SPARSE MIXTURE MODELS INSPIRED BY ANOVA DECOMPOSITIONS

JOHANNES HERTRICH†, FATIMA ANTAROU BA†, AND GABRIELE STEIDL‡

Abstract. Inspired by the analysis of variance (ANOVA) decomposition of functions, we propose a Gaussian-uniform mixture model on the high-dimensional torus which relies on the assumption that the function that we wish to approximate can be well explained by limited variable interactions. We consider three model approaches, namely wrapped Gaussians, diagonal wrapped Gaussians, and products of von Mises distributions. The sparsity of the mixture model is ensured by the fact that its summands are products of Gaussian-like density functions acting on low-dimensional spaces and uniform probability densities defined on the remaining directions. To learn such a sparse mixture model from given samples, we propose an objective function consisting of the negative log-likelihood function of the mixture model and a regularizer that penalizes the number of its summands. For minimizing this functional we combine the Expectation Maximization algorithm with a proximal step that takes the regularizer into account. To decide which summands of the mixture model are important, we apply a Kolmogorov-Smirnov test. Numerical examples demonstrate the performance of our approach.

Key words. sparse mixture models, ANOVA decomposition, wrapped Gaussian distribution, von Mises distribution, approximation of high-dimensional probability density functions, Kolmogorov-Smirnov test

AMS subject classifications. 62H30, 62H12, 65D15, 65C60, 62H10

1. Introduction. Most high-dimensional real-world systems are dominated by a small number of low-complexity interactions [55]. This is the background of extensive research on how to represent functions acting on high-dimensional data by functions defined on lower-dimensional spaces. Approaches include active subspace methods [14, 15, 21] and random features [11, 26, 37, 48, 57].

This paper was inspired by the analysis of variance (ANOVA) decomposition of functions [10, 25, 28, 34, 38], which decomposes a function uniquely into a sum of functions depending on the different variable combinations. In practice it can often be assumed that the significant part of a function can be explained by the simultaneous interactions of only a small number of variables, which is also in the spirit of [19]. The amazing result of Potts and Schmischke in [4, 47, 46] shows that high-dimensional functions with a sparse ANOVA decomposition can be reconstructed using their approximation in the Fourier domain by rather few samples \( t^i \in \mathbb{T}^d \) and \( f(t^i) \in \mathbb{R} \), \( i = 1, \ldots, N \). While the theory relies mainly on uniformly sampled points on the high-dimensional torus \( \mathbb{T}^d \) or on \([0,1]^d\), also real-world data sets can be approximated in a way that beats state-of-the-art methods such as the gradient boosting machine [23], random forests [23], sparse random features [26], and local learning regression neural networks [33].

In this paper, we assume that we are given high-dimensional samples \( (x^i)_i \) from a distribution with unknown probability density function \( f \) rather than interpolation knots \( (t^i, f(t^i))_i \). Since attribute ranking can, for instance, be used to remove unimportant variables immediately and to reduce the dimensionality of the problem, we concentrate on functions where each variable has an influence but not simultaneously with all others. Then, we are interested in mixture models with sparse components in the sense that they depend only on the data in smaller dimensions. We propose to learn such a mixture model by minimizing a penalized negative log-likelihood function in connection with a Kolmogorov-Smirnov test to find the active variables in the summands of the mixture model. Once a mixture model is fitted,
a natural way to identify the influence of attributes to the outcome is obtained by adjusting samples to the appropriate summands of the mixture.

Our model appears to be opposite to some recently introduced mixture models whose components rely on projections into sparse subspaces of the high-dimensional data space, such as mixtures of probabilistic PCAs (MPPCA) [54], high-dimensional data clustering (HDDC) [7], high-dimensional mixture models for unsupervised image denoising (HDMI) [30], and PCA-GMMs [27]. For more information, see Remark 2.4.

The paper is organized as follows: in Section 2, we first recall the sparse ANOVA decomposition on the \(d\)-dimensional torus. Then, we introduce appropriate sparse mixture models having components that are products of a Gaussian-like density function on the \(n\)-dimensional torus and a uniform density on the \((d-n)\)-dimensional torus. We propose three Gaussian-like settings, namely the wrapped normal distribution, the diagonal wrapped normal distribution, and products of von Mises distributions. Further, we discuss the ANOVA decomposition of the mixture models based on the notion of identifiable parameterized families of functions. Section 3 deals with the learning of the sparse mixture model. Based on an objective function consisting of the negative log-likelihood function penalized by a sparsity term for the number of coefficients, we propose to apply an Expectation Maximization (EM) algorithm in combination with a proximal step and a Kolmogorov-Smirnov test. Section 4 demonstrates the performance of our model by several examples. The code is available online\(^1\). Appendix A summarizes the EM algorithms for tree Gaussian-like mixture models, and Appendix B briefly shows how the Kolmogorov-Smirnov test works.

2. ANOVA decomposition and mixture models. Let \([d] := \{1, \ldots, d\}\), with the convention that \([0] = \emptyset\), and let \(\mathcal{P}([d])\) be the power set of \([d]\). Furthermore, for \(u \subseteq [d]\), we write \(u^c := [d] \setminus u\) and \(x_u := (x_i)_{i \in u}\). By \(T^d = \mathbb{R}^d / \mathbb{Z}^d = [0,1]^d\), we denote the \(d\)-dimensional torus and by \(I_d\) the \(d \times d\) identity matrix.

