Statistical Papers (2025) 66:114
https://doi.org/10.1007/s00362-025-01723-9

REGULAR ARTICLE

# Handling skewness and directional tails in model-based clustering

**Cristina Tortora**[1] · **Antonio Punzo**[2] · **Brian C. Franczak**[3]

Received: 21 May 2024 / Revised: 6 May 2025 / Published online: 4 July 2025


## Abstract

Model-based clustering is a powerful approach used in data analysis to unveil underlying patterns or groups within a data set. However, when applied to clusters that exhibit skewness, heavy tails, or both, the classification of data points becomes more challenging. In this study, we introduce two models considering two component-wise transformations of the observed data within a mixture of multiple scaled contaminated normal (MSCN) distributions. MSCN distributions are designed to enable a different tail behavior in each dimension and directional outlier detection in the direction of the principal components. Using the transformed MSCN distributions as components of a mixture, we obtain model-based clustering techniques that allow for 1) flexible cluster shapes in terms of skewness and kurtosis and 2) component-wise and directional outlier detection. We assess the efficacy of the proposed techniques by comparing them with model-based clustering methods that perform global or component-wise outlier detection using simulated and real data sets. This comparative analysis aims to demonstrate which practical clustering scenarios using the proposed MSCN-based approaches are advantageous.

**Keywords** EM algorithm · Multiple scaled distributions · Contaminated normal distribution · Data transformations · Model-based clustering

✉ Antonio Punzo
antonio.punzo@unict.it

Cristina Tortora
cristina.tortora@sjsu.edu

[1] Department of Mathematics and Statistics, San José State University, One Washington square, San José, California 95192, USA

[2] Dipartimento di Economia e Impresa, Università di Catania, Catania, Sicily, Italy

[3] Department of Mathematics and Statistics, MacEwan University, Edmonton, Alberta, Canada

## 1 Introduction

Cluster analysis, or clustering, refers to the process of identifying the underlying groups (or clusters) within a data set. The term model-based clustering refers to using a finite mixture model for clustering. Traditionally, the most popular model-based clustering approach assumes that each group in the data follows a multivariate normal distribution (see Sect. 2.1 of McNicholas 2016, for a discussion of the historical development of this model). Unfortunately, the assumption of normally distributed clusters is typically unrealistic, as, in real-world scenarios, groups can be skewed, leptokurtic, or both.

The statistical literature on model-based clustering has two main approaches to handling clusters with non-normal features. The first approach utilizes a flexible parametric distribution to parameterize the feature(s) directly. There are many examples of such models; for example, Peel and McLachlan (2000) introduces a mixture of $t$ (Mt) distributions for groups of observations with longer than normal or heavy tails. Azzalini (2005) and Lin (2009) discuss mixtures of skew-normal distributions to handle data sets with asymmetric features, Azzalini and Capitanio (2003) and Lin (2010) propose mixtures of skew-$t$ (MSt) distributions to handle both heavy-tailed and asymmetric clusters of observations, Franczak et al. (2014) develop a mixture of shifted asymmetric Laplace (MSAL) distributions as an alternative to Lin (2009), and Browne and McNicholas (2015) present a mixture of generalized hyperbolic (MGH) distributions. Note that all of the distributions above are multivariate. There are other notable examples of finite mixtures of skewed distributions; for example, one can consider the mixture of variance-gamma distributions (McNicholas et al. 2017) and the mixture of canonical fundamental skew-$t$ distributions (Lee and McLachlan 2016) for a more detailed study on finite mixtures of skewed distributions, see Davila et al. (2018). The second approach for handling asymmetry is to apply a transformation to the observed data. Widespread transformations like the Box-Cox (Box and Cox 1964), the logarithmic, or square-root transformations can mitigate skewness and suppress some of the non-normal features of the observed data, making the application of a mixture of normal (MN) distributions more suitable. A recent example of such an approach is given in Zhu and Melnykov (2018), where the authors introduce a clustering framework that considers a Manly transformation (Manly 1976) of the observed data within the MN distributions. Herein, we refer to this model as a Manly transformed mixture.

As an alternative to Mt and MSt distributions, one may be interested in identifying the observation(s) leading to the heavier-than-normal tails. In this context, we may now consider these observations spurious points or mild outliers. To do so, we follow the recommendation of Davies and Gather (1993) and define these observations with respect to a reference distribution. Punzo and McNicholas (2016) summarizes approaches that assume the reference distribution is multivariate normal within a model-based clustering framework. In addition, Punzo and McNicholas (2016) develop a mixture of contaminated normal distributions (MCN; see Sect. 2.1 for details). The MCN distribution performs model-based clustering and outlier detection simultaneously, returning a dual classification of the observations based on their group membership and whether they are outlying points. However, if the data set con-

tains asymmetric features, this approach will not be suitable. As such, Morris et al. (2019) introduces a mixture of contaminated shifted asymmetric Laplace (MCSAL) distributions that perform the same dual classification as the MCN distributions but assume the reference distribution is shifted asymmetric Laplace (SAL). Instead of parameterizing skewness directly, one can also consider a transformation within the contaminated mixture modeling framework. The most notable example of such an approach is given in Melnykov et al. (2021), where the authors introduce the mixture of power-transformed contaminated normal (MPTCN) distributions. The MPTCN performs the same dual classification mentioned above but accounts for skewness via a power transformation within MCN distributions.

A drawback to utilizing any of the distributions above is that the shape of the hyper-contours may be restrictive. This is because the tail behavior of the data set(s) of interest is modeled using common parameters across all data dimensions. To overcome this issue, Forbes and Wraith (2014) propose a generalized multivariate normal variance-mean mixture (Barndorff-Nielsen et al. 1982) that accommodates varying levels of excess kurtosis on each principal component (PC) of the data. This leads to alternative shapes for the hyper-contours of the distribution of interest. Herein, we refer to any distribution that utilizes this generalization as being multiple scaled. Note that Forbes and Wraith (2014) use this generalization to develop a mixture of multiple scaled-$t$ (MSt) distributions.

The concept of a multiple scaled distribution can be viewed as an extension of the multivariate normal variance-mean mixture, characterized by two fundamental elements:

1. The decomposition of the scale matrix, $\boldsymbol{\Sigma}$, using eigenvalues and eigenvectors matrices $\boldsymbol{\Lambda}$ and $\boldsymbol{\Gamma}$, respectively.
2. The use of a random variable $W$ independently for each dimension of the space spanned by the columns of $\boldsymbol{\Gamma}$, that is, for each PC.

This approach has been extended to include several of the distributions cited above. For example, Franczak et al. (2015) develop a multiple scaled SAL (MSSAL) distribution. A few papers focus on the generalized hyperbolic (GH) distribution, Wraith and Forbes (2015) propose a multiple scaled GH (MSGH) distribution, Tortora et al. (2019) introduce a coalesced GH distribution that utilizes the MSGH distribution. Punzo and Bagnato (2022) discusses multiple scaled distributions within the context of allometric studies, and Mahdavi et al. (2024) proposes the skew multiple scaled mixtures of multivariate normal distributions. Within the contaminated setting, Punzo and Tortora (2021) extend the multiple scaling concept to include the contaminated normal (CN) distribution and introduce the multiple scaled contaminated normal (MSCN) distribution (see Sect. 2.2 for more information) and Tortora et al. (2024) propose a multiple scaled contaminated MSSAL (MSCMSSAL) distribution. The MSCN and MSCMSSAL allow directional outlier detection independently in the direction of the PCs; however, the MSCMSSAL assumes that each PC follows a contaminated asymmetric Laplace distribution. Thus, unlike the MSCN distribution, the MSCMSSAL can account for skewness in the direction of each PC. However, due to a complex parameter estimation procedure for the eigenvector matrix (see Sect.

4.3 of Tortora et al. 2024, for details), the MSCMSSAL distribution is not used for model-based clustering. As such, we propose two novel mixtures of MSCN distributions that utilize transformations of the observed data to account for asymmetry on each PC. Like the MSCMSSAL, these models account for varying levels of excess kurtosis on each PC and can be used for outlier detection, but, unlike the MSCMSSAL, we are also able to use these models for clustering, rectifying the aforementioned drawback of the MSCMSSAL.

The remainder of the paper is organized as follows. In Sect. 2 we provide the necessary background material to develop the proposed models. In Sect. 3 we present the novel transformations of the MSCN distribution. Specifically, Sect. 3.1 describes a Manly transformed MSCN (MTMSCN) distribution, and Sect. 3.2 describes a power-transformed MSCN (PTMSCN) distribution. A parameter estimation framework and other algorithmic considerations are presented in Sect. 4. In Sect. 5 we evaluate the proposed mixtures using both simulated and real data sets, and in Sect. 6 we conclude with a summary and a discussion of future work. Appendices A and B explain the data simulation process and give the parameters used in the simulation study, respectively. Finally, Appendices C and D contains additional results from the simulation study discussed in Sect. 5.

## 2 Background

### 2.1 Mixtures of contaminated normal distributions

The contaminated normal (CN) distribution proposed by Tukey (1960) is a mixture of two normal distributions, where both distributions have the same location, but one has an inflated covariance matrix to model mild outliers. Formally, the probability density function (pdf) of a $d$-variate random vector $\boldsymbol{X} = (X_1, \ldots, X_d)^\top$ following a CN distribution can be written as

$$f_{\text{CN}}\left(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}, \alpha, \eta\right) = \alpha f_{\text{N}}\left(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}\right) + (1-\alpha) f_{\text{N}}\left(\boldsymbol{x} \mid \boldsymbol{\mu}, \eta\boldsymbol{\Sigma}\right), \tag{1}$$

where $f_{\text{N}}\left(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}\right)$ is the pdf of the multivariate normal distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$, $\alpha \in (0,1)$ is the proportion of good observations, and $\eta > 1$ is the degree of contamination which captures the increase in variability due to the presence of mild outliers. The constraint $\eta > 1$ is used to guarantee model identifiability. Alternatively, one can refer to $\alpha$ and $\eta$ as the tailedness parameters of the distributions, allowing the distribution to be leptokurtic. More details on the kurtosis of the CN distribution as a function of $\alpha$ and $\eta$ can be found in Appendix G of Bagnato et al. (2017).

In applications, the constraint $\alpha \in (0.5, 1)$ is typically used to ensure that at least half of the observations are considered "good" and that the model can be used for outlier detection (Punzo and McNicholas 2016).

Punzo and McNicholas (2016) develop the MCN distributions for model-based clustering. Formally, the density of the MCN distributions is a convex combination of

$G$ multivariate CN density functions. As such, one can write the density of an MCN distribution as

$$f_{\text{MCN}}(\boldsymbol{x} \mid \boldsymbol{\vartheta}) = \sum_{g=1}^{G} \pi_g f_{\text{CN}}(\boldsymbol{x} \mid \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \alpha_g, \eta_g), \tag{2}$$

where $\boldsymbol{\vartheta}$ is the vector of all model parameters, $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_G)$, such that $0 < \pi_g \leq 1$ and $\sum_{g=1}^{G} \pi_g = 1$, are the mixing proportions, $f_{\text{CN}}(\boldsymbol{x} \mid \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \alpha_g, \eta_g)$ is the pdf of the CN distribution defined in (1), and all other model parameters are as defined for (1), $g = 1, \ldots, G$.

### 2.2 Mixtures of multiple scaled contaminated normal distributions

To address the limitations of the MCN distribution discussed in Sect. 1, Punzo and Tortora (2021) introduces a mixture of MSCN distributions (MMSCN). Formally, the pdf of the MMSCN can be written as:

$$\begin{aligned} f_{\text{MMSCN}}(\boldsymbol{x} \mid \boldsymbol{\vartheta}) &= \sum_{g=1}^{G} \pi_g f_{\text{MSCN}}(\boldsymbol{x} \mid \boldsymbol{\mu}_g, \boldsymbol{\Gamma}_g, \boldsymbol{\Lambda}_g, \boldsymbol{\alpha}_g, \boldsymbol{\eta}_g) \\ &= \sum_{g=1}^{G} \pi_g \prod_{h=1}^{d} f_{\text{CN}}\left(\left[\boldsymbol{\Gamma}_g^{\top}(\boldsymbol{x} - \boldsymbol{\mu}_g)\right]_h \mid 0, \Lambda_{hg}, \alpha_{hg}, \eta_{hg}\right), \end{aligned} \tag{3}$$

where $d$, in addition to be the number of available variables, is also the number of PCs, $\pi_g$ and $\boldsymbol{\vartheta}$ are defined as for (2), $\boldsymbol{\mu}_g$ is the location parameter for the observed data, $\boldsymbol{\Gamma}_g$ is a matrix of eigenvectors such that $\left[\boldsymbol{\Gamma}_g^{\top}(\boldsymbol{x} - \boldsymbol{\mu}_g)\right]_h$ represents the $h$th element of the PC transform of $\boldsymbol{x}$, $\boldsymbol{\Lambda}_g$ is a diagonal matrix of eigenvalues with elements $\Lambda_{1g}, \ldots, \Lambda_{dg}$, $\boldsymbol{\alpha}_g = (\alpha_{1\,g}, \ldots, \alpha_{dg})^{\top}$ and $\boldsymbol{\eta}_g = (\eta_{1\,g}, \ldots, \eta_{dg})^{\top}$ give, respectively, the proportion of good points and the degree of contamination in each PC. It follows from (3) that the MSCN has marginal CN distributions on each PC; see Punzo and Tortora (2021) for details.

Compared to the MCN, the MMSCN allows for directional outlier detection separately in the direction of each PC. This provides a versatile framework for modeling complex data distributions, particularly in scenarios where traditional elliptical distributions may not adequately capture the underlying variability and tail behaviors present in the data.

### 2.3 Two transformations to account for asymmetry

As noted in Sect. 1, there are two principal approaches to model-based clustering when dealing with asymmetric clusters: (1) directly modeling skewness using mixture component distributions that account for skewness, or (2) applying a suitable cluster-specific transformation to the observed data. With respect to the second

approach, Zhu and Melnykov (2018) suggests using a Manly transformation, formally defined as

$$\mathscr{M}\left(\boldsymbol{X} \mid \boldsymbol{\lambda}\right) = \left(\frac{e^{\lambda_1 x_1} - 1}{\lambda_1}, \ldots, \frac{e^{\lambda_d x_d} - 1}{\lambda_d}\right)^{\top}, \tag{4}$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_d)$, with $\lambda_h \neq 0$ for $h = 1, \ldots, d$, represents the transformation parameter vector. If $\lambda_h = 0$, the $h$th variable remains untransformed. Zhu and Melnykov (2018) further demonstrate that the Manly transformation of $\boldsymbol{X}$, as defined in (4), can be incorporated into a mixture modeling framework by assuming $\mathscr{M}\left(\boldsymbol{X} \mid \boldsymbol{\lambda}_g\right) \sim \mathrm{N}_d\left(\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g\right)$, for $g = 1, \ldots, G$, meaning that in the $g$th group a Manly transformation of $\boldsymbol{X}$ follows a normal distribution. This model is referred to as the mixture of Manly transformed distributions.

Along a similar trajectory, Melnykov et al. (2021) consider a power transformation of $\boldsymbol{X}$ from some non-symmetric multivariate statistical distribution. The power transformation for the generic $h$th dimension, $h = 1, \ldots, d$, is defined as

$$\mathscr{T}\left(X_h \mid \lambda_h\right) = \begin{cases} \lambda_h^{-1}\left(\left(X_h + 1\right)^{\lambda_h} - 1\right), & \lambda_h \neq 0,\ X_h \geq 0 \\ \log\left(X_h + 1\right), & \lambda_h = 0,\ X_h \geq 0 \\ \left(2 - \lambda_h\right)^{-1}\left(1 - \left(1 - X_h\right)^{2-\lambda_h}\right), & \lambda_h \neq 2,\ X_h < 0 \\ -\log\left(1 - X_h\right), & \lambda_h = 2,\ X_h < 0 \end{cases}, \tag{5}$$

such that $\mathscr{T}\left(\boldsymbol{X} \mid \boldsymbol{\lambda}\right) = \left(\mathscr{T}\left(X_1 \mid \lambda_1\right), \ldots, \mathscr{T}\left(X_d \mid \lambda_d\right)\right)^{\top}$. Melnykov et al. (2021) go on to develop the MPTCN by assuming that $\mathscr{T}\left(\boldsymbol{X} \mid \boldsymbol{\lambda}_g\right) \sim \mathrm{CN}(\boldsymbol{x} \mid \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \alpha_g, \eta_g)$, where $\mathrm{CN}(\boldsymbol{x} \mid \boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g, \alpha_g, \eta_g)$ refers to the multivariate CN distribution with density given in (1), for $g = 1, \ldots, G$.

## 3 Marginal transformations for the MSCN distribution

In this section, we introduce two marginally transformed MSCN distributions. The proposed models have the same advantages as the MSCN, i.e., they offer flexible tails that allow for directional outlier detection and different downweighting of outlying observations in each PC; however, they can also account for skewness. For each model, we assume that the transformed data follow the MSCN distribution in each mixture component.

### 3.1 Manly transformed MSCN distribution

The first model is obtained by assuming, in the generic $g$ cluster with, that

$$\mathscr{M}\left(\boldsymbol{X} \mid \boldsymbol{\lambda}_g\right) \sim \mathrm{MSCN}(\boldsymbol{\mu}_g, \boldsymbol{\Gamma}_g, \boldsymbol{\Lambda}_g, \boldsymbol{\alpha}_g, \boldsymbol{\eta}_g), \tag{6}$$

i.e., that a Manly transformation of $\boldsymbol{X}$ follows an MSCN distribution with the parameter set defined for (3). Following this definition, we then assume that the observed

data can be modeled using a novel mixture of Manly transformed MSCN (MMTMSCN) distributions with pdf

$$\begin{aligned} f_{\text{MMTMSCN}}(\boldsymbol{x} \mid \boldsymbol{\vartheta}) &= \sum_{g=1}^{G} \pi_g \prod_{h=1}^{d} f_{\text{CN}}\left(\left[\boldsymbol{\Gamma}_g^{\top}\left(\mathscr{M}(\boldsymbol{x} \mid \boldsymbol{\lambda}_g) - \boldsymbol{\mu}_g\right)\right]_h \mid 0, \Lambda_{hg}, \alpha_{hg}, \eta_{hg}\right) \prod_{l=1}^{d} \mathrm{J}(x_l \mid \lambda_{lg}) \\ &= \sum_{g=1}^{G} \pi_g f_{\text{MSCN}}\left(\mathscr{M}(\boldsymbol{x} \mid \boldsymbol{\lambda}_g) \mid \boldsymbol{\mu}_g, \boldsymbol{\Gamma}_g, \boldsymbol{\Lambda}_g, \boldsymbol{\alpha}_g, \boldsymbol{\eta}_g\right) \prod_{l=1}^{d} \mathrm{J}(x_l \mid \lambda_{lg}) \\ &= \sum_{g=1}^{G} \pi_g f_{\text{MTMSCN}}(\boldsymbol{x} \mid \boldsymbol{\vartheta}_g), \end{aligned} \tag{7}$$

where all model parameters are as defined for (3), $\mathscr{M}(\boldsymbol{x} \mid \boldsymbol{\lambda}_g)$ is the Manly transformation of $\boldsymbol{X}$ defined for (4), and $\mathrm{J}(x_l \mid \lambda_{lg})$ is the Jacobian associated with the transformation from the MSCN distribution for $g = 1, \ldots, G$ and $l = 1, \ldots, d$ (cf. Zhu and Melnykov 2018).

### 3.2 Power transformed MSCN distribution

Building on the procedure outlined in Sect. 3.1, we introduce a second novel assumption. Specifically, within the th cluster, for, we assume that

$$\mathscr{T}(\boldsymbol{X} \mid \boldsymbol{\lambda}_g) \sim \text{MSCN}(\boldsymbol{\mu}_g, \boldsymbol{\Gamma}_g, \boldsymbol{\Lambda}_g, \boldsymbol{\alpha}_g, \boldsymbol{\eta}_g), \tag{8}$$

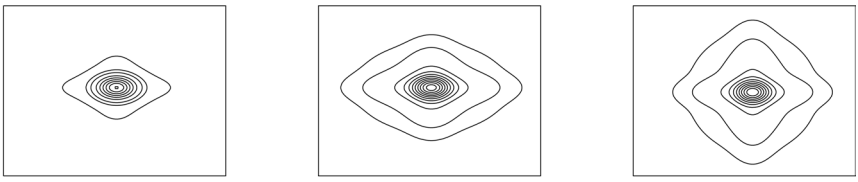
i.e., that a power transformation of $\boldsymbol{X}$ follows an MSCN distribution with the parameter set defined for (3). It follows that the density of the observed random vector $\boldsymbol{X}$ can be modeled using a mixture of power-transformed MSCN (MPTMSCN) distributions with pdf

$$\begin{aligned} f_{\text{MPTMSCN}}(\boldsymbol{x} \mid \boldsymbol{\vartheta}) &= \sum_{g=1}^{G} \pi_g \prod_{h=1}^{d} f_{\text{CN}}\left(\left[\boldsymbol{\Gamma}_g^{\top}\left(\mathscr{T}(\boldsymbol{x} \mid \boldsymbol{\lambda}_g) - \boldsymbol{\mu}_g\right)\right]_h \mid 0, \Lambda_{hg}, \alpha_{hg}, \eta_{hg}\right) \prod_{l=1}^{d} \mathrm{J}(x_l \mid \lambda_{lj}) \\ &= \sum_{g=1}^{G} \pi_g f_{\text{MSCN}}\left(\mathscr{T}(\boldsymbol{x} \mid \boldsymbol{\lambda}_g) \mid \boldsymbol{\mu}_g, \boldsymbol{\Gamma}_g, \boldsymbol{\Lambda}_g, \boldsymbol{\alpha}_g, \boldsymbol{\eta}_g\right) \prod_{l=1}^{d} \mathrm{J}(x_l \mid \lambda_{lj}) \\ &= \sum_{g=1}^{G} \pi_g f_{\text{PTMSCN}}(\boldsymbol{x} \mid \boldsymbol{\vartheta}_g), \end{aligned} \tag{9}$$

where all parameters are as defined for (7).

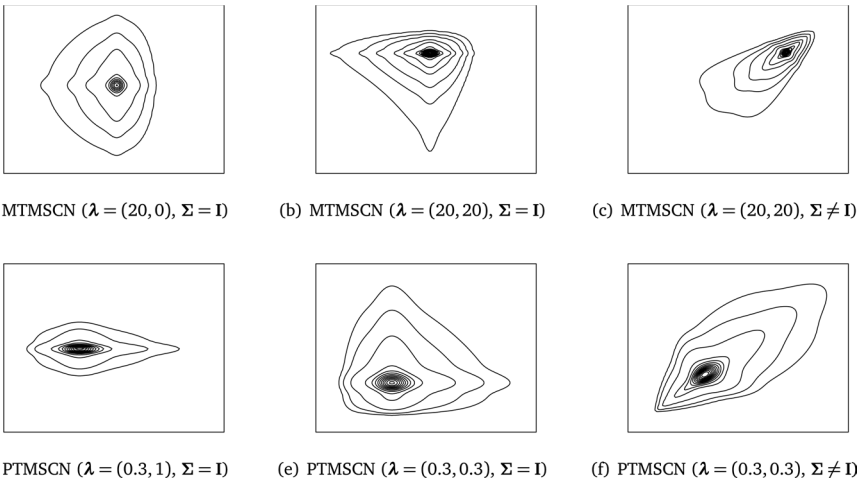
Figures 1 and 2 show examples of contour plots obtained from the MTMSCN and PTMSCN distributions. Specifically, Fig. 1 shows the effect of $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$ when $\boldsymbol{\Sigma} = \mathrm{I}$ under symmetry (i.e., when $\boldsymbol{\lambda} = (0, 0)$ for the MTMSCN and $\boldsymbol{\lambda} = (1, 1)$ for the PTMSCN). In Fig. 1a and b, $\boldsymbol{\eta} = (5, 5)^{\top}$. In the former, $\boldsymbol{\alpha} = (0.99, 0.99)^{\top}$ which results in almost no outliers, as $\boldsymbol{\alpha}$ decreases to $(0.6, 0.6)^{\top}$ the tails become wider (Fig. 1b). In Fig. 1c $\boldsymbol{\alpha} = (0.6, 0.6)^{\top}$ and $\boldsymbol{\eta} = (20, 5)^{\top}$, which is reflected in the longer tail present for one of the PCs.

Figure 2 shows contour plots of the MTMSCN and PTMSCN distributions for varying $\boldsymbol{\lambda}$. The first row corresponds to the MTMSCN distributions, and the second



(a) $\boldsymbol{\alpha} = (0.99, 0.99)^\top$, $\boldsymbol{\eta} = (5, 5)^\top$ (b) $\boldsymbol{\alpha} = (0.6, 0.6)^\top$, $\boldsymbol{\eta} = (5, 5)^\top$ (c) $\boldsymbol{\alpha} = (0.6, 0.6)^\top$, $\boldsymbol{\eta} = (20, 5)^\top$

**Fig. 1** Contour plots of MTMSCN and PTMSCN distributions under symmetry (i.e. when $\boldsymbol{\lambda} = (0, 0)$ for the MTMSCN and $\boldsymbol{\lambda} = (1, 1)$ for the PTMSCN), $\boldsymbol{\Sigma} = \mathbf{I}$, and the varying values of $\boldsymbol{\alpha}$ and $\boldsymbol{\eta}$ given in the sub-captions



(a) MTMSCN ($\boldsymbol{\lambda} = (20, 0)$, $\boldsymbol{\Sigma} = \mathbf{I}$) (b) MTMSCN ($\boldsymbol{\lambda} = (20, 20)$, $\boldsymbol{\Sigma} = \mathbf{I}$) (c) MTMSCN ($\boldsymbol{\lambda} = (20, 20)$, $\boldsymbol{\Sigma} \neq \mathbf{I}$)

(d) PTMSCN ($\boldsymbol{\lambda} = (0.3, 1)$, $\boldsymbol{\Sigma} = \mathbf{I}$) (e) PTMSCN ($\boldsymbol{\lambda} = (0.3, 0.3)$, $\boldsymbol{\Sigma} = \mathbf{I}$) (f) PTMSCN ($\boldsymbol{\lambda} = (0.3, 0.3)$, $\boldsymbol{\Sigma} \neq \mathbf{I}$)

**Fig. 2** Contour plots of MTMSCN distribution, first row, and PTMSCN distribution, second row, with $\boldsymbol{\alpha} = (0.6, 0.6)^\top$ and $\boldsymbol{\eta} = (20, 5)^\top$. In the first column, $\boldsymbol{\Sigma} = \mathbf{I}$ and one PC is skewed, in the second column, $\boldsymbol{\Sigma} = \mathbf{I}$ and both PCs are skewed, and in the third column, both PCs are skewed with a correlation of 0.5

to the PTMSCN distribution. In the first column (Fig. 2a and d), $\boldsymbol{\Sigma} = \mathrm{I}$ and only one principal component is skewed as $\boldsymbol{\lambda} = (20, 0)$ for the MTMSCN distribution and $\boldsymbol{\lambda} = (0.3, 1)$ for the PTMSCN distribution. In the second column (Fig. 2b and e), skewness is introduced in both PCs by setting $\boldsymbol{\lambda} = (20, 20)$ for the MTMSCN and $\boldsymbol{\lambda} = (0.3, 0.3)$ for the PTMSCN. As the two components are uncorrelated, the skewness is in the direction of the axes. The effect of correlation can be seen in the third column (Fig. 2c and f), where the off-diagonal elements of $\boldsymbol{\Sigma}$ are set to 0.5. Note that these directions of skewness are chosen for illustrative purposes; the models can account for skewness in all directions.

## 4 Implementation

This section discusses the computational considerations made when implementing the MTMCSN and PTMSCN distributions. Specifically, Sect. 4.1 gives details on the parameter estimation scheme used to fit the proposed models, Sect. 4.2 describe the used algorithm, Sect. 4.3 discusses relevant issues with the proposed parameter estimation scheme, e.g., how to initialize the algorithm and monitor for convergence, and Sect. 4.4 discusses model selection after convergence.

### 4.1 Parameter estimation

To determine the maximum likelihood (ML) estimates for model parameters of interest, we employ a variant of the classical expectation-maximization (EM) algorithm (Dempster et al. 1977), the expectation conditional maximization (ECM) algorithm (Meng and Rubin 1993). The EM algorithm and its variants are natural approaches for ML estimation when dealing with incomplete data (McLachlan and Krishnan 2008). Effectively, the EM algorithm iterates between two steps, an E-step and an M-step. On the E-step, we compute the expected value of the complete-data log-likelihood. On the M-step, we maximize the expected value of the complete-data log-likelihood with respect to the model parameters of interest. The ECM differs from the traditional EM algorithm as it uses computationally simpler conditional maximization (CM) steps instead of one M-step to update the model parameters in disjoint subsets rather than as a whole.

We encounter two levels of data incompleteness for the proposed MMTMSCN and MPTMSCN distributions. The first level of incompleteness stems from the uncertainty about the component membership of each observation. To address this, we use an indicator vector $\boldsymbol{z}_i = (z_{i1}, \ldots, z_{ig})$, where $z_{ig} = 1$ if $\boldsymbol{x}_i$ belongs to component $g$, and $z_{ig} = 0$ otherwise, for $g = 1, \ldots, G$. This type of incompleteness is common in the context of mixture models. The second level of incompleteness arises from the uncertainty of knowing whether the $i$th transformed and projected observation $\left[\boldsymbol{\Gamma}_g^\top \left(\mathscr{T}\left(\boldsymbol{x}_i \mid \boldsymbol{\lambda}_g\right) - \boldsymbol{\mu}_g\right)\right]_h$ is either good or bad on the $h$th PC in the $g$th cluster, for $i = 1, \ldots, n$, $h = 1, \ldots, d$, and $g = 1, \ldots, G$. This level of incompleteness is governed by an $n \times d \times G$ indicator array with elements $v_{ihg}$, where $v_{ihg} = 1$ if the $h$th PC of the $i$th transformed and projected observation within cluster $g$ is deemed a "good" observation, and $v_{ihg} = 0$ otherwise, for $i = 1, \ldots, n$, $h = 1, \ldots, d$, and $g = 1, \ldots, G$.

Therefore, for the MMTMSCN and MPTMSCN distributions, the complete-data is comprised of the observed $\boldsymbol{x}_i$, the $z_{ig}$, and the $v_{ihg}$, for $i = 1, \ldots, n$, $h = 1, \ldots, d$, and $g = 1, \ldots, G$. Using (7) and (9), we can derive the complete-data likelihood functions for the marginally transformed MSCN distributions of interest. As the structure of the complete-data likelihood functions for the MMTMSCN and MPTMSCN distributions are identical, except for the transformation, we only formally state the complete-data likelihood for the MPTMSCN. This function can be written as

$$L_c(\boldsymbol{\vartheta}) = \prod_{i=1}^{n}\prod_{g=1}^{G}\left\{\pi_g \prod_{h=1}^{d}\left[\alpha_{hg} f_{\mathrm{N}}\left(\left[\boldsymbol{\Gamma}_g^{\top}\left(\mathscr{T}\left(\boldsymbol{x}_i \mid \boldsymbol{\lambda}_g\right)-\boldsymbol{\mu}_g\right)\right]_h \mid 0, \Lambda_{hg}\right)\right]^{v_{ihg}}\right.$$
$$\left.\times\left[\left(1-\alpha_{hg}\right) f_{\mathrm{N}}\left(\left[\boldsymbol{\Gamma}_g^{\top}\left(\mathscr{T}\left(\boldsymbol{x}_i \mid \boldsymbol{\lambda}_g\right)-\boldsymbol{\mu}_g\right)\right]_h \mid 0, \eta_{hg}\Lambda_{hg}\right)\right]^{(1-v_{ihg})} \times \prod_{l=1}^{d} \mathrm{J}\left(x_{il} \mid \lambda_{lg}\right)\right\}^{z_{ig}}, \tag{10}$$

where all model parameters are as defined for (9). It follows that the complete-data likelihood for the MMTMSCN will be identical to that given in (10), except with the Manly transformation of $\boldsymbol{x}_i$, $i = 1, \ldots, n$. For the MPTMSCN, the corresponding complete-data log-likelihood can be written as

$$l_c(\boldsymbol{\vartheta}) = \sum_{g=1}^{G}\left[l_{c,1g}(\pi_g) + l_{c,2g}(\boldsymbol{\alpha}_g) + l_{c,3g}(\boldsymbol{\theta}_g) + l_{c,4g}(\boldsymbol{\lambda}_g)\right], \tag{11}$$

where $\boldsymbol{\theta}_g = (\boldsymbol{\mu}_g, \boldsymbol{\Gamma}_g, \boldsymbol{\Lambda}_g, \boldsymbol{\eta}_g)$ and

$$l_{c,1g}(\pi_g) = \sum_{i=1}^{n} z_{ig} \ln \pi_g, \tag{12}$$

$$l_{c,2g}(\boldsymbol{\alpha}_g) = \sum_{i=1}^{n}\sum_{h=1}^{d} z_{ig}\left[v_{ihg} \ln \alpha_{hg} + (1 - v_{ihg}) \ln\left(1 - \alpha_{hg}\right)\right], \tag{13}$$

$$l_{c,3g}(\boldsymbol{\theta}_g) = -\frac{1}{2}\sum_{i=1}^{n}\sum_{h=1}^{d} z_{ig}\left\{\ln \Lambda_{hg} + (1 - v_{ihg}) \ln \eta_{hg}\right.$$
$$\left.+\left(v_{ihg} + \frac{1 - v_{ihg}}{\eta_{hg}}\right)\frac{\left[\boldsymbol{\Gamma}_g^{\top}\left(\mathscr{T}\left(\boldsymbol{x}_i \mid \boldsymbol{\lambda}_g\right)-\boldsymbol{\mu}_g\right)\right]_h^2}{\Lambda_{hg}}\right\}, \tag{14}$$

$$l_{c,4g}(\boldsymbol{\lambda}_g) = \sum_{i=1}^{n}\sum_{l=1}^{d} z_{ig} \ln \mathrm{J}\left(x_{il} \mid \lambda_{lg}\right). \tag{15}$$

An advantage of applying a transformation that aims to obtain a symmetric distribution is that a parameter estimation scheme based on the EM algorithm will be very similar to that of the target (symmetric) distribution. For example, Zhu and Melnykov (2018) shows that the EM algorithm for the Manly mixture is analogous to that of a mixture of normal distributions (e.g., compare Sect 2.3 of Zhu and Melnykov 2018 to Section 2.2.1 of McNicholas 2016). The only caveat is that the transformation parameter $\boldsymbol{\lambda}_g = (\lambda_{1\,g}, \ldots, \lambda_{dg})$, for $g = 1, \ldots, G$, has to be estimated using a numerical procedure. Similarly, Melnykov (2021, Sect. 2.1) show that the proposed EM algorithm for the MPTCN is almost identical to the ECM algorithm used for an MCN distribution (see Sect 4.1 of Punzo and McNicholas 2016). Again, with the caveat that the transformation parameter $\boldsymbol{\lambda}_g = (\lambda_{1\,g}, \ldots, \lambda_{dg})$, for $g = 1, \ldots, G$, has to be estimated using a numerical procedure. Therefore, the ECM algorithm for the

proposed MMTMSCN and MPTMSCN distributions will be similar to that given in Sect 3.2 of Punzo and Tortora (2021). However, there is still the issue of estimating the transformation parameter $\boldsymbol{\lambda}_g$, $g = 1, \ldots, G$, for the MMTMSCN and MPTMSCN. For both models, we follow the suggestions of Zhu and Melnykov (2018) and Melnykov et al. (2021) and use a general-purpose optimization procedure based on the Nelder-Mead method (Nelder and Mead 1965).

### 4.2 A ECM algorithm for MPTMSCN distributions

As eluded to at the end of Sect. 4.1, the updates for the ECM algorithms for the novel MPTMSCN and MMTMSCN distributions will only differ with respect to the utilized transformation of the observed data. So, for the sake of brevity, we only outline the proposed ECM algorithm for the MPTMSCN distributions.

#### 4.2.1 E-step

On the E-step of the proposed ECM algorithm for the MPTMSCN distributions, we can write

$$E\left(Z_{ig} \,|\boldsymbol{x}_i\right) = \frac{\dot{\pi}_g f_{\text{PTMSCN}}\left(\boldsymbol{x}_i \mid \dot{\boldsymbol{\vartheta}}_g\right)}{f_{\text{MPTMSCN}}\left(\boldsymbol{x}_i \mid \dot{\boldsymbol{\vartheta}}\right)} := \ddot{z}_{ig} \tag{16}$$

and

$$E\left(V_{ihg}|\boldsymbol{x}_i, Z_{ig} = 1\right) = \frac{\dot{\alpha}_{hg} f_{\text{N}}\left(\left[\dot{\boldsymbol{\Gamma}}_g^{\top}\left(\mathscr{T}\left(\boldsymbol{x}_i \mid \dot{\boldsymbol{\lambda}}_g\right) - \dot{\boldsymbol{\mu}}_g\right)\right]_h \mid 0, \dot{\Lambda}_{hg}\right)}{f_{\text{CN}}\left(\left[\dot{\boldsymbol{\Gamma}}_g^{\top}\left(\mathscr{T}\left(\boldsymbol{x}_i \mid \dot{\boldsymbol{\lambda}}_g\right) - \dot{\boldsymbol{\mu}}_g\right)\right]_h \mid 0, \dot{\Lambda}_{hg}, \dot{\alpha}_{hg}, \dot{\eta}_{hg}\right)} := \ddot{v}_{ihg}, \tag{17}$$

where $f_{\text{PTMSCN}}(\cdot)$ is as defined in (9), $f_N(\cdot)$ is the density of a normal distribution, $f_{\text{CN}}(\cdot)$ is the density of the contaminated normal distribution, all parameters are as defined for (9), and the super-scripted single dot and double dots on top of the parameters stand for estimates at the previous and current iterations, respectively.

#### 4.2.2 CM-step 1

On the first CM-step of the proposed ECM algorithm, we update $\pi_g$, $\boldsymbol{\alpha}_g$, $\boldsymbol{\mu}_g$, and $\boldsymbol{\Lambda}_g$. Formally, the ML estimates for these parameters are

$$\ddot{\pi}_g = \frac{\ddot{n}_g}{n}, \tag{18}$$

$$\ddot{\alpha}_{hg} = \frac{1}{\ddot{n}_g} \sum_{i=1}^{n} \ddot{z}_{ig} \ddot{v}_{ihg}, \tag{19}$$

$$\ddot{\mu}_{hg} = \sum_{i=1}^{n} \left[ \dot{\boldsymbol{\Gamma}}_g \ddot{\widetilde{\boldsymbol{W}}}_{ig} \dot{\boldsymbol{\Gamma}}_g^{\top} \mathscr{T}\left(\boldsymbol{x}_i \mid \dot{\boldsymbol{\lambda}}_g\right) \right]_h \tag{20}$$

$$\ddot{\Lambda}_{hg} = \frac{1}{\ddot{n}_g} \sum_{i=1}^{n} \ddot{z}_{ig} \left( \ddot{v}_{ihg} + \frac{1 - \ddot{v}_{ihg}}{\dot{\eta}_{hg}} \right) \left[ \dot{\boldsymbol{\Gamma}}_g^{\top} \left( \mathscr{T}\left(\boldsymbol{x}_i \mid \dot{\boldsymbol{\lambda}}_g\right) - \ddot{\boldsymbol{\mu}}_g \right) \right]_h^2, \tag{21}$$

where $\ddot{n}_g = \sum_{i=1}^{n} \ddot{z}_{ig}$ and

$$\ddot{\widetilde{\boldsymbol{W}}}_{ig} = \operatorname{diag}\left\{ \frac{\ddot{z}_{ig} \left( \ddot{v}_{i1g} + \frac{1 - \ddot{v}_{i1g}}{\dot{\eta}_{1g}} \right)}{\sum_{i=1}^{n} \ddot{z}_{ig} \left( \ddot{v}_{i1g} + \frac{1 - \ddot{v}_{i1g}}{\dot{\eta}_{1g}} \right)}, \ldots, \frac{\ddot{z}_{ig} \left( \ddot{v}_{idg} + \frac{1 - \ddot{v}_{idg}}{\dot{\eta}_{dg}} \right)}{\sum_{i=1}^{n} \ddot{z}_{ig} \left( \ddot{v}_{idg} + \frac{1 - \ddot{v}_{idg}}{\dot{\eta}_{dg}} \right)} \right\}.$$

#### 4.2.3 CM-step 2

On the second CM-step of the proposed ECM algorithm, we update $\boldsymbol{\eta}_g$ as follows

$$\ddot{\eta}_{hg} = \max\left\{ \eta^*, \sum_{i=1}^{n} \frac{\ddot{z}_{ig}\left(1 - \ddot{v}_{ihg}\right)}{\sum_{l=1}^{n} \ddot{z}_{lj}\left(1 - \ddot{v}_{lhg}\right)} \frac{\left[ \dot{\boldsymbol{\Gamma}}_g^{\top} \left( \mathscr{T}\left(\boldsymbol{x}_i \mid \dot{\boldsymbol{\lambda}}_g\right) - \ddot{\boldsymbol{\mu}}_g \right) \right]_h^2}{\ddot{\boldsymbol{\Lambda}}_{hg}} \right\},$$

with $\eta^* = 1.001$.

#### 4.2.4 CM-step 3

On the third CM step, the update of $\boldsymbol{\Gamma}_g$, $g = 1, \ldots, G$, is obtained as

$$\ddot{\boldsymbol{\Gamma}}_g = \arg\min_{\boldsymbol{\Gamma}_g} \sum_{i=1}^{n} \operatorname{trace}\left[ \boldsymbol{\Gamma}_g \left( \ddot{\boldsymbol{W}}_{ig} \ddot{\boldsymbol{\Lambda}}_g \right)^{-1} \boldsymbol{\Gamma}_g^{\top} \ddot{z}_{ig} \left( \mathscr{T}\left(\boldsymbol{x}_i \mid \dot{\boldsymbol{\lambda}}_g\right) - \ddot{\boldsymbol{\mu}}_g \right) \left( \mathscr{T}\left(\boldsymbol{x}_i \mid \dot{\boldsymbol{\lambda}}_g\right) - \ddot{\boldsymbol{\mu}}_g \right)^{\top} \right], \tag{22}$$

where

$$\ddot{\boldsymbol{W}}_{ig} = \operatorname{diag}\left\{ \left( \ddot{v}_{i1g} + \frac{1 - \ddot{v}_{i1g}}{\ddot{\eta}_{1g}} \right)^{-1}, \ldots, \left( \ddot{v}_{idg} + \frac{1 - \ddot{v}_{idg}}{\ddot{\eta}_{dg}} \right)^{-1} \right\}.$$

#### 4.2.5 CM-step 4

On the fourth CM step, $\ddot{\boldsymbol{\lambda}}_g$, namely the update of $\boldsymbol{\lambda}_g$, is obtained numerically. As stated in Sect. 4.1, we use a general-purpose optimization procedure based on the Nelder-Mead method (Nelder and Mead 1965). We implement this method using the R function `optim()`.

### 4.3 Computational details

Despite its widespread use and effectiveness for handling incomplete data, the EM algorithm and its variants are recognized for their sensitivity to initial values and convergence speed (Biernacki et al. 2000; Karlis and Xekalaki 2003; Shireman et al. 2017). Recent research efforts, such as those by Michael and Melnykov (2016) and You et al. (2023), have focused on enhancing the initialization process of the EM algorithm. The challenge is amplified when dealing with data containing outliers, as discussed in studies like Cuesta-Albertos et al. (2008). In this paper, we use partition around medoids (PAM; Kaufman and Rousseeuw 1990) to initialize the proposed methods. Convergence in the ECM algorithm is assessed using the Aitken stopping criterion (Aitken 1926). Further details regarding the Aitken stopping criterion within model-based clustering are elaborated in McNicholas et al. (2010).

Upon convergence, we determine the cluster memberships and perform component-wise directional outlier detection for all observations. First, we assign $\boldsymbol{x}_i$ to the $g$th cluster using the maximum *a posteriori* (MAP) classification of $\widehat{z}_{ig}$, which is given by

$$\text{MAP}\left(\widehat{z}_{ig}\right) = \begin{cases} 1 & \text{if } \max_k\{\widehat{z}_{ik}\} \text{ occurs at component } k = g \\ 0 & \text{otherwise.} \end{cases}$$

Then, within the $g$th cluster, $\boldsymbol{x}_i$ is labelled as 'good' with respect to the $h$th PC if $\widehat{v}_{ihg} > 0.5$, for $i = 1, \ldots, n$, $h = 1, \ldots, d$, and $g = 1, \ldots, G$. In these rules, $\widehat{z}_{ig}$ and $\widehat{v}_{ihg}$ are the values of $\ddot{z}_{ig}$ and $\ddot{v}_{ihg}$ obtained at convergence of the proposed ECM algorithm. This two-level classification-detection process operates without additional distributional assumptions or subjective thresholds (Punzo and Tortora 2021).

### 4.4 Model selection

When selecting a model from a set of candidates, criteria such as the Akaike Information Criterion (AIC; Akaike 1998) and the Bayesian Information Criterion (BIC; Schwarz 1978) are commonly employed. Formally, the AIC and BIC are given by

$$\text{AIC} = -2l(\widehat{\boldsymbol{\vartheta}}) + 2q, \tag{23}$$

$$\text{BIC} = -2l(\widehat{\boldsymbol{\vartheta}}) + q \log n, \tag{24}$$

respectively, where $\widehat{\boldsymbol{\vartheta}}$ are the estimated parameter values at convergence of the proposed ECM algorithm, $l(\widehat{\boldsymbol{\vartheta}})$ is the associated observed-data log-likelihood, and $q$ is the number of free parameters in the model. The AIC and BIC serve as quantitative measures to balance model fit and complexity, helping to identify the most appropriate model. Consequently, when comparing models, lower values of the AIC or BIC indicate better model performance, with the choice between the two criteria often dependent on the specific context and theoretical considerations of the research problem. Many other criteria have also been proposed, and some studies have been conducted to compare these information criteria in model-based clustering of complete data sets; see, for example, Tran and Tortora (2021) and Tong and Tortora (2023) for the MSCN distribution and Akogul and Erisoglu (2016) for the mixture of normal distributions. In this paper, we consider both the AIC and BIC.

## 5 Applications

The proposed distributions offer distinct advantages in the context of clustering and directional outlier detection. Therefore, we compare the performance of the proposed techniques to other mixtures of contaminated distributions using both simulated (Sect. 5.1) and a real data set (Sect. 5.2). Specifically, we consider the following distributions: the MCN (Punzo and McNicholas 2016), the MMSCN (Punzo and Tortora 2021), and the MPTCN (Melnykov et al. 2021). All the analyses are performed in R. The MCN distributions are fitted using the R package **ContaminatedMixt** (Punzo et al. 2018), the MMSCN distributions are fitted using the R package **MSclust** (Tortora et al. 2024), and the MPTCN distributions are fitted using R code available from the authors upon request.

### 5.1 Simulation study

For each method, we measure the ability to recover the cluster partition using the adjusted Rand index (ARI; Hubert and Arabie 1985), which corrects the Rand index (Rand 1971) for chance and has an expected value of 0 under random partitions and an expected value of 1 under perfect agreement (for more information, see Steinley 2004). Since all the clustering techniques used in the simulation study also assign labels to any outlying points, the outliers are included in the ARI calculation too. In the scenarios when outliers are included in the data generation process, to assess outlier detection, we rely on the true positive rate (TPR), measuring the proportion of outliers correctly detected, and the false positive rate (FPR), measuring the proportion of good observations incorrectly detected as outliers.

#### 5.1.1 Simulation design

To assess the method's performance, we use three sample sizes: $n = 110$, $n = 550$ and $n = 1100$ with $\pi_1 = \pi_2 = 0.18$, $\pi_3 = 0.27$, and $\pi_4 = 0.37$. We then generate data from the proposed distributions and the competitors; see Appendix A for details

on data generation from the proposed distributions. Moreover, we consider three additional scenarios. In the first scenario, named GH/MSGH, three clusters are generated from a mixture of GH distributions and one cluster from an MSGH distribution. The code for generating data from GH and MSGH is available in the R package **MixGHD** (Tortora et al. 2021). The other two additional scenarios include outliers; specifically, in scenario NOT (normal outliers transformed), the clusters are generated from a multivariate normal distribution, 5% of the points are substituted with outliers, and then each cluster is transformed using the Manly transformation. In the last scenario, NTO (normal transformed outliers), instead, the clusters are generated from a multivariate normal distribution, each cluster is transformed using the Manly transformation, and then 5% of the points are substituted with outliers. For each scenario, we simulated 50 data sets. In Fig. 3, we provide an example of a data set generated from each scenario. Note that the acronym given in the title of each panel indicates the generating distribution.

In each panel of Fig. 3, the different colors and symbols represent the clusters of observations. In the last two panels, the squares indicate the outliers. The parameter sets used for each scenario are given in Appendix B.

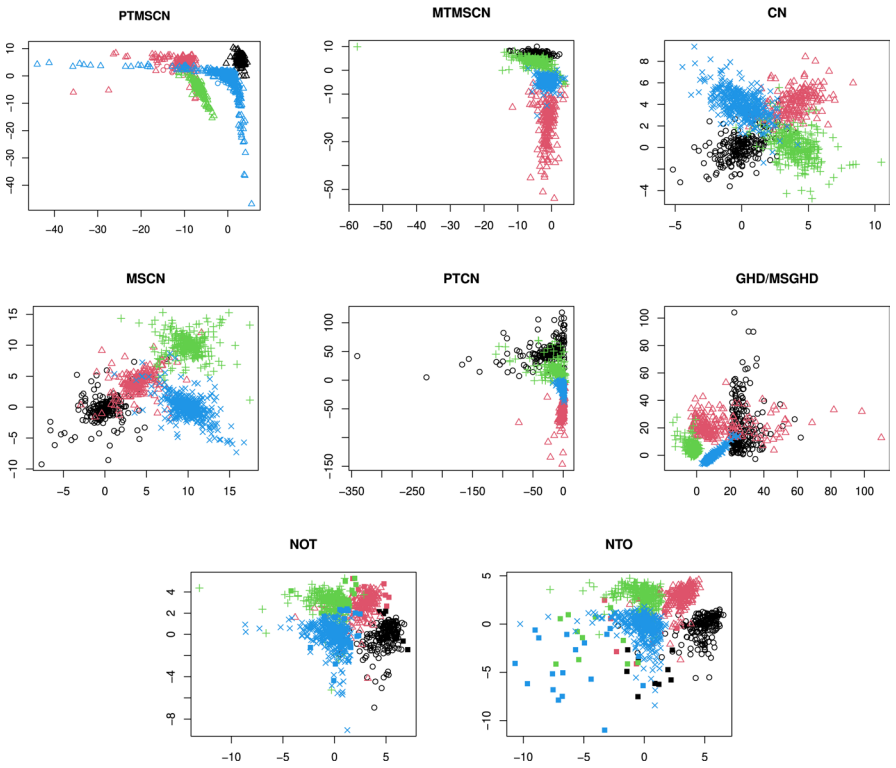



**Fig. 3** Examples of data sets obtained using the different data generation settings. Colors and symbols represent the clusters, and the squares in the NOT and NTO plots represent the outliers

### 5.1.2 Simulation results

For the sake of space, we only report the results for $n = 1100$; the results for $n = 110$ and $n = 550$ are shown in Appendices C and D, respectively. Table 1 shows the average ARI for each of the considered methods when fitted to the data sets simulated from each of the considered generating distributions. No single method outperforms the others, which shows the need for multiple techniques. As expected, all of the methods have similar performances on the data generated from the MCN and MMSCN distributions. When the clusters are skewed, the performance of methods based on symmetric clusters, that is, MCN and MMSCN, deteriorates, except for PTCN-generated data. MPTMSCN and MMTMSCN are similar, with MMTMSCN performing slightly better. On data generated from GH/MSGH, the methods based on the power transformation, i.e. the MPTMSCN and MPTCN, give the best average ARI values.

It is well documented that larger sample sizes are needed to accurately estimate the parameters governing tail behavior (see Punzo and Bagnato 2021; Tomarchio et al. 2022, 2020; Tortora et al. 2024). Through a comparison of the average ARI values given in Tables 1, 8, and 13, we can see that the average ARI values given in Table 8 are very similar for all of the considered methods. However, as the sample size increases, we start to see more differences in the reported ARI values. With respect to the proposed mixtures, this implies that smaller sample sizes make differences in tail behavior less evident, and therefore the models behave similarly for all of the considered generating distributions. This observation is also supported by the similarities in likelihood values and, therefore, by the corresponding AIC and BIC values.

Tables 2 and 3 show the proportion of times the AIC and BIC select the correct model across the 50 data sets generated for each scenario of interest. The indices perform similarly; they tend to select the model that was actually used to generate the data in most cases.

When the data are generated from the mixture of GH/MSGH and NTO, both the AIC and BIC select MPTMSCN distributions for 80% of the simulated data sets; in the NOT case, both indices always select the MMTMSCN distributions. This typically corresponds to a very high average ARI value.

**Table 1** Average ARI values for each considered method across the 50 data sets of size 1100 simulated from each generating component distribution. Bold-faced values represent the highest average ARI value for each scenario

| | Considered method | | | | |
|---|---|---|---|---|---|
| Generating Distribution | MPTMSCN | MMTMSCN | MCN | MMSCN | MPTCN |
| PTMSCN | **0.88** | 0.84 | 0.77 | 0.78 | **0.88** |
| MTMSCN | 0.65 | **0.67** | 0.65 | 0.66 | 0.66 |
| CN | 0.73 | 0.73 | **0.75** | 0.74 | 0.74 |
| MSCN | 0.81 | 0.85 | 0.84 | **0.86** | 0.84 |
| PTCN | 0.75 | 0.82 | **0.83** | 0.80 | 0.81 |
| GH/MSGH | 0.79 | 0.72 | 0.71 | 0.70 | **0.82** |
| NOT | 0.75 | **0.78** | 0.64 | 0.61 | 0.75 |
| NTO | 0.66 | **0.67** | 0.57 | 0.54 | 0.62 |

**Table 2** The proportion of times (in %) the AIC selects one of the considered methods across the 50 data sets of size 1100 simulated from each generating component distribution

| | Considered method | | | | |
|---|---|---|---|---|---|
| Generating Distribution | MPTMSCN | MMTMSCN | MCN | MMSCN | MPTCN |
| PTMSCN | 96 | 0 | 0 | 0 | 4 |
| MTMSCN | 0 | 100 | 0 | 0 | 0 |
| CN | 0 | 14 | 86 | 0 | 0 |
| MSCN | 0 | 18 | 0 | 82 | 0 |
| PTCN | 10 | 22 | 0 | 2 | 66 |
| GH/MSGH | 80 | 10 | 0 | 0 | 10 |
| NOT | 0 | 100 | 0 | 0 | 0 |
| NTO | 80 | 10 | 0 | 0 | 10 |

**Table 3** The proportion of times (in %) the BIC selects one of the considered methods across the 50 data sets of size 1100 simulated from each generating component distribution

| | Considered method | | | | |
|---|---|---|---|---|---|
| Generating Distribution | MPTMSCN | MMTMSCN | MCN | MMSCN | MPTCN |
| PTMSCN | 96 | 0 | 0 | 0 | 4 |
| MTMSCN | 0 | 100 | 0 | 0 | 0 |
| CN | 0 | 0 | 100 | 0 | 0 |
| MSCN | 0 | 18 | 0 | 82 | 0 |
| PTCN | 8 | 20 | 8 | 0 | 64 |
| GH/MSGH | 80 | 10 | 0 | 0 | 10 |
| NOT | 0 | 100 | 0 | 0 | 0 |
| NTO | 80 | 10 | 0 | 0 | 10 |

**Table 4** Average TPR and FPR for each considered method across the 50 data sets of size 1100 simulated from each generating component distribution. Bold-faced values represent the best-returned average TPR and FPR for the NOT and NT0 scenarios

| | | Considered method | | | | |
|---|---|---|---|---|---|---|
| | Generating Distribution | MPTMSCN | MMTMSCN | MCN | MMSCN | MPTCN |
| TPR | NOT | 0.20 | 0.14 | 0.35 | **0.42** | 0.30 |
| | NTO | 0.72 | 0.55 | 0.84 | **0.90** | 0.34 |
| FPR | NOT | 0.05 | **0.01** | 0.13 | 0.21 | 0.09 |
| | NTO | 0.07 | **0.05** | 0.11 | 0.21 | 0.10 |

As previously mentioned, smaller sample sizes make the task of estimating the tailedness parameters more difficult. This leads to more similar log-likelihood values across models, and, as a result, the considered model selection criteria penalize the methods with more parameters in favor of the simpler model. Comparing Table 3 with Tables 10 and 15, we can see that the BIC, which applies a bigger penalty than AIC (for $n > 7$), tends to select the most parsimonious model, the MCN, for the smallest considered sample size. The effect is less evident for the AIC (compare Tables 2, 9 and 14), which implies that AIC is better for smaller sample sizes.

When outliers have been introduced in the data sets, we also report the average TPR and FPR in Tables 4, 11, and 16. Notably, the results show that for the NOT scenario, where the outliers were added before transformation, it was more challenging

to detect these points compared to the NTO scenario, where the outliers were added after transformation. Again, we also observe the impact of changing sample size with respect to parameterizing the tails of the distributions. Specifically, the reported TPR values tend to decrease as the sample size decreases. For the largest considered sample size, the MMSCN distributions give the best TPR but the worst FPR, indicating that too many observations were flagged as outliers. A comparison of these results also shows that while the best-performing method in terms of FPR varies, all the transformation-based methods always have lower average FPR values. This implies that observations that belong to a skewed tail are likely being flagged as outliers when using the MCN or MMSCN distributions.

Notably, the MPTMSCN and MMTMSCN distributions tend to return lower average FPR values compared to the MCN, MMSCN, and MPTCN distributions. Combined with the reported average TPR values, these results imply that the proposed models effectively handle outliers without excessive detection or oversight.

Table 5 gives the number of parameters estimated for each model and the average time elapsed in seconds to run each model across 50 data sets per scenario. All the models share the parameters $\boldsymbol{\pi}$, $\boldsymbol{\mu}_g$, and $\boldsymbol{\Sigma}_g$, with $g = 1, \ldots, G$; obtaining $q^* = (G-1) + (G \times d) + (G \times d(d+1)/2)$ free model parameters. The MCN requires the estimation of two additional scalars for each component: $\alpha_g$ and $\eta_g$; this leads to $q_{\text{MCN}} = q^* + 2G$ free model parameters. The MMSCN requires the estimation of two additional $d$-dimensional vectors for each component: $\boldsymbol{\alpha}_g$ and $\boldsymbol{\eta}_g$; leading to $q_{\text{MMSCN}} = q^* + 2(G \times d)$. Compared to the MCN, the MPTCN requires the estimation of the $d$-dimensional vector $\boldsymbol{\lambda}_g$ for each component, giving $q_{\text{MPTCN}} = q_{\text{MCN}} + (G \times d)$ free model parameters. Compared to the MMSCN, the MPTMSCN and MMTMSCN also require the estimation of the $d$-dimensional $\boldsymbol{\lambda}_g$ for each component, leading to $q_{\text{MPTMSCN}} = q_{\text{MMTMSCN}} = q_{\text{MMSCN}} + (G \times d)$.

Table 5 shows that the MCN has the lowest average run time across all scenarios. Significant run-time increases are observed across all other models due to the estimation schemes for the component-wise orthogonal matrices, transformation parameters, or both. Interestingly, the MMTMSCN and MPTCN have similar overall average run times, with the MMTMSCN being more efficient than the MPTCN for

**Table 5** Average time elapsed (in seconds) to run each of the considered methods across the 50 data sets of size 1100 simulated from each generating component distribution. The first row shows the number of parameters ($q$) estimated per method, and the last row is the average run time across all scenarios

| | Considered method | | | | |
|---|---|---|---|---|---|
| Generating Distribution | MPTMSCN | MMTMSCN | MCN | MMSCN | MPTCN |
| $q$ | 47 | 47 | 31 | 39 | 39 |
| PTMSCN | 1108.29 | 211.01 | 2.08 | 119.94 | 287.42 |
| MTMSCN | 329.46 | 109.13 | 0.52 | 195.99 | 163.23 |
| CN | 291.49 | 99.44 | 0.51 | 174.92 | 136.28 |
| MSCN | 304.52 | 109.79 | 0.51 | 175.29 | 118.99 |
| PTCN | 156.76 | 73.71 | 0.45 | 194.29 | 72.97 |
| GHD/MSGHD | 210.64 | 98.33 | 0.49 | 191.73 | 68.77 |
| NOT | 237.55 | 107.08 | 0.50 | 198.17 | 60.36 |
| NTO | 204.13 | 106.87 | 0.51 | 187.34 | 49.02 |
| Average elapsed time | 303.92 | 102.42 | 0.60 | 280.79 | 116.19 |

**Table 6** Number of parameters ($q$) per method, AIC and BIC obtained from fitting all the considered methods to the wholesale data set. The bold-faced values are the lowest AIC and BIC values across all fitted models

| | MPTMSCN | MMTMSCN | MCN | MMSCN | MPTCN |
|---|---|---|---|---|---|
| $q$ | 91 | 91 | 59 | 79 | 71 |
| AIC | **2219.53** | 2375.84 | 3093.40 | 3083.33 | 2241.98 |
| BIC | 2542.38 | 2698.70 | 3334.52 | 3406.18 | **2532.14** |

**Table 7** The ARI values and number of misclassified observations (m) obtained fitting all of the considered methods to the wholesale data set. The bold-faced value is the best ARI

| | MPTMSCN | MMTMSCN | MCN | MMSCN | MPTCN |
|---|---|---|---|---|---|
| ARI | **0.55** | 0.38 | 0.25 | 0.41 | 0.49 |
| m | 56 | 85 | 109 | 89 | 66 |

the first four scenarios. Tables 12 and 17 give the average time elapsed in seconds when $n = 110$ and $n = 550$, respectively. For all methods, the average elapsed times tend to decrease as the value of $n$ reduces. Notably, when we halve $n$, the resulting decrease in elapsed time is more than half of the original duration. This suggests that the relationship between $n$ and elapsed time is not linear, with increasing reductions as $n$ becomes smaller. This shows that the number of observations greatly impacts the run times.

In summary, our results indicate that the difference between the classification performance of the considered methods and the performance of the AIC and BIC is more obvious for bigger sample sizes. The proposed novel transformation-based models tend to perform better when the data are skewed, as one would expect. The AIC and BIC can both be used for model selection, but we suggest using the AIC for small sample sizes.

### 5.2 Real data: the wholesale data set

The wholesale data set, available within the R package **tclust** (Fritz et al. 2012), comprises annual expenditures, measured in monetary units, on $d = 6$ product categories for $n = 440$ customers of a wholesale distributor in Portugal (Abreu et al. 2011). The categories include fresh, milk, grocery, frozen, detergent paper (DP), and delicatessen products. Additionally, the dataset contains two nominal variables: region (Lisboa, Porto, or other) and channel (hotel/restaurant/café or retail). While consumption patterns do not vary significantly across regions, they differ notably between channels. The goal of this analysis is to segment customers according to their spending patterns and to compare these segments with the channel variable. Accordingly, we set $G = 2$.

Table 6 shows the AIC, BIC, and number of parameters per method. In this data set, the methods based on the power transformation perform the best. The AIC selects the MPTMSCN, and the BIC selects the MPTCN. As remarked in the simulations study, the BIC tends to over-penalize; in fact, the method chosen using the AIC gives the best performance in terms of ARI, as shown in Table 7. The MPTCN gives the second-best ARI.

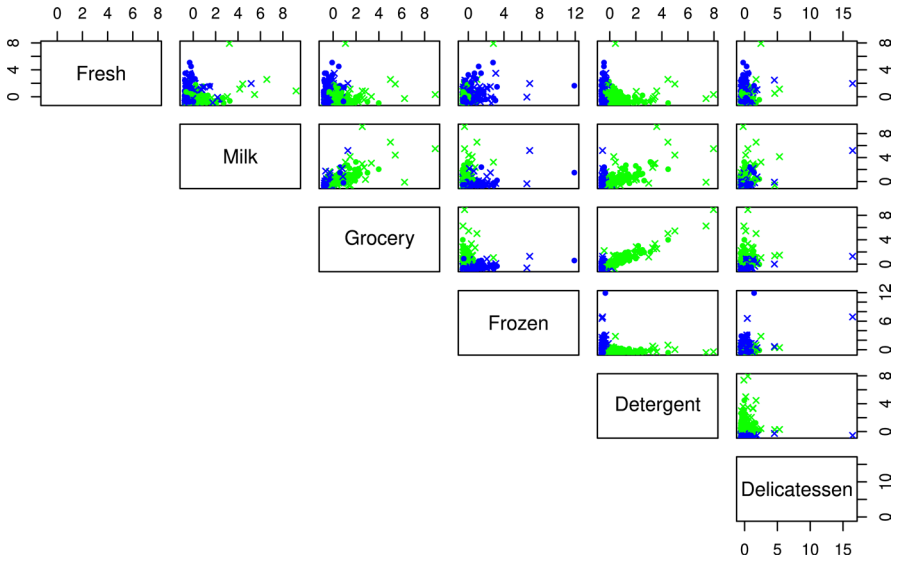



**Fig. 4** Scatter plot matrix for the wholesale data set, where the color represents the clusters and the + represents the outliers for at least one PC

Figure 4 shows the scatter plot matrix of the wholesale data. The color represents the clusters obtained using the MPTMSCN and the + represents the outliers for at least one PC. The blue cluster represents wholesale customers whose channel was Hotels, Restaurants, or Cafes, and the green cluster represents wholesale customers whose channel was Retail. The difference in annual monetary units spent is particularly evident in Milk, Grocery, and Detergent products, where wholesale customers whose channel was Retail spent more. Further, it is evident that the clusters are skewed and have many outliers.

## 6 Discussion

This paper presents two innovative modeling paradigms that consider transformations of the observed data within a mixture of multiple scaled contaminated normal (MSCN) distributions. Leveraging an eigen-decomposition of the scale matrix, the MSCN distribution offers a notable advantage: its adaptable tail behavior on the principal components. Further, the MSCN distribution performs outlier detection in the direction of the principal components of the data. Our modeling paradigms facilitate the identification of skewed clusters and perform directional outlier detection for groups of data with asymmetric features. We evaluated the efficacy of these methods using both simulated and real data sets.

A current limitation of the proposed method is its reliance on complete data, necessitating a prior imputation procedure if the data set contains missing values. This constraint could be addressed by adopting the procedures described in Tong and Tortora (2022) and Tong and Tortora (2023), which offers a viable solution to handle

missing data. In addition, the method does not account for the inclusion of covariates in the analysis, posing a second drawback. This limitation could be mitigated by implementing a similar approach to that proposed by Mazza and Punzo (2020). Incorporating covariates into the analysis could yield more nuanced insights and be more suitable for other diverse real-world scenarios. These extensions can significantly enhance our techniques' versatility and robustness, making them more effective tools for outlier detection and data analysis in practical settings.

## Appendix

### A. Simulating data from the MMTMSCN and MPTMSCN distributions

The inverse transformation can be used to pseudo-randomly generate data from the MMTMSCN and MPTMSCN distributions. Specifically, $\boldsymbol{Y}$ is generated form a mixture of CN distributions, then, within each cluster, the inverse Manly transformation can be obtained by defining

$$\mathscr{M}^{-1}\left(Y_h \mid \lambda_h\right) = \begin{cases} \lambda_h^{-1}\log(Y_h\lambda_h+1), & \lambda_h \neq 0 \\ Y_h, & \lambda_h = 0 \end{cases}, \tag{25}$$

and setting $\boldsymbol{X} = \left(\mathscr{M}^{-1}\left(Y_1 \mid \lambda_1\right), \ldots, \mathscr{M}^{-1}\left(Y_d \mid \lambda_d\right)\right)^\top$.

Similarly, the inverse power transform can be defined as

$$\mathscr{T}^{-1}\left(Y_h \mid \lambda_h\right) = \begin{cases} (Y_h\lambda_h+2)^{\frac{1}{\lambda_h}}-1, & \lambda_h \neq 0, Y_h \geq 0 \\ e^{Y_h}-1, & \lambda_h = 0, Y_h \geq 0 \\ 1-(1-Y_h(2-\lambda_h))^{\frac{1}{2-\lambda_h}}, & \lambda_h \neq 2, Y_h < 0 \\ 1-e^{-Y_h}, & \lambda_h = 2, Y_h < 0 \end{cases}. \tag{26}$$

We can then set $\boldsymbol{X} = \left(\mathscr{T}^{-1}\left(Y_1 \mid \lambda_1\right), \ldots, \mathscr{T}^{-1}\left(Y_d \mid \lambda_d\right)\right)^\top$.

### B. Parameters utilized in the simulation study

The parameter sets used to generate the data in each scenario considered in the simulation study of Sect. 5.1 are given below. Unless otherwise specified, we used

$$\boldsymbol{\Sigma}_1 = \boldsymbol{\Sigma}_2 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix} \text{ and } \boldsymbol{\Sigma}_3 = \boldsymbol{\Sigma}_4 = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}.$$

#### Mixture of PTMSCN distribution

$\boldsymbol{\lambda}_1 = (1.8, 1)$, $\boldsymbol{\mu}_1 = (5,5)^\top$, $\boldsymbol{\alpha}_1 = (0.7, 0.6)^\top$, $\boldsymbol{\eta}_1 = (2, 20)^\top$,

$\boldsymbol{\lambda}_2 = (1.6, 1.6)$, $\boldsymbol{\mu}_2 = (0, 0)^\top$, $\boldsymbol{\alpha}_2 = (0.9, 0.6)^\top$, $\boldsymbol{\eta}_2 = (10, 20)^\top$, then add $(-10, 5)^\top$ to every $\boldsymbol{x}_i$
$\boldsymbol{\lambda}_3 = (1.6, 1)$, $\boldsymbol{\mu}_3 = (5, 5)^\top$, $\boldsymbol{\alpha}_3 = (0.6, 0.9)^\top$, $\boldsymbol{\eta}_3 = (20, 2)^\top$, then add -10 to every value
$\boldsymbol{\lambda}_4 = (1.6, 1.6)$, $\boldsymbol{\mu}_4 = (0, 0)^\top$, $\boldsymbol{\alpha}_4 = (0.6, 0.8)^\top$, $\boldsymbol{\eta}_4 = (20, 2)^\top$.

#### Mixture of MTMSCN distribution

$\boldsymbol{\lambda}_1 = (0.06, 0.1)$, $\boldsymbol{\mu}_1 = (-4, 10)^\top$, $\boldsymbol{\alpha}_1 = (0.6, 0.6)^\top$, $\boldsymbol{\eta}_1 = (2, 10)^\top$,
$\boldsymbol{\lambda}_2 = (0.06, 0.08)$, $\boldsymbol{\mu}_2 = (-1, -10)^\top$, $\boldsymbol{\alpha}_2 = (0.9, 0.8)^\top$, $\boldsymbol{\eta}_2 = (10, 20)^\top$,
$\boldsymbol{\lambda}_3 = (0.06, 0.06)$, $\boldsymbol{\mu}_3 = (-4, 4)^\top$, $\boldsymbol{\alpha}_3 = (0.6, 0.6)^\top$, $\boldsymbol{\eta}_3 = (20, 10)^\top$,
$\boldsymbol{\lambda}_4 = (0.05, 0.06)$, $\boldsymbol{\mu}_4 = (-1, -4)^\top$, $\boldsymbol{\alpha}_4 = (0.9, 0.8)^\top$, $\boldsymbol{\eta}_4 = (4, 10)^\top$.

#### Mixture of CN disttributions

$\boldsymbol{\mu}_1 = (0, 0)^\top$, $\alpha_1 = 0.8$, $\eta_1 = 4$,
$\boldsymbol{\mu}_2 = (4, 4)^\top$, $\alpha_2 = 0.7$, $\eta_2 = 3$,
$\boldsymbol{\mu}_3 = (4, 0)^\top$, $\alpha_3 = 0.6$, $\eta_3 = 5$,
$\boldsymbol{\mu}_4 = (0, 4)^\top$, $\alpha_4 = 0.7$, $\eta_4 = 3$.

#### Mixture of MSCN distributions

$\boldsymbol{\mu}_1 = (0, 0^\top)$, $\boldsymbol{\alpha}_1 = (0.8, 0.6)^\top$, $\boldsymbol{\eta}_1 = (13, 14)^\top$,
$\boldsymbol{\mu}_2 = (4, 4)^\top$, $\boldsymbol{\alpha}_2 = (0.7, 0.7)^\top$, $\boldsymbol{\eta}_2 = (12, 20)^\top$,
$\boldsymbol{\mu}_3 = (10, 10)^\top$, $\boldsymbol{\alpha}_3 = (0.8, 0.6)^\top$, $\boldsymbol{\eta}_3 = (9, 14)^\top$,
$\boldsymbol{\mu}_4 = (10, 0)^\top$, $\boldsymbol{\alpha}_4 = (0.7, 0.7)^\top$, $\boldsymbol{\eta}_4 = (12, 10)^\top$.

#### Mixture of PTCN

$\boldsymbol{\lambda}_1 = (1.8, 0.5)$, $\boldsymbol{\mu}_1 = (-4, 10)^\top$, $\alpha_1 = 0.6$, $\eta_1 = 10$,
$\boldsymbol{\lambda}_2 = (1.6, 1.6)$, $\boldsymbol{\mu}_2 = (-1, -10)^\top$, $\alpha_2 = 0.9$, $\eta_2 = 20$,
$\boldsymbol{\lambda}_3 = (1.6, 0.5)$, $\boldsymbol{\mu}_3 = (-4, 4)^\top$, $\alpha_3 = 0.6$, $\eta_3 = 20$,
$\boldsymbol{\lambda}_4 = (1.6, 1.6)$, $\boldsymbol{\mu}_4 = (-1, -4)^\top$, $\alpha_4 = 0.9$, $\eta_4 = 4$.

#### Mixtures of GH distributions and MSGH distributions

Cluster 1 was obtained by generating from a GH distribution with diagonal matrix $\boldsymbol{\Sigma}$ and parameters: $\boldsymbol{\mu}_1 = (20, 0)^\top$, $\boldsymbol{\alpha}_1 = (4, 10)^\top$, $\sigma_{11} = 1$, $\sigma_{12} = 5$, $\omega_1 = 1$, and $\lambda_1 = 0.5$.

For clusters 2 to 4:

$$\boldsymbol{\Sigma}_2 = \boldsymbol{\Sigma}_3 = \boldsymbol{\Sigma}_4 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}.$$

Cluster 2 was obtained generating from an MSGH distribution with parameters: $\boldsymbol{\mu}_2 = (-5, 10)^\top$, $\boldsymbol{\alpha}_2 = (10, 7)^\top$, $\boldsymbol{\omega_2} = (1, 2)$, $\boldsymbol{\lambda}_2 = (0.5, 0.8)$. Cluster 3 and 4

were obtained generating from a GH distribution with parameters: $\boldsymbol{\mu}_3 = (0,0)^\top$, $\boldsymbol{\alpha}_3 = (-1,2)^\top, \omega_3 = 1, \lambda_3 = 0.5, \boldsymbol{\mu}_4 = (5,-5)^\top, \boldsymbol{\alpha}_4 = (2,2)^\top, \omega_4 = 1, \lambda_4 = 0.5$.

#### Mixture of normal distributions with outliers Manly transformed (NOT)

$\boldsymbol{\lambda}_1 = (0.6, 0.6)$, $\boldsymbol{\mu}_1 = (0,0)^\top$, then added $(5,0)^\top$ to every $\boldsymbol{x}_i$
$\boldsymbol{\lambda}_2 = (0.6, 0.6)$, $\boldsymbol{\mu}_2 = (0,0)^\top$, then added $(3,3)^\top$ to every $\boldsymbol{x}_i$
$\boldsymbol{\lambda}_3 = (0.6, 0.6)$, $\boldsymbol{\mu}_3 = (0,0)^\top$, then added $(0,3)^\top$ to every $\boldsymbol{x}_i$
$\boldsymbol{\lambda}_4 = (0.5, 0.5)$, $\boldsymbol{\mu}_4 = (0,0)^\top$.

5% of the points in each cluster have been substituted with points generated uniformly between -5 and 5 before transformation.

#### Mixture of normal distributions Manly transformed with outliers (NTO)

$\boldsymbol{\lambda}_1 = (0.6, 0.6)$, $\boldsymbol{\mu}_1 = (0,0)^\top$, then added $(5,0)^\top$ to every $\boldsymbol{x}_i$
$\boldsymbol{\lambda}_2 = (0.6, 0.6)$, $\boldsymbol{\mu}_2 = (0,0)^\top$, then added $(3,3)^\top$ to every $\boldsymbol{x}_i$
$\boldsymbol{\lambda}_3 = (0.6, 0.6)$, $\boldsymbol{\mu}_3 = (0,0)^\top$, then added $(0,3)^\top$ to every $\boldsymbol{x}_i$
$\boldsymbol{\lambda}_4 = (0.5, 0.5)$, $\boldsymbol{\mu}_4 = (0,0)^\top$.

5% of the points in each cluster have been substituted with points generated uniformly between $\min(\boldsymbol{X}) + 0.1\min(\boldsymbol{X})$ and 0 after transformation.

## C. Results for the considered methods when $n = 110$

This appendix gives the average ARI, TPR, FPR, and elapsed run times (in seconds) and percentage of times AIC and BIC select the method for each of the considered methods fitted to 50 data sets of size 110 simulated from the generating distributions described in Sect. 5 and Appendix B. Tables 8, 9, 10, 11, 12.

**Table 8** Average ARI values for each considered method across the 50 data sets of size $n = 110$ simulated from each generating component distribution. Bold-faced values represent the highest average ARI value for each scenario. Note that the row gives the generating distribution

| | Considered method | | | | |
|---|---|---|---|---|---|
| Generating Distribution | MPTMSCN | MMTMSCN | MCN | MMSCN | MPTCN |
| PTMSCN | 0.79 | 0.84 | 0.81 | 0.79 | **0.81** |
| MTMSCN | 0.65 | 0.65 | **0.66** | 0.65 | 0.65 |
| CN | 0.66 | 0.66 | **0.70** | 0.66 | 0.66 |
| MSCN | 0.79 | 0.79 | **0.81** | 0.78 | 0.77 |
| PTCN | 0.70 | 0.80 | **0.81** | 0.80 | 0.75 |
| GHD | **0.76** | 0.74 | 0.73 | 0.74 | **0.76** |
| NOT | **0.71** | **0.71** | 0.64 | 0.61 | 0.68 |
| NTO | 0.64 | **0.66** | 0.56 | 0.55 | 0.61 |

**Table 9** The proportion of times (in %) the AIC selects one of the considered methods across the 50 data sets of size $n = 110$ simulated from each generating component distribution

| | Considered method | | | | |
|---|---|---|---|---|---|
| Generating distribution | MPTMSCN | MMTMSCN | MCN | MMSCN | MPTCN |
| PTMSCN | 38 | 36 | 14 | 2 | 10 |
| MTMSCN | 9 | 36 | 45 | 6 | 4 |
| CN | 4 | 10 | 86 | 0 | 0 |
| MSCN | 12 | 20 | 34 | 30 | 4 |
| PTCN | 21 | 12 | 50 | 2 | 15 |
| GHD | 80 | 2 | 4 | 2 | 12 |
| NOT | 26 | 18 | 40 | 8 | 8 |
| NTO | 16 | 14 | 60 | 4 | 6 |

**Table 10** The proportion of times (in %) the BIC selects one of the considered methods across the 50 data sets of size $n = 110$ simulated from each generating component distribution

| | Considered method | | | | |
|---|---|---|---|---|---|
| Generating distribution | MPTMSCN | MMTMSCN | MCN | MMSCN | MPTCN |
| PTMSCN | 18 | 8 | 70 | 0 | 4 |
| MTMSCN | 0 | 0 | 96 | 4 | 0 |
| CN | 0 | 0 | 100 | 0 | 0 |
| MSCN | 0 | 0 | 96 | 2 | 2 |
| PTCN | 4 | 2 | 92 | 0 | 2 |
| GHD | 10 | 0 | 86 | 2 | 2 |
| NOT | 0 | 0 | 100 | 0 | 0 |
| NTO | 0 | 0 | 100 | 0 | 0 |

**Table 11** Average TPR and FPR for each considered method across the 50 data sets of size $n = 110$ simulated from each generating distribution. Bold-faced values represent the best returned average TPR and FPR for the NOT and NT0 scenarios

| | | Considered method | | | | |
|---|---|---|---|---|---|---|
| | Generating distribution | MPTMSCN | MMTMSCN | MCN | MMSCN | MPTCN |
| TPR | NOT | 0.07 | 0.11 | 0.24 | **0.27** | 0.03 |
| | NTO | 0.15 | 0.21 | **0.61** | 0.59 | 0.22 |
| FPR | NOT | 0.02 | 0.02 | 0.11 | 0.14 | **0.01** |
| | NTO | **0.02** | 0.04 | 0.11 | 0.15 | 0.03 |

**Table 12** Average time elapsed (in seconds) to run each of the considered methods across the 50 data sets of size $n = 110$ simulated from each generating component distribution. The first row shows the number of parameters ($q$) estimated per method, and the last row is the average run time across all scenarios

| | Considered method | | | | |
|---|---|---|---|---|---|
| Generating distribution | MPTMSCN | MMTMSCN | MCN | MMSCN | MPTCN |
| $q$ | 47 | 47 | 31 | 39 | 39 |
| PTMSCN | 40.83 | 12.25 | 0.25 | 4.31 | 14.41 |
| MTMSCN | 30.27 | 10.51 | 0.07 | 15.92 | 16.97 |
| CN | 12.76 | 10.07 | 0.08 | 16.53 | 6.03 |
| MSCN | 19.55 | 8.61 | 0.07 | 17.36 | 7.02 |
| PTCN | 24.84 | 9.40 | 0.07 | 16.40 | 8.73 |
| GHD | 14.84 | 7.93 | 0.07 | 16.92 | 5.54 |
| NOT | 9.57 | 6.95 | 0.07 | 16.98 | 2.93 |
| NTO | 8.23 | 8.31 | 0.07 | 14.86 | 3.13 |
| Average elapsed time | 17.76 | 9.05 | 0.07 | 16.22 | 7.21 |

## D. Results for the considered methods when $n = 550$

This appendix gives the average ARI, TPR, FPR, and elapsed run times (in seconds) and percentage of times AIC and BIC select the method for each of the considered methods fitted to 50 data sets of size 550 simulated from the generating distributions described in Sect. 5 and Appendix B. Tables 13, 14, 15, 16, 17.

**Table 13** Average ARI values for each considered method across the 50 data sets of size $n = 550$ simulated from each generating distribution. Bold-faced values represent the highest average ARI value for each scenario. Note that the row gives the generating component distribution

| | Considered method | | | | |
|---|---|---|---|---|---|
| Generating distribution | MPTMSCN | MMTMSCN | MCN | MMSCN | MPTCN |
| PTMSCN | **0.87** | 0.87 | 0.82 | 0.79 | **0.87** |
| MTMSCN | 0.66 | 0.66 | 0.65 | 0.66 | **0.67** |
| CN | 0.72 | 0.72 | **0.75** | 0.74 | 0.71 |
| MSCN | 0.81 | 0.84 | 0.84 | **0.85** | 0.83 |
| PTCN | 0.76 | 0.82 | **0.84** | 0.81 | 0.81 |
| GH/MSGH | 0.78 | 0.71 | 0.70 | 0.69 | **0.80** |
| NOT | 0.74 | **0.76** | 0.63 | 0.59 | 0.72 |
| NTO | 0.68 | **0.69** | 0.58 | 0.56 | 0.64 |

**Table 14** The proportion of times (in %) the AIC selects one of the considered methods across the 50 data sets of size $n = 550$ simulated from each generating component distribution

| | Considered method | | | | |
|---|---|---|---|---|---|
| Generating distribution | MPTMSCN | MMTMSCN | MCN | MMSCN | MPTCN |
| PTMSCN | 84 | 12 | 0 | 0 | 4 |
| MTMSCN | 2 | 88 | 0 | 6 | 4 |
| CN | 4 | 10 | 86 | 0 | 0 |
| MSCN | 2 | 32 | 0 | 66 | 0 |
| PTCN | 2 | 6 | 0 | 0 | 92 |
| GH/MSGH | 76 | 0 | 0 | 0 | 24 |
| NOT | 4 | 94 | 0 | 0 | 2 |
| NTO | 56 | 30 | 0 | 0 | 14 |

**Table 15** The proportion of times (in %) the BIC selects one of the considered methods across the 50 data sets of size $n = 550$ simulated from each generating component distribution

| | Considered method | | | | |
|---|---|---|---|---|---|
| Generating distribution | MPTMSCN | MMTMSCN | MCN | MMSCN | MPTCN |
| PTMSCN | 84 | 12 | 0 | 0 | 4 |
| MTMSCN | 0 | 64 | 32 | 2 | 2 |
| CN | 0 | 0 | 100 | 0 | 0 |
| MSCN | 2 | 18 | 34 | 46 | 0 |
| PTCN | 2 | 4 | 12 | 0 | 82 |
| GH/MSGH | 76 | 0 | 0 | 0 | 24 |
| NOT | 4 | 76 | 18 | 0 | 2 |
| NTO | 40 | 22 | 32 | 0 | 6 |

**Table 16** Average TPR and FPR for each considered method across the 50 data sets of size $n = 550$ simulated from each generating distribution. Bold-faced values represent the best returned average TPR and FPR for the NOT and NT0 scenarios

| | | Considered method | | | | |
|---|---|---|---|---|---|---|
| | Generating distribution | MPTMSCN | MMTMSCN | MCN | MMSCN | MPTCN |
| TPR | NOT | 0.15 | **0.41** | 0.16 | 0.34 | 0.14 |
| | NTO | 0.59 | 0.77 | **0.83** | 0.46 | 0.72 |
| FPR | NOT | 0.04 | **0.02** | 0.13 | 0.22 | 0.05 |
| | NTO | 0.07 | **0.04** | 0.11 | 0.20 | 0.09 |

**Table 17** Average time elapsed (in seconds) to run each of the considered methods across the 50 data sets of size $n = 550$ simulated from each generating component distribution. The first row shows the number of parameters ($q$) estimated per method, and the last row is the average run time across all scenarios

| | Considered method | | | | |
|---|---|---|---|---|---|
| Generating distribution | MPTMSCN | MMTMSCN | MCN | MMSCN | MPTCN |
| $q$ | 47 | 47 | 31 | 39 | 39 |
| PTMSCN | 442.53 | 113.82 | 1.23 | 38.78 | 95.93 |
| MTMSCN | 142.17 | 46.09 | 0.26 | 95.54 | 73.41 |
| CN | 124.20 | 50.11 | 0.26 | 84.66 | 56.16 |
| MSCN | 130.27 | 52.59 | 0.26 | 84.67 | 59.77 |
| PTCN | 96.62 | 53.57 | 0.24 | 95.65 | 31.89 |
| GH/MSGH | 99.30 | 51.84 | 0.26 | 88.55 | 28.33 |
| NOT | 100.02 | 50.98 | 0.25 | 93.21 | 27.51 |
| NTO | 100.14 | 50.89 | 0.25 | 88.26 | 22.87 |
| Average elapsed time | 109.36 | 51.29 | 0.25 | 89.52 | 41.16 |

**Acknowledgements** This work was supported by *NSF grant No. 2209974* (Tortora), and by a Discovery Grant from the Natural Sciences and Engineering Research Council of Canada (Franczak, No. RGPIN-2017-04676), and by the Italian Ministry of University and Research (MUR) under the PRIN 2022 grant No.2022XRHT8R (CUP: E53D23005950006), as part of 'The SMILE Project: Statistical Modelling and Inference to Live the Environment', funded by the European Union – Next Generation EU (Punzo).



## References

Abreu NGCFM d et al (2011) Analise do perfil do cliente recheio e desenvolvimento de um sistema promocional. Master's thesis

Aitken A (1926) A series formula for the roots of algebraic and transcendental equations. Proc R Soc Edinburgh 45(1):14–22

Akaike H (1998) Information theory and an extension of the maximum likelihood principle. In: Parzen E, Tanabe K, Kitagawa G (eds) Selected papers of Hirotugu Akaike. Springer, New York, pp 199–213

Akogul S, Erisoglu M (2016) A comparison of information criteria in clustering based on mixture of multivariate normal distributions. Math Comput Appl 21(3):34

Azzalini A (2005) The skew-normal distribution and related multivariate families. Scand J Stat 32(2):159–188

Azzalini A, Capitanio A (2003) Distributions generated by perturbation of symmetry with emphasis on a multivariate skew t-distribution. J R Stat Soc Ser B: Stat Methodol 65(2):367–389

Bagnato L, Punzo A, Zoia MG (2017) The multivariate leptokurtic-normal distribution and its application in model-based clustering. Can J Stat 45(1):95–119

Barndorff-Nielsen O, Kent J, Sørensen M (1982) Normal variance-mean mixtures and z distributions. Int Stat Rev / Revue Internationale de Statistique 50(2):145–159

Biernacki C, Celeux G, Govaert G (2000) Assessing a mixture model for clustering with the integrated completed likelihood. IEEE Trans Pattern Anal Mach Intell 22(7):719–725
Box GE, Cox DR (1964) An analysis of transformations. J R Stat Soc Ser B: Stat Methodol 26(2):211–243
Browne RP, McNicholas PD (2015) A mixture of generalized hyperbolic distributions. Can J Stat 43(2):176–198
Cuesta-Albertos J, Matrán C, Mayo-Iscar A (2008) Robust estimation in the normal mixture model based on robust clustering. J R Stat Soc Ser B: Stat Methodol 70(4):779–802
Davies L, Gather U (1993) The identification of multiple outliers. J Am Stat Assoc 88(423):782–792
Davila VH L, Cabral CRB, Zeller CB (2018) Finite Mixture of Skewed Distributions. Springer Publishing Company, Incorporated, 1st edition, Berlin
Dempster AP, Laird NM, Rubin DB (1977) Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc Ser B 39(1):1–38
Forbes F, Wraith D (2014) A new family of multivariate heavy-tailed distributions with variable marginal amounts of tailweights: application to robust clustering. Stat Comput 24(6):971–984
Franczak B, Browne RP, McNicholas PD (2014) Mixtures of shifted asymmetric laplace distributions. IEEE Trans Pattern Anal Mach Intell 36(6):1149–1157
Franczak BC, Tortora C, Browne RP, McNicholas PD (2015) Unsupervised learning via mixtures of skewed distributions with hypercube contours. Pattern Recognit Lett 58:69–76
Fritz H, Garcia-Escudero LA, Mayo-Iscar A (2012) tclust: an R package for a trimming approach to cluster analysis. J Stat Softw 47(12):1–26
Hubert L, Arabie P (1985) Comparing partitions. J Classif 2(1):193–218
Karlis D, Xekalaki E (2003) Choosing initial values for the EM algorithm for finite mixtures. Comput Stat Data Anal 41:577–590
Kaufman L, Rousseeuw PJ editors (1990) Finding groups in data: an introduction to cluster analysis. Wiley Series in Probability and Statistics. Wiley, Hoboken
Lee SX, McLachlan GJ (2016) Finite mixtures of canonical fundamental skew t-distributions: the unification of the restricted and unrestricted skew t-mixture models. Stat Comput 26(3):573–589
Lin T-I (2009) Maximum likelihood estimation for multivariate skew normal mixture models. J Multivar Anal 100:257–265
Lin T-I (2010) Robust mixture modeling using multivariate skew t distributions. Stat Comput 20(3):343–356
Mahdavi A, Desmond AF, Jamalizadeh A, Lin T-I (2024) Skew multiple scaled mixtures of normal distributions with flexible tail behavior and their application to clustering. J Classif 41:620–649
Manly BF (1976) Exponential data transformations. J R Stat Soc Ser D: Stat 25(1):37–42
Mazza A, Punzo A (2020) Mixtures of multivariate contaminated normal regression models. Stat Pap 61(2):787–822
McLachlan GJ, Krishnan T (2008) The EM algorithm and extensions. Wiley Interscience, New York
McNicholas PD (2016) Mixture model-based classification. Chapman & Hall/CRC Press, Boca Raton
McNicholas P, Murphy T, McDaid A, Frost D (2010) Serial and parallel implementations of model-based clustering via parsimonious Gaussian mixture models. Second Special Issue Stat Algorithm Softw 54(3):711–723
McNicholas S, McNicholas PD, Browne RP (2017) A mixture of variance-gamma factor analyzers. Springer International Publishing, Cham
Melnykov Y, Zhu X, Melnykov V (2021) Transformation mixture modeling for skewed data groups with heavy tails and scatter. Comput Stat 36:61–78
Meng X-L, Rubin DB (1993) Maximum likelihood estimation via the ECM algorithm: a general framework. Biometrika 80(2):267–278
Michael S, Melnykov V (2016) An effective strategy for initializing the em algorithm in finite mixture models. Adv Data Anal Classif 10:563–583
Morris K, Punzo A, Blostein M, McNicholas PD (2019) Asymmetric clusters and outliers: mixtures of multivariate contaminated shifted asymmetric Laplace distributions. Comput Stat Data Anal 132:145–166
Nelder JA, Mead R (1965) A simplex method for function minimization. Comput J 7(4):308–313
Peel D, McLachlan GJ (2000) Robust mixture modelling using the t distribution. Stat Comput 10:339–348
Punzo A, Bagnato L (2021) The multivariate tail-inflated normal distribution and its application in finance. J Stat Comput Simul 91(1):1–36
Punzo A, Bagnato L (2022) Multiple scaled symmetric distributions in allometric studies. Int J Biostat 18(1):219–242

Punzo A, McNicholas PD (2016) Parsimonious mixtures of multivariate contaminated normal distributions. Biom J 58(6):1506–1537
Punzo A, Tortora C (2021) Multiple scaled contaminated normal distribution and its application in clustering. Stat Model 21(4):332–358
Punzo A, Mazza A, McNicholas PD (2018) ContaminatedMixt: an R package for fitting parsimonious mixtures of multivariate contaminated normal distributions. J Stat Softw 85(10):1–25
Rand WM (1971) Objective criteria for the evaluation of clustering methods. J Am Stat Assoc 66(336):846–850
Schwarz G (1978) Estimating the dimension of a model. Annals Stat 6(2):461
Shireman E, Steinley D, Brusco MJ (2017) Examining the effect of initialization strategies on the performance of Gaussian mixture modeling. Behav Res Methods 49(1):282–293
Steinley D (2004) Properties of the Hubert-Arabie adjusted Rand index. Psychol Methods 9(3):386–396
Tomarchio SD, Punzo A, Bagnato L (2020) Two new matrix-variate distributions with application in model-based clustering. Comput Stat Data Anal 152:107050
Tomarchio SD, Bagnato L, Punzo A (2022) Model-based clustering via new parsimonious mixtures of heavy-tailed distributions. AStA Adv Stat Anal 106:315–347
Tong H, Tortora C (2022) Model-based clustering and outlier detection with missing data. Adv Data Anal Classif 16(1):5–30
Tong H, Tortora C (2023) Missing values and directional outlier detection in model-based clustering. J Classif 41:480–513
Tortora C, Franczak BC, Browne RP, McNicholas PD (2019) A mixture of coalesced generalized hyperbolic distributions. J Classif 36:26–57
Tortora C, Browne RP, El Sherbiny A, Franczak BC, McNicholas PD (2021) Model-based clustering, classification, and discriminant analysis using the generalized hyperbolic distribution: MixGHD R package. J Stat Softw 98(3):1–24
Tortora C, Franczak BC, Bagnato L, Punzo A (2024) A Laplace-based model with flexible tail behavior. Comput Stat Data Anal 192:107909
Tortora C, Punzo A, Tran L (2024) MSclust: multiple-scaled clustering. R Package Version 1:4
Tran L, Tortora C (2021) How many clusters are best? Investigating model selection in robust clustering. In JSM Proceedings, Statistical Learning and Data Science Section. American Statistical Association. Alexandria
Tukey JW (1960) A survey of sampling from contaminated distributions. In: Olkin I, Ghurye SG, Hoeffding W, Madow WG, Mann HB (eds) Contributions to probability and statistics: essays in honor of Harold hotelling. Stanford University Press, Stanford, pp 448–485
Wraith D, Forbes F (2015) Location and scale mixtures of Gaussians with flexible tail behaviour: properties, inference and application to multivariate clustering. Comput Stat Data Anal 90:61–73
You J, Li Z, Du J (2023) A new iterative initialization of em algorithm for gaussian mixture models. Plos One 18(4):e0284114
Zhu X, Melnykov V (2018) Manly transformation in finite mixture modeling. Comput Stat Data Anal 121:190–208