Bayesian clustering of skewed and multimodal data using geometric skewed normal distributions

Bayesian Clustering of skewed and multimodal data using geometric skewed normal distributions

Edoardo Redivo*[1,3], Hien Nguyen†[2], and Mayetri Gupta‡[3]

[1]School of Economics, Management, and Statistics, Università di Bologna

[2]Department of Mathematics and Statistics, La Trobe University

[3]School of Mathematics and Statistics, University of Glasgow

June 17, 2020

Abstract

Model-based clustering approaches generally assume that the observations to be clustered are generated from a mixture of distributions, each component of the mixture corresponding to a particular parametric distribution. Most commonly, the underlying distribution is assumed to be normal, which is inadequate for many situations, for example when skewness or multimodality is present within the components. The problem is intensified when the data dimension increases, leading to inaccurate groupings and incorrect inference. A new Bayesian model-based clustering approach is proposed, that can handle a variety of complexities in the data, based on a recently introduced family of geometric skew normal distributions. The performance of this methodology is illustrated through a number of simulation studies and applications to a number of datasets from genomics and medicine.

Keywords: Model-based clustering; Markov Chain Monte Carlo; Mixtures of distributions; Genome-wide Association Studies; Image Segmentation.

*edoardoredivo@gmail.com

†h.nguyen5@latrobe.edu.au

‡mayetri.gupta@glasgow.ac.uk

1 Introduction

Clustering is a widely used tool in exploratory data analysis, which is increasingly being applied to datasets of massive size and complexity in a variety of applications, ranging from the fields of biomedicine and genetics to those of demographics, sports, environmental and social media studies. Most model-based techniques approach clustering through finite parametric mixtures of distributions (McLachlan and Peel, 2000), which are commonly based on the Gaussian mixture model and its variations (Fraley and Raftery, 2002).

In many applications, however, the densities of individual clusters cannot be accurately modelled by symmetric or unimodal distributions, motivating efforts to move beyond the restrictiveness of the normal model. A multivariate t-distribution can be represented as a multivariate Gaussian scale-mixture model, that leads to a natural extension of the normal mixture to a t-mixture that can allow for heavier-tailed distributions in clustering (Andrews and McNicholas, 2012; McLachlan and Peel, 2000). With heterogeneous data exhibiting asymmetric features, skewed extensions to the elliptical distributions, such as the multivariate skewed normal (MSN) and skew-t (MST) distributions, have been developed (Azzalini, 2005). Mixtures of these distributions have proved promising in the context of clustering (Fruhwirth-Schnatter and Pyne, 2010; Vrbik and McNicholas, 2014). The MSN distribution can be generated by conditioning a multivariate normal variable (say Y_1) on another univariate or multivariate normal variable (say Y_0); varying the form or dimensions of Y_1 and Y_0, lead to a number of variants of the MSN. A unifying framework for a variety of MSN and MST distributions has been presented in Lee and McLachlan (2013). One attraction behind using the MST in clustering is its flexibility in allowing for skewness and kurtosis, while retaining mathematical tractability, so that methods such as the EM algorithm (Dempster et al., 1977), Monte Carlo, or straightforward Markov chain Monte Carlo approaches (Fruhwirth-Schnatter and Pyne, 2010) can be used.

Multivariate heavy-tailed distributions are not limited to the t-distribution. For example, Forbes and Wraith (2014) explored the scale mixture representation by introducing multidimensional weights and a decomposition of the Gaussian covariance matrix, which allows for arbitrary correlations between dimensions. The family of distributions they define includes the generalized t-distribution and the Pearson type-VII family. Other approaches include Browne and McNicholas (2015), who used mixtures of generalized hyperbolic distributions, and O'Hagan et al. (2016), who employed mixtures of multivariate normal inverse Gaussian distributions, to allow for skewness and heavy-tails in the data.

The flexibility of mixture distributions to model a wide variety of data scenarios has also led to the ap-

proach of modelling a mixture of mixtures, where the non-Gaussian cluster distributions themselves are allowed to be multimodal, and modelled by Gaussian mixtures (Jordan and Jacobs, 1994). Estimation of mixture-of-mixtures models is challenging due to identifiability issues, since exchanging subcomponents between clusters on a lower level gives different cluster distributions, while the density of the overall mixture remains the same. Identifiability can be ensured by additional information, such as placing strong constraints on the locations and covariance matrices of the components (e.g. Zio et al., 2007). Some others (Li, 2005; Hennig, 2010) take a different approach, where a standard Gaussian mixture is first fitted to the data, then sub-clusters determined by successively merging components according to a criterion. However, Malsiner-Walli et al. (2017) show that such two-step approaches may tend to miss the cluster structure. They instead propose to make the mixture-of-mixtures models identifiable within a Bayesian framework through a hierarchical prior construction; the choice of hyperparameters is guided by a variance decomposition of the data. This is closely related to hierarchical Bayesian nonparametric approaches based on Dirichlet process mixtures (Escobar and West, 1995; Teh et al., 2006) and species mixture models (Argiento et al., 2014). The mixture of Gaussian mixtures approach would be attractive for applications, where clusters are not necessarily homogeneous, or the data contains a number of outliers, and has the potential for better interpretability than an overfitted mixture. This model however, being completely based on normal distributions, somewhat lacks the flexibility of allowing for skewness and heavy tails in the components.

In a variety of data sets arising from complex scientific experiments, the distributions deviate far from normality, displaying aspects of skewness, heavy tails, and also, possibly, outliers and multimodality. It would thus be attractive to use a probability model that can incorporate these features, to perform more accurate clustering. Most traditional clustering methods do not assume that the cluster kernel is multimodal; often, in fact, the prime motivation for clustering is to partition data into groups centered around the individual modes. In his famous volume, Everitt (1974) provides three definitions for a cluster, of which the first is, "A cluster is a set of entities which are alike, and entities from different clusters are not alike." This definition does not tie cluster components to distributional modes. We find that this definition is likely to be prescribed to by most cluster analysts; even those that propound the use of unimodal density regions as the definition of a cluster.

Furthermore, recent literature regarding the modality of mixture models additionally dismisses the notion of synonymousness of clusters and unimodal densities. For example, the work of Ray and Ren (2012) demonstrates that a two component mixture of normal distributions in d dimensions can have up to $d + 1$

modes. Thus, if one believes that the two densities are representative of clusters, then we must accept that the maximum a posteriori region that maps to each cluster may be multimodal. Extending upon this work, Amendola et al. (2019) demonstrate that when we consider $k > 2$ components, the number of possible modes of the resulting density function grows factorially in k and d. A corollary of both results is this: if you partition the d-dimensional real space using the maximum a posteriori (MAP) clustering rule (that is, assign each point of the space to a cluster corresponding to the greatest product of component prior probability and conditional density), then there is no guarantee that any of the partitions will be unimodal. That is to simply say, Gaussian mixture models do not produce unimodal clusters. The partitionings are not unimodal with respect to a well-specified Gaussian mixture measure, and they cannot be guaranteed to be unimodal with respect to any other underlying measure. Regardless of whether the component densities are multimodal or not, the clusterings using the usual MAP rule will always have the potential of creating multimodal partitionings of the space from which the data under analysis arises. Thus, the conclusion is that advocating for the justification of unimodality with respect to clusters via mixture modelling is unreasonable.

In fact there are methods under which one can guarantee unimodal clusters, in some sense, that is strongly related to mixture model-based clustering. These methods are mode-seeking clustering algorithms such as the mean-shift algorithm (Cheng, 1995; Li et al., 2007) and the DBSCAN algorithm (Ester et al., 1996). Both of the algorithms above use the modality of a model of the generative density function as the objective of the clustering processes. This is quite different to the objective of the maximum a posteriori model-based approach, which does not take modality of the underlying partitions into account.

We therefore see no philosophical problem with allowing for multimodality when defining clusters, even when using unimodal component densities. Furthermore, in line with Everitt (1974), we believe that the ultimate aim of clustering is to retrieve some underlying taxonomy that represents similarity among the observations in the same cluster.

Multimodality within clusters has been observed in a variety of scenarios. For example, measurement restrictions placed on experimental data (such as image intensities at a fixed resolutions) may result in a minor grouping of data points within clusters. Measurement errors of discrete valued signals may yield multimodality, which was a complicating factor in classifying cancer patients from mass spectrometry imaging data on tissue microarrays (Mascini et al., 2018). This is also observed in many datasets in molecular biology and ecology (Thiem et al., 2007; Lampert and Tlusty, 2013), astronomy and astrophysics (Einasto et al., 2012) and also in the image reconstruction literature (Richards, 2012) and the noisy image segmentation and

quantisation examples presented in Section 6.

Here, we introduce a novel clustering method based on a characterisation of clusters that may be skewed, heavy-tailed, and possibly multimodal, while retaining moderate levels of parametrisation, mathematical tractability, and identifiability. The model is adapted from a recently introduced distribution based on a convolution of normal and geometric densities, called the Geometric Skew Normal (GSN) distribution (Kundu, 2017). The probability density function (PDF) of the GSN distribution can be unimodal or multimodal, symmetric or asymmetric, thin- or heavy-tailed, can be written as an infinite mixture of normal distributions, and reduces to a normal distribution in the limit. Although the maximum likelihood estimators (MLEs) for a GSN cannot be obtained in an explicit form, using a latent variable formulation allows us to construct an MCMC-based algorithm to estimate the joint posterior distribution of its parameters.

In this paper, we explore the qualities of the GSN family that are relevant for purposes of clustering, and set up a Bayesian framework and methodology for its estimation, in the univariate and multivariate cases. This is extended to a model for a mixture of GSN distributions, and developing a procedure for its estimation via an MCMC technique. We also address the questions of model identifiability and model selection. The article is organized as follows. We first describe the GSN model in its univariate and multivariate versions, illustrating its potential for modelling a variety of distributional shapes (Section 2), and discuss parameter estimation in a Bayesian framework (Section 3). Next, we extend the model to a mixture of GSN distributions (henceforth termed GSNM) with the aim of applying it to clustering, and discuss aspects of model estimation, cluster identification and model selection in Section 4. This is followed by a performance comparison of clustering using the GSNM against other mixture-model based clustering methods, in a variety of simulation studies, in Section 5. Finally, the GSNM is applied to three real-life examples in Sections 6 and 7, demonstrating various aspects of its effectiveness in clustering heavily non-normal data.

2 The geometric skew normal distribution

The univariate GSN distribution is a three-parameter distribution introduced by Kundu (2014) as a generalization of the normal distribution, which allows for a large degree of flexibility in terms of skewness, kurtosis and multimodality. Let X_i $(i \in \mathbb{N})$ be independent and identically distributed (i.i.d.) as Normal (μ, σ^2), and

$N \sim$ Geometric (p), independently of each other, where

$$P(N = n) = p(1-p)^{n-1}, \qquad n = 1, 2, \ldots. \tag{1}$$

If we define

$$X = \sum_{i=1}^{N} X_i, \tag{2}$$

then X is said to follow a geometric skew normal distribution with parameters μ, σ^2, and p, denoted as GSN(μ, σ^2, p). The PDF of the GSN distribution is derived in Kundu (2014) as the marginalization of the joint density of X (defined in (2)) and N (defined in (1)), with respect to X. That is,

$$\begin{aligned} f_{X,N}(x,n) &= p(N=n)\, f(x|n) \\ &= p(1-p)^{n-1} \frac{1}{(2\pi\, n\sigma^2\,)^{1/2}} \exp\left(-\frac{1}{2\,n\,\sigma^2}(x-n\mu)^2\right). \end{aligned}$$

$$\begin{aligned} \text{Then,} \qquad f_X(x) &= \sum_{n=1}^{\infty} f_{X,N}(x,n) \\ &= \sum_{n=1}^{\infty} p(1-p)^{n-1} \frac{1}{(2\pi\, n\sigma^2\,)^{1/2}} \exp\left(-\frac{1}{2\,n\,\sigma^2}(x-n\mu)^2\right). \end{aligned}$$

We now explore how changing the values of the three parameters of the GSN distribution leads to flexibility in the shape of the resulting PDF of X. First, in the limit as p tends to 1, the distribution of X reduces to a normal distribution. With μ fixed at 0, we only obtain symmetric distributions around 0, with σ^2 determining the variability or dispersion and p determining the degree of heavy-tailedness (kurtosis) of the distribution. Figure 1 shows that if we fix $\mu = 0$ and $\sigma^2 = 1$ and decrease p, the resulting symmetric distribution has heavier tails (as noted in Kundu (2014)). If we want to obtain a symmetric distribution around a value of x, different from zero, we need to change the value of μ, and fix p close to 1. This works on a visual level, but technically the sign of μ determines the sign of the skewness: when $\mu > 0$ the distribution will be positively skewed, and when $\mu < 0$ the distribution will be negatively skewed. But if p is close enough to 1, the skewness will be imperceptible.