We are interested in the additive decomposition of integrable functions \(f : T^d \rightarrow \mathbb{R}\) into lower-dimensional components
\[
(2.1) \quad f(x) = \sum_{u \subseteq [d]} f_u(x_u), \quad f_u : T^{|u|} \rightarrow \mathbb{R}.
\]
In general, such a decomposition is not unique. We rely on two special decompositions, namely the ANOVA decomposition, which was the motivation for this work, and sparse mixture models. In the following we introduce both concepts and explain their relation.

ANOVA decomposition. For any integrable function \(f : T^d \rightarrow \mathbb{R}\), there exists a unique decomposition of the form (2.1), the so-called analysis of variance (ANOVA) decomposition determined by
\[
(2.2) \quad f_u := (P_u f) - \sum_{v \subseteq u \setminus u} (-1)^{|u| - |v|} P_v f,
\]

where
\[
(P_u f)(x) := \int_{T^{|u|}} f(x) \, dx_{u^c}.
\]

The following proposition recalls that the ANOVA decomposition of a function that is the sum of lower-dimensional functions can only contain summands acting on the same subspaces.

**Proposition 2.1.** Let \(W \subseteq \mathcal{P}([d])\). Then, a function \(f\) of the form
\[
(2.3) \quad f(x) = \sum_{w \in W} g_w(x_w)
\]

\(^1\)https://github.com/johertrich/Sparse_Mixture_Models
has an ANOVA decomposition of the form

\[ f = \sum_{u \subseteq W} f_u, \]

where \( W \) denotes the set \( \{ u \subseteq w : w \in W \} \).

Proof. Let \( T_u \) denote the linear operator which maps a function \( f \) to its ANOVA component \( f_u \). Then, we have for the function in (2.3) that

\[ f_u = T_u f = \sum_{w \in W} T_u g_w, \]

and it remains to show that for \( w \) not containing \( u \), it holds that \( T_u g_w = 0 \). Let \( u \) be not contained in \( w \). Using (2.2) and the facts that \( g_w = P_w g_w \) and \( P_u P_v = P_{v \cap u} \), we obtain

\[ T_u g_w = \sum_{v \subseteq u} (-1)^{|u| - |v|} P_v g_w = \sum_{v \subseteq u} (-1)^{|u| - |v|} P_v P_w g_w = \sum_{v \subseteq u} (-1)^{|u| - |v|} P_{v \cap w} g_w. \]

If \( v \cap w = 0 \), then the assertion follows since \( \sum_{v \subseteq u} (-1)^{|u| - |v|} = 0 \). Otherwise, for \( n := |u \setminus w| > 0 \), we obtain that

\[
T_u g_w = \sum_{v_1 \subseteq u \cap w} \sum_{v_2 \subseteq u \setminus w} (-1)^{|u| - |v_1| + |v_2|} P_{v_1} g_w
= \sum_{v_1 \subseteq u \cap w} (-1)^{|u| - |v_1|} P_{v_1} g_w \sum_{v_2 \subseteq u \setminus w} (-1)^{|v_2|}
= \sum_{v_1 \subseteq u \cap w} (-1)^{|u| - |v_1|} P_{v_1} g_w \sum_{j=0}^{n} \binom{n}{j} (-1)^j = 0,
\]

which proves the thesis.

In real-world applications, it is often the case that the decomposition (2.1) does not contain all subsets of \( |d| \) but only a smaller number of subsets \( U \subset \mathcal{P}(|d|) \) which have cardinality not larger than some \( n \ll d \) or that \( f \) can be at least well approximated by a sparse ANOVA decomposition

\[ \sum_{u \subseteq U} f_u(x_u). \]

Several authors examined the reconstruction of functions having such a sparse ANOVA approximation from values \( (t_i, f(t_i)) \), \( i = 1, \ldots, N \), of \( f \). The setting in this paper is different.

Remark 2.2 (Setting of this paper). We deal with non-negative functions \( f : \mathbb{T}^d \to \mathbb{R} \) on the \( d \)-dimensional torus \( \mathbb{T}^d \) fulfilling \( \|f\|_{L_1(\mathbb{T}^d)} = 1 \) and consider them as probability density functions of a certain random variable \( X : \Omega \to \mathbb{T}^d \). Instead of sampled function values, we assume that we are given samples \( x_i, i = 1, \ldots, N \), of the distribution with density \( f \), i.e., realizations of the random variable \( X \). This means that in contrast to the \( t_i \), the samples \( x^i \) inherit the properties of \( f \). If the \( t_i \), \( i = 1, \ldots, N \), are uniformly sampled, then clearly \( f(t^i) \) times \( t^i \) may serve as samples, i.e., the \( t^i \) must be weighted with the values \( f(t^i) \).

