NIASRA

NATIONAL INSTITUTE FOR APPLIED STATISTICS RESEARCH AUSTRALIA

National Institute for Applied Statistics Research Australia

University of Wollongong, Australia

Working Paper

06-23

Mixture Modeling with Normalizing Flows for Spherical Density Estimation

Tin Lok James Ng and Andrew Zammit-Mangion

National Institute for Applied Statistics Research Australia, University of Wollongong, Wollongong NSW 2522, Australia T: +61 2 42215076. E: karink@uow.edu.au

Mixture Modeling with Normalizing Flows for Spherical Density Estimation

Tin Lok James Ng[1] and Andrew Zammit-Mangion[2]

[1]School of Computer Science and Statistics, Trinity College Dublin, Ireland

[2]School of Mathematics and Applied Statistics, University of Wollongong, Australia

Abstract

Normalizing flows are objects used for modeling complicated probability density functions, and have attracted considerable interest in recent years. Many flexible families of normalizing flows have been developed. However, the focus to date has largely been on normalizing flows on Euclidean domains; while normalizing flows have been developed for spherical and other non-Euclidean domains, these are generally less flexible than their Euclidean counterparts. To address this shortcoming, in this work we introduce a mixture-of-normalizing-flows model to construct complicated probability density functions on the sphere. This model provides a flexible alternative to existing parametric, semiparametric, and nonparametric, finite mixture models. Model estimation is performed using the expectation maximization algorithm and a variant thereof. The model is applied to simulated data, where the benefit over the conventional (single component) normalizing flow is verified. The model is then applied to two real-world data sets of events occurring on the surface of Earth; the first relating to earthquakes, and the second to terrorist activity. In both cases, we see that the mixture-of-normalizing-flows model yields a good representation of the density of event occurrence.

1 Introduction

Finite mixture models are widely used to model and analyze heterogeneous data. Of the several variants that appear in the literature, the parametric finite mixture models are the most popular; efficient estimation strategies to fit them to data are widely available, and their theoretical properties are well understood. It is common to model the mixture components using flexible density functions; for example both finite mixtures of skew normal densities as well as finite mixtures of t-densities have been used (Frühwirth-Schnatter and Pyne, 2010; Lin et al., 2014; Hejblum et al., 2019), while on the sphere both von Mises-Fisher densities and Kent densities have been used (Banerjee et al., 2005; Peel et al., 2001; Gopal and Yang, 2014) as building blocks. The mixture component densities of these models are reasonably flexible, however they still rely on strong model assumptions that may not be satisfied in some practical settings. Semiparametric and nonparametric finite mixture models have been proposed in order to relax the strong assumptions implicit to fully parametric mixture models. The semiparametric and nonparametric models replace one or more

of the mixture components by a nonparametric density with possible constraints such as symmetry and log-concavity (Chang and Walther, 2007; Hunter et al., 2007; Bordes and Vandekerkhove, 2010; Levine et al., 2011). The process of fitting semiparametric and nonparametric mixture models tends to be more computationally intensive than that of fitting parametric ones.

An alternative, straightforward mechanism for modeling complicated probability distributions is the normalizing flow. Normalizing flows only require the specification of a simple "base" distribution (e.g., a standard normal or a uniform distribution) and a series of invertible and differentiable transformations (see, e.g., Papamakarios et al., 2021). The density of a sample can be evaluated by computing the product of the density of the transformed sample under the base distribution and the associated change in volume induced by the series of transformations. The latter term is the product of the absolute Jacobian determinants for each transformation. Many flexible families of normalizing flows on the Euclidean space have been developed (Kobyzev et al., 2020), some of which exhibit a "universal property" (Huang et al., 2018; Jaini et al., 2019; Ng and Zammit-Mangion, 2023), in the sense that they can be used to approximate a large class of density functions arbitrarily well.

Despite their popularity and efficacy, normalizing flows have a weakness: they require large and/or deep architectures to approximate complex target distributions with arbitrary precision (Cornish et al., 2020). To address this shortcoming, Izmailov et al. (2020) proposed to model the reference density function as a mixture of Gaussian density functions with unknown mean and covariance parameters, while Dinh et al. (2019) proposed a framework involving domain partitioning and locally invertible functions. In the latter approach, the transport map is no longer required to be fully invertible but only piecewise invertible, leading to greater flexibility. Ciobanu (2021) took a different approach, and proposed using a mixture-of-normalizing-flows model, where each component of the mixture model is a density parameterized by a normalizing flow with its own parameters. Pires and Figueiredo (2020) proposed a variational mixture-of-normalizing-flows model, which combines the flexibility of normalizing flows with the ability to exploit class-membership structure. Model fitting is done via optimization of a variational objective, for which the variational posterior over class membership latent variables is parameterized by a neural network. While normalizing flows have been extensively studied and utilized on the Euclidean domain, there are several cases where the data should be treated and analyzed as elements of a non-Euclidean manifold. A popular example is directional statistics which involves the analysis of data on the unit sphere, and this is the case we focus on in this work. Spherical data arise in many application domains including gene expression analysis (Banerjee et al., 2005), protein bioinformatics (Mardia et al., 2022), and astronomy (Jupp, 1995). While normalizing flows on the sphere and more general manifolds have been considered (e.g., Gemici et al., 2016; Rezende et al., 2020), they are generally less flexible than their Euclidean counterparts. This is a direct consequence of the difficulties that arise when working on arbitrary manifolds with complex geometric structure, which in turn leads to both modeling and computational challenges.

To address these limitations, in this work we propose adapting the mixtures-of-normalizing-flows models that have been developed for the Euclidean domain (Pires and Figueiredo, 2020; Ciobanu, 2021) for use on a widely used non-Euclidean manifold: the sphere. Specifically, we develop a mixture modeling framework where each mixture component is a spherical normalizing flow. Our spherical normalizing flows are constructed using exponential map flows (Sei, 2013; Rezende et al., 2020; Ng and Zammit-Mangion, 2022);

however, the proposed framework can be adapted to any spherical normalizing flow. We give the requisite background on normalizing flows on Euclidean spaces and spheres in Section 2. In Section 3 we then present the proposed mixture modeling framework along with the expectation-maximization (EM) algorithm (and a variant thereof) for model estimation, which we implement efficiently using mini-batch stochastic gradient descent. In this section we also briefly discuss approaches for mixture-components-order selection. We showcase the proposed methodology through a simulation study, density estimation of earthquake events, and density estimation of terrorist activity on the surface of Earth in Section 4. Section 5 discusses potential future research directions and concludes.

2 Background

Given two probability measures $\mu_0(\cdot)$ and $\mu_1(\cdot)$ defined on spaces $\mathcal{X}$ and $\mathcal{Z}$, respectively, a transport map $T : \mathcal{X} \to \mathcal{Z}$ is said to push forward $\mu_0(\cdot)$ to $\mu_1(\cdot)$ if, for any Borel subset $B \subset \mathcal{Z}$,

$$\mu_1(B) = \mu_0(T^{-1}(B)), \tag{1}$$

where the inverse $T^{-1}(\cdot)$ is set valued; specifically, $T^{-1}(\boldsymbol{z}) = \{\boldsymbol{x} \in \mathcal{X} : T(\boldsymbol{x}) = \boldsymbol{z}\}$. For an injective transport map $T(\cdot)$, (1) can be re-formulated as $\mu_0(A) = \mu_1(T(A))$, for any Borel subset $A \subset \mathcal{X}$. Suppose that the measures $\mu_0(\cdot), \mu_1(\cdot)$ are absolutely continuous with respect to the Lebesgue measure, with densities $f_0(\cdot)$ and $f_1(\cdot)$, respectively. If the map $T(\cdot)$ is bijective with a differentiable inverse $T^{-1}(\cdot)$, we obtain the familiar change-of-variables formula