Suppose we fix μ ($\neq 0$) and σ^2 with the ratio $|\mu/\sigma| < \eta$, η being "small" (approximately less than 4, observing that this distinction only works at a visual level). Then, as p decreases, the distribution will be

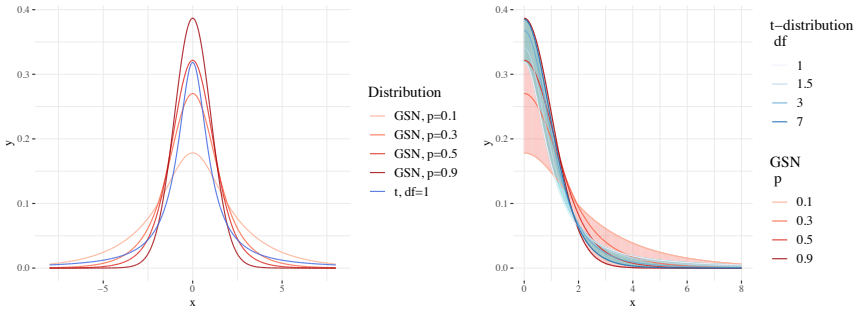

Figure 1: Plot of the GSN with $\mu = 0$, $\sigma^2 = 1$ and varying p, compared to a t-distribution. The right panel shows how the tails compare when the degrees of freedom of the t vary.

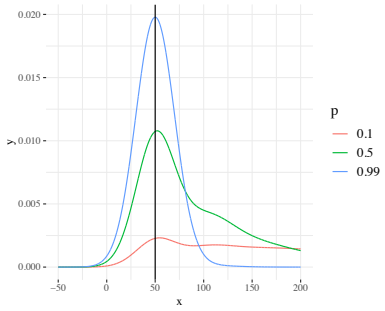

Figure 2: Plot of the GSN with $\mu = 50$, $\sigma = 20$ and varying p.

more spread on the right of μ (if $\mu > 0$) or on the left of μ (if $\mu < 0$). See, for example, Figure 2. If p is sufficiently small, multimodality appears, in addition to the skewness, with the smaller modes being very close to the maximum of the density, which is in turn close to μ. See, for example, Figure 3. The greatest mode of the GSN, with $\mu \neq 0$, will always be slightly (imperceptibly for high values of p) to the right of μ (if $\mu > 0$) or to the left of μ (if $\mu < 0$). If the ratio $|\mu/\sigma| \geq \eta$ (i.e., is "large"), as p decreases, multimodality appears (with no previous appreciable skewness) with modes that reproduce skewness, the number of modes increasing as p decreases. See, for example, Figure 4. These modes are only to the right of the greatest mode, if μ is positive, (and to the left if μ is negative), and they decrease in relevance (i.e., their local maxima are smaller) as we move to the extremes. Upon closer inspection, it can be seen that as the ratio $|\mu/\sigma|$ increases, the number of modes increases. Thus, for a fixed ratio $|\mu/\sigma|$ and fixed p we obtain a distribution with a fixed number of modes (Figure 15 in the online Supplement). The ratio $|\mu/\sigma|$ has a direct influence on the number of modes of the PDF: as $|\mu/\sigma|$ increases, the components (normal distributions) of which the GSN distribution is made become more separated and this results in a greater number of modes,

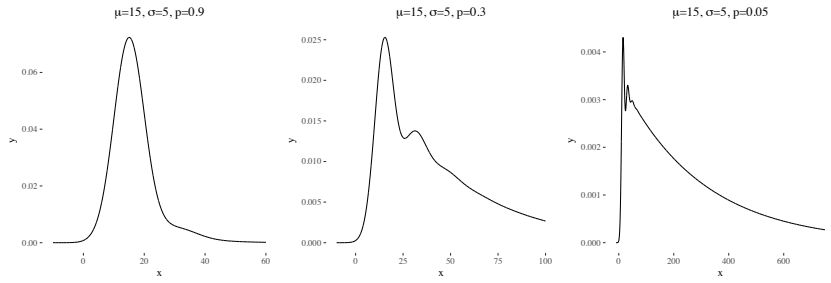

Figure 3: Plot of the GSN with "small" $|\mu/\sigma|$: as p decreases first skewness increases, then multimodality appears in addition to the skewness.

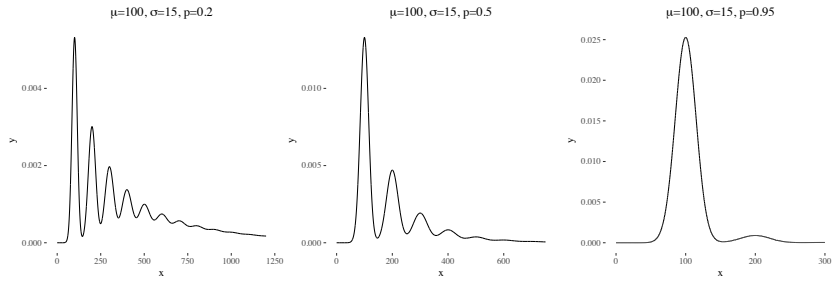

Figure 4: Plot of the GSN with "large" $|\mu/\sigma|$: as p decreases multimodality appears.

which goes up to a certain number of non-negligible modes. With a ratio $|\mu/\sigma|$ already past the point at which "all" the modes have appeared, as $|\mu|$ increases (with fixed σ^2), the maximum is shifted towards $|\mu|$ and the distance between the modes increases (see, for example, Figure 16 in the online Supplement).

2.1 Multivariate GSN

The multivariate geometric skew normal distribution (Kundu, 2017) is presented as a simple extension of the univariate case. Let MVN_d denote a multivariate normal distribution of dimension $d \in \mathbb{N}$. Next, suppose that

$$\boldsymbol{X}_i \ (i \in \mathbb{N}) \text{ are i.i.d.} \sim \text{MVN}_d\left(\boldsymbol{\mu}, \boldsymbol{\Sigma}\right).$$

Then, $\boldsymbol{X} = \sum_{i=1}^{N} \boldsymbol{X}_i$ is said to follow a d-dimensional geometric skew normal distribution with parameters p, $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, and is denoted as $\text{MGSN}_d(\boldsymbol{\mu}, \boldsymbol{\Sigma}, p)$. The joint PDF of $(\boldsymbol{X}, N)$ can be written as:

$$\begin{aligned} f_{\boldsymbol{X},N}(\boldsymbol{x}, n) &= p(N = n)\, f(\boldsymbol{x}|n) \\ &= p(1-p)^{n-1} \frac{1}{(2\pi n)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2n}(\boldsymbol{x} - n\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - n\boldsymbol{\mu})\right). \end{aligned}$$

The PDF of $\boldsymbol{X}$ is obtained by marginalising the joint density over N, and can be seen, as in the univariate case, as a mixture model with mixing proportions given by the probability mass function (PMF) of N:

$$\begin{aligned} f_{\boldsymbol{X}}(\boldsymbol{x}) &= \sum_{n=1}^{\infty} f_{\boldsymbol{X},N}(\boldsymbol{x}, n) \\ &= \sum_{n=1}^{\infty} p(1-p)^{n-1} \frac{1}{(2\pi n)^{d/2}|\boldsymbol{\Sigma}|^{1/2}} \exp\left(-\frac{1}{2n}(\boldsymbol{x}-n\boldsymbol{\mu})^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-n\boldsymbol{\mu})\right). \end{aligned}$$

The distribution, as in the univariate case, can result in skew and multimodal densities: a pair of examples of the bivariate case, is shown in Figure 5. It is worth noting that the normal components which make up the GSN will always lie on the $x = y$ line, as shown for the bivariate case. In general the means of the normal components will be on the line $\boldsymbol{z} = \alpha\,\boldsymbol{\mu}\,(\alpha \in \mathbb{R})$. The matrix $\boldsymbol{\Sigma}$ only defines the dependence structure within each normal component.

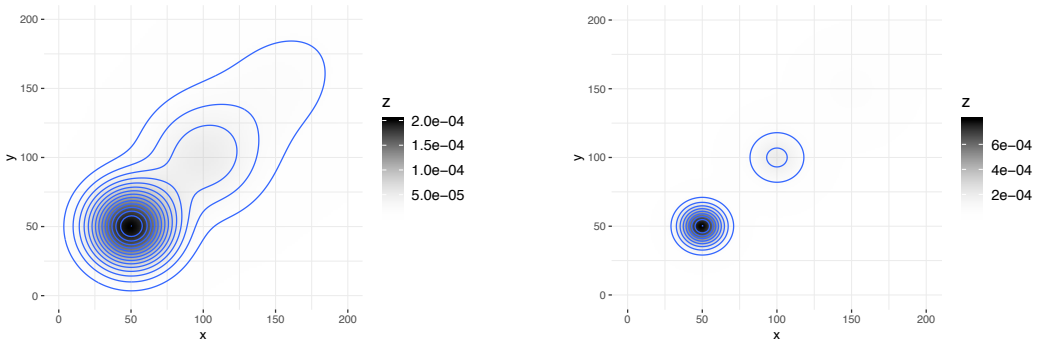

Figure 5: (a) Contour plot of a bivariate GSN with skewness [$\boldsymbol{\mu} = (50, 50)$, $\boldsymbol{\Sigma} = \text{diag}(400, 400)$, $p = 0.5$]; (b) Contour plot of a bivariate GSN with multimodality [$\boldsymbol{\mu} = (50, 50)$, $\boldsymbol{\Sigma} = \text{diag}(100, 100)$, $p = 0.5$].

3 Model-fitting and parameter estimation

Due to the intractability of the likelihood, Kundu (2014) proposed the use of an EM algorithm to estimate the parameters of the GSN. However, with a multivariate GSN, the increase in dimensionality of the parameter space massively increases the computational cost of maximising the likelihood, using an EM approach. Instead, Kundu (2017) proposed a profile log-likelihood approach, first fixing the value of p as known, using EM to estimate the MLEs of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ for the fixed value of p, say, $\hat{\boldsymbol{\mu}}(p)$ and $\hat{\boldsymbol{\Sigma}}(p)$, and finally finding the MLE of p by maximising the profile log-likelihood function of p, $l(p, \hat{\boldsymbol{\mu}}(p), \hat{\boldsymbol{\Sigma}}(p))$ with respect to p. In this article, we instead take a fully Bayesian approach to set up the model, and use a Markov chain Monte

Carlo procedure to estimate all parameters jointly. By using a latent variable formulation, as described in the following sections, we avoid the necessity of presuming any parameter values in advance, and the use of a two-step procedure in estimation. First, we describe the components of the fully Bayesian model parametrization.

3.1 Priors

To set up a Bayesian model to estimate the parameters of the GSN, we first need to specify all prior distributions. Priors may allow for varying degrees of informativeness, but for model-based clustering, such priors cannot be improper. MCMC in a mixture model involves sampling component-specific parameters from their conditional posterior distributions, which may reduce to the prior distributions if no observations are sampled from a component, and impropriety of priors would make such sampling impossible.

For p, we choose a Beta (c, d) prior, which can represent a wide range of prior beliefs, and can be set as non-informative when $c = d = 1$. The beta prior is conjugate to the likelihood, leading to a beta full conditional posterior distribution for p. The priors for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are chosen to be multivariate Gaussian, and inverse-Wishart, respectively. These distributions are also conjugates, allowing full conditional distributions to be derived, and reducing the computational cost of fitting the model. Two possible choices are (i) independent priors for $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$, and (ii) hierarchical priors, with the prior distribution of $\boldsymbol{\mu}$ dependent on $\boldsymbol{\Sigma}$, through its variance parameter. We choose the second setting, where the full conditional distribution of $\boldsymbol{\Sigma}$ does not depend on $\boldsymbol{\mu}$, and the reduced dependence between parameters allows for a more efficient MCMC algorithm. The prior distribution of $\boldsymbol{\mu}$ is conditional on $\boldsymbol{\Sigma}$, its variance being equal to $\boldsymbol{\Sigma}/k_0$, where k_0, usually set between 0 and 1, inflates the variance, resulting in a weakly informative prior for $\boldsymbol{\mu}$. We assume $\boldsymbol{\mu} \mid \boldsymbol{\Sigma} \sim \text{MVN}_d\left(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}/k_0\right)$, and the prior distribution for $\boldsymbol{\Sigma}$ is taken as: $\boldsymbol{\Sigma} \sim$ Inverse-Wishart $(\nu_0, \boldsymbol{\Lambda}_0^{-1})$.

3.1.1 Hyperparameter settings for the priors

In all applications, the hyperparameters have been set at values that result in minimally informative priors, however, these can also be chosen to reflect the amount of available information. The default prior distribution for p is Beta$(1, 1)$. For $\boldsymbol{\mu}$, the prior is a normal with mean $\mathbf{0}$ and the parameter k_0, which inflates the variance dependent on $\boldsymbol{\Sigma}$, is chosen to be 10^{-3}. The inverse-Wishart prior of $\boldsymbol{\Sigma}$ has parameter $\nu_0 = d+1$ and $\boldsymbol{\Lambda}_0 = \text{diag}(\mathbf{1})$. This value for the prior of $\boldsymbol{\Sigma}$ is considered non-informative (Gelman et al., 2014), because

the correlation in $\boldsymbol{\Sigma}$ marginally has a uniform distribution. Sensitivity analyses, carried out by repeating the analyses while changing the values of all hyperparameters within a moderate range, exhibited imperceptible differences in the results. Hence, these hyperparameter values were set as fixed for all analyses that follow. A section of the sensitivity analyses, pertaining to the hyperparameters $k_0,\ \nu_0$, $\boldsymbol{\Lambda}_0^{-1}$, and (c, d) is presented in Tables 9, 10 and 11 in the online Supplementary materials.