Sparse mixture models. In this paper, we aim to find an approximation of \( f \in L_1(\mathbb{T}^d) \) by a mixture model from samples of the corresponding distribution; see [43] for an introduction to mixture models. To this end, let \( \Delta_K := \{ \alpha \in \mathbb{R}^{K_0}_+ : \alpha^T 1_K = 1 \} \), with \( 1_K \) the vector
consisting of \( K \) entries equal to 1, be the probability simplex, and let \( \text{SPD}(d) \) be the cone of symmetric, positive definite matrices. Assume that \( f \) can be approximated by mixture models of the form

\[
p(x|\alpha, \vartheta) = \sum_{k=1}^{K} \alpha_k p_{u_k}(x_{u_k} | \vartheta_k),
\]

where \( u_k \in U \subset \mathcal{P}([d]), \alpha = (\alpha_k)_{k=1}^{K} \in \Delta_K, \vartheta = (\vartheta_k)_{k=1}^{K} \), and the \( p_{u_k} \) are probability density functions on \( \mathbb{T}^n, n = |u_k| \). Note that the index sets \( u_k \) are in general not pairwise different, i.e., \( u_k = u_l \) can appear for \( k \neq l \). If \( U \) contains only sets of small cardinality, then we call (2.4) a sparse mixture model. Indeed, the density \( p_u \) determines the distribution of a \( \mathbb{T}^d \)-valued random variable \( X = (X_u, X_{u^c}) \) characterized by

\[
(X_u, X_{u^c}) \sim p_u(\cdot | \vartheta) \times \mathcal{U}_{\mathbb{T}^d-|u|}.
\]

This class of distributions includes for \( u = \emptyset \) the uniform distribution on \( \mathbb{T}^d \).

In this paper, we need the (absolutely continuous) normal or Gaussian distribution on \( \mathbb{R}^n \) having the density function

\[
\mathcal{N}(x|\mu, \Sigma) = (2\pi)^{-\frac{n}{2}} |\Sigma|^{-\frac{1}{2}} \exp \left( -\frac{1}{2} (x - \mu)^T \Sigma^{-1} (x - \mu) \right)
\]

with mean \( \mu \in \mathbb{R}^n \) and \( \Sigma \in \text{SPD}(n) \). This distribution has many characterizing properties, which unfortunately cannot be transferred to a “normal distribution” on manifolds; see, e.g., [36]. In this paper, we restrict our attention to the normal-like distributions \( p_{u_k} \) on the \( (n = |u_k|) \)-dimensional torus \( \mathbb{T}^n \) listed in the following example.

\textbf{Example 2.3.} We focus on mixture models on \( \mathbb{T}^d \) with low-dimensional components from one of the following distributions on \( \mathbb{T}^n, n \ll d \):

i) the wrapped normal distribution

\[
p_{\mathcal{N}}(x|\mu, \Sigma) = \sum_{l \in \mathbb{Z}^n} \mathcal{N}(x + l|\mu, \Sigma) = \mathcal{N}_{w}(x|\mu, \Sigma),
\]

where \( \mu \in \mathbb{T}^n, \Sigma \in \text{SPD}(n) \). Note that \( \mathcal{N}_{w}(\mu, \Sigma) \) is characterized by the distribution of \( X - \lfloor X \rfloor \), where \( X \sim \mathcal{N}(\mu, \Sigma) \). This formula allows us to easily draw samples from \( \mathcal{N}_{w}(\mu, \Sigma) \).

ii) the diagonal wrapped normal distribution

\[
p_{\mathcal{D}}(x|\mu, \sigma^2) = p_{\mathcal{N}}(x|\mu, \text{diag}(\sigma^2))
\]

\[
= \sum_{l \in \mathbb{Z}^n} \prod_{j=1}^{n} \mathcal{N}(x_j + l_j|\mu_j, \sigma_j^2) = \prod_{j=1}^{n} \mathcal{N}_{w}(x_j|\mu_j, \sigma_j^2),
\]

where \( (\mathcal{N}) \) \( \mathcal{N} \) is the univariate (wrapped) Gaussian density function and \( \sigma^2 \in \mathbb{R}_{>0}^n \).

iii) the von Mises distribution on \( \mathbb{T}^n \) with parameters \( \mu \in \mathbb{T}^n \) and \( \kappa \in \mathbb{R}_{>0}^n \) is the restriction of the probability density function of an isotropic normal distribution to the unit circle, and it has the probability density function

\[
p_{\text{M}}(x|\mu, \kappa) = \prod_{j=1}^{n} \frac{1}{I_0(|\kappa_j|)} \exp \left( \kappa_j \cos \left( 2\pi (x_j - \mu_j) \right) \right),
\]

where \( I_0 \) is the modified Bessel function of first kind of order 0.
The wrapped normal distribution inherits by definition several properties of the normal distribution in \( \mathbb{R}^n \). For example, for independent \( X \sim \mathcal{N}_w(\mu, \Sigma) \), \( Y \sim \mathcal{N}_w(\mu', \Sigma') \), we directly obtain that \( X + Y \sim \mathcal{N}_w(\mu + \mu', \Sigma + \Sigma') \). Similarly, we get that any marginal of \( X \) is again a wrapped normal distribution. Other properties of the normal distribution are not transferred to the wrapped case. For example, on a circle it holds that the von Mises distribution maximizes the entropy and not the wrapped normal distribution; see [31]. Indeed the von Mises distribution with parameters \((\mu, \kappa)\) is very similar to the one-dimensional wrapped normal distribution with parameters \((\mu, \sigma^2)\), where the parameters are related via
\[
\frac{I_1(\kappa)}{I_0(\kappa)} = \exp\left(-\frac{(2\pi)^2 \sigma^2}{2}\right);
\]
see [32]. Thus, the von Mises distribution is often used in place of the wrapped normal distributions with the benefit of a reduced complexity for evaluating the density function and estimating the parameters; see, e.g., [8, 20, 35]. Unfortunately, there is no multivariate counterpart for this approximation. Finally, we mention that there also exist extensions of the von Mises distribution to the (non-tensor) multivariate case on \( \mathbb{T}^d \); see [39, 41, 42]. Unfortunately, the normalization constants of these multivariate von Mises distributions have in general no closed form, and the numerical approximation is very expensive.