$$f_0(\boldsymbol{x}) = f_1(T(\boldsymbol{x}))|\det(\nabla T(\boldsymbol{x}))|, \quad \boldsymbol{x} \in \mathcal{X}, \tag{2}$$

which expresses a complicated probability density $f_0(\cdot)$ in terms of a simpler density $f_1(\cdot)$ and a transport map $T(\cdot)$. The reference density $f_1(\cdot)$ typically has no unknown parameter, and the multivariate standard normal density and the uniform density on a compact domain are common choices. Marzouk et al. (2016) discuss various strategies for parameterizing the transport map $T(\cdot)$, which has also been done using deep learning models (e.g., Papamakarios et al., 2017; Kobyzev et al., 2020). It has been proved that under mild conditions arbitrarily complex probability density functions can be well approximated by neural network based transport maps (Huang et al., 2018; Ng and Zammit-Mangion, 2023). Recent approaches introduce more flexibility by defining $T(\cdot)$ as a composition of multiple transformations, that is, as $T(\cdot) \equiv T^{(K)} \circ \cdots \circ T^{(1)}(\cdot)$, where $T^{(k)}(\cdot)$ transforms $\boldsymbol{z}^{(k-1)}$ into $\boldsymbol{z}^{(k)}$, with $\boldsymbol{z}^{(0)} \equiv \boldsymbol{x}$ and $\boldsymbol{z}^{(K)} \equiv \boldsymbol{z}$. The composition of multiple transformations is called a normalizing flow in the machine learning literature. Given two bijective maps $T^{(1)}(\cdot), T^{(2)}(\cdot)$ with differentiable inverses, their composition $T^{(2)} \circ T^{(1)}(\cdot)$ remains bijective with a differentiable inverse. The Jacobian determinant of the resulting composition remains computationally tractable since, by the chain rule,

$$\det(\nabla(T^{(2)} \circ T^{(1)}(\boldsymbol{x}))) = \det(\nabla T^{(1)}(\boldsymbol{x}))\det(\nabla T^{(2)}(T^{(1)}(\boldsymbol{x}))), \quad \boldsymbol{x} \in \mathcal{X}.$$

While normalizing flows have largely been studied and used in Euclidean spaces, data are often naturally described on Riemannian manifolds such as spheres and tori. A number of normalizing flow techniques for Riemannian manifolds have been proposed; some of these, such as the technique proposed by Gemici et al. (2016), involve projecting to the Euclidean space before projecting back to the original manifold. These

approaches, however, lead to singularities if the manifold is not diffeomorphic to $\mathbb{R}^d$, as in the case of the sphere. Projections can be avoided by constructing normalizing flows directly on the manifold of interest. Rezende et al. (2020) proposed constructing normalizing flows on spheres and tori using Möbius transformations and spherical splines. Mathieu and Nickel (2020) proposed constructing continuous normalizing flows for Riemannian manifolds by solving ordinary differential equations on manifolds. Here, we focus on the exponential map flow, which was first proposed by Sei (2013), then extended by Rezende et al. (2020), and then adapted to a (spherical) spatial point process setting by Ng and Zammit-Mangion (2022).

Let $\phi(\cdot)$ be a *wrapping potential function* (see Sei (2013) for a definition), and let $\exp_{\boldsymbol{x}}(\cdot)$ denote the *exponential map*; then $\exp_{\boldsymbol{x}}(\nabla\phi(\boldsymbol{x}))$ is a valid exponential map flow for $\boldsymbol{x} \in \mathbb{S}^{d-1}$. Let $p \in \mathbb{Z}^+$, $\beta_i > 0, \boldsymbol{m}_i \in \mathbb{S}^{d-1}$, and $\eta_i > 0$ for $i = 1, \ldots, p$, such that $\sum_{i=1}^{p} \eta_i = 1$. The wrapping potential function we use in this work is given by (Rezende et al., 2020; Ng and Zammit-Mangion, 2022)

$$\phi(\boldsymbol{x}) = \sum_{i=1}^{p} \frac{\eta_i}{\beta_i} e^{\beta_i(\cos d(\boldsymbol{x}, \boldsymbol{m}_i)-1)}, \quad \boldsymbol{x} \in \mathbb{S}^{d-1}, \tag{3}$$

where $\beta_i, \boldsymbol{m}_i$ and $\eta_i, i = 1, \ldots, p$, are model parameters that need to be estimated. For an intuitive description of exponential maps and their behavior on the sphere, see Ng and Zammit-Mangion (2022). A normalizing flow on the sphere can be constructed by stringing together several exponential map flows of the form $\exp_{\boldsymbol{x}}(\nabla\phi(\boldsymbol{x}))$ through composition.

We note that while we restrict ourselves to the sphere, Cohen et al. (2021) has generalized the exponential map flow to arbitrary Riemannian manifolds. While the parameterization of the normalizing flows in Cohen et al. (2021) leads to a universal property where arbitrary c-concave functions on compact manifolds can be approximated arbitrarily well, their construction leads to a piecewise smooth map which is not differentiable everywhere, and hence difficult to fit in practice.

3 Methodology

3.1 Mixture-of-normalizing-flows model for spherical data

We consider the case where the target measure $\mu_0(\cdot)$ and reference measure $\mu_1(\cdot)$ admit densities with respect to the Lebesgue measure on $\mathbb{S}^{d-1}$, and model the target density as a mixture of G component densities. Let $f_{0,g}(\cdot)$ be the g-th mixture component of the target density of interest, and τ_g the corresponding weight, where $\tau_g \geq 0, g = 1, \ldots, G$, and $\sum_{g=1}^{G} \tau_g = 1$. We model $f_0(\cdot)$ as

$$f_0(\boldsymbol{x}) = \sum_{g=1}^{G} \tau_g f_{0,g}(\boldsymbol{x}), \quad \boldsymbol{x} \in \mathbb{S}^{d-1},$$

where each $f_{0,g}(\cdot), g = 1, \ldots, G$, is constructed using a normalizing flow. For ease of exposition we consider the case where the normalizing flows have the same functional form, but different parameters; that is, the case where $f_{0,g}(\cdot) \equiv f(\cdot\,; \Theta_g), g = 1, \ldots, G$, where Θ_g is the set of parameters corresponding to the g-th component. Let the reference measure for each component be the uniform density. From (2) we then have that

$$f_{0,g}(\boldsymbol{x}) \equiv f(\boldsymbol{x}; \Theta_g) \propto |\det(\nabla T(\boldsymbol{x}; \Theta_g))|, \quad \boldsymbol{x} \in \mathbb{S}^{d-1},$$

where for $\mathbb{S}^2$ the constant of proportionality is equal to $\frac{1}{4\pi}$, and where we have explicitly notated the dependence of the transport map $T(\cdot)$ on the component-specific parameters Θ_g. Now, consider a set of K transport maps, $T^{(1)}(\cdot\,;\Theta_{g,1}),\ldots,T^{(K)}(\cdot\,;\Theta_{g,K})$ parameterized using parameters collected in $\Theta_g \equiv \{\Theta_{g,1},\ldots,\Theta_{g,K}\}$. We model the transport map for each mixture component as a composition of K maps:

$$T(\boldsymbol{x};\Theta_g) = T^{(K)} \circ \cdots \circ T^{(1)}(\boldsymbol{x};\Theta_g), \quad g = 1,\ldots,G,$$

where each $T^{(k)}(\cdot\,;\Theta_{g,k}) = \exp_{\boldsymbol{x}}(\nabla\phi(\boldsymbol{x};\Theta_{g,k}))$ is an exponential map with wrapping potential function of the form (3). Specifically, this wrapping potential function is given by

$$\phi(\boldsymbol{x};\Theta_{g,k}) = \sum_{i=1}^{p} \frac{\eta_i^{(g,k)}}{\beta_i^{(g,k)}} e^{\beta_i^{(g,k)}(\cos d(\boldsymbol{x},\boldsymbol{m}_i^{(g,k)})-1)}, \quad \boldsymbol{x} \in \mathbb{S}^{d-1}. \tag{4}$$

Therefore, for $g = 1,\ldots,G$, $\Theta_g \equiv \{\beta_i^{(g,k)}, \boldsymbol{m}_i^{(g,k)}, \eta_i^{(g,k)} : k = 1,\ldots,K; i = 1,\ldots,p\}$ are the model parameters for the g-th mixture component that together construct a composition of K radial flows, where the k-th map in the composition has model parameters $\{\beta_i^{(g,k)}, \boldsymbol{m}_i^{(g,k)}, \eta_i^{(g,k)} : i = 1,\ldots,p\}$. Note that since we are assuming that each mixture component has the same functional form, we are fixing the number of layers K and the number of basis functions p that are used for each mixture component; this choice is made for convenience and is not a model requirement. In our previous work (Ng and Zammit-Mangion, 2022), we showed that a small value of p (i.e., $p = 1$ or $p = 2$) and a moderate to large value of K (between 20 and 40) led to good performance in practice; we found that this was the case with our mixture-of-normalizing-flows model as well (see Section 4).

Let $\Theta \equiv \{\Theta_1,\ldots,\Theta_G\}$ and $\boldsymbol{\tau} \equiv (\tau_1,\ldots,\tau_G)'$. The problem of density estimation on the sphere using our mixture-of-normalizing-flows model reduces to the problem of estimating Θ and $\boldsymbol{\tau}$ from data. We do this using the EM and related algorithms, that are often used when fitting mixture models.

3.2 Parameter estimation using the EM Algorithm

In this section we present both the standard EM algorithm, as well as an adaptation of it that is often referred to as the hard EM algorithm (e.g., Samdani et al., 2012). Both algorithms could be used to fit the mixture-of-normalizing-flows model; however the hard EM algorithm is more computationally efficient and offers a solution to the problem of determining the order (i.e., the number of components) of the mixture.

3.2.1 The standard (soft) EM algorithm

Given N observations $\mathbf{X} \equiv (\boldsymbol{x}_1,\ldots,\boldsymbol{x}_N)$ where each $\boldsymbol{x}_j \in \mathbb{S}^{d-1}, j = 1,\ldots,N$, the likelihood function for the unknown parameters is given by

$$L(\Theta,\boldsymbol{\tau};\mathbf{X}) = \prod_{j=1}^{N}\left(\sum_{g=1}^{G}\tau_g f(\boldsymbol{x}_j;\Theta_g)\right). \tag{5}$$

To facilitate estimation with the EM algorithm, we introduce the latent variables $\mathbf{Z} \equiv (\boldsymbol{z}_1,\ldots,\boldsymbol{z}_N)$ that denote the latent assignment of observations to mixture components, where $\boldsymbol{z}_j \equiv (z_{j,1},\ldots,z_{j,G})'$ and $z_{j,g} = 1$ if observation j belongs to the g-th mixture component and $z_{j,g} = 0$ otherwise (see, e.g., Bishop, 2006,

Ch. 9). Then, the so-called complete-data likelihood function is given by

$$L_c(\Theta, \boldsymbol{\tau}; \mathbf{X}, \mathbf{Z}) = \prod_{j=1}^{N} \prod_{g=1}^{G} \big(\tau_g f(\boldsymbol{x}_j; \Theta_g)\big)^{z_{j,g}}. \tag{6}$$

Taking logarithms of (6) we obtain the complete data log-likelihood function:

$$\ell_c(\Theta, \boldsymbol{\tau}; \mathbf{X}, \mathbf{Z}) \equiv \log L_c(\Theta, \boldsymbol{\tau}; \mathbf{X}, \mathbf{Z}) = \sum_{j=1}^{N} \sum_{g=1}^{G} z_{j,g} \big(\log \tau_g + \log f(\boldsymbol{x}_j; \Theta_g)\big). \tag{7}$$

The EM algorithm can now be readily applied to predict the latent mixture component assignments $\mathbf{Z}$ in the E-step, and to estimate the unknown model parameters Θ and $\boldsymbol{\tau}$ in the M-step. At the $(t+1)$-th iteration, the E-step computes $\hat{\pi}_{j,g}^{(t+1)} \equiv \mathbb{P}(z_{j,g} = 1 \mid \boldsymbol{x}_j, \hat{\Theta}^{(t)}, \hat{\boldsymbol{\tau}}^{(t)})$, that is, the probability of the latent assignment for each observation when conditioning on the data $\boldsymbol{x}_j$ and the estimates of the model parameters at the t-th iteration ($\hat{\Theta}^{(t)}$ and $\hat{\boldsymbol{\tau}}^{(t)}$):

$$\hat{\pi}_{j,g}^{(t+1)} = \frac{\hat{\tau}_g^{(t)} f(\boldsymbol{x}_j; \hat{\Theta}_g^{(t)})}{\sum_{h=1}^{G} \hat{\tau}_h^{(t)} f(\boldsymbol{x}_j; \hat{\Theta}_h^{(t)})}, \quad j = 1, \ldots, N; g = 1, \ldots, G. \tag{8}$$

We collect these latent probability parameters in the $N \times G$ matrix $\hat{\boldsymbol{\Pi}}^{(t+1)}$.

At the $(t+1)$-th iteration, the M-step maximizes the conditional expectation of the complete data log-likelihood in (7) with respect to the parameters Θ and $\boldsymbol{\tau}$, where the expectation is taken with respect to the conditional distribution of $\mathbf{Z}$ given the observations $\mathbf{X}$ and parameter estimates $\hat{\Theta}^{(t)}, \hat{\boldsymbol{\tau}}^{(t)}$. This expectation, which we denote by $Q(\{\Theta, \boldsymbol{\tau}\}; \{\hat{\Theta}^{(t)}, \hat{\boldsymbol{\tau}}^{(t)}\})$, is given by

$$Q(\{\Theta, \boldsymbol{\tau}\}; \{\hat{\Theta}^{(t)}, \hat{\boldsymbol{\tau}}^{(t)}\}) = \mathbb{E}\big(\ell_c(\Theta, \boldsymbol{\tau}; \mathbf{X}, \mathbf{Z}) \mid \mathbf{X}, \hat{\Theta}^{(t)}, \hat{\boldsymbol{\tau}}^{(t)}\big) \tag{9}$$

$$= \sum_{j=1}^{N} \sum_{g=1}^{G} \hat{\pi}_{j,g}^{(t+1)} \big(\log \tau_g + \log f(\boldsymbol{x}_j; \Theta_g)\big). \tag{10}$$

The update for τ_g, obtained by maximizing (10) with respect to τ_g, can be derived analytically for $g = 1, \ldots, G$:

$$\hat{\tau}_g^{(t+1)} = \frac{\sum_{j=1}^{N} \hat{\pi}_{j,g}^{(t+1)}}{N}, \quad g = 1, \ldots, G. \tag{11}$$

The optimization of the conditional expected complete data log-likelihood (10) with respect to Θ does not result in a closed-form update rule for $\hat{\Theta}^{(t+1)}$, and therefore the optimization needs to be done numerically. As with $\boldsymbol{\tau}$, the parameters Θ for each mixture component can be updated separately. Specifically, we define

$$Q_g(\Theta_g; \hat{\Theta}_g^{(t)}) \equiv \sum_{j=1}^{N} \hat{\pi}_{j,g}^{(t+1)} \log f(\boldsymbol{x}_j; \Theta_g). \quad g = 1, \ldots, G, \tag{12}$$

and optimize $Q_g(\Theta_g; \hat{\Theta}_g^{(t)})$ with respect to $\Theta_g, g = 1, \ldots, G$. To facilitate this step we use mini-batch stochastic gradient descent (SGD) with automatic differentiation (AutoDiff) in *PyTorch* (Paszke et al., 2017). Mini-batch SGD updates model parameters using small batches of data rather than the entire dataset. Hence, each step performs a descent based on an unbiased estimate of the gradient rather than the exact gradient; however it is more computationally efficient than exact gradient descent, and the introduced

Algorithm 1 Soft (standard) EM Algorithm
Input: $G, \{\boldsymbol{x}_j\}_{j=1}^N$
Output: Estimated parameters $\hat{\Theta}, \hat{\boldsymbol{\tau}}$, and the estimated component probabilities, $\hat{\boldsymbol{\Pi}}$

```
Initialise {Θ̂_g^(0), τ̂_g^(0)}_{g=1}^G
Set t = 0
do
    E-Step
    for j = 1, ..., N do
        for g = 1, ..., G do
            Compute π̂_{j,g}^(t+1) according to (8)
        end for
    end for

    M-Step
    for g = 1, ..., G do
        Compute τ̂_g^(t+1) according to (11)
        Find Θ̂_g^(t+1) by optimizing (12) with respect to Θ_g using SGD with AutoDiff
    end for

    t ← t + 1
while Not Converged

Π̂ ← Π̂^(t)
τ̂ ← τ̂^(t)
Θ̂ ← Θ̂^(t)
```

stochasticity helps to avoid local optima. As in Ng and Zammit-Mangion (2022), we found that SGD works very well with this model. Once estimates for $\Theta_g, g = 1, \ldots, G$, are found, the E-step is repeated, followed by the M-step again, and so on until convergence. We summarise this EM algorithm, which we term the *soft* EM algorithm to contrast it with the *hard* EM algorithm that we discuss next, in Algorithm 1.

3.2.2 The hard EM algorithm

The hard EM algorithm, also known as the classification maximization algorithm, is a practical alternative to the standard (soft) EM algorithm. In the soft EM algorithm, the E-step is used to find the full conditional distribution of $z_{j,g}$, for $j = 1, \ldots, N$ and $g = 1, \ldots, G$. In the hard EM algorithm, this distribution is summarized as a Kronecker delta function centered at the mode of the true conditional distribution. That is, at the $(t+1)$-th iteration, the E-step approximates $\mathbb{P}(z_{j,g} = 1 \mid \boldsymbol{x}_j, \hat{\Theta}^{(t)}, \hat{\boldsymbol{\tau}}^{(t)}) \approx \mathbb{I}(z_{j,g} = \hat{z}_{j,g}^{(t+1)})$, where

$$\hat{z}_{j,g}^{(t+1)} = \begin{cases} 1 & g = \underset{g'}{\operatorname{argmax}} \{\hat{\pi}_{j,g'}^{(t+1)}\}_{g'=1}^G \\ 0 & \text{otherwise,} \end{cases} \tag{13}$$

for $j = 1, \ldots, N$, and $g = 1, \ldots, G$, where $\hat{\pi}_{j,g}^{(t+1)}$ is defined in Equation (8). We collect these estimated mixture component assignments in the $N \times G$ matrix $\hat{\mathbf{Z}}^{(t+1)}$. The M-step is done similarly to the standard EM algorithm, except that now expectations in (9) are taken with respect to this degenerate conditional distribution rather than the full (true) conditional distribution. For the M-step, the update for $\hat{\boldsymbol{\tau}}$ becomes

$$\hat{\tau}_g^{(t+1)} = \frac{N_g^{(t+1)}}{N}, \quad g = 1, \ldots, G, \tag{14}$$

Algorithm 2 Hard EM Algorithm
Input: $G, \{\boldsymbol{x}_j\}_{j=1}^N$,
Output: Estimated parameters $\hat{\Theta}, \hat{\boldsymbol{\tau}}$, and the estimated mixture component assignments, $\hat{\mathbf{Z}}$

```
Initialise {Θ̂_g^(0), τ̂_g^(0)}_{g=1}^G
Set t = 0
do
    E-Step
    for i = 1, ..., N do
        for g = 1, ··· , G do
            Compute π̂_{j,g}^(t+1) according to (8)
            Compute ẑ_{j,g}^(t+1) according to (13)
        end for
    end for

    M-Step
    for g = 1, ..., G do
        Compute τ̂_g^(t+1) according to (14)
        Find Θ̂_g^(t+1) by optimizing (15) with respect to Θ_g using SGD with AutoDiff
    end for

    t ← t + 1
while Not Converged

Ẑ ← Ẑ^(t)
τ̂ ← τ̂^(t)
Θ̂ ← Θ̂^(t)
```

where $N_g^{(t+1)} \equiv |\{j : \hat{z}_{j,g}^{(t+1)} = 1\}|$ is the number of observations allocated to the g-th mixture component. The update $\hat{\Theta}^{(t+1)}$ can be done separately for each mixture component g by optimizing the following objective function

$$Q_g(\Theta_g; \hat{\Theta}_g^{(t)}) = \sum_{\left\{j:\hat{z}_{j,g}^{(t+1)}=1\right\}} \log f(\boldsymbol{x}_j; \Theta_g) \tag{15}$$

with respect to $\Theta_g, g = 1, \ldots, G$. Note how (15) involves a summation over $N_g^{(t+1)}$ rather than over N points as in (12). This leads to a simpler optimization with SGD that in turn leads to improved computational efficiency (and, in practice, also more stable estimates). Furthermore, the hard EM algorithm offers a simple solution to the problem of determining the number of mixture components; this is discussed in Section 3.3. The hard EM algorithm is summarized in Algorithm 2.

3.3 Order Selection

Having a strategy to determine the number of mixture components G in a finite mixture model is important. Many frequentist and Bayesian approaches have been proposed to select the optimal G for finite mixtures of parametric distributions; these include modified likelihood ratio tests (Dacunha-Castelle and Gassiat, 1999; Gassiat, 2002), bootstrapping (McLachlan, 1987), information criteria approaches (Spiegelhalter et al., 2002; Drton and Plummer, 2017), classification-based information criteria approaches (Biernacki et al., 2000), and marginal-likelihood-based methods (Chib, 1995; Green, 1995). Determining the number of components for mixtures of normalizing flows is more challenging since in addition to choosing G, one also needs to choose

K (the number of layers for the normalizing flows) and p (the number of basis functions constructing the wrapping potential functions). While one may try to adapt some of the existing order selection approaches mentioned above to this context to find, jointly, optimal values for G, K and p, these new approaches would need to be investigated both theoretically and practically. This lies beyond the scope of this paper. A potential computational bottleneck we envision with several of these approaches is that they typically require comparisons across a large number of fitted models; this could be prohibitive with mixture-of-normalizing-flows models.

In this work we sideline the issue of tuning K, p, and especially G as follows. First, we leverage the good results we obtained from empirical studies with a conventional (single component) normalizing flow on $\mathbb{S}^2$ with $p = 1$ and $K = 20$ to fix p and K to those values, respectively. Second, for G, we fit the mixture-of-normalizing-flows model with a large number of mixture components (we set $G = 10$ in our simulation studies and $G = 20$ in our studies on real data) using the hard EM algorithm, and then remove mixture components that have no observations allocated to them. This approach to mixture modeling is commonly referred to as 'mixture overfitting' (e.g., Rousseau and Mengersen, 2011; van Havre et al., 2015) and takes advantage of a tendency of mixture models to not assign observations to superfluous mixture components. We find this approach works well with our mixture-of-normalizing-flows model in the empirical studies of Section 4.

4 Empirical Studies

This section showcases the mixture-of-normalizing flows model on $\mathbb{S}^2$ for a variety of point patterns. Section 4.1 is a simulation study using a known density function that illustrates the potential benefit of having a mixture of normalizing flows rather than a single component normalizing flow when modeling data on the sphere. Sections 4.2 and 4.3 then show how our model can fit complex densities on the surface of Earth well through the use of an earthquake dataset and a terrorism dataset, respectively.

4.1 Simulation Study

In this section we conduct a simulation experiment to compare the mixture-of-normalizing-flows model to the standard, single component, normalizing flow model (Rezende et al., 2020) on $\mathbb{S}^2$. We generate synthetic data by simulating random observations from a mixture of von Mises-Fisher (vMF) densities on $\mathbb{S}^2$:

$$\sum_{j=1}^{J} \pi_j f_{\text{vMF}}(\cdot\,; \mu_j, \kappa_j),$$

where J is the number of mixture components for the mixture of vMF densities, $\pi_j, j = 1, \ldots, J$, are the mixture weights, and where $f_{\text{vMF}}(\cdot\,; \mu, \kappa)$ is the density of a vMF distribution with mean direction μ and concentration parameter κ. The density $f_{\text{vMF}}(\cdot\,; \mu, \kappa)$ converges to the uniform distribution on $\mathbb{S}^2$ when κ goes to 0, and becomes increasingly concentrated at μ as κ becomes larger. We randomly draw the mean directions from the uniform distribution on $\mathbb{S}^2$, and randomly draw the concentration parameters from the exponential distribution with rate parameter λ. We consider four different simulation setups by varying the number of mixture components J and the rate paramter λ; specifically, we consider the cases

Table 1: Average and empirical standard deviation (in parentheses) of the 20 L^1 distances between the true density function and the estimated density function in the simulation experiment for each simulation setting.

Model	$J = 10, \lambda = 10^{-2}$	$J = 10, \lambda = 10^{-3}$	$J = 20, \lambda = 10^{-2}$	$J = 20, \lambda = 10^{-3}$
NF	1.42 (0.04)	1.44 (0.04)	1.50 (0.05)	1.67 (0.07)
Mix NF	0.60 (0.03)	0.67 (0.04)	0.88 (0.15)	0.94 (0.16)

$\{J, \lambda\} = \{10, 10^{-2}\}$, $\{J, \lambda\} = \{10, 10^{-3}\}$, $\{J, \lambda\} = \{20, 10^{-2}\}$, and $\{J, \lambda\} = \{20, 10^{-3}\}$. We note that the expected value of κ under the simulation is inversely proportional to the rate parameter λ.

We fit both the mixture-of-normalizing-flows model with $G = 10$ mixture components, and the standard (single component) normalizing flow model (Rezende et al., 2020) to the simulated datasets. For both models, we set the number of compositions to $K = 20$ and let $p = 1$ in the wrapping potential function (4). We fit the (single component) normalizing flow model using the "committee of networks" approach adopted by Ng and Zammit-Mangion (2022) in the point process setting. This strategy involves training several models (in our case 50) with random initializations, and then averaging their outputs; this was necessary since the fit of a normalizing flow tends to be highly sensitive to the initial parameter settings. We fit the mixture-of-normalizing-flows model using the hard EM algorithm; a committee of networks was not needed with the mixture-of-normalizing-flows model, likely because of the relative ease with which it can fit complicated densities due to the increased model flexibility. It took approximately 30 seconds to train the (single component) normalizing flow model, and approximately 15 minutes to train the mixture-of-normalizing-flows model; this increase in computing time was expected given that the mixture-of-normalizing-flows model contains an order of magnitude more parameters to estimate than the (single component) normalizing flow model.

For each simulation setting (combination of J and λ), we repeated the simulation and fitting procedure 20 times, and computed the average and empirical standard deviation of the L^1 distance between the true and the estimated density functions for both approaches. The results are shown in Table 1: The mixture-of-normalizing-flows model consistently outperforms the (single component) normalizing flow model. Note that both the (single component) normalizing flow model and the mixture-of-normalizing-flows model perform better for the "simpler" density functions, corresponding to when J is smaller and when λ is larger (which in turn leads to smaller mixture concentration parameters).

4.2 Earthquake locations

To showcase the flexibility of the mixture-of-normalizing-flows model we also apply it real data that exhibits a complex structure. In our first setting we consider the locations of 7354 known earthquake events with body-wave magnitude above 6.0 that occurred between the years 1960 to 2018 across the globe. These data were extracted from the Geocoded Disasters (GDIS) dataset (Rosvold and Buhaug, 2021). For each mixture component, we set the number of compositions to $K = 20$ and let $p = 1$ in the wrapping potential function (4). We fit the mixture-of-normalizing-flows model to the data with $G = 20$ mixture components with the hard EM algorithm. The hard EM algorithm identified 17 non-empty mixture components.

The density (relative to the uniform distribution on the unit sphere) using our model is shown in Figure 1. The two orientations of Earth shown are chosen to depict the two most active regions on Earth: the western

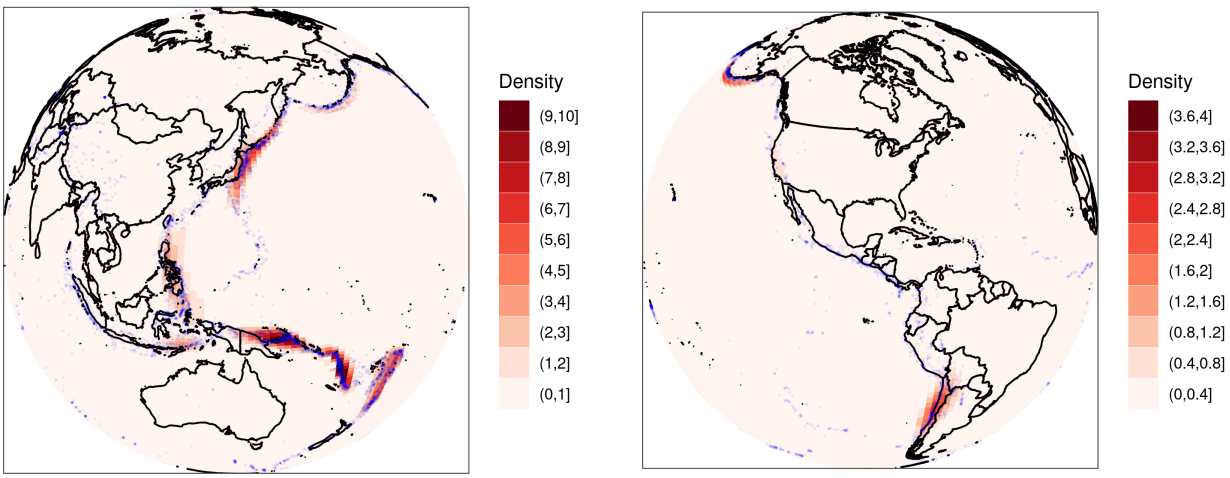

Figure 1: Earthquake events (blue dots) and estimated density of earthquake locations obtained using the mixture-of-normalizing-flows model with $G = 20$ mixture components (red shading). (Left panel) View of Earth centered on 140°E. (Right panel) View of Earth centered on 90°W.

edge of the Pacific plate (left panel) and the western edge of the South American plate (right panel). Earthquakes largely occur where the tectonic plates meet, and thus predominantly follow long narrow paths along the surface of Earth; using a parametric model to fit the density of such data is extremely challenging (and would not be possible using most conventional parametric mixture models). Yet, we can see that the estimated density using the mixture-of-normalizing-flows model provides a very good fit to the data. We also show the fitted density at a selection of locations on Earth in Figure 2; note how the mixture-of-normalizing-flows model is able to easily pick up multiple modes on the sphere of varying orientation, scale, and intensity.

We also attempted to fit a (single component) normalizing flows model fitted to the data using a "committee of networks" approach, where we average 50 models trained with random initializations. The fit we obtained, however, was inadequate and a poor representation of the data.

4.3 Terrorism event locations

In our second setting we consider the locations of 8378 known terrorist events with known locations that occurred in the year 2020 across the globe. The data was extracted from the Global Terrorism Database (GTD).[1] Like the earthquake dataset, these data are challenging to fit as terrorist activity tends to be highly clustered and largely influenced by geopolitical borders. For each mixture component, we set the number of compositions to $K = 20$ and let $p = 1$ in the wrapping potential function (4). We fit the mixture-of-normalizing-flows model to the dataset with $G = 20$ mixture components using the hard EM algorithm. In this case, the hard EM algorithm identified 11 non-empty mixture components.

The density (relative to the uniform distribution on the unit sphere) using our model is shown in Figure 3. Similar to the earthquake dataset, we observe that the estimated density provides a good fit to the data. The (single component) normalizing flow model was also fitted to the dataset using a "committee of networks" approach; here, as with the earthquake data, the normalizing flow failed to achieve a reasonable fit to the

[1] `https://www.start.umd.edu/gtd/`

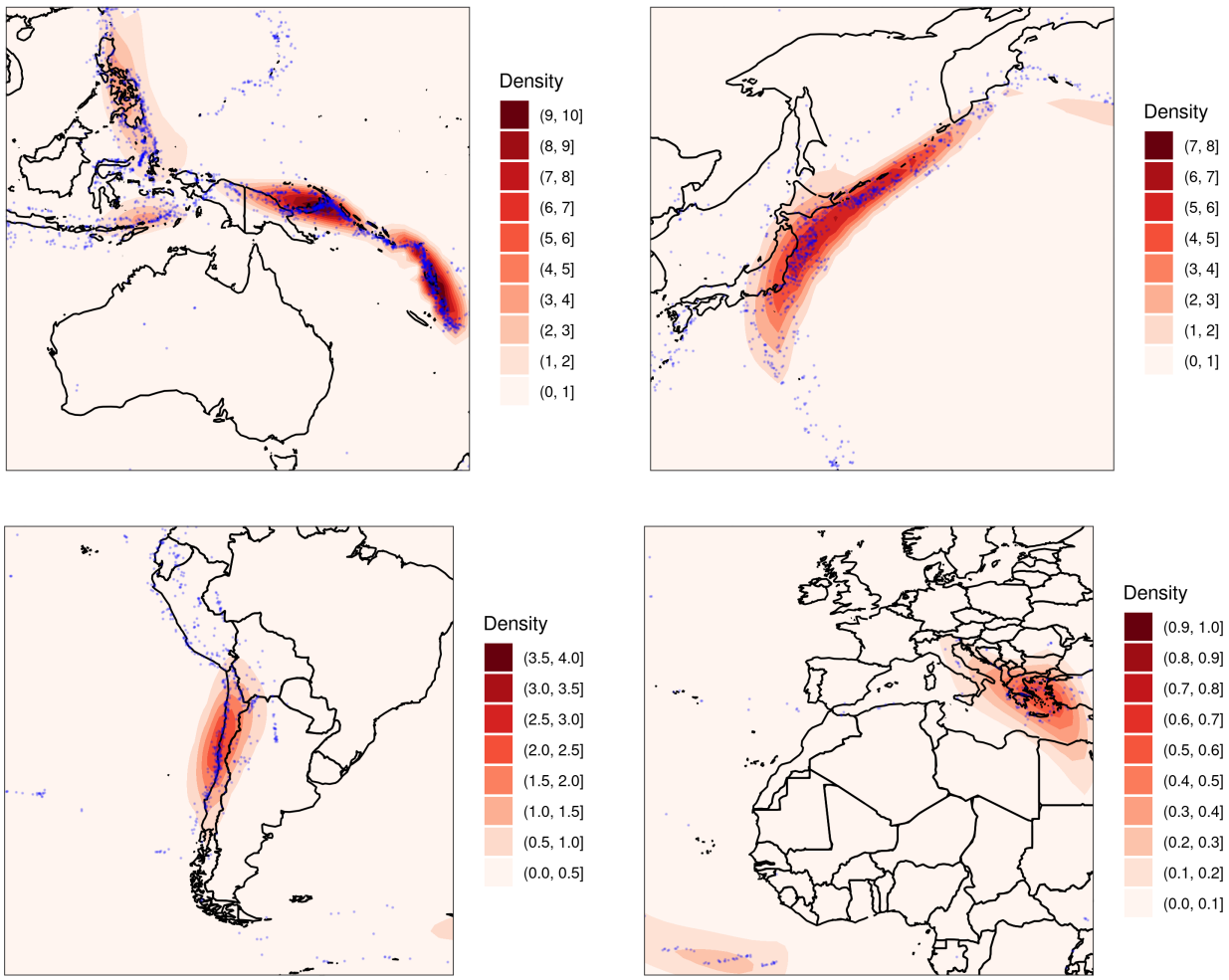

Figure 2: Earthquakes between 1960 to 2018 (blue dots) and the corresponding estimated density obtained using the mixture-of-normalizing-flows model with $G = 20$ mixture components (red shading). (Top-left panel) Earthquakes in the region of the Pacific Islands, Papua New Guinea, Indonesia, and the Philippines. (Top-right panel) Earthquakes in the region of Japan and the Kuril Islands. (Bottom-left panel) Earthquakes in South America. (Bottom-right panel) Earthquakes in Europe and off the Liberian coast in the Atlantic Ocean.

data.

5 Conclusion

In this work we present a mixture modeling framework with normalizing flows for spherical density estimation. The proposed approach offers an attractive alternative to existing parametric and nonparametric mixture modeling approaches by leveraging the computational efficiency and the representational power obtained when constructing deep hierarchies through function composition. The proposed EM algorithms are computationally efficient and scalable when used with mini-batch stochastic gradient descent.

An important consideration with mixture models that we have not explored is parameter identifiability. A necessary condition for identifiability is that there is a one-to-one map between the model parameters and the corresponding probability law (up to permutations of the mixture components). There is a long history of research on the identifiability conditions for finite mixtures of parametric distributions (Teicher, 1961, 1963; Barndorff-Nielsen, 1965; Holzmann et al., 2006), and identifiability has been proved for various parametric

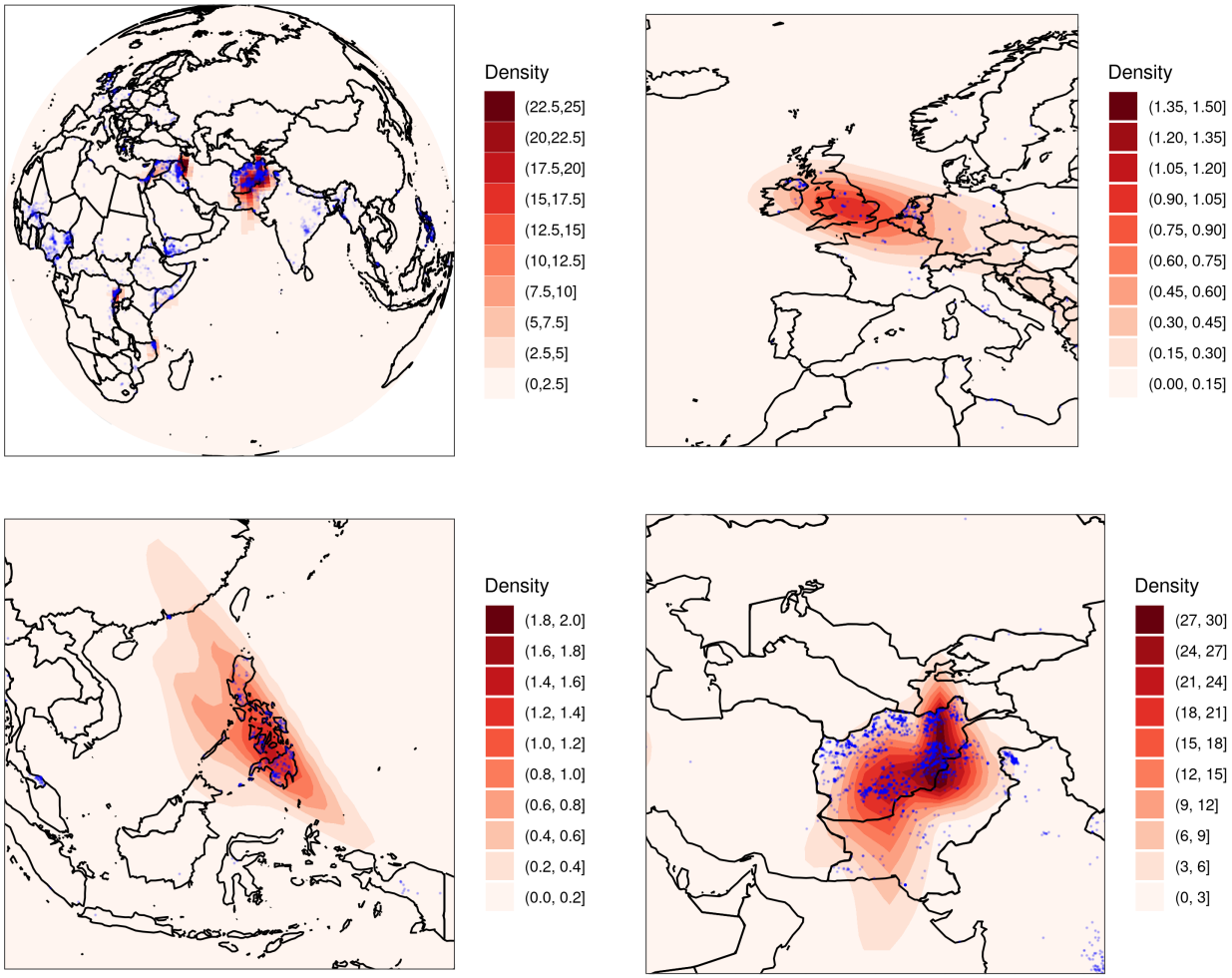

Figure 3: Reported terrorist events in the year 2020 (blue dots) and the corresponding estimated density obtained using the mixture-of-normalizing-flows model with $G = 20$ mixture components (red shading). (Top-left panel) View of Earth centered on 60°E. (Top-right panel) Events in Europe. (Bottom-left panel) Events in the Philippines. (Bottom-right panel) Events in Afghanistan.

families under various conditions. More recently, there has been an increased interest in the identifiability of nonparametric mixture models where each mixture component comes from a flexible, nonparametric family of probability distributions. Identifiability has been established for some of these models under various structural assumptions such as independence of marginal distributions and symmetry (Hall and Zhou, 2003; Hall et al., 2005; Hunter et al., 2007; Levine et al., 2011; D'Haultfoeuille and Février, 2015) and more general conditions (Aragam et al., 2020). Studying conditions under which identifiability holds for a mixture-of-normalizing-flows model is a challenging task. In order for the mixture-of-normalizing-flows model to be identifiable, each mixture component needs to be identifiable itself. Many popular normalizing flows models are typically parameterized by (deep) neural networks where the formulation of identifiability conditions is still in its early days (Phuong and Lampert, 2020; Bona-Pellissier et al., 2021).

While the focus in this paper is on density estimation on spherical domains, the proposed methodology can be extended to more general manifolds. There are several open questions that warrant future research efforts. Identifiability, as discussed above, is one such open question. Another open question concerns interpretability: unlike parametric mixture models, the interpretation of the mixture components in a

mixture-of-normalizing-flows model is challenging. Further, computationally-efficient ways for determining the optimal number of mixture components also needs to be addressed; this is challenging with any mixture model, but is particularly challenging with the mixture-of-normalizing-flows model where there are trade-offs between the number of mixture components and the complexity of each mixture component (via the number of compositions and the number of basis functions per wrapping potential function). Finally, model-based approaches for density estimation have the advantage that they lend themselves well to uncertainty quantification. With our mixture-of-normalizing-flows model one could, for example, employ the bootstrap (e.g., Ng and Zammit-Mangion, 2022). One could also develop Bayesian methods to recover posterior distributions over the model parameters as well as, potentially, the number of mixture components.

Acknowledgements

Andrew Zammit-Mangion's research was supported by the Australian Research Council (ARC) Discovery Early Career Research Award DE180100203.

References

Aragam, B., Dan, C., Xing, E. P., and Ravikumar, P. (2020), "Identifiability of nonparametric mixture models and Bayes optimal clustering," *Annnals of Statistics*, 48, 2277–2302.

Banerjee, A., Dhillon, I. S., Ghosh, J., and Sra, S. (2005), "Clustering on the Unit Hypersphere Using von Mises-Fisher Distributions," *Journal of Machine Learning Research*, 6, 1345–1382.

Barndorff-Nielsen, O. (1965), "Identifiability of mixtures of exponential families," *Journal of Mathematical Analysis and Applications*, 12, 115–121.

Biernacki, C., Celeux, G., and Govaert, G. (2000), "Assessing a mixture model for clustering with the integrated completed likelihood," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 719–725.

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

Bona-Pellissier, J., Bachoc, F., and Malgouyres, F. (2021), "Parameter identifiability of a deep feedforward ReLU neural network," Online: https://arxiv.org/abs/2112.12982.

Bordes, L. and Vandekerkhove, P. (2010), "SEMIPARAMETRIC TWO-COMPONENT MIXTURE MODEL WITH A KNOWN COMPONENT: A CLASS OF ASYMPTOTICALLY NORMAL ESTIMATORS," *Mathematical Methods of Statistics*, 19, 22–41.

Chang, G. T. and Walther, G. (2007), "Clustering with mixtures of log-concave distributions," *Computational Statistics & Data Analysis*, 51, 6242–6251.

Chib, S. (1995), "Marginal Likelihood from the Gibbs Output," *Journal of the American Statistical Association*, 90, 1313–1321.

Ciobanu, S. (2021), "Mixtures of Normalizing Flows," in *Proceedings of ISCA 34th International Conference on Computer Applications in Industry and Engineering*, eds. Shi, Y., Hu, G., Yuan, Q., and Goto, T., pp. 82–90.

Cohen, S., Amos, B., and Lipman, Y. (2021), "Riemannian Convex Potential Maps," in *Proceedings of the 38th International Conference on Machine Learning*, eds. Meila, M. and Zhang, T., pp. 2028–2038.

Cornish, R., Caterini, A., Deligiannidis, G., and Doucet, A. (2020), "Relaxing Bijectivity Constraints with Continuously Indexed Normalising Flows," in *Proceedings of the 37th International Conference on Machine Learning*, eds. Daumé III, H. and Singh, A., Proceedings of Machine Learning Research, pp. 2133–2143.

Dacunha-Castelle, D. and Gassiat, E. (1999), "Testing the order of a model using locally conic parametrization: population mixtures and stationary ARMA processes," *The Annals of Statistics*, 27, 1178–1209.

Dinh, L., Sohl-Dickstein, J., Pascanu, R., and Larochelle, H. (2019), "A RAD approach to deep mixture models," Online: https://openreview.net/pdf?id=HJeZNLIt_4.

Drton, M. and Plummer, M. (2017), "A Bayesian information criterion for singular models," *Journal of the Royal Statistical Society B*, 79, 323–380.

D'Haultfoeuille, X. and Février, P. (2015), "Identification of mixture models using support variations," *Journal of Econometrics*, 189, 70–82.

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

Gassiat, E. (2002), "Likelihood ratio inequalities with applications to various mixtures," *Annales de l'Institut Henri Poincare B*, 38, 897–906.

Gemici, M., Rezende, D. J., and Mohamed, S. (2016), "Normalizing Flows on Riemannian Manifolds," Online: https://arxiv.org/abs/1611.02304.

Gopal, S. and Yang, Y. (2014), "Von Mises-Fisher Clustering Models," in *Proceedings of the 31st International Conference on Machine Learning*, eds. Xing, E. P. and Jebara, T., pp. 154–162.

Green, P. J. (1995), "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika*, 82, 711–732.

Hall, P., Neeman, A., Pakyari, R., and Elmore, R. (2005), "Nonparametric inference in multivariate mixtures," *Biometrika*, 92, 667–678.

Hall, P. and Zhou, X.-H. (2003), "Nonparametric Estimation of Component Distributions in a Multivariate Mixture," *The Annals of Statistics*, 31, 201–224.

Hejblum, B. P., Alkhassim, C., Gottardo, R., Caron, F., and Thiébaut, R. (2019), "Sequential Dirichlet process mixtures of multivariate skew t-distributions for model-based clustering of flow cytometry data," *The Annals of Applied Statistics*, 13, 638–660.

Holzmann, H., Munk, A., and Gneiting, T. (2006), "Identifiability of Finite Mixtures of Elliptical Distributions," *Scandinavian Journal of Statistics*, 33, 753–763.

Huang, C.-W., Krueger, D., Lacoste, A., and Courville, A. (2018), "Neural Autoregressive Flows," in *Proceedings of the 35th International Conference on Machine Learning*, eds. Dy, J. and Krause, A., pp. 2078–2087.

Hunter, D. R., Wang, S., and Hettmansperger, T. P. (2007), "Inference for mixtures of symmetric distributions," *The Annals of Statistics*, 35, 224–251.

Izmailov, P., Kirichenko, P., Finzi, M., and Wilson, A. G. (2020), "Semi-Supervised Learning with Normalizing Flows," in *Proceedings of the 37th International Conference on Machine Learning*, eds. Daumé III, H. and Singh, A., pp. 3165–3176.

Jaini, P., Selby, K. A., and Yu, Y. (2019), "Sum-of-Squares Polynomial Flow," in *Proceedings of the 36th International Conference on Machine Learning*, eds. Chaudhuri, K. and Salakhutdinov, R., pp. 3009–3018.

Jupp, P. E. (1995), "Some applications of directional statistics to astronomy," in *New Trends in Probability and Statistics, Vol. 3*, eds. Tiit, E. M., Kollo, T., and Niemi, H., Utrecht, The Netherlands: De Gruyter, pp. 123–133.

Kobyzev, I., Prince, S. J., and Brubaker, M. A. (2020), "Normalizing flows: An introduction and review of current methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43, 3964–3979.

Levine, M., Hunter, D. R., and Chauveau, D. (2011), "Maximum smoothed likelihood for multivariate mixtures," *Biometrika*, 98, 403–416.

Lin, T.-I., Ho, H. J., and Lee, C.-R. (2014), "Flexible Mixture Modelling Using the Multivariate Skew-t-Normal Distribution," *Statistics and Computing*, 24, 531–546.

Mardia, K. V., Barber, S., Burdett, P. M., Kent, J. T., and Hamelryck, T. (2022), "Mixture models for spherical data with applications to protein bioinformatics," in *Directional Statistics for Innovative Applications*, eds. SenGupta, A. and Arnold, B., Singapore: Springer, pp. 15–32.

Marzouk, Y., Moselhy, T., Parno, M., and Spantini, A. (2016), "Sampling via Measure Transport: An Introduction," in *Handbook of Uncertainty Quantification*, eds. Ghanem, R., Higdon, D., and Owhadi, H., Cham, Switzerland: Springer International Publishing, pp. 1–41.

Mathieu, E. and Nickel, M. (2020), "Riemannian Continuous Normalizing Flows," in *Advances in Neural Information Processing Systems 33*, online: https://proceedings.neurips.cc/paper/2020.

McLachlan, G. J. (1987), "On Bootstrapping the Likelihood Ratio Test Stastistic for the Number of Components in a Normal Mixture," *Journal of the Royal Statistical Society C*, 36, 318–324.

Ng, T. L. J. and Zammit-Mangion, A. (2022), "Spherical Poisson point process intensity function modeling and estimation with measure transport," *Spatial Statistics*, 50, 100629.

— (2023), "Non-homogeneous Poisson process intensity modeling and estimation using measure transport," *Bernoulli*, 29, 815–838.

Papamakarios, G., Nalisnick, E., Rezende, D. J., Mohamed, S., and Lakshminarayanan, B. (2021), "Normalizing Flows for Probabilistic Modeling and Inference," *Journal of Machine Learning Research*, 22, 1–64.

Papamakarios, G., Pavlakou, T., and Murray, I. (2017), "Masked Autoregressive Flow for Density Estimation," in *Advances in Neural Information Processing Systems 30*, online: https://proceedings.neurips.cc/paper/2017.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017), "Automatic Differentiation in PyTorch," in *Advances in Neural Information Processing Systems 30: Workshop on Autodiff*, online: https://openreview.net/forum?id=BJJsrmfCZ.

Peel, D., Whiten, W. J., and McLachlan, G. J. (2001), "Fitting Mixtures of Kent Distributions to Aid in Joint Set Identification," *Journal of the American Statistical Association*, 96, 56–63.

Phuong, M. and Lampert, C. H. (2020), "Functional vs. parametric equivalence of ReLU networks," in *International Conference on Learning Representations*, online: https://openreview.net/forum?id=Bylx-TNKvH.

Pires, G. and Figueiredo, M. A. T. (2020), "Variational Mixtures of Normalizing Flows," in *European Symposium on Artificial Neural Networks (ESANN)*, online: https://www.esann.org/sites/default/files/proceedings/2020/ES2020-188.pdf.

Rezende, D. J., Papamakarios, G., Racaniere, S., Albergo, M., Kanwar, G., Shanahan, P., and Cranmer, K. (2020), "Normalizing Flows on Tori and Spheres," in *Proceedings of the 37th International Conference on Machine Learning*, eds. Daumé III, H. and Singh, A., pp. 8083–8092.

Rosvold, E. and Buhaug, H. (2021), "GDIS, a global dataset of geocoded disaster locations," *Scientific Data*, 8, 1–7.

Rousseau, J. and Mengersen, K. (2011), "Asymptotic behaviour of the posterior distribution in overfitted mixture models," *Journal of the Royal Statistical Society B*, 73, 689–710.

Samdani, R., Chang, M.-W., and Roth, D. (2012), "Unified expectation maximization," in *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, eds. Fosler-Lussier, E., Riloff, E., and Bangalore, S., pp. 688–698.

Sei, T. (2013), "A Jacobian Inequality for Gradient Maps on the Sphere and Its Application to Directional Statistics," *Communications in Statistics - Theory and Methods*, 42, 2525–2542.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and Van Der Linde, A. (2002), "Bayesian measures of model complexity and fit," *Journal of the Royal Statistical Society B*, 64, 583–639.

Teicher, H. (1961), "Identifiability of Mixtures," *The Annals of Mathematical Statistics*, 32, 244–248.

— (1963), "Identifiability of Finite Mixtures," *The Annals of Mathematical Statistics*, 34, 1265–1269.

van Havre, Z., White, N., Rousseau, J., and Mengersen, K. (2015), "Overfitting Bayesian Mixture Models with an Unknown Number of Components," *PLoS ONE*, 10, e0131739.