3.2 Posterior distributions and their estimation

The PDF of the MGSN, as given in (3), is characterized as an infinite sum, which would normally lead to an intractable likelihood to compute for the purpose of estimation. However, as we show here, a latent variable formulation of the model leads to an efficient MCMC sampler for model-fitting.

First, we introduce the latent variables n_i ($i = 1, .., m$), one for each corresponding $\boldsymbol{x}_i$, (m being the sample size). With these latent variables, the complete data likelihood for $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ becomes

$$f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m, \boldsymbol{n}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, p) = \prod_{i=1}^{m} p(1-p)^{n_i-1} \left|\boldsymbol{\Sigma}\right|^{-\frac{1}{2}} (2\pi n_i)^{-\frac{d}{2}} \exp\left(-\frac{1}{2\, n_i} \left(\boldsymbol{x}_i - n_i\boldsymbol{\mu}\right)^T \boldsymbol{\Sigma}^{-1} \left(\boldsymbol{x}_i - n_i\boldsymbol{\mu}\right)\right), \quad (3)$$

with $\boldsymbol{n} = (n_1, \ldots n_m)$. The variable n_i can be interpreted as controlling the assignment of each observation coming from a MGSN, to a normal component with parameters $(n_i\boldsymbol{\mu},\ n_i\boldsymbol{\Sigma})$ and mixing proportion equal to $p(1-p)^{n_i-1}$. From now on, the matrix $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)$, of dimension $d \times m$, will indicate the sample, where d is the dimension of each sample observation $\boldsymbol{x}_i$ ($i = 1, .., m$). Moreover, $\boldsymbol{\mu}$ is a d-dimensional vector, $\boldsymbol{\Sigma}$ a $d \times d$ matrix, p a scalar parameter and $\boldsymbol{n}$ an m-dimensional vector. From (3), we can take advantage of the fact that the likelihood resembles the product of a multivariate normal likelihood and a geometric likelihood, and if n_i were known for each $\boldsymbol{x}_i$, it would be straightforward to obtain full conditional posterior distributions of standard form for $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$ and p, in a Gibbs sampler. However, since $\boldsymbol{n}$ is unobserved, we need an additional step to build a full sampling algorithm to obtain the joint posterior distribution of all unknown quantities in the model. For each observation in the sample there is a latent variable n_i and thus its posterior distribution depends only on $\boldsymbol{x}_i$, along with the other parameters of the model. The posterior

distribution of n_i can be derived, up to a normalization constant, as

$$\begin{aligned} p(n_i|\boldsymbol{x}_i, \boldsymbol{\mu}, \boldsymbol{\Sigma}, p) &\propto p(n_i)\, p(\boldsymbol{x}_i, n_i|\boldsymbol{\mu}, \boldsymbol{\Sigma}, p) \\ &\propto (1-p)^{n_i-1}\, n_i^{-\frac{d}{2}}\, \exp\left(-\frac{1}{2\,n_i}\,(\boldsymbol{x}_i - n_i\boldsymbol{\mu})^T\,\boldsymbol{\Sigma}^{-1}\,(\boldsymbol{x}_i - n_i\boldsymbol{\mu})\right). \end{aligned}$$

Including an updating step for $\boldsymbol{n}$ leads to a hybrid Gibbs-Metropolis-Hastings algorithm for sampling the full set of parameters and latent variables. For brevity, we present the full algorithm, along with derivations of the posterior distributions, in the online Supplement. Henceforth, we will use the term GSN to refer to both univariate and multivariate forms of the geometric skew normal distribution.

4 Clustering using the GSN

Statistical clustering is often performed, in the model-based setting, by fitting a finite mixture of distributions (McLachlan and Peel, 2000), through likelihood-based methods such as expectation-maximization algorithms or Markov chain Monte Carlo (MCMC) methods (Diebolt and Robert, 1994). Clustering using MCMC, in the Bayesian framework, involves sampling a partition of observations from the set of all possible partitions.

4.1 GSN Mixture model

As seen in Section 2, the GSN distribution generalises the normality assumption for a data distribution by allowing for skewness and multimodality. One of the drawbacks in most existing model-based clustering algorithms, for instance those based on Gaussian-like component distributions, is the inability to allow for varying shapes and features within individual clusters. Bringing the flexibility of the GSN to the mixture model framework has the potential to allow for more diversity in features of individual components.

We now describe the framework of clustering using the GSN distribution. Model-based clustering makes use of a mixture model (of a finite number of k components with mixing proportions w_j, with $j = 1, \cdots, k$). We assume that the component densities are GSN. The population distribution of $\boldsymbol{x}_i$, are assumed to be in the form of a mixture of GSN distributions (GSNM):

$$p(\boldsymbol{x}_i \,|\, \boldsymbol{\theta}, \boldsymbol{w}) = \sum_{j=1}^{k} w_j\, p(\boldsymbol{x}_i \,|\, \boldsymbol{\theta}_j) = \sum_{j=1}^{k}\sum_{n=1}^{\infty} w_j\, \frac{p_j(1-p_j)^{n-1}}{(2\pi n)^{d/2}|\boldsymbol{\Sigma}_j|^{1/2}} e^{-\frac{1}{2n}(\boldsymbol{x}_i - n\boldsymbol{\mu}_j)^T\boldsymbol{\Sigma}_j^{-1}(\boldsymbol{x}_i - n\boldsymbol{\mu}_j)}, \tag{4}$$

where the notation $\boldsymbol{\theta}_j = (\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j, p_j)$ has been introduced to indicate the parameter set corresponding to component j, ($j = 1, \cdots, k$), and $\boldsymbol{w} = (w_1, \ldots, w_k)$.

Statistical inference for mixture distributions is often complicated by a lack of identifiability of the component-specific parameters, marginally, as the model likelihood is invariant under the permutation of the indices of its components. Identifiability is not a huge concern in Bayesian inference, as it can be imposed through priors or bypassed for prediction (Marin and Robert, 2007). However, it frequently impacts crucial aspects of Bayesian computation, through label-switching and convergence to the posterior distribution. In the following section, we discuss the identifiability of the GSN distributions and their finite mixtures.

4.2 Identifiability of GSN mixtures

Since the GSN distribution can be interpreted as a scaled mixture of normals, a mixture of GSN distributions can be considered a mixture of mixture distributions, which can potentially suffer from identifiability issues, and hence, problems in inference. Even in the Bayesian case of multi-level mixtures, therefore, evidence of identifiability, at least in terms of component-specific parameters across levels, is desirable. As Malsiner-Walli et al. (2017) state, exchanging subcomponents between lower-level clusters can lead to different cluster distributions while the higher level mixture likelihood remains identical, thus necessitating additional constraints in order to make the likelihood identifiable. They resolve this issue through a hierarchical prior setting, ensuring that the component distributions of the upper level are invariant to permutations, and within each cluster, prior distributions have a (hierarchical) block independence structure. In the case of GSN mixtures, however, we can show that even without such constraints, the distribution is identifiable.

In determining identifiability of GSN mixtures, we will need to use some definitions and results from van der Vaart (1998), and Yang and Wu (2014).

Definition 1. (van der Vaart, 1998, Sec. 5.5). Let $\mathcal{F} = \left\{ f_{\boldsymbol{\theta}} : f\left(\boldsymbol{x}; \boldsymbol{\theta}\right), \boldsymbol{x} \in \boldsymbol{X}, \boldsymbol{\theta} \in \boldsymbol{\Theta} \right\}$ be a family of probability density functions over the support $\boldsymbol{X}$, indexed by the parameter $\boldsymbol{\theta} \in \boldsymbol{\Theta}$. We say that the family $\mathcal{F}$ is identifiable, with respect to $\boldsymbol{\Theta}$, if for every $\boldsymbol{\theta}_0 \neq \boldsymbol{\theta}$, there exists an $\boldsymbol{x}^* \in \boldsymbol{X}$ such that $f_{\boldsymbol{\theta}}\left(\boldsymbol{x}^*\right) \neq f_{\boldsymbol{\theta}_0}\left(\boldsymbol{x}^*\right)$.

We are firstly concerned with the identifiability of the family of GSN densities (Kundu, 2017):

$$\mathcal{F} = \left\{ f_{\boldsymbol{\theta}} : f\left(x; p, \mu, \sigma^2\right) = \sum_{k=1}^{\infty} \frac{p\left(1-p\right)^{k-1}}{\sqrt{k\sigma^2}} \phi\left(\frac{x - k\mu}{\sqrt{k\sigma^2}}\right), x \in \mathbb{R}, \boldsymbol{\theta} = \left(p, \mu, \sigma^2\right) \in \boldsymbol{\Theta} \right\},$$

where $\boldsymbol{\Theta} = (0,1) \times \mathbb{R} \times (0,\infty)$. Here $\phi(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}z^2\right)$ is the standard normal density function. Our goal is first to prove the identifiability of $\mathcal{F}$, and then identifiability of the class of finite mixtures of $\mathcal{F}$.

Now, making use of results from Yang and Wu (2014), we can prove the following identifiability results.

Theorem 1. *The functional class $\mathcal{F}$ is identifiable in the sense of Definition 1.*

Theorem 2. *The class of finite mixtures of $\mathcal{F}$ is identifiable in the sense of Definition 1.*

The complete proofs for Theorems 1 and 2 are presented in the Appendix. It is important to note here that the identifiability of the GSN mixture model, although providing a sound theoretical basis for its use, does not necessarily imply computational identifiability in model-fitting. This is because by specification, the mixture components are exchangeable across cluster labels and therefore one cannot meaningfully estimate a component-specific posterior distribution for component k without defining component k to be in some way different from the others. We discuss this issue further in Section 5.

4.3 Fitting the GSN mixture model

In the Bayesian framework, fitting the GSN mixture model is an extension of fitting an individual GSN, and is facilitated by introducing latent variables $\boldsymbol{z} = (z_1, \ldots, z_m)$, one for each observation, which acts as an (unobserved) indicator for the component to which each observation belongs. We set $z_i = j$ if $\boldsymbol{x}_i$ comes from component j ($j = 1, \cdots, k$). As previously, we also have the latent variables $\boldsymbol{n} = (n_1, \ldots, n_m)$. The complete data likelihood for the mixture distribution then becomes

$$\begin{aligned} p(\boldsymbol{x} \,|\, \boldsymbol{\theta}, \boldsymbol{w}, \boldsymbol{z}, \boldsymbol{n}) &= \prod_{i=1}^{m} w_{z_i} \, f(\boldsymbol{x}_i, n_i \,|\, \boldsymbol{\theta}_{z_i}) \\ &= \prod_{i=1}^{m} w_{z_i} \, p_{z_i} (1-p_{z_i})^{n_i - 1} \, |\boldsymbol{\Sigma}|^{-\frac{1}{2}} (2\pi n_i)^{-\frac{d}{2}} \, \exp\left(-\frac{1}{2\,n_i} \left(\boldsymbol{x}_i - n_i \boldsymbol{\mu}_{z_i}\right)^T \boldsymbol{\Sigma}_{z_i}^{-1} \left(\boldsymbol{x}_i - n_i \boldsymbol{\mu}_{z_i}\right) \right). \end{aligned}$$

The mixing proportions $\boldsymbol{w} = (w_1, \cdots, w_k)$ are given a Dirichlet $(\alpha_1, \cdots, \alpha_k)$ prior. Let us define the indicator variable $\mathbb{1}_{[z_i = j]}$, to take the value 1 (0) if observation $\boldsymbol{x}_i$ is (is not) allocated to component j ($j = 1, \cdots, k$). The full conditional posterior distribution for $\boldsymbol{w}$ is a Dirichlet $(\alpha_1 + l_1, \cdots, \alpha_k + l_k)$, where $l_j = \sum_{i=1}^{m} \mathbb{1}_{[z_i = j]}$. Defining $\boldsymbol{q}_i = (q_{i1}, \cdots, q_{ik})$ to be the posterior probability distribution of z_i, gives:

$$q_{ij} = p(z_i = j \,|\, w_j, \boldsymbol{\theta}_j, \boldsymbol{x}_i) = \frac{w_j \, f(\boldsymbol{x}_i, n_i \,|\, \boldsymbol{\theta}_j)}{\sum_{g=1}^{k} w_g f(\boldsymbol{x}_i, n_i \,|\, \boldsymbol{\theta}_g)}.$$

The full conditional posterior distribution of $\boldsymbol{\theta}_j$, the parameters of the j-th GSN distributed component, are derived in an almost identical way as in the MCMC algorithm for the GSN, the only difference is that the full conditional posteriors depend on l_j (as defined earlier), instead of m; on $\mathbb{1}_{[z_i=j]}\boldsymbol{x}_i$, instead of the whole sample $\boldsymbol{X}$; and on $\mathbb{1}_{[z_i=j]}n_i$, instead of the whole vector $\boldsymbol{n}$. The priors for component specific parameters are chosen to have the same hyperparameters for each component, and as described in Section 3.1.1. The only additional hyperparameters are the ones for $\boldsymbol{w}$, that is, α_j $(j = 1, \cdots, k)$, which would be usually chosen to be constant (e.g. 1), but could be adapted to reflect any prior knowledge of cluster sizes.

The full conditional posterior for p_j is a beta distribution; $\sum_{i=1}^{m} \mathbb{1}_{[z_i=j]}n_i - l_j$ determines where most of the density concentrates. If $\sum_{i=1}^{m} \mathbb{1}_{[z_i=j]}n_i$ is much greater than l_j, the mode of the density will be closer to 0 (the distribution being right skewed). If $\sum_{i=1}^{m} \mathbb{1}_{[z_i=j]}n_i = l_j$, the full conditional posterior will depend only on l_j, and will have much of its density towards values close to 1 (i.e. will be left skewed). $\boldsymbol{\mu}_{mj}$ can be interpreted as a weighted average of the prior mean $\boldsymbol{\mu}_0$ and the sample mean of component j, $(\sum_{i=1}^{m} \mathbb{1}_{[z_i=j]}\boldsymbol{x}_i)/(\sum_{i=1}^{m} \mathbb{1}_{[z_i=j]}n_i)$, with the weights dependent on the relative sizes of the sample and informativeness of the prior for $\boldsymbol{\mu}_j$. The scale parameter in the full conditional distribution for $\boldsymbol{\Sigma}_j$ does not depend on $\boldsymbol{\mu}_j$, reducing the dependence between parameters and allowing for more efficient MCMC. Details of the hybrid Gibbs-Metropolis-Hastings sampling algorithm are presented in the online Supplementary Materials. At every iteration, the order of computation (solely based on the number of parameters to be updated) is $mk + m + 4k$.

The MCMC technique for fitting the GSN mixture may be prone to label-switching, as is the case with most Bayesian mixture models, which can cause complications in posterior inference, especially when the number of clusters is large. If the MCMC chain mixed well enough and a sufficiently large number of samples were drawn, one could obtain the same posterior distributions for each component-specific parameter, thus making it impossible to compute posterior summaries accurately. We therefore use a postprocessing step, along the lines proposed in Gelman et al. (2014), by relabeling the samples to each correspond to a specific component before calculating the posterior summaries. In our examples, it was sufficient to do this empirically through observing the traceplots of posterior samples. More formally, one could use a loss function that maximises the similarity of the sampled parameters (e.g. $\boldsymbol{\mu}_k$) across iterations within clusters.

4.4 Model selection and diagnostics

A problem intrinsic to model-based clustering is model choice, or determining the number of components that is most appropriate for a specific data set. A frequently used Bayesian criterion, the Bayes factor, involves the computation of the marginal distributions, that provide evidence in favour of a particular model. This is not feasible in a clustering setting, as the latent variables $\boldsymbol{z}$, representing the cluster assignment of each observation, cannot be integrated out of the joint density to obtain the marginal. In Bayesian mixture models, marginal likelihoods are intractable to compute—the latent variables $\boldsymbol{z}$, representing the cluster assignment of each observation, cannot be integrated out of the joint density—and thus numerical or analytical approximations are necessary (Friel et al., 2017). The Bayesian information criterion or BIC (Schwarz, 1978), is often used in a Bayesian context as it provides a rough asymptotic approximation to the posterior model probability that forms the basis for model choice (Kass and Raftery, 1995). Mixture models, however, have singularities in the Fisher Information matrix at the boundary of the parameter space when considering models with k and $(k-1)$ components (for any integer k), hence the log-likelihood function, even in large samples, cannot be approximated by a quadratic form. This means that there may no longer be a connection between the Bayesian marginal likelihood function and a Gaussian integral, and therefore no longer a clear justification for using the BIC for model comparison (Drton and Plummer, 2017).

A second issue in the MCMC setting is that in the BIC, the likelihood of the models has to be evaluated at the MLE of the parameters, but using a Bayesian procedure, we can only evaluate it at the (estimated) posterior mode. An alternative would be to use a posterior simulation-based version of BIC, such as the BICM (Raftery et al., 2007). BICM makes use of the sequence of loglikelihoods l_t, $(t = 1, \ldots, T)$ over T MCMC iterations, and is defined as, $BICM = 2\hat{l}_{max} - \hat{d}\log(n)$, where $\hat{l}_{max} = \bar{l} + s_l^2$ is the estimated maximum achievable loglikelihood ($\bar{l},\ s_l^2$ being the sample mean and variance of the l_t's); and $\hat{d} = 2s_l^2$. We investigate the use of both the BIC and BICM in our numerical applications.

We also consider a recently proposed alternative, the Widely Applicable Information Criterion (or Watanabe Akaike Information Criterion), denoted as WAIC (Watanabe, 2010). This criterion attempts to estimate the out-of-sample expectation and is made of two parts: the computed log pointwise posterior predictive density and a correction for the effective number of parameters. The first part is given by:

$$\text{lppd} = \sum_{i=1}^{m} \log \left(\frac{1}{T} \sum_{t=1}^{T} f(\boldsymbol{x}_i \mid \boldsymbol{\theta}^{(t)}) \right),$$

where T is the total number of MCMC draws, and $f(\boldsymbol{x}_i \mid \boldsymbol{\theta}^{(t)})$ is the likelihood of observation i evaluated at parameters $\boldsymbol{\theta}^{(t)}$, drawn at iteration t of the MCMC chain; in this case $\boldsymbol{\theta}^{(t)}$ is made of $\left(\boldsymbol{n}^{(t)}_{z_i^{(t)}}, \boldsymbol{\mu}^{(t)}_{z_i^{(t)}}, \boldsymbol{\Sigma}^{(t)}_{z_i^{(t)}}, p^{(t)}_{z_i^{(t)}}\right)$. The second part of the criterion is denoted as p_{WAIC_2}, defined as:

$$p_{\text{WAIC}_2} = \sum_{i=1}^{m} \text{var}_{t=1}^{T}(\log p(\boldsymbol{x}_i \mid \boldsymbol{\theta}^{(t)})),$$

where $\text{var}_{t=1}^{T}$ stands for sample variance (unbiased) computed over all iterations of the MCMC chain. The WAIC is then defined as: $\text{WAIC} = -2\text{lppd} + 2p_{\text{WAIC}_2}$. The model with the lowest value of WAIC is considered preferable. A selection procedure, that turns out to be more parsimonious in some cases, is to instead choose the preferred k to be the one at which the WAIC reaches its first local minimum.

Another important aspect of model-fitting is assessing the model goodness of fit. Along the lines proposed by Gelman et al. (2014), we carried out posterior predictive model checks to assess the fit of the GSNM model. It is important to note, however, that posterior predictive model checks are only designed to assess whether aspects of the data distribution in posterior predictive samples have a corresponding similarity with the original data, so do not have the power to detect incorrect classification. In addition, we also implement a multivariate Kolmogorov-Smirnov (K-S) goodness of fit test along the lines proposed by O'Hagan et al. (2016). To avoid the difficulties of calculating a multivariate K-S test, they constructed empirical cumulative distribution functions (ECDFs) based on the log-density values of the underlying data and data simulated under the fitted model. We adapt their procedure into a Bayesian setting, comparing the ECDF of the log-densities of the original data under the optimal model (based on the posterior mean) to the ECDF of log-densities of samples generated from the posterior predictive distribution. This provides a useful visual diagnostic; if the model provides a good fit, these ECDFs should resemble one another.

5 Simulation studies

We now describe numerical studies that were conducted to test the performance of the GSNM model in clustering applications.

We generated bivariate ($d = 2$) data sets from 3 component ($k = 3$) mixtures, each consisting of 1000 observations ($m = 1000$). A first group of data sets was generated from a mixture of normals model. The component parameters were obtained as follows. $\boldsymbol{\Sigma}$ was sampled from an inverse-Wishart distribution

(with parameters $\nu_0 = d + 1$ and $\mathbf{\Lambda}_0 = \text{diag}(\mathbf{1})$), and $\boldsymbol{\mu}$ from a normal distribution with mean $\mathbf{0}$ and three settings of variance: diag($\mathbf{2}$), diag($\mathbf{5}$) and diag($\mathbf{10}$), which were selected to give three broad categories of different levels of overlap in the components and were therefore named low, medium and high. The mixing proportions, $\boldsymbol{w}$, were obtained by scaling to unit length (making its sum equal to 1) a sample of 3 elements from a Binomial$(n = 10, p = 0.5)$, with a unit added to avoid obtaining zeros. The second group of data sets were simulated from a mixture of multivariate t-distributions. They were obtained in the same way as the mixture of normals, with an additional parameter, ν_k, for the component-specific degrees of freedom, sampled from a Binomial$(n = 30, p = 0.2) + 1$. A third group was obtained from mixtures of GSN distributed components. The mixing proportions were obtained as in the normal mixture simulation. The parameter $\boldsymbol{\mu}$ was selected manually, $\mathbf{\Sigma}$ was generated from an inverse-Wishart (with $\nu_0 = d + 1$ and $\mathbf{\Lambda}_0^{-1} = \boldsymbol{I}_d \times$selected constant), and p was generated from a uniform distribution (with changing boundaries). Fifteen parameter settings were considered and categorised into 3 sets: representing low, medium and high degrees of overlap in the components. For each model, and each of the categories (low, medium and high), 5 different parameter settings were obtained, and for each setting, 5 data sets were simulated. The parameter settings for all simulated data sets are recorded in Tables 5, 6 and 7 in the online Supplementary Materials.

Aside from the GSN clustering algorithm (referred to as `gsnclust`), 3 other clustering methods were used to compare clustering results: `mclust` (Fraley and Raftery, 2002), which uses normal component densities and was fit via the EM algorithm; `teigen` (Andrews and McNicholas, 2012), which is a model-based clustering algorithm which assumes t-distributed components and was fit by an ECM (expectation-conditional maximization) algorithm, and a Gibbs sampling algorithm to fit a normal mixture model (referred to as `mnclust`). All algorithms were implemented in `R`.

In order to compare clustering solutions, a point estimate had to be computed from the posterior samples of clusterings. Any evidence of label-switching would have to be first taken into account, by a post-processing step after observing the traceplots of the posterior samples and doing any necessary relabelling of samples (Gelman et al., 2014). In our simulations and data applications, label-switching was never observed. Next, the proportion of times each observation was allocated to each cluster, in the post burn-in samples, was determined, and each observation was allocated to the cluster with the highest posterior probability. Two clustering situations were taken into account: one where k, the number of components, was assumed known; and one where it had to be chosen. In the latter case, all 4 algorithms were left to determine the correct number of clusters by comparing models whose number of components ranged from 1 to 10. Algo-

rithms `mclust` and `teigen` determined the best model using the BIC. For the two MCMC algorithms, the BIC, BICM and WAIC were used. The results were evaluated with three summary statistics:

- Accuracy: proportion of observations assigned to the correct clusters. To compute this metric, a relabelling of cluster calls given by the algorithms was carried out– this considered all possible permutations of label names, using the one that maximized accuracy.
- Adjusted Rand Index (ARI): a measure of similarity between two clustering solutions for a certain set of observations, its upper boundary being 1 (Hubert and Arabie, 1985). It was calculated comparing the results given by each algorithm considered and the true cluster assignments.
- Variation of Information (VI): a measure of distance between two clustering solutions (Meilă, 2003). Its lower bound is 0, attained when two clusterings are equal. As in the Adjusted Rand Index, it was used to compare clusterings obtained by used algorithms to the true cluster assignments.

5.1 When k is known and fixed

The results of applying each method under different model settings are summarized in Figure 6.

5.1.1 Normal mixture simulation

There is a negative trend in accuracy as the level of overlap increases, for each of the 4 model-fitting methods. However, there was very little difference between the four methods and this was also expected as they can all accommodate for the normal mixture model, which is assumed in `mnclust` and `mclust`. The algorithms `teigen` and `gsnclust` have it as limiting cases (i.e., when $\nu_k \to \infty$ and when $p_k \to 1$). The VI and ARI reinforced the same conclusion: no method performed consistently better than the others.

5.1.2 t-distributed mixture simulation

In this case, there did not seem to be a great deal of difference among the performance of the different algorithms. In 8 out of the 15 parameter settings, the range in average accuracy differs by less than 1%. In the case of medium overlap, `gsnclust` appeared to perform better than the other algorithms, indicating robustness to model misspecification. It may seem slightly counterintuitive that `teigen` did not perform better than `mnclust` in this case, but there could be a number of reasons for this. Due to the deterministic

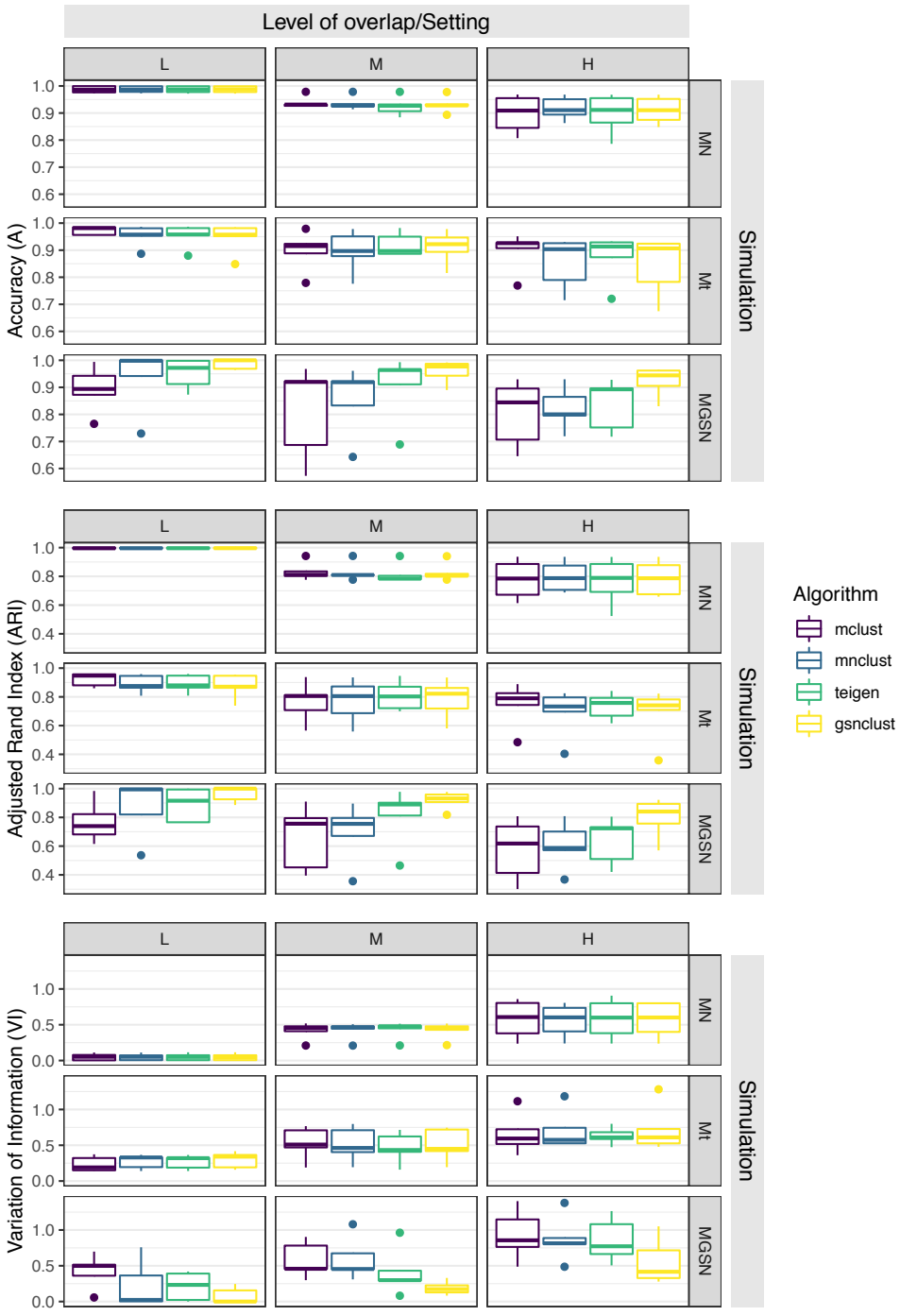

Figure 6: Comparisons of Accuracy, VI and ARI for four different clustering methods (`mclust`, `mnclust`, `teigen` and `gsnclust`) under different generative models. Each row within a 3×3 block corresponds to a different generative multivariate mixture model (MN: normal, Mt: t, MGSN: geometric skew normal). The columns correspond to different levels of overlap between clusters: low (L), medium (M) and high (H). Each boxplot is for 25 data sets generated under a total of 5 different model settings.

nature and gradient descent form of the EM algorithm implemented for `teigen`, it has no general ability to escape poor local maxima or saddle points in the search of more globally optimal solutions. In contrast, the estimation process that underlies the MCMC algorithm randomizes the estimation at each step, which makes it better able to escape local optima and produce higher quality estimates. One additional aspect that makes it more difficult for `teigen` than `mnclust` is its larger parameter space that includes the additional parameters that characterize the degrees of freedom of the t-distributions. These parameters introduce additional local optima to the likelihood function and complicate any estimation procedure that is used for model-fitting. As previously, ARI and VI gave very similar rankings of the methods. An example of the different clustering results for this study is given in Figure 17 in the online Supplement. In this case we can see how similarly the methods perform, with the exception of `mclust`, which in this case is not able to pick up the heavy tails of the component densities.

5.1.3 GSN mixture simulation

In the simulation from the GSN mixture model, `gsnclust` performed significantly better under almost all settings, in terms of accuracy, ARI and VI. This was evident in data simulated under both low and high levels of overlap between clusters. By inspection of the accuracy results, in only 1 out of the 10 cases, the range between methods was less than 1%. The mean improvement in accuracy for `gsnclust` compared to `teigen`, `mnclust` and `mclust`, respectively was 5, 6 and 9%.

We next highlight two particular cases that showed features of multimodality and skewness in the data. In Figures 7 and 8, which show the clustering results for two data sets, it is apparent that `gsnclust` was the only method which correctly identified the multimodal clusters. In both simulated data sets, two of the three components presented some multimodality, which was caused by values of p ranging from 0.8 to 0.95. They were chosen to highlight a feature of the GSN distribution that was unable to be picked up by the other models, reflecting the case where bigger clusters in the data may contain distinct subclusters. In summary, the GSNM fitted the data significantly better than other methods when the data generation and estimation model matched, and performed well and sometimes better even when the estimation model was misspecified. The closest competitor was the t-mixture, which, however, performed significantly worse under model misspecification and completely missed the multimodality in the data.

We also assessed the accuracy of the estimated model parameters in each scenario. Table 8 (in the online Supplement) gives posterior 95% credible intervals for the parameters of `gsnclust` when applied to data

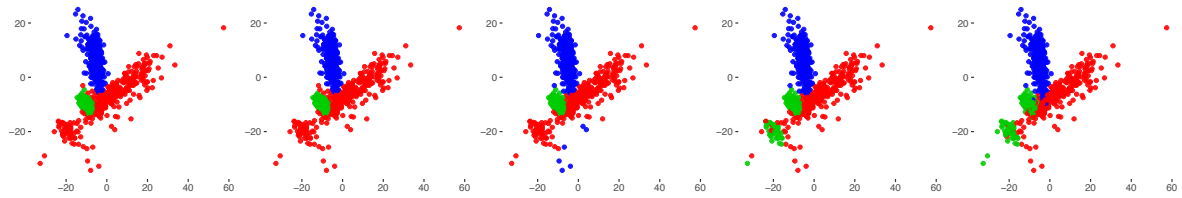

Figure 7: Clustering results for simulated GSN mixture when the true value of $k = 3$ (Simulation setting H1). Panels from left to right show the clustering from: (i) MNclust (ii) `MClust` (iii) `teigen` (iv) GSNclust and (v) True clusters.

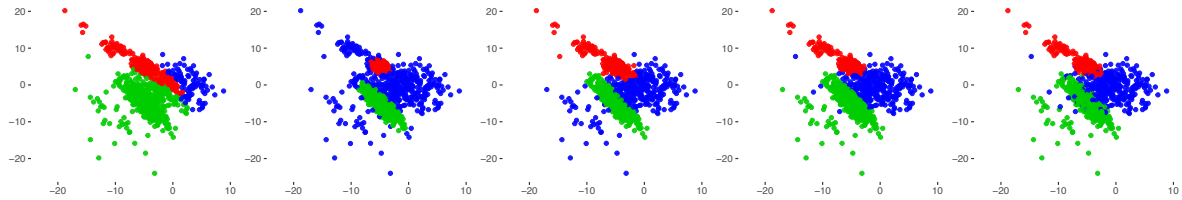

Figure 8: Clustering results for simulated GSN mixture when the true value of $k = 3$ (Simulation setting H4). Panels from left to right show the clustering from: (i) MNclust (ii) `MClust` (iii) `teigen` (iv) GSNclust and (v) True clusters.

simulated under three models (normal, t and GSN) over a variety of 8 settings, ranging from low to high cluster overlaps. All credible intervals are seen to include the true values well within them, even when models are misspecified. The credible interval for p, when data is simulated under a normal mixture, almost converges onto 1, as expected. Similarly, the component-specific $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ parameters are estimated close to the truth (approximating the mean and variance parameters in case of the normal).

`mnclust` and `gsnclust`, being the two MCMC-based methods, were also compared in terms of computional speed. We carried out a study with varying data set dimensions, and sample sizes (per cluster) between 2000 and 20,000, with k fixed at 3. The computing times, on a 3.2 GHz Intel® Core™ i5 processor with 8 GB RAM on a computing cluster, for `gsnclust`, ranged from 4.5 to 6.5 min for 1000 iterations, compared to a range of 3 to 4.5 minutes for `mnclust`, with the increase being roughly linear as the data dimension increased from 3 to 15 (the sample size was fixed at 2000 here). The time increase was larger as the sample size increased, from 3 minutes to about 190 minutes as the sample size increased to 20,000. The increase was slightly sharper for `gsnclust` than `mnclust`, however, there did not appear to be a substantial time saving for `mnclust` (Figure 18 in the online Supplement).

5.2 When k is unknown

When k was assumed unknown, we used information criteria, as described in Section 4.4, to select the number of components which seemed to best fit the data. To select the number of components in the GSNM (the true value of k being 3), four criteria were put to the test: the BIC, BICM, minimum WAIC, and the first local minimum in the WAIC. Five datasets each were simulated from the low-overlap settings (L1-L5) for normal, t and GSN mixtures (see Tables 5, 6 and 7 in the online Supplementary Materials). Table 1 gives a graphical summary of the proportion of times (out of 25 simulations) different values of k were selected under each model. All criteria except the BICM seem to overall determine the number of clusters correctly when data is simulated from a normal model. In the case of t-mixtures, the WAIClm performs most accurately, although the performance of the BIC is very close. In the GSN case, all except BICM seem to overfit slightly, with BIC overfitting the least. BICM seems to consistently underestimate the number of clusters in both normal and GSN mixtures, and in the t, it overfits almost half of the time.

The BIC thus appeared to be best overall for model selection, followed closely by the WAIClm (local minimum), based on the proportion of data sets in which k was correctly identified. For WAIClm, the selected k often tended towards larger values ($k > 5$) compared to the BIC, where in almost 75% of cases, the selected k was less than or equal to 5. (This seems to agree with findings in Friel et al. (2017).) The BIC may still be preferable to use, in spite of its unclear correspondence to the marginal likelihood, although a more comprehensive study may be necessary before conclusive evidence can be gained on this.

6 Illustrative data sets

6.1 SNP Genotyping data

Variation in the DNA sequences of individuals are compared by recording Single Nucleotide Polymorphisms (SNPs), sites which differ from a *reference* genome. Usually the variation in a single location is between two nucleotides (*alleles*). Each SNP site is assigned to one of three categories (a process called genotype calling) by considering multiple individuals at once (Lin et al., 2008). For each SNP, two variables are recorded, that relate to binding of a single-stranded DNA of the individual to two probes containing the complementary sequence for both alleles, adjacent to the SNP site. Homozygous individuals (having the same allele in both chromosomes) are expected to have high values for one of the two variables, while heterozygous individuals

		Effective k				
	Selection method	2	***3***	4	5	> 5
MN simulation	BIC	0	96	4	0	0
	BICM	72	28	0	0	0
	WAIC	0	96	4	0	0
	WAIClm	0	100	0	0	0
Mt simulation	BIC	0	24	48	24	4
	BICM	36	20	4	24	16
	WAIC	0	8	24	32	36
	WAIClm	0	44	36	16	4
GSNM simulation	BIC	0	0	16	44	40
	BICM	84	4	0	4	8
	WAIC	0	0	0	0	100
	WAIClm	0	0	12	12	76

Table 1: Selected number of components for different criteria for the MN, Mt and GSNM simulations; results are in percentages. Effective k refers to the actual number of used components (the chosen model can have a higher number of available components). The true value of k for this study is 3.

are expected to form a cluster in between the two homozygous genotypes. We considered a subset of data from a genome-wide association study relating to bone fractures (Estrada et al., 2012), conducted using Illumina microarrays. This contains observations on around 5000 individuals and 400 SNPs, and calls given by Illumina, which uses a proprietary algorithm for clustering followed by a manual curating process. The curated results were used as a benchmark for the accuracy of the calls. To compare the performance of `gsnclust` with normal and t-distributed mixtures in this scenario, we picked 2 SNPs, where the original Illumina clustering method did not give ideal results.

6.2 Image segmentation

This dataset is from the UCI machine learning database repository (Dua and Graff, 2017), contributed by the Vision group at the University of Massachusetts. A subset of this data was analyzed in the paper by Li (2005). Each sample represents a 3×3 block of an image. Each class of image blocks contains 300 samples- for our analysis, we focus on 4 classes: *brickface*, *path*, *cement*, and *sky*. From the original dataset of 19 variables (that include features such as location information of the blocks, which we ignore for the present purposes), we extract 5 variables corresponding to continuous-valued measurements in the HIS colour space (variables $6, 8, 17, 18, 19$). The data are centered and scaled before analysis for comparability, however, we keep variables in the original coordinates, in contrast to the projection on principal components used in Li

(2005). Our goal was to determine how accurately the 4 image classes may be recovered from the data using `gsnclust` as well as other algorithms. Scatterplots of the data (Fig 19 in the online Supplement) indicate that at least two true image classes do not have unimodal distributions and most are severely non-normal.

6.3 Photographic image reconstruction and quantisation

The image used in this analysis is a well known test image from the public domain, commonly referred to as *Baboon* (Andrews and Patterson, 1976). The original image is a 500×480 colour image (actually, of a mandrill). For the analysis the image has been compressed and resized—via the `resize` function in the `imager` R package—to a 200×200 image, in order to shorten computation time. The structure of image data can be described as three-dimensional, with pixels as units and RGB (red, green, blue) intensities as variables, so in our case the data set has 40000 units and 3 variables. RGB intensities are discrete in a scale from 0 to 255; before applying the methods these have been standardised, to facilitate the convergence of the two MCMC methods.

Figure 9 shows the *Baboon* image and pairs plot of the data. The many modes that appear from the 2D density estimation offer a theoretical justification for applying clustering methods that allow for multimodal components. In the RGB space, colours change gradually and thus no defined boundaries exist that group together pixels with similar colours. Moreover, if we want a number of clusters, which is smaller than the number of modes, then, necessarily, more than one mode will fall within the same cluster. Applying clustering methods to RGB image data means simplifying the image, that is, reducing the number of colours. After applying a clustering method, each pixel that has been assigned to the same cluster can be given the same colour, which in our case was decided to be the mean of the cluster. In this way, we obtain a simplified rendition of the image with as many colours as the number of components. These colours are the means of the component distributions fitted to the component pixels, so the image will maintain colour fidelity to the original.

7 Results

In this section, we describe the results on applying `gsnclust` to the three data sets described previously. The results were compared to another MCMC algorithm, which used normal component densities (`mnclust`), and three model-based algorithms `mclust`, `teigen`, and a mixture of multivariate skew-t

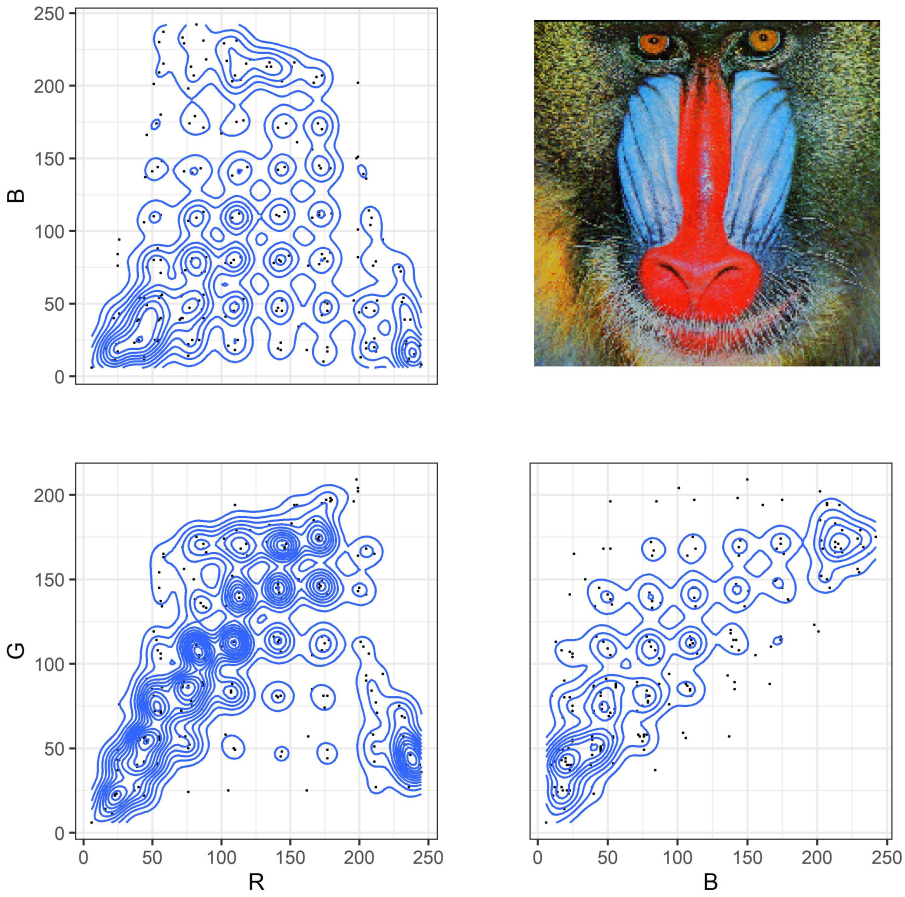

Figure 9: *Baboon* image (200 × 200 pixels) with pairs plot of the RGB intensities (integers on a scale from 0 to 255) with 2D density estimation.

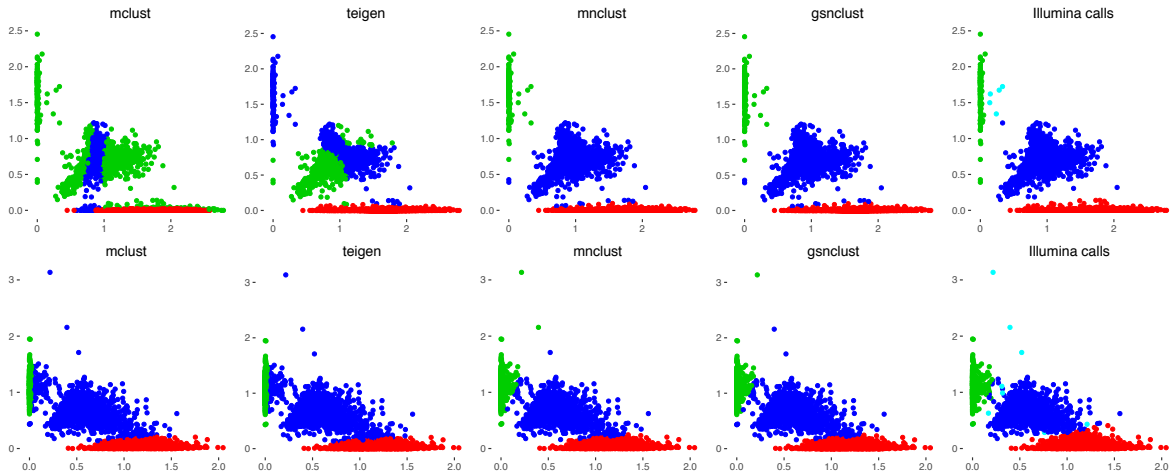

Figure 10: Genotype calls respectively for SNPs: rs11586958 (top row), and rs6103938 (bottom row), given by Illumina and the four clustering algorithms considered.

distributions (Wang et al., 2009).

7.1 Genotyping data clustering

The clustering results on two SNPs, `rs11586958` and `rs6103938`, are presented in Figure 10. The results of `gsnclust` and `mnclust` were more similar to the results given by Illumina, however, they were also able to allocate certain previously non-assigned calls, (where the uncertainty for the call was deemed too high) to visually appropriate clusters. `mclust` and `teigen` were not able to detect the clusters in either data sets correctly and were outperformed by the MCMC algorithms. This is especially evident in the case of SNP `rs11586958`, where the cluster sizes are extremely unbalanced.

Table 12 (in the online Supplement) gives the 95% central posterior credible intervals for the parameters of the fitted GSNM model, showing the distinctness of the cluster parameters. The estimated credible intervals for each p are almost identically 1, and the estimated values of $\boldsymbol{n}$ are almost identical to 1 which indicates why the GSNM performs similarly to the Gaussian mixture. The GSN results in this and the following examples were based on a sample of 10,000 iterations—varying this number between 5000 and 30,000 did not have any significant impact on the results and standard MCMC convergence diagnostics did not give any indications of non-convergence.

7.2 Image segmentation example

Mixtures of Gaussians (`mclust` and MCMC-based), t-distributions (`teigen`), and skew-t distributions were fitted to the data to detect the image classes, along with the GSNM. The skew-t mixture was fitted using the `R` package `EMMIXskew`. First, ignoring our knowledge of the true total number of clusters k (which should be 4), we ran each algorithm over values of k, starting from $k = 3$, and using the BIC to select an optimum number of clusters. For `mclust`, the BIC attained a minimum under a model of 23 clusters. `teigen` and the MCMC-based Gaussian mixture model chose models with $k = 9$, and the skew-t mixture chose $k = 6$. The skew-t mixture was sometimes numerically unstable at values of k above 9. The GSN mixture was the most parsimonious, choosing $k = 4$. Table 2 compares the classifications resulting from the optimal models from each method versus the truth and versus the GSNM. The algorithm `gsnclust` gives results closest to the true clustering, and the Skew-t and Gaussian mixtures are the next best in terms of clustering, although there is still a quite large difference in classification, as shown by the ARI and VI. Figure 12 shows that GSN is the only model that can pick up the two multimodal clusters (`brickface` and `cement`) accurately; all the other methods split these clusters into multiple classes. It can also be seen that both `teigen` and the skew-t mixture choose a heavy-tailed cluster- bottom panels 2 and 3 of Figure 12 (comprising of one mode of the `brickface` cluster) rather than including the secondary mode. Methods other than `gsnclust` also have difficulties in classifying the points in the overlapping parts of the distributions, and even allowing for more clusters, do not find the correct subgroups.

When we set $k = 4$ (the truth), the results for the other methods improve, but `gsnclust` is still the most accurate. The mixture of normal distributions, both in EM and MCMC, had the lowest correct classification rates (64% and 65%); at least one of the clusters is split incorrectly (Figure 20 in the online Supplement). `teigen` (70%) does better but is unable to capture the spread of the `cement` cluster due to possible multimodality. `EMMIXskew` (72%) has a similar limitation, and additionally cannot capture the range of variability in the `path` cluster, in terms of both the hue and value variables. The results from `gsnclust` (75%) come closest to the truth– it identifies the two multimodal clusters (`brickface` and `cement`) almost perfectly. Finding the 95% central posterior credible intervals for the parameters indicates that for p_2 (value of p corresponding to `cement`), this is $(0.791, 0.892)$, which represents a large deviation from normality of data points within this cluster (Table 3). For the other clusters, the CIs for p all lie between 0.96 to 1. The posterior densities for $\boldsymbol{w}$ and $\boldsymbol{\mu}$ indicate a slight propensity of the GSNM to underestimate the

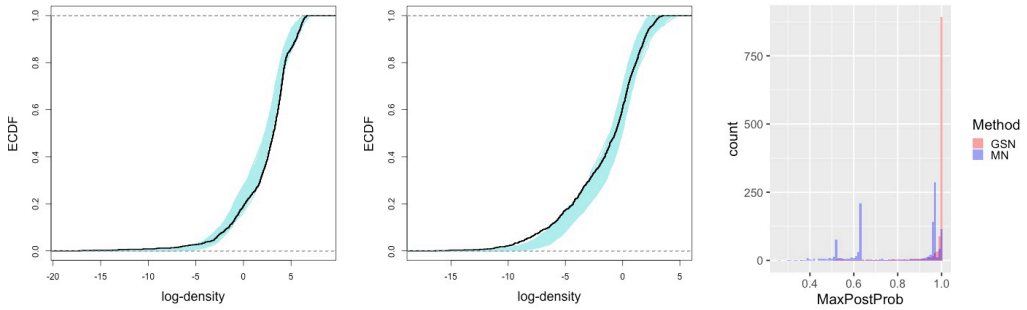

Figure 11: Adapted multivariate K-S goodness of fit plot for image segmentation data under (i) the GSNM model and (ii) the multivariate Gaussian mixture model. The dark central line represents the ECDF of the original data and the light coloured lines represents the ECDFs of 1000 posterior predictive data sets. Panel (iii) gives the overlying histograms of posterior cluster allocation probabilities for the GSNM and multivariate normal mixture models.

number of points in the `cement` cluster, and slightly overestimate points in the `path` cluster. The credible intervals for the cluster-specific means, $\boldsymbol{\mu}_k$, are narrow and do not overlap across clusters, indicating that each cluster possesses distinct features. The values of $\boldsymbol{n}$ are estimated in the range of 1 and 8; although a large number of n's are identical to 1, there are a substantial number of values in the range of 2 to 8.

The adapted K-S goodness of fit plots in Figure 11 show that the GSNM model overall provides an excellent fit to the data, with a slight tendency to place lower mass in the regions of data with medium log-density. The optimal Gaussian mixture, with $k = 9$, gives a worse model fit, showing deviations from the model in regions of low to medium log-density. The third panel of Figure 11 shows the histogram of maximum posterior cluster allocation probabilities for both models. GSNM overall has a much higher number of probabilities near 1, indicating higher cluster stability than the Gaussian mixture. Further posterior predictive checks with `gsnclust` (as in the previous example), did not indicate evidence of model violations (PPP-values ranging from 0.51 to 0.68).

Table 2: Comparing classifications of the image data using different methods to the true classes (Columns 3-7) and comparing the GSN-based clustering to the other methods (Columns 8-11).

Metric	Optimum	`mclust`	`mnclust`	`teigen`	Skew-t	GSN	`mclust` vs. GSN	`mnclust` vs. GSN	`teigen` vs. GSN	Skew-t vs. GSN
ARI	1	0.2689	0.6096	0.5200	0.6168	0.6304	0.2421	0.5846	0.4925	0.6215
VI	0	2.7427	1.6108	1.6602	1.3104	1.1970	2.7701	1.5180	1.7259	1.4147

The MCMC algorithm underlying `gsnclust` is highly scalable to large data sets. The time taken to

Table 3: Posterior credible intervals for component-specific parameters for the 4 components of the GSNM model for the image segmentation data.

K	1		2		3		4	
Class	`brickface`		`cement`		`path`		`sky`	
	2.5%	97.5%	2.5%	97.5%	2.5%	97.5%	2.5%	97.5%
μ_{1k}	-0.496	-0.461	0.349	0.501	0.004	0.179	-0.401	-0.319
μ_{2k}	-0.502	-0.466	0.022	0.129	1.004	1.622	-0.410	-0.334
μ_{3k}	1.508	1.556	-0.087	-0.037	-0.735	-0.623	-1.122	-1.058
μ_{4k}	-1.020	-0.896	-0.223	-0.185	-0.017	0.348	1.202	1.371
μ_{5k}	-0.903	-0.857	-0.237	-0.203	-0.296	-0.069	1.313	1.553
p_k	0.987	1.000	0.791	0.892	0.969	1.000	0.986	1.000

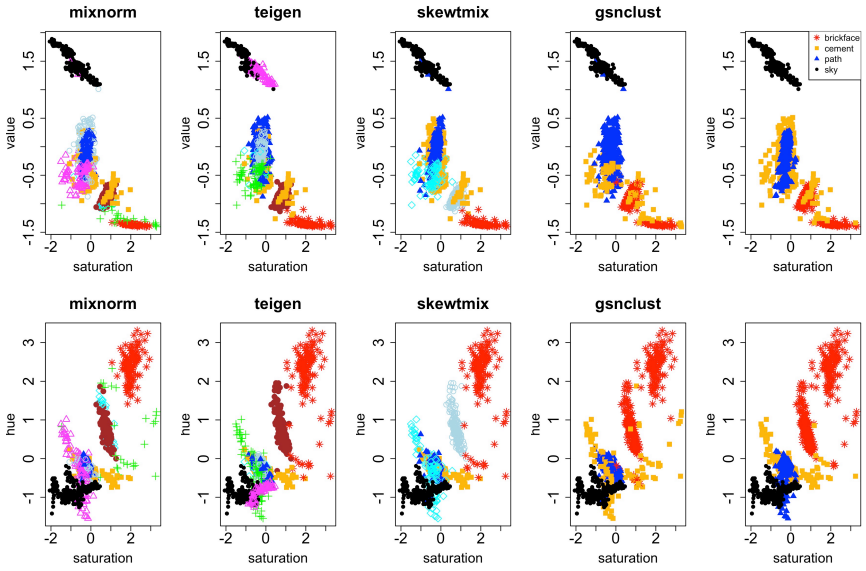

Figure 12: Clustering image segmentation data, with optimal k, according to BIC, for mixtures of (i) Gaussians ($k = 9$) (ii) t ($k = 9$), (iii) skew-t ($k = 6$) and (iv) `gsnclust` ($k = 4$), compared to (v) the true classes of the image blocks.

run 10,000 iterations for this data, ranged from 50 min ($k = 3$) to 130 min ($k = 15$), a rough linear increase with k. The comparative times under differing data and parameter space dimensions are shown in Figure 32 in the online Supplement.

7.3 Photographic image analysis application

The same five methods, as used in Section 7.2, have been applied to the *Baboon* image shown in Figure 9, with 5, 10, 15 and 20 components. The resulting simplified images are shown in Figure 13. From a qualitative point of view we can notice the similarity of the algorithms with $k = 5$ and the fact that the image is already recognisable, albeit the limited number of colours available (corresponding to the number of clusters) is noticeable. With greater values of k, `EMMIXskew` results are very poor in terms of fidelity to the original image, in contrast to all other methods, which already, with 10 components, capture most semantically recognisable features of the original image (Figure 9). This means that even a relatively low number of cluster colours is enough to output a good simplification of the original, with any increase giving only marginal benefits, barely noticeable from a qualitative point of view.

To more formally judge the quality of the renditions, for every pixel in the output image, the sum of squared deviations from the original RGB colours has been computed. By plotting the output images according to this error, we can observe the areas of the image that have been more poorly rendered. These error plots are shown in Figure 14, which refer to the output images of Figure 13. These pixel errors have been computed using the standardised data set. From Figure 14 we can notice how the error is not uniform across the image and it affects in particular contours and some areas, where the colour is not exact, such as the eyes. In general the magnitude of the errors seems to decrease as k increases.

The pixel errors (sum of squared deviations over the three RGB variables for every pixel) of an output image have been summed in order to produce a summary error metric, similar to the MSE, as it is the sum of squared deviations over the sample–in this case the pixels of the image. In Table 4, the MSE results are shown. In addition to the *Baboon* image shown in Figure 13, and referred to as "Original", the metrics resulting from applying the same methods to three modifications of the *Baboon* image are also included. Two of the modified images have been derived by adding random Gaussian noise to the initial image, with standard deviations of 30 and 2–these are referred to as 'Noise (sd=30)' and 'Noise (sd=2)' in Table 4. The rationale behind these modifications is to assess performance in image reconstruction when the input image

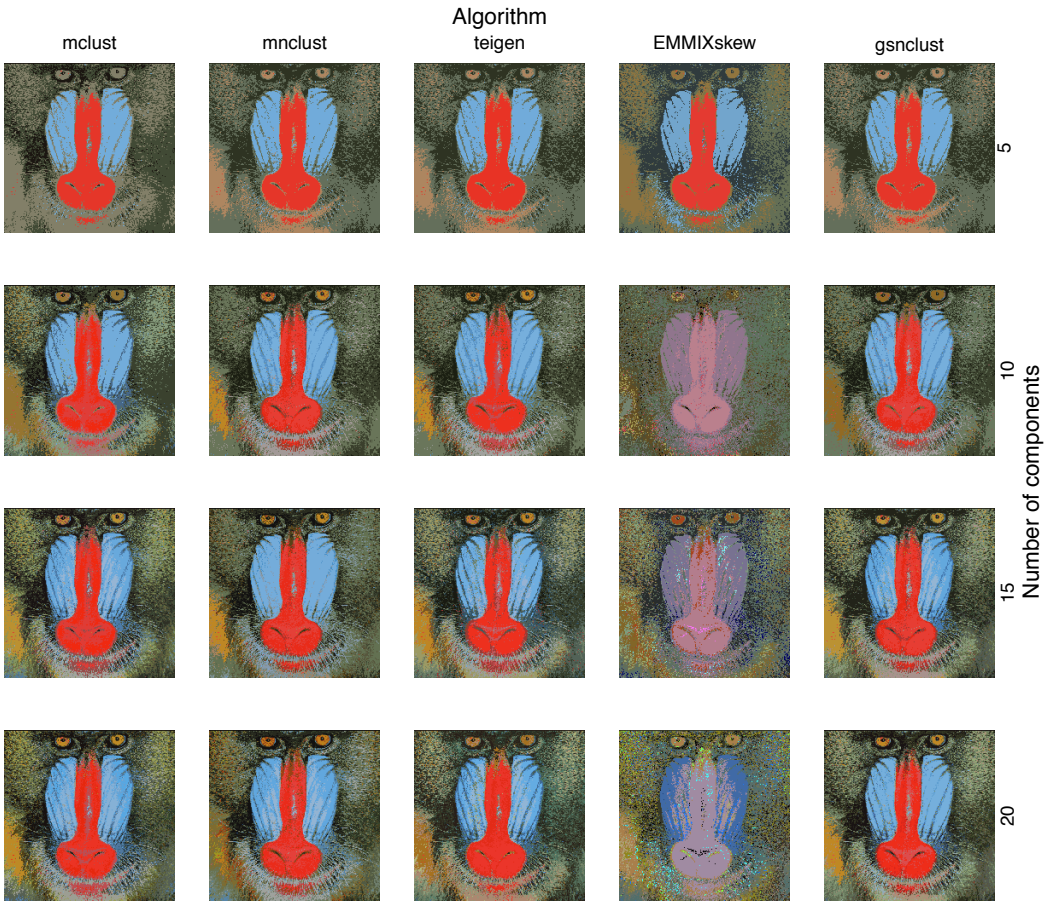

Figure 13: Renditions of the *Baboon* image in Figure 9 given by `mclust`, `mnclust`, `teigen`, `EMMIXskew` and `gsnclust` (columns), with the number of components being 5, 10, 15 and 20 (rows).

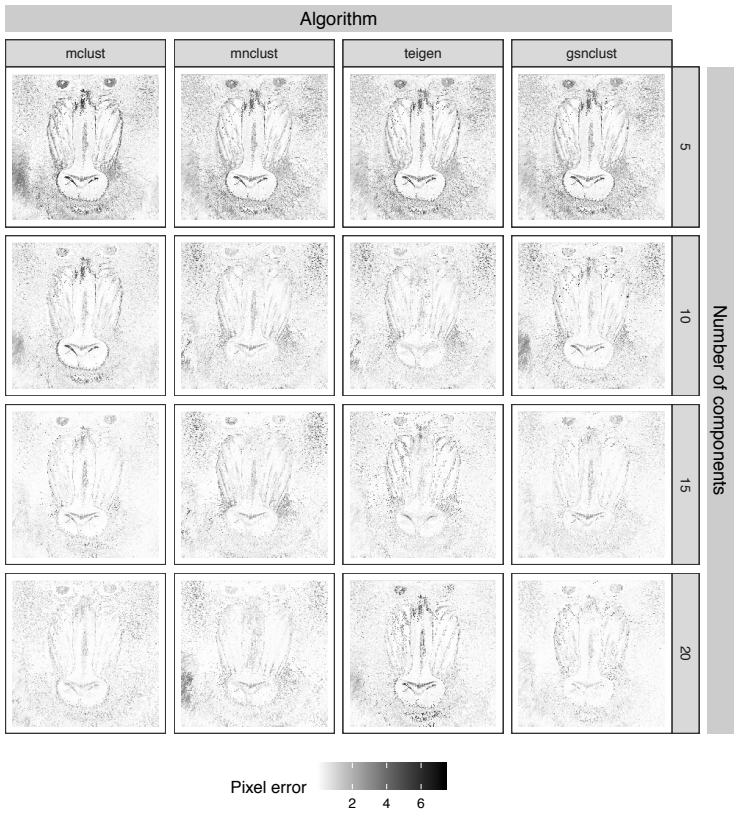

Figure 14: Images representing the error of the renditions of Figure 13. Each pixel of 200×200 matrices has been coloured based the sum of the squared differences with the original image over the 3 variables–RGB intensities. (The scale is that of the standardised variables.) `EMMIXskew` has been excluded from this plot because, given the scale of magnitude of its errors, its presence would have made it difficult to compare the other methods.

data is a deteriorated version of the original. With the same goal of evaluating image recovery the *Baboon* image has also been blurred (via the `isoblur` function in the R package `imager`); this modified image is referred to as 'Blurred' in Table 4. The visual output, including the input image with pairs plots, output images and error plots, are presented in the online Supplement, Figures 22–31. The errors and MSE for the modified images have been computed with respect to the original *Baboon* image (Figure 9).

Table 4: Summary error metric contrasting the performance of the 5 algorithms for the *Baboon* image in its original form and for three other modifications.

k	`mclust`	`mnclust`	`teigen`	`EMMIXskew`	`gsnclust`	
5	32862	31982	32281	38007	31987	
10	20930	21213	20849	98927	22035	
15	12498	17956	17421	91198	14053	Original
20	15128	15803	19294	270535	12462	
5	32920	44641	37872	68740	44640	
10	28919	37578	36953	54469	36499	
15	30270	32359	30184	44205	33885	Noise (sd=30)
20	27000	29365	22944	39654	28844	
5	29817	32278	32214	38802	32308	
10	18842	21326	18324	74926	21371	
15	13904	13746	15772	68307	13266	Noise (sd=2)
20	16217	12943	12307	81997	15218	
5	57433	57356	58742	58509	57292	
10	46574	46134	46606	55594	46169	
15	46422	42454	46501	52424	45332	Blurred
20	45732	42127	43766	49655	42318	

Regarding performance, the best result in terms of MSE is reached by `gsnclust` with $k = 20$ in the original image, with the next best result for `mclust` with $k = 15$. As already noted for this image all methods except for `EMMIXskew` perform reasonably well. It is worth noting that increasing the number of components does not always work in obtaining a better image as the MSE for both `mclust` and `mnclust` increases when k goes from 15 to 20. With the two noisy images the similarity of the algorithms in terms of MSE is confirmed, with the difference that `teigen` reaches a better result than the other algorithms

with $k = 20$ for the more noisy (sd=30) image. The blurred image is the only one where `EMMIXskew` achieves results similar to the other algorithms, but again the best results are reached by `mnclust` and `gsnclust` when $k = 20$. Figure 31 in the online Supplement shows the slight differences in the features of clustering between the 5 methods (more pronounced in the case of `EMMIXskew`), for the case of the noisy *Baboon* image (sd=2) when $k = 20$. For this same case the cluster assignments have been compared by computing ARI and VI; the two matrices giving the pairwise comparisons between methods for the two metrics are shown in Table 13 in the Supplement. These measures confirm the presence of similarity among the clustering solutions, with the mean ARI, excluding `EMMIXskew`, being 0.46.

Algorithms `mclust` and `teigen`, by default, fit multiple models within their respective families and choose the best in terms of BIC. This can be seen as an advantage with respect to other models that only fit one model, thus `mclust` and `teigen` have been set to only run their most unrestricted model (denoted respectively 'VVV' and 'UUUU'). For `teigen` however, this model did not converge—even when allowing for very high tolerances in the convergence parameters—for the noisy image with sd=30 and $k = 10, 15, 20$ and thus the least restrictive models which attained convergence has been used ('CUUU', 'UCCC', and 'CCCU', respectively).

8 Summary and discussion

This paper explores a recently introduced family of distributions, the geometric skew normal (GSN), which has the ability to describe a wide variety of data, with features such as asymmetry and multimodality. A Bayesian approach to fit such models through a hybrid MCMC algorithm is proposed. Further, the model is expanded to a mixture of GSN (GSNM) distributions, providing a model-based clustering methodology for non-normally distributed data. The methodology was tested in simulation studies and applications to real datasets from the fields of genomics and image analysis, where it performed as well as, or better than, four existing methods for clustering based on normal, t and skew-t-distributions, and with higher-dimensional datasets, was able to differentiate features that others failed to detect. It is important to note that multimodality in the GSN almost always ensures that the primary mode of the component has much higher mass than secondary modes, in principle allowing for "neighborhoods" of data within a cluster that are similar to the central part of the distribution. The flexibility of the GSN in capturing asymmetry and multimodality comes with a slightly higher computational cost compared to fitting normal mixtures, but with increasingly

powerful computing infrastructure, this should not be a limitation.

The motivation for using the GSN mixtures is three-fold: first, it provides a natural way, based on two well-known distributions: the Gaussian and geometric, to handle a range of data features, from symmetric and Gaussian to skewed and multimodal. Second, due to the underlying Gaussian framework, it is identifiable, which in turn leads to stability in MCMC-based estimation. It also reduces to a Gaussian mixture distribution as a limiting case. It may be argued that the multimodality allowed for in the GSN is limited to a specific type, being dependent on cluster means. Nevertheless, it appeared that in a number of applications, where the within-component distributions appeared multimodal, the approach was able to detect scientifically interpretable clusters without over-fitting, and provided inference that was more satisfactory than those obtained by other approaches. Third, it appeared that, with a moderate increase in dimension (the highest dimension considered here was 5, in the image segmentation data set), the MCMC approach used to fit the GSNM was stable enough to detect cluster-specific features with a lower number of clusters, in contrast to some of the EM-based methods, where even increasing the number of clusters was not sufficient to improve performance in cluster allocation, and in addition, made the estimation procedure unstable. The GSN mixture, by allowing for deviations from symmetry and unimodality within the cluster, appeared in these cases more likely to allocate observations to correct clusters. This result needs to be interpreted with caution, however, as there is no guarantee that if the data dimension is very high, the GSNM will still continue to perform well, especially as an increase in dimensionality and the size of the parameter space is a known obstacle to MCMC convergence, to which the GSNM model-fitting algorithm, like any other MCMC algorithm, is likely to be prone. Furthermore, as with any numerical study, the interpretation of the results need to be restricted to scenarios that are similar to those that we assessed, and our findings may not hold outside the scope of the studied examples.

Although there are similarities, there is also a clear difference between the usual hierarchical construction and the GSN mixture. Firstly, a hierarchical mixture model is defined as a finite mixture of finite mixtures of probability models (Jordan and Jacobs, 1994). Each of the probability models in the hierarchical construction are unrestricted and thus the final probability model is highly unidentifiable, as one may switch probability models between branches of the mixture tree and not modify the probability measure that is constructed. Furthermore, the model is finite in nature, and there are only a finite number of leaf probability models in the tree, although the number of branches may take on any arbitrary structure. The models that are ultimately constructed lack identifiability and the estimation of such models is almost guaranteed to be only

locally optimal due to the large number of parameters required to characterize the large tree structure. The GSN mixture can be considered as a restricted two-layer infinite mixture of Gaussian probability models. This is due to the fact that the GSN models can be viewed as infinite mixtures of Gaussian components, where each component has a mean and probability that must scale with the geometric law. Since we have the restriction on the mean and probability of each component, one can establish identifiability of each of the infinite number of components of a GSN model to one of its top layer components due to the model identifiability. This guarantees that the models have a manageable structure that can be used to simplify estimation and inference, as we have leveraged throughout this work.

Although the GSNM appears to provide a promising framework for clustering in complex datasets, it is important to be aware of several caveats and limitations in its application. As an MCMC algorithm to fit a complex model, convergence needs to be monitored carefully to make sure that the results are valid for posterior inference. Even though the model structure admits Gaussian mixtures as a limiting case, the computational complexity compared to fitting Gaussian mixtures is higher, due to the additional geometric parameter, and therefore it may not be profitable to use the GSNM unless it presents a significant improvement to a normal mixture model fit. There also is more work needed to establish an ideal criterion for judging an appropriate number of clusters for a GSNM. When the number of clusters was presumed unknown, it was shown that the BIC or WAIC could usually detect the correct number of clusters within the GSN mixture model. However, more detailed simulation studies also demonstrated that these model selection criteria, in Bayesian mixture models, may not always give the most parsimonious choices in practice. More research is needed in this direction before any one criterion can be generally recommended for use.

The flexibility of the GSN model leads to questions on whether other families of distributions, based on infinite mixtures of Gaussians through convolution, can be effectively used for data with features that cannot be accounted for by existing models, while at the same time preserving mathematical tractability and interpretability. A possible extension in this direction was shown through a recent proposal of Roozegar and Nadarajah (2017), who consider replacing the geometric model by a power distribution. The GSN uses a geometric model in the convolution of distributions, however, this assumption can be conceivably extended further, with the aim of capturing other aspects of cluster components that may be of scientific interest.

Acknowledgments

The authors are grateful to Douglas Kiel and David Karasik for the genotyping application, and to Florence Forbes, Alessandra Guglielmi, Sonia Petrone, Vincent Macaulay, and Debasis Kundu, for helpful discussions. Hien Nguyen is supported by the Australian Research Council grants DE170101134 and DP180101192.

References

Amendola, C., Engström, A., and Haase, C. (2019). Maximum number of modes of Gaussian mixtures. *Information and Inference: A Journal of the IMA*. iaz013.

Andrews, H. and Patterson, C. L. (1976). Singular value decomposition (SVD) image coding. *IEEE transactions on Communications*, 24(4):425–432.

Andrews, J. L. and McNicholas, P. D. (2012). Model-based clustering, classification, and discriminant analysis via mixtures of multivariate t-distributions. *Stat. Comput.*, 22(5):1021–1029.

Argiento, R., Cremaschi, A., and Guglielmi, A. (2014). A density-based algorithm for cluster analysis using species sampling Gaussian mixture models. *J. Comput. Graph. Statist.*, 23(4):1126–1142.

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

Browne, R. and McNicholas, P. (2015). A mixture of generalized hyperbolic distributions. *Canadian J. Stat.*, 43(2):176–198.

Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Trans. Pattern Anal. Mach. Intell.*, 17(8):790799.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Stat. Soc. B*, 39(1):1–38.

Diebolt, J. and Robert, C. P. (1994). Estimation of finite mixture distributions through Bayesian sampling. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 56(2):pp. 363–375.

Drton, M. and Plummer, M. (2017). A Bayesian information criterion for singular models. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 79(2):323–380.

Dua, D. and Graff, C. (2017). UCI machine learning repository. `http://archive.ics.uci.edu/ml`.

Einasto, M., Vennik, J., Nurmi, P., Tempel, E., Ahvensalmi, A., Tago, E., Liivamägi, L. J., Saar, E., Heinämäki, P., Einasto, J., and Martínez, V. J. (2012). Multimodality in galaxy clusters from SDSS DR8: substructure and velocity distribution. *A&A*, 540:A123.

Escobar, M. D. and West, M. (1995). Bayesian density estimation and inference using mixtures. *J. Amer. Statist. Assoc.*, 90(430):577–588.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD96, page 226231. AAAI Press.

Estrada, K., Rivadeneira, F., and et al. (2012). Genome-wide meta-analysis identifies 56 bone mineral density loci and reveals 14 loci associated with risk of fracture. *Nat. Genet.*, 44(5):491–501.

Everitt, B. (1974). *Cluster analysis.* Heinemann Educational Publishers, London.

Forbes, F. and Wraith, D. (2014). A new family of multivariate heavy-tailed distributions with variable marginal amounts of tailweight: application to robust clustering. *Stat. Comput.*, 24(6):971–984.

Fraley, C. and Raftery, A. E. (2002). Model-based clustering, discriminant analysis, and density estimation. *J. Amer. Statist. Assoc.*, 97(458):611–631.

Friel, N., McKeone, J. P., Oates, C. J., and Pettitt, A. N. (2017). Investigation of the widely applicable Bayesian information criterion. *Stat. Comput.*, 27(3):833–844.

Fruhwirth-Schnatter, S. and Pyne, S. (2010). Bayesian inference for finite mixtures of univariate and multivariate skew-normal and skew-t distributions. *Biostatistics*, 11(2):317–336.

Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., and Rubin, D. B. (2014). *Bayesian Data Analysis*. Texts in Statistical Science Series. Chapman & Hall, London, 3 edition.

Hennig, C. (2010). Methods for merging Gaussian mixture components. *Adv. Data Anal. Classif.*, 4(1):3–34.

Hubert, L. and Arabie, P. (1985). Comparing partitions. *J. Classif.*, 2(1):193–218.

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

Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *J. Amer. Statist. Assoc.*, 90(430):773–795.

Kundu, D. (2014). Geometric skew normal distribution. *Sankhya B*, 76(2):167–189.

Kundu, D. (2017). Multivariate geometric skew-normal distribution. *Statistics*, 51(6):1377–1397.

Lampert, A. and Tlusty, T. (2013). Resonance-induced multimodal body-size distributions in ecosystems. *Proceedings of the National Academy of Sciences*, 110(1):205–209.

Lee, S. X. and McLachlan, G. J. (2013). On mixtures of skew normal and skew t-distributions. *Adv. Data Anal. Classif.*, 7(3):241–266.

Li, J. (2005). Clustering based on a multilayer mixture model. *J. Comput. Graph. Statist.*, 14(3):547–568.

Li, J., Ray, S., and Lindsay, B. (2007). A nonparametric statistical approach to clustering via mode identification. *Journal of Machine Learning Research*, 8:1687–1723.

Lin, Y., Tseng, G. C., Cheong, S. Y., Bean, L. J., Sherman, S. L., and Feingold, E. (2008). Smarter clustering methods for SNP genotype calling. *Bioinformatics*, 24(23):2665–2671.

Malsiner-Walli, G., Frühwirth-Schnatter, S., and Grün, B. (2017). Identifying mixtures of mixtures using Bayesian estimation. *J. Comput. Graph. Statist.*, 26(2):285–295.

Marin, J.-M. and Robert, C. P. (2007). *Bayesian Core: A Practical Approach to Computational Bayesian Statistics (Springer Texts in Statistics)*. Springer-Verlag, Berlin.

Mascini, N. E., Teunissen, J., Noorlag, R., Willems, S. M., and Heeren, R. M. A. (2018). Tumor classification with MALDI-MSI data of tissue microarrays: A case study. *Methods*, 151:21–27.

McLachlan, G. and Peel, D. (2000). *Finite mixture models*. Wiley-Interscience, Hoboken.

Meilă, M. (2003). Comparing clusterings by the variation of information. In Schölkopf, B. and Warmuth, M. K., editors, *Learning Theory and Kernel Machines*, pages 173–187. Springer Berlin.

O'Hagan, A., Murphy, T. B., Gormley, I. C., McNicholas, P. D., and Karlis, D. (2016). Clustering with the multivariate normal inverse gaussian distribution. *Comput. Stat. Data Anal.*, 93:18–30.

Raftery, A., Newton, M., M. Satagopan, J., and Krivitsky, P. (2007). Estimating the integrated likelihood via posterior simulation using the harmonic mean identity. *Bayesian Statistics*, 8:1–45.

Ray, S. and Ren, D. (2012). On the upper bound of the number of modes of a multivariate normal mixture. *J. Multivariate Anal.*, 108:41 – 52.

Richards, J. A. (2012). *Remote Sensing Digital Image Analysis: An Introduction*. Springer Publishing Company, Incorporated, 5th edition.

Roozegar, R. and Nadarajah, S. (2017). The power series skew normal class of distributions. *Commun. Stat-Theor. M.*, 46(22):11404–11423.

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

Teh, Y. W., Jordan, M. I., Beal, M. J., and Blei, D. M. (2006). Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

Thiem, S., Kentner, D., and Sourjik, V. (2007). Positioning of chemosensory clusters in E. coli and its relation to cell division. *The EMBO Journal*, 26(6):1615–1623.

van der Vaart, A. W. (1998). *Asymptotic Statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge.

Vrbik, I. and McNicholas, P. D. (2014). Parsimonious skew mixture models for model-based clustering and classification. *Comp. Stat. Data Anal.*, 71:196 – 210.

Wang, K., Ng, S., and McLachlan, G. (2009). Multivariate skew t mixture models: applications to fluorescence-activated cell sorting data. In Shi, H., Zhang, Y., Bottema, M., Lovell, B., and Maede, A., editors, *Conference of Digital Image Computing: Techniques and Applications, Melbourne*, pages 526–531. Los Alamitos, California: IEEE Computer Society.

Watanabe, S. (2010). Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *J. Mach. Learn. Res.*, 11:3571–3594.

Yang, L. and Wu, X. (2014). A new sufficient condition for identifiability of countably infinite mixtures. *Metrika*, 77(3):377–387.

Zio, M. D., Guarnera, U., and Rocci, R. (2007). A mixture of mixture models for a classification problem: The unity measure error. *Comp. Stat. Data Anal.*, 51(5):2573 – 2585.