The following remark highlights the difference of our approach to another kind of “sparse” mixture models in the literature.

**Remark 2.4 (An “opposite” sparse mixture model).** In a Gaussian setting, our sparse mixture model replaces the inverse covariance matrices \( \Sigma^{-1} \) in the summands of the mixture model by special matrices of low rank \(|u|\). This is opposite to replacing the covariance matrices \( \Sigma \) themselves by low-rank matrices as done, e.g., in [24, 50]. In other words, our paper addresses sparsity in the time domain, while the other authors consider the Fourier domain.

In [27] (see also [7, 30]), the authors considered the so-called PCA-GMMs. These are Gaussian mixture models where, up to a rotation, the summand densities are associated with random variables distributed as
\[
(X_u, X_{w'}) \sim \mathcal{N}(\mu, \Sigma) \times \mathcal{N}(0, \sigma^2 I),
\]
where \( \sigma^2 > 0 \) is a small fixed parameter which may account for noise in the data. Now, wrapping \( X \) around the \( d \)-dimensional torus yields that \( Y = X - \lfloor X \rfloor \) is distributed as
\[
(Y_u, Y_{w'}) \sim \mathcal{N}_w(\mu, \Sigma) \times \mathcal{N}_w(0, \sigma^2 I).
\]
For \( \sigma^2 \to \infty \) this distribution converges to
\[
\mathcal{N}_w(\mu, \Sigma) \times \mathcal{U}_{\mathbb{T}^d - \lfloor u \rfloor},
\]
which is exactly how the components of our model (2.4) are defined. Thus, in contrast to the PCA-GMM model, we have \( \sigma^2 \to \infty \) instead of a small or vanishing \( \sigma \).

Finally, we are interested in the ANOVA decomposition of our mixture models. To this end, we consider functions \( h_u(\cdot|\vartheta) : \mathbb{T}^{|u|} \to \mathbb{R} \) depending on \( u \subseteq [d] \) and \( \vartheta \in \Theta_u \) for some parameter space \( \Theta_u \). We say that a family of probability density functions
\[
\mathcal{H} = \{ h_u(\cdot|\vartheta) : u \subseteq [d], \vartheta \in \Theta_u \}
\]
is closed under projections, if for any \( u, v \in [d], \vartheta \in \Theta_u \), there exists \( \tilde{\vartheta} \in \Theta_{v \cap u} \) such that
\[
h_{u \cap v}(x_{v \cap u}|\tilde{\vartheta}) = P_{v \cap u} h_u(\cdot|\vartheta).
\]
In other words, marginals of functions in \( \mathcal{H} \) have the same form. As already mentioned, the family of wrapped Gaussians is closed under projection. Clearly this holds also true for families of direct products of univariate distributions.

Furthermore, the family \( \mathcal{H} \) is called identifiable, if its elements are linearly independent in the vector space of all functions on \( T^d \), i.e., if for any \( K \in \mathbb{N} \) and for \( \alpha_1, \ldots, \alpha_K \in \mathbb{R}, \sum_{k=1}^K \alpha_k h_{u_k}(x_{u_k}|\theta_{u_k}) = 0 \) implies that \( \alpha_k = 0 \) for all \( k = 1, \ldots, K \); see [52, 56]. It is known that the multivariate Gaussian family on \( \mathbb{R}^d \) is identifiable [16, 56]. Moreover, the univariate wrapped normal distribution [29] and the von Mises distribution on \( T \) are identifiable [22]. By the results in [53], also diagonal wrapped normal distributions in ii) and the products of von Mises distributions in iii) are identifiable. On the other hand, whether the wrapped normal distribution on \( T^d \) in i) is identifiable or not appears to be an open problem.

Then, we have the following proposition on the ANOVA decomposition of mixture models.

**Proposition 2.5.** Let \( W \subseteq \mathcal{P}([d]) \) and \( \mathcal{H} = \{ h_u(\cdot|\varnothing) : u \subseteq [d], \varnothing \in \Theta_u \} \) be an identifiable family of probability density functions that is closed under projections. Further, let \( g_w, w \in W \), be a linear combination of functions from \( \{ h_w(\cdot|\varnothing_l) : \varnothing_l \in \Theta_w \} \) with positive coefficients. Then, a function \( f \) of the form

\[
f(x) = \sum_{w \in W} g_w(x_w)
\]

has the ANOVA decomposition

\[
f = \sum_{u \in \bar{W}} f_u
\]

with \( f_u \neq 0 \) for all \( u \in \bar{W} \), where \( \bar{W} \) denotes the set \( \{ v \subseteq w : w \in W \} \).

**Proof.** By Proposition 2.1 we know already that

\[
f = \sum_{u \in W} f_u,
\]

so it remains to show that none of these summands vanishes. Assume on the contrary that there exists \( u \in \bar{W} \) such that \( f_u = 0 \). By (2.2) we have

\[
0 = f_u = P_u f + \sum_{v \subseteq u} (-1)^{|u| - |v|} P_v f.
\]

Since \( \mathcal{H} \) is closed under projection and the \( g_w \) are positive linear combinations of functions from \( \{ h_w(\cdot|\varnothing) : \varnothing \in \Theta_w \} \), we have \( g_w = \sum_{l=1}^{K_1} \alpha_{w,l} h_w(\cdot|\hat{\varnothing}_l), \alpha_{w,l} > 0 \), and therefore

\[
P_u f = \sum_{w \in W} P_u g_w = \sum_{w \in W, u \subseteq w} P_u g_w + \sum_{w \in W, u \not\subseteq w} P_u g_w
\]

\[
= \sum_{w \in W, u \subseteq w} \sum_{l=1}^{K_1} \alpha_{w,l} P_u h_w(\cdot|\hat{\varnothing}_l) + f_1
\]

\[
= \sum_{k=1}^{K} \alpha_k h_u(\cdot|\varnothing_k) + f_1
\]

for some \( K \in \mathbb{N} \), positive coefficients \( \alpha_k \in \mathbb{R}_{>0} \), \( f_1 \in \text{span}\{ h_v(\cdot|\varnothing) : v \subseteq u, \varnothing \in \Theta_v \} \), and pairwise distinct \( \varnothing_k, k = 1, \ldots, K \). Since \( u \in \bar{W} \), we have that \( K > 0 \). Moreover, for \( v \not\subseteq u \)
it holds that
\[ P_v f = \sum_{w \in \mathcal{W}} P_v g_w = f_2 \in \text{span}\{ h_{\vartheta}(\cdot | \vartheta) : \vartheta \subseteq u, \vartheta \in \Theta_v \}. \]

Putting the last two formulas together, we obtain in (2.5) that
\[ 0 = \sum_{k=1}^{K} \alpha_k h_u(\cdot | \vartheta_k) + f_3, \]
where \( f_3 \in \text{span}\{ h_v(\cdot | \vartheta) : v \subseteq u, \vartheta \in \Theta_v \} \). Now the identifiability of \( \mathcal{H} \) yields that \( \alpha_k = 0 \) for all \( k = 1, \ldots, K \), which is a contradiction.

3. Learning sparse mixture models. Our approach for learning a sparse mixture model consists of three items. First, we need a rough approximation of the involved index sets \( u_k \), which is explained in Section 3.3. Then, we consider an objective function consisting of the log-likelihood of the corresponding mixture model and an additional term that penalizes a high number of summands, supporting further sparsity of the mixture model. To minimize this objective function we propose a combination of a proximal step and the EM algorithm. The proximal step is considered in Section 3.1 and the EM algorithm in Section 3.2.

Let \( N \) observations \( \mathcal{X} = (x_1, \ldots, x_N) \in \mathbb{R}^{d,N} \) with non-negative real-valued weights \( \mathcal{W} = (w_1, \ldots, w_N) \) be given. For simplicity, we assume that \( \sum_{i=1}^{N} w_i = N \). Then, the weighted negative log-likelihood function of the mixture model (2.4) is given by
\[
(3.1) \quad \mathcal{L}(\alpha, \vartheta | \mathcal{X}) = -\sum_{i=1}^{N} w_i \log \left( \sum_{k=1}^{K} \alpha_k p_{u_k}(x_i | \vartheta_k) \right).
\]

Since we intend to obtain a sparse mixture model, we propose to minimize instead of \( \mathcal{L} \) the penalized function
\[
(3.2) \quad \mathcal{L}_{\lambda}(\alpha, \vartheta) := \mathcal{L}(\alpha, \vartheta | \mathcal{X}) + \lambda \|\alpha\|_0 + \iota_{\Delta_K}(\alpha), \quad \lambda > 0,
\]
with the zero “norm” defined as \( \|\alpha\|_0 := |\{k : \alpha_k > 0\}| \) and the indicator function \( \iota_{\Delta_K}(\alpha) \) such that \( \iota_{\Delta_K}(\alpha) = 0 \) if \( \alpha \in \Delta_K \) and \( \iota_{\Delta_K}(\alpha) = +\infty \) otherwise. Here we suppose that the \( u_k \subseteq [d] \), \( k = 1, \ldots, K \), are fixed. In Section 3.3, we will suggest a heuristic for determining appropriate sets \( u_k \). We propose to minimize (3.2) by alternating between the EM steps for \( \mathcal{L} \) and a proximity step for the function
\[
(3.3) \quad h(\alpha) := \|\alpha\|_0 + \iota_{\Delta_K}(\alpha).
\]

More precisely, we will iterate
\[
(3.4) \quad (\alpha^{(r+\frac{1}{2})}, \vartheta^{(r+1)}) = \text{EM}(\alpha^{(r)}, \vartheta^{(r)}),
\]
\[
(3.5) \quad \alpha^{(r+1)} = \text{prox}_{\gamma h}(\alpha^{(r+\frac{1}{2})}), \quad \gamma > 0,
\]
where \( \text{prox}_{\gamma h}(\cdot) \) is defined according to (3.6).

In the following section, we consider the proximity step before we explain the EM algorithm for our mixture models with components from Example 2.3.
3.1. Proximal algorithm. The proximal operator $\text{prox}_{\gamma g}$ for a proper, lower semi-continuous function $g: \mathbb{R}^K \rightarrow \mathbb{R} \cup \{+\infty\}$ and $\gamma > 0$ is defined as

$$\text{prox}_{\gamma g}(x) := \arg\min_{y \in \mathbb{R}^d} \left\{ \frac{1}{2\gamma} \|x - y\|^2 + g(y) \right\}.$$  

Note, that for a non-convex function $g$, a minimizer is not necessarily unique such that the proximal operator might be set-valued. For the non-convex function $g$ in (3.3), we can compute a proximum using the following lemma.

**Lemma 3.1.** Let $\alpha \in \Delta_K$, and assume without loss of generality that $\alpha_1 \leq \cdots \leq \alpha_K$. Let $h$ be defined as in (3.3). Then, the following holds true:

i) An element $\hat{\alpha} \in \text{prox}_{\gamma h}(\alpha)$ is given by $\hat{\alpha}_J = 0$ and

$$\hat{\alpha}_{J^C} = \alpha_{J^C} + \frac{1}{|J^C|} \sum_{k \in J} \alpha_k,$$

where $J = [K_0], J^C$ is defined by $J^C = [K] \setminus J$, and

$$K_0 \in \arg\min_{n \in \{0, \ldots, K-1\}} g(n), \quad g(n) = \frac{1}{2\gamma} \left( \frac{\sum_{k=1}^{n} \alpha_k}{K - n} \right)^2 + \frac{1}{2\gamma} \sum_{k=1}^{n} \alpha_k^2 - n.$$  

ii) Assume that $\alpha_i = 0$ for some $i \in \{1, \ldots, K\}$. Then, it holds for any $\hat{\alpha} \in \text{prox}_{\gamma h}(\alpha)$ that $\hat{\alpha}_i = 0.$

**Proof.** First we note that for $\hat{\alpha} \in \text{prox}_{\gamma h}(\alpha)$ it holds that

$$\hat{\alpha} \in \arg\min_{y \in \Delta_K} \left\{ \frac{1}{2\gamma} \|\alpha - y\|^2 + ||y||_0 \right\}.$$  

With $J := \{k \in [K] : \hat{\alpha}_k = 0\}$, this can be rewritten as

$$\hat{\alpha}_{J^C} \in \arg\min_{y \in \Delta_{J^C}} \left\{ \frac{1}{2\gamma} ||\alpha_{J^C} - y||^2 \right\},$$

which is the orthogonal projection of $\alpha_{J^C}$ onto $\Delta_{J^C}$. Since $\alpha_k \geq 0$ for all $k$ and $||\alpha_{J^C}||_1 \leq 1$, this projection is given by

$$\hat{\alpha}_{J^C} = \alpha_{J^C} + \frac{1}{|J^C|} \sum_{k \in J} \alpha_k.$$  

In particular, we have that

$$||\hat{\alpha} - \alpha||^2 = |J^C| \left( \frac{\sum_{k \in J} \alpha_k}{|J^C|} \right)^2 + \sum_{k \in J} \alpha_k^2 = \left( \frac{\sum_{k \in J} \alpha_k}{|J^C|} \right)^2 + \sum_{k \in J} \alpha_k^2.$$  

(a) Thanks to the previous calculations, the definition of proximal operators, and the fact that $J \subsetneq [K]$ (otherwise $\hat{\alpha} = 0 \notin \Delta_K$), we only have to show that $J$ given by (3.7) is a minimizer of

$$\min_{J \subsetneq [K]} \left( \frac{\sum_{k \in J} \alpha_k}{|J^C|} \right)^2 + \frac{1}{2\gamma} \sum_{k \in J} \alpha_k^2 + |J^C|.$$  


Due to the monotonicity of (3.8) in $\alpha_k$, $k \in J$, and $\alpha_1 \leq \cdots \leq \alpha_K$, there exists a minimizer of the form $J = \{K_0\}$ for

$$K_0 \in \arg\min_{n \in \{0, \ldots, K-1\}} g(n), \quad g(n) = \frac{1}{2\gamma} \left( \sum_{k=1}^{n} \alpha_k \right)^2 + \frac{1}{2\gamma} \sum_{k=1}^{n} \alpha_k^2 - n.$$ 

(b) Let $\alpha_k = 0$ and assume that $\hat{\alpha}_k > 0$, where $\hat{\alpha}_k \in \text{prox}_{\gamma(\|\cdot\|_0 + \Delta_K)}(\alpha)$. If there exists no $l \in [K]$ with $\alpha_l > 0$ and $\hat{\alpha}_l = 0$, then it holds that

$$\frac{1}{2\gamma} \|\hat{\alpha} - \alpha\|^2 + \|\hat{\alpha}\|_0 > \|\alpha\|_0,$$

which is a contradiction to the definition of the proximal operator. Thus, we can assume that such an $l \in [K]$ exists with $\alpha_l > 0$, $\hat{\alpha}_l = 0$. Now, define $\hat{\alpha}$ with $\hat{\alpha}_k = 0$, $\hat{\alpha}_l = \hat{\alpha}_k$, and $\hat{\alpha}_j = \hat{\alpha}_j$, for $j \in [K] \setminus \{k, l\}$. Then, it holds by the definition of the proximal operator that

$$0 \geq \frac{1}{2\gamma} \|\hat{\alpha} - \alpha\|^2 + \|\hat{\alpha}\|_0 - \frac{1}{2\gamma} \|\hat{\alpha} - \alpha\|^2 - \|\hat{\alpha}\|_0 = \frac{1}{2\gamma} (\hat{\alpha}_k^2 + (\alpha_l)^2 - (\hat{\alpha}_l - \alpha_l)^2) = \frac{1}{2\gamma} (\hat{\alpha}_k^2 + (\alpha_l)^2 - (\hat{\alpha}_k - \alpha_l)^2).$$

Since $\hat{\alpha}_k, \alpha_l \in (0, 1]$ we have that $|\hat{\alpha}_k - \alpha_l| < \max(\hat{\alpha}_k, \alpha_l)$, which implies that the right-hand side of the above equation is strictly greater than 0. This is a contradiction, and the proof is completed.

**Remark 3.2.** Since sorting the components of the vector $\alpha$ can be done in $O(K \log(K))$, Lemma 3.1 implies that we can compute an element of $\text{prox}_{\gamma h}(\alpha)$ in $O(K \log(K))$. In particular, the computation of the prox step is very cheap compared with the EM-step.

### 3.2. EM Algorithm

For minimizing the negative log-likelihood function (3.1) we apply an EM algorithm; see [9, 17] and for a good brief introduction also [36]. We need two different variants of this algorithm, namely for products of von Mises distributions and for the wrapped Gaussians.

Let $X^1, \ldots, X^N$ be i.i.d. random variables distributed according to $p_X(\cdot | \alpha, \vartheta)$, and let $X = (X^1, \ldots, X^N)$. Given a realization $X^* = (x^1, \ldots, x^N) \in \mathbb{R}^{d,N}$ of $X$, the common idea of the EM algorithm for finding a maximizer of the log-likelihood function

$$\mathcal{L}(\alpha, \vartheta | X^*) = \prod_{i=1}^{N} p_X(x^i | \alpha, \vartheta)$$

is to introduce an artificial, hidden random variable $Z$ and to perform the following two steps:

**E-Step:** For a fixed estimate $(\alpha^{(r)}, \vartheta^{(r)})$ of $(\alpha, \vartheta)$, we approximate the log-likelihood function $\log \left( p_{X,Z}(X, Z | \alpha, \vartheta) \right)$ of the unknown joint realization $(X, Z)$ by the so-called $Q$-function

$$Q((\alpha, \vartheta), (\alpha^{(r)}, \vartheta^{(r)})) = E_{(\alpha^{(r)}, \vartheta^{(r)})} \left( \log \left( p_{X,Z}(X, Z | \alpha, \vartheta) \right) | X = X^* \right),$$

where the expectation value is taken with respect to the probability distribution associated with the mixture model $p(\cdot | \alpha^{(r)}, \vartheta^{(r)})$.

**M-Step:** We update $\vartheta$ by maximizing the $Q$-function

$$(\alpha^{(r+1)}, \vartheta^{(r+1)}) \in \arg\max_{\alpha, \vartheta \in \Delta_K \times \Theta} \{Q((\alpha, \vartheta), (\alpha^{(r)}, \vartheta^{(r)}))\}.$$
A convergence analysis of the EM algorithm via the Kullback-Leibler proximal point algorithms was given in [12, 13, 36]. These convergence results apply also to our special mixture models in the following paragraphs.

**Proposition 3.3.** Let the sequence \((\alpha^{(r)}, \vartheta^{(r)})_r\) be generated by the above EM steps. Then, the sequence of negative log-likelihood values \(L(\alpha^{(r)}, \vartheta^{(r)} | \mathcal{X})\) is decreasing.

The EM algorithm for minimizing the log-likelihood function of the mixture model with products of von Mises functions as summands can be realized by a standard approach [3, 40] for mixture models, which uses a special hidden random variable \(Z\). In particular, the maximum in the M-step can be computed analytically. Unfortunately, with this approach, the M-step has no analytical solution in the wrapped Gaussians setting, therefore we have to choose the hidden variable \(Z\) in a different way. We describe both approaches in the following.

**EM algorithm for products of von Mises distributions.** For mixture models, it is common to choose hidden variables \(Z_{ik}\) with \(Z_{ik} = 1\) if \(X_i\) belongs to the \(k\)-th component of the mixture model and \(Z_{ik} = 0\) otherwise. Let \(X = (x^i)\), and \(Z = (z_{i,k})_{i,k,l}\) be (unknown) joint realizations.

**E-Step:** It can be shown (see [17, 36]) that with the so-called complete weighted log-likelihood function

\[
\ell(\alpha, \vartheta | \mathcal{X}, Z) := \sum_{i=1}^N w_i \sum_{k=1}^K z_{ik} \log(\alpha_k p_{uk}(x^i | \vartheta_k)),
\]

the \(Q\) function reads as

\[
Q((\alpha, \vartheta), (\alpha^{(r)}, \vartheta^{(r)})) = E_{(\alpha^{(r)}, \vartheta^{(r)})} \left( \ell(\alpha, \vartheta | \mathcal{X}, Z) | \mathcal{X} = \mathcal{X} \right) = \sum_{i=1}^N w_i \sum_{k=1}^K \beta^{(r)}_{i,k} \log(\alpha_k p_{uk}(x^i | \vartheta_k)),
\]

with

\[
\beta^{(r)}_{i,k} = \frac{\alpha^{(r)}_k p_{uk}(x^i | \vartheta^{(r)}_k)}{\sum_{j=1}^K \alpha^{(r)}_j p_{uj}(x^i | \vartheta^{(r)}_j)}.
\]

Note that \(\beta^{(r)}_{i,k}\) is an estimate of \(z_{ik}\), therefore it can be seen as the probability that \(x^i\) arises from the \(k\)-th summand of the mixture model.

The optimization in the M-step can be done separately for \(\alpha\) and \(\vartheta\). This results in the EM algorithm for mixture models given in Algorithm 1.

It remains to maximize the function in the second M-step. For the von Mises model in Example 2.3 iii), this can be done analytically as described in the following.

**M-Step:** For products of von Mises distributions, the log-density functions in each component of the mixture model are given by

\[
\log \left( p_{uk}(x_{uk}) \right) = \sum_{j \in u_k} \log \left( p_{M}(x_j | \mu_{j,k}, \kappa_{j,k}) \right).
\]

Then

\[
(\mu^{(r+1)}_k, \kappa^{(r+1)}_k) = \arg\max_{\mu_k, \kappa_k} \left\{ \sum_{i=1}^N w_i \sum_{j \in u_k} \beta^{(r)}_{i,k} \log(p_{M}(x^i | \mu_{j,k}, \kappa_{j,k})) \right\}
\]
Algorithm 1 EM algorithm for mixture models.

Input: \((x^1, \ldots, x^N) \in \mathbb{T}^{d,N}, (w_1, \ldots, w_N) \in \mathbb{R}^N\) and initial estimates \(\alpha^{(0)}, \vartheta^{(0)}\).

for \(r = 0, 1, \ldots\) do

E-Step: for \(k = 1, \ldots, K\) and \(i = 1, \ldots, N\), compute

\[
\beta^{(r)}_{i,k} = \frac{\alpha^{(r)}_k p_{uk}(x^i | \vartheta^{(r)}_k)}{\sum_{j=1}^K \alpha^{(r)}_j p_{uj}(x^i | \vartheta^{(r)}_j)}
\]

M-Step: for \(k = 1, \ldots, K\), compute

\[
\alpha^{(r+1)}_k = \frac{1}{N} \sum_{i=1}^N w_i \beta^{(r)}_{i,k},
\]

\[
\vartheta^{(r+1)}_k = \arg \max_{\vartheta_k} \left\{ \sum_{i=1}^N w_i \beta^{(r)}_{i,k} \log(p_{uk}(x^i | \vartheta_k)) \right\}.
\]

end for

decouples for \(j\). For the univariate von Mises distribution, the log-maximum likelihood estimator is well known \([31]\), and we obtain

\[
\mu^{(r+1)}_{j,k} = \frac{1}{2\pi} \arctan^* \left( \frac{S^{(r)}_{j,k}}{C^{(r)}_{j,k}} \right), \quad \kappa^{(r+1)}_{j,k} = A^{-1}(R^{(r)}_{j,k}),
\]

where

\[
A(\kappa) := \frac{\int_0^\kappa}{\phi_0(\kappa)},
\]

\[
C^{(r)}_{j,k} := \sum_{i=1}^N w_i \beta^{(r)}_{i,k} \cos(2\pi x^i_j), \quad S^{(r)}_{j,k} := \sum_{i=1}^N w_i \beta^{(r)}_{i,k} \sin(2\pi x^i_j),
\]

\[
R^{(r)}_{j,k} := \frac{1}{N\alpha^{(r+1)}_k} \sqrt{(S^{(r)}_{j,k})^2 + (C^{(r)}_{j,k})^2},
\]

and \(\arctan^* : \mathbb{R} \times \mathbb{R} \rightarrow [0, 2\pi)\) with \((S, C) \mapsto \arctan^* \left( \frac{S}{C} \right)\) denotes the "quadrant-specific" inverse of the tangent defined by

\[
\arctan^* \left( \frac{S}{C} \right) = \begin{cases} 
\arctan \left( \frac{S}{C} \right), & \text{if } C > 0, \quad S \geq 0, \\
\pi, & \text{if } C = 0, \quad S > 0, \\
\arctan \left( \frac{S}{C} \right) + \pi, & \text{if } C < 0, \\
\arctan \left( \frac{S}{C} \right) + 2\pi, & \text{if } C > 0, \quad S < 0, \\
\frac{3\pi}{2}, & \text{if } C = 0, \quad S < 0, \\
\text{undefined}, & \text{if } C = S = 0.
\end{cases}
\]

It is known that \(A\) is a strictly increasing, strictly concave function with derivative \(A'(\kappa) = (1 - A(\kappa) - A^2(\kappa)) > 0\), and it has the limits \(A(\kappa) \rightarrow 0\) for \(\kappa \rightarrow 0\) and \(A(\kappa) \rightarrow 1\) for \(\kappa \rightarrow \infty\); see \([31]\). Thus, we can compute the updates of \(\kappa\) using Newton’s method. The whole EM algorithm is summarized in Algorithm 4 in the appendix.
EM algorithm for wrapped Gