = 2$ and $p = q = 1$. The covariate variables (x_i, r_i) are simulated such that $x_i = r_i = (1, x_i)^T$ where x_i is simulated uniformly over the interval $(-1, 1)$. For each generated sample, we fit SαSMoE. The estimated values are averaged over 100 trials.

The true value of parameters, average of estimated values for parameters across simulation iterations (as estimated value for parameter), and for each model are given

in Table 1. We also consider the empirical coverage probability (ECP) of the 95% confidence intervals for the estimated parameters. The ECP is defined as the proportion of simulated confidence intervals that contain the true parameter value. Furthermore, the empirical mean squared error (EMSE) is calculated. For a parameter such as τ, with jth estimate $\hat{\tau}^{(j)}$ across M simulations, for $j = 1, \ldots, M$, EMSE is defined as

$$\text{EMSE} = \frac{1}{M}\sum_{j=1}^{M}(\tau - \hat{\tau}^{(j)})^2,$$

where M is the number of simulation runs performed.

Table 1: Parameter estimates (EMSE) and the ECP index for sample sizes $n = 200$ and $n = 400$.

Component	Parameter	True value	Estimated value	ECP	Estimated value	ECP
			$n = 200$		$n = 400$	
Component 1	δ_{10}	0	-1.622 (0.1232)	0.928	1.052 (0.0581)	0.951
	δ_{11}	8	11.893 (10.8958)	0.931	9.912 (8.1406)	0.948
	β_{10}	-0.05	-0.201 (0.0681)	0.926	-0.134 (0.0335)	0.949
	β_{11}	2	2.274 (0.0751)	0.930	2.103 (0.0392)	0.952
	α_1	1.4	1.272 (0.0008)	0.929	1.288 (0.0004)	0.950
	γ_1	0.250	0.279 (0.0615)	0.925	0.273 (0.0296)	0.947
Component 2	β_{20}	0.05	0.549 (0.0253)	0.927	0.185 (0.0125)	0.953
	β_{21}	-2	-1.515 (0.0328)	0.932	-1.801 (0.0161)	0.949
	α_2	1.85	1.412 (0.0023)	0.928	1.537 (0.0011)	0.951
	γ_2	0.707	0.582 (0.0640)	0.930	0.613 (0.0317)	0.948

The analysis of Table 1 reveals that the estimation method performs well, with the accuracy and precision of the estimates generally improving as the sample size increases.

The estimated values are reasonably close to the true parameter values across most components. For example, for Component 1, the true value of β_{11} is 2, and the estimated values are 2.284 for $n = 200$ and 2.143 for $n = 400$. Similarly, for Component 2, the true value of β_{21} is -2, and the estimated values are -1.515 and -1.801 for $n = 200$ and $n = 400$, respectively. The estimation of parameters such as δ_{10} and δ_{11} also shows good agreement, with the estimated values for $n = 400$ being closer to the true values than for $n = 200$.

A key observation is the relationship between sample size and the quality of the estimates. As expected in statistical simulations, the EMSE values consistently decrease when the sample size increases from $n = 200$ to $n = 400$. For instance, the EMSE for δ_{10} decreases from 0.1232 to 0.0581. This trend is evident across all parameters, underscoring the improved precision of the model as more data are included.

The ECP values, which indicate the proportion of times the true parameter value falls within the confidence interval of the estimate, are consistently high, all above 0.925. This suggests that the confidence intervals generated by the estimation method are reliable and provide good coverage. Notably, the ECP values also show a slight increase from $n = 200$ to $n = 400$, moving closer to the nominal 0.95 level, which is a desirable outcome indicating the robustness of the method. As previously mentioned,

the $\boldsymbol{\delta}_G$ parameters in the last component (here component $G = 2$) are zero and therefore are not estimated and are not included in Table 1.

To evaluate the performance of the MoE models (i.e., NMoE, TMoE, and SαSMoE) in terms of clustering, we calculated the adjusted Rand index (ARI; Hubert and Arabie, 1985) for each simulation as a performance metric. We note that the expected value of ARI is 0, and that a value of 1 indicates perfect classification. The average ARIs for NMoE, TMoE, and SαSMoE were 0.417, 0.591, and 0.749 for $n = 200$, and 0.548, 0.653, and 0.825 for $n = 400$, respectively. These findings demonstrate that when the data components follow SαS distributions, the SαSMoE model performs noticeably better in clustering the data compared to NMoE and TMoE. Moreover, the ARI values for all three models increase with the larger sample size, suggesting better clustering performance with more data points. The superior clustering performance of SαSMoE likely stems from its ability to better capture the underlying data structure when the true distributions are SαS, leading to more accurate estimation of component-specific parameters and thus better separation of the clusters. It should be noted that all calculations pertaining to the NMoE and TMoE models have been conducted using the `meteorits` `R` package Chamroukhi et al. (2019).

4.5 Simulation 2

In this subsection, we compare three methods-NMoE, TMoE, and SαSMoE-and present a simulation study to demonstrate the effectiveness of the proposed method.

Similar to Yao et al. (2014) and Zarei (2024), we consider independently and identically distributed samples $\{(x_{1i}, y_i), i = 1, \ldots, n\}$ generated from the model:

$$Y = \begin{cases} 4 + X_1 + \epsilon_1 & \text{if } Z = 1 \\ -3 - X_1 + \epsilon_2 & \text{if } Z = 2, \end{cases}$$

where Z is a component indicator for Y, with $X_1 \sim N(0, 1)$, and ϵ_1 and ϵ_2 are model errors. Furthermore, for the second component, $\boldsymbol{\delta}_2 = (0, 7)^T$. We consider the following three cases for the error density of ϵ_1 and ϵ_2 with a sample size of $n = 300$:

Case I: $\epsilon_1 \sim N(0, 1)$ and $\epsilon_2 \sim N(0, 1)$ (standard normal distribution).

Case II: $\epsilon_1 \sim t_3$ and $\epsilon_2 \sim t_3$ (t-distribution with 3 degrees of freedom).

Case III: $\epsilon_1 \sim S(1.4, 0, 0.7, 0)$ and $\epsilon_2 \sim S(1.7, 0, 1.23, 0)$ (SαS distribution).

Case I represents a standard scenario with normally distributed errors. Case II introduces heavy-tailed errors, often leading to mild outliers due to the heavy tails of the t-distribution. Case III considers extreme outliers due to the heavy tails of the SαS distribution. In fact, in case III, since $\alpha < 2$, the variance of the distribution is infinite, and there will be gross outliers in the generated data.

For each model and in each iteration of the simulation, we calculate the ARI value. The averaged ARI values across the cases are as follows:

- **Case I:** SαSMoE: 0.851, NMoE: 0.842, TMoE: 0.841,
- **Case II:** SαSMoE: 0.753, NMoE: 0.595, TMoE: 0.764,
- **Case III:** SαSMoE: 0.740, NMoE: 0.515, TMoE: 0.613.

As expected, the presence of outliers, particularly in Case III, degrades the performance of all models. However, the SαSMoE model demonstrates robustness to outliers, especially in Case III, where it significantly outperforms the other models. Since the

normal distribution is a special case of the stable distribution, and by adjusting the α parameter of the stable distribution, it can effectively model both mild and gross outlier errors (in the data), the SαSMoE model exhibits good accuracy and competes favorably with the other models across all considered cases.

5 Real data analysis

The dataset analyzed pertains to the tone perception data originally presented by Cohen (1984). These data have been analyzed by Song et al. (2014) and Chamroukhi (2016). In the context of regression, Song et al. (2014) proposed a mixture of Laplace regressions. This model was later extended by Nguyen and McLachlan (2016) to a mixture of experts, which they named the Laplace mixture of linear experts (hereafter LMoE).

In the tone perception experiment, a pure fundamental tone was played for a trained musician, who then adjusted an electronically generated tone with overtones added at a specific stretching ratio (denoted as "stretch ratio" = 2). This ratio aligns with the harmonic structures typically found in traditional pitched instruments. The musician's task was to tune an adjustable tone to match the octave above the fundamental frequency, resulting in "tuned" measurements that reflect the ratio of the adjusted tone to the fundamental. The dataset comprises $n = 150$ pairs. Similar to Chamroukhi (2016) predictor $x_i = r_i$ (for $i = 1, \ldots, 150$) is the actual tone ratio and the response y_i is the perceived tone ratio.

The BIC values for $G = 2$ and $G = 3$ are 208.954 and 192.824, respectively, which indicates that the optimal number of components for our mixture of experts model is 2. These results are consistent with the methodologies employed by Nguyen and McLachlan (2016) and Chamroukhi (2016).

The estimated values for the stability indices, α_1 and α_2, are 1.377 and 1.097, respectively. The estimated common parameters for the MoE models applied to the tone perception dataset are summarized in Table 3.

Based on Table 3, the estimated parameter values across the different MoE models (NMoE, LMoE, TMoE, and SαSMoE) appear to be quite similar for several parameters, particularly $\hat{\beta}_{10}$ and $\hat{\beta}_{21}$. However, there are notable differences in the estimated scale parameters ($\hat{\sigma}_1$ and in particular $\hat{\sigma}_2$) and parameters ($\hat{\delta}_{10}$ and $\hat{\delta}_{11}$), suggesting that the choice of model has a significant impact on the estimation of these specific characteristics. The LMoE model, lacks estimates for the scale parameters, as indicated by the dashes. For more comparison, see Figure 2 which shows the data scatter plot with the estimated regression lines produced by the different MoE models.

To compare the parameter estimates obtained from different methods, we employed the bootstrap approach. Specifically, we generated 100 bootstrap samples by sampling with replacement from the original dataset. For each bootstrap sample, parameter estimates were computed using each method, and the standard deviation of these estimates across the 100 replications was calculated as a measure of precision. These bootstrap standard errors are reported in parentheses in Table 3. Based on the bootstrap standard errors, the SαSMoE method yields the most precise estimates overall. TMoE performs well in several cases but is generally outperformed by SαSMoE, while

NMoE and LMoE show higher variability. Bootstrap is a scientifically sound and widely accepted method for estimating standard errors in such settings.

Table 2: The values of the BIC index in clustering tone perception dataset with different mixture of experts methods. Bold number highlight the best performance.

	NMoE	LMoE	TMoE	SαSMoE
BIC	122.805	146.132	202.827	**208.954**

Table 3: Estimated common parameters for the MoE models applied to the tone perception dataset (bootstrap-based standard errors in parentheses).

Parameter	NMoE	LMoE	TMoE	SαSMoE
$\hat{\beta}_{10}$	1.913 (0.652)	1.935 (0.487)	1.927 (0.115)	1.906 (0.112)
$\hat{\beta}_{11}$	0.043 (0.197)	0.025 (0.168)	0.037 (0.105)	0.049 (0.205)
$\hat{\beta}_{20}$	-0.029 (0.956)	0.004 (0.833)	0.002 (1.213)	0.012 (1.178)
$\hat{\beta}_{21}$	0.995 (0.992)	0.997 (0.987)	0.999 (1.0310)	0.996 (0.074)
$\hat{\sigma}_1$	0.0471 (0.128)	-	0.0430 (0.178)	0.0673 (0.169)
$\hat{\sigma}_2$	0.1373 (0.186)	-	0.0025 (0.079)	0.0005 (0.007)
$\hat{\delta}_{10}$	-2.682 (2.191)	-0.421 (1.433)	-0.219 (1.086)	-2.717 (0.981)
$\hat{\delta}_{11}$	0.793 (1.089)	0.091 (0.923)	0.026 (0.543)	0.805 (0.556)

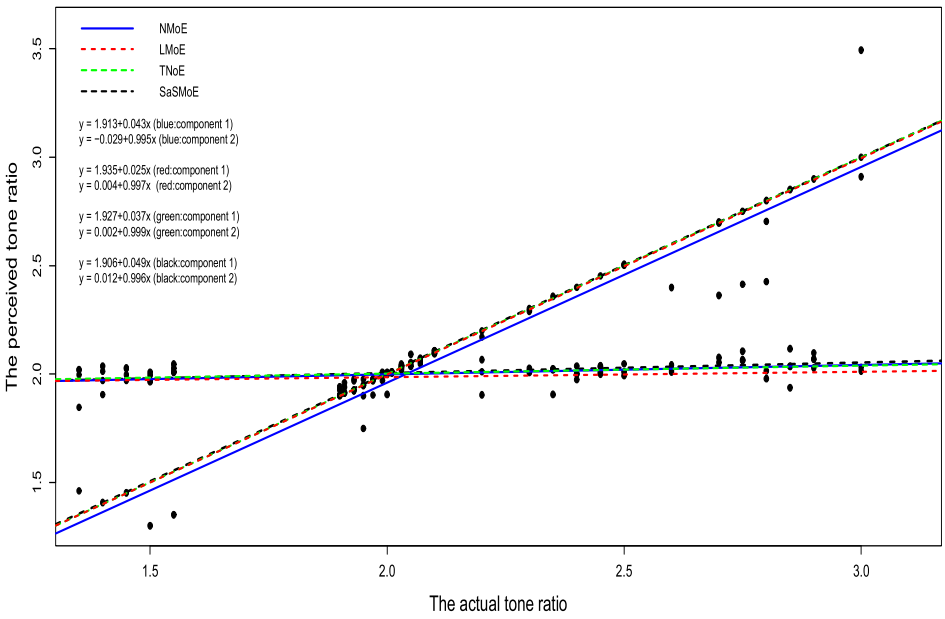

Figure 2: Scatter plot with fitted regression lines of NMoE, LMoE, TMoE, and SαSMoE, which approximately are same.

6 Discussion and conclusions

In this study, we have expanded the normal mixture of experts framework to incorporate the SαS mixture of experts model, which effectively addresses the presence of both mild and extreme outliers in the error terms. In this innovative model, the maximum likelihood estimates for the parameters (excluding α_g, $g = 1, \ldots, G$) are derived using a standard EM algorithm. The parameter α_g is estimated through Monte Carlo integral and interpolation. This methodology effectively mitigates the computational challenges associated with the EM algorithm in the context of SαSMoE.

The proposed model offers greater flexibility compared to existing alternatives; it encompasses NMoE models when $\alpha_g = 2$ and permits heavier tails in the response variable distributions as α_g deviates from 2 for $g = 1, \ldots, G$.

Both simulation studies and analyses of real dataset have validated the efficacy of our proposed method. The Sα*lpha*SMoE framework is particularly adept at clustering data characterized by normal distributions as well as those containing outliers or noise, rendering it a more realistic option than traditional NMoE models. Nonetheless, the parameter α_g, which influences the tail behavior of each mixture component, requires numerical methods for estimation, introducing a computational burden in terms of processing time for our algorithm. For example, the proposed SαSMoE algorithm requires more time than the classical TMoE due to Monte Carlo integration, α-stable sampling, and grid search. For the analysis of tone data, the average runtime was about 44.64 seconds compared to 0.55 seconds for TMoE, yet the method remains practical for moderate-scale applications. The additional cost reflects the robustness of the procedure in handling heavy-tailed and non-Gaussian noise structures.

Future research directions may involve developing algorithms that accommodate asymmetric outlier data and investigating novel estimation techniques for α_g. For this purpose, Bayesian methods can be employed.

Acknowledgements

I am very grateful to the reviewers for their constructive and insightful comments on this manuscript. These suggestions were instrumental in clarifying the content and strengthening the arguments.

References

Bai, X., Yao, W. and Boyer, J.E. (2012). Robust fitting of mixture regression models. *Computational Statistics & Data Analysis*, **56**(7):2347–2359.

Celeux, G. and Diebolt, J. (1985). The SEM algorithm: A probabilistic teacher algorithm derived from the EM algorithm for the mixture problem. *Computational Statistics*, **2**(1):73–82.

Chamroukhi, F. (2016). Robust mixture of experts modeling using the t distribution. *Neural Networks*, **79**:20–36.

Chamroukhi, F., Lecocq, F. and Bartcus, M. (2019). Meteorits: Mixtures-of-experts modeling for complex and non-normal. *R package version 0.1.1. https://github.com/fchamroukhi/MEteorits.*

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

Farcomeni, A. and Punzo, A. (2020). Robust model-based clustering with mild and gross outliers. *Test*, **29**(4):989–1007.

Goldfeld, S.M. and Quandt, R.E. (1973). A Markov model for switching regression. *Journal of Econometrics*, **3**(1):3–15.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, **2**(1):193–218.

Jacobs, R.A., Jordan, M.I., Nowlan, S.J. and Hinton, G.E. (1991). Adaptive mixtures of local experts. *Neural Computation*, **3**(1):79–87.

Jain, A.K. and Dubes, R.C. (1988). *Algorithms for Clustering Data*. New York: Prentice and Hall.

Jiang, W. and Tanner, M.A. (1999). On the identifiability of mixtures-of-experts. *Neural Networks*, **12**(9):197–220.

Jordan, M.I. and Jacobs, R.A. (1994). Hierarchical mixtures of experts and the EM algorithm. *Computational Statistics*, **6**(2):181–214.

Kong, A., McCullagh, P., Meng, X.L., Nicolae, D. and Tan, Z. (2009). A theory of statistical models for Monte Carlo integration. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, **65**(3):585–604.

Kring, S., Rachev, S.T., Höchstötter, M. and Fabozzi, F.J. (2009). Estimation of α-stable sub-Gaussian distributions for asset returns. *Risk Assessment*, **12**(2):111–152.

Lévy, P. (1925). *Calcul des probabilites*. Paris: Gauthier-Villars.

Mandelbrot, B. (1961). Stable paretian random functions and the multiplicative variation of income. *Econometrica*, **29**(4), 517– 543.

Mandelbrot, B. (1963b). New methods in statistical economics. *The Journal of Political Economy*, **71**(5):421–440.

McLachlan, G.J. and Peel, D. (2000). *Finite Mixture Models.* New York: John Wiley & Sons.

Nguyen, H.D. and McLachlan, G.J. (2016). Laplace mixture of linear experts. *Computational Statistics & Data Analysis*, **93**:177–191.

Nolan, J.P. (2009). *Stable Distributions: Models for Heavy Tailed Data*. Boston:Birkhauser.

Ritter, G. (2014). *Robust Cluster Analysis and Variable Selection*. Vol. 137, Chapman & Hall/CRC Monographs on Statistics & Applied Probability.

Roche, A. (2011). EM algorithm and variants: An informal tutorial. *arXiv preprint arXiv:1105.1476*.

Salas-Gonzalez, D., Kuruoglu, E.E. and Ruiz, D.P. (2009). Finite mixture of α-stable distributions. *Digital Signal Processing*, **19**(2):250–264.

Samorodnitsky, G. and Taqqu, M.S. (1994). *Stable Non-Gaussian Random Processes*. New York: Chapman and Hall.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, **6**(2):461–464.

Song, W., Yao, W. and Xing, Y. (2014). Robust mixture regression model fitting by Laplace distribution. *Computational Statistics & Data Analysis*, **71**(10):128–137.

Yao, W., Wei, Y. and Yu, C. (2014). Robust mixture regression using the t-distribution. *Computational Statistics & Data Analysis*, **71**(1):116–127.

Yuksel, S.E., Wilson, J.N. and Gader, P.D. (2012). Twenty years of mixture of experts. *IEEE Transactions on Neural Networks and Learning Systems*, **28**(8):1177–1193.

Zarei, S. (2024). Robust mixture of regression models using the symmetric α-stable distribution. *Communications in Statistics-Simulation and Computation*, **54**(10):3879–3897

Zarei, S. and Mohammdpour, A. (2020). Pseudo-stochastic EM for sub-Gaussian α-stable mixture models. *Digital Signal Processing*, **99**:102671.

Zarei, S., Mohammadpour, A., Ingrassia, S. and Punzo, A. (2019). On the use of the sub-Gaussian α-stable distribution in the cluster-weighted model. *Iranian Journal of Science and Technology, Transactions A: Science*, **43**(3):1059–1069.

Appendix

Suppose $Y \sim S(\alpha, 0, \gamma, \mu(\boldsymbol{x}, \boldsymbol{\beta}))$. Therefore, $Y \stackrel{d}{=} \mu(\boldsymbol{x}, \boldsymbol{\beta}) + \sqrt{P}U$, where U is a univariate zero-mean normal random variable with variance $\sigma^2 = 2\gamma^2$ and $P \sim S(\frac{\alpha}{2}, 1, (\cos(\frac{\pi\alpha}{4}))^{\frac{2}{\alpha}}, 0)$ is a positive stable random variable, where P and U are independent. To compute $E_1 = E\Big(P^{-1}|y; \alpha, 0, \gamma, \mu(\boldsymbol{x}, \boldsymbol{\beta})\Big)$, we ought to calculate

$$f(p|y) = \frac{f(y,p)}{f(y)} = \frac{f_P(p|\alpha) f(y|p)}{\int_0^\infty f_P(p|\alpha) f(y|p) dp}.$$

Since $Y|P = p \sim N(\mu(\boldsymbol{x}, \boldsymbol{\beta}), p\sigma^2)$, we have

$$E_1 = \frac{\int_0^\infty p^{-1/2-1} f_P(p|\alpha) \exp\{\frac{-(y-\mu(\boldsymbol{x},\boldsymbol{\beta}))^2}{2p\sigma^2}\} dp}{\int_0^\infty p^{-1} f_P(p|\alpha) \exp\{\frac{-(y-\mu(\boldsymbol{x},\boldsymbol{\beta}))^2)}{2p\sigma^2}\} dp},$$

or

$$E_1 = \frac{E_P\left(P^{-1/2-1}\exp\{\frac{-(y-\mu(\boldsymbol{x},\boldsymbol{\beta}))^2}{2P\sigma^2}\}\right)}{E_P\left(P^{-1}\exp\{\frac{-(y-\mu(\boldsymbol{x},\boldsymbol{\beta}))^2)}{2P\sigma^2}\}\right)},$$

where E_P refer to expectation value with respect to random variable P. For approximating E_1, we use a Monte Carlo integration (Kong et al., 2003) method by generating B samples form the probability density function of P and computing the elements of under integral. According to Monte Carlo integration technique $E_P(g(P)) = \sum_{b=1}^{B} g(p_b)/B$, as $B \to \infty$. If $p_1^{mc}, \ldots, p_B^{mc}$ be a random sample from $f_P(p|\alpha)$, then the approximate value of E_1 is

$$\frac{\sum_{b=1}^{B} p_b^{mc(-1/2-1)}\exp\{\frac{-(y-\mu(\boldsymbol{x},\boldsymbol{\beta}))^2)}{2p_b^{mc}\sigma^2}\}}{\sum_{b=1}^{B} p_b^{mc(-1/2)}\exp\{\frac{-(y-\mu(\boldsymbol{x},\boldsymbol{\beta}))^2)}{2p_b^{mc}\sigma^2}\}}.$$

We take $B = 3000$ and update $e_{pig}^{(t)}$, in the iteration t for $i = 1, \ldots, n$ and $g = 1, \ldots, G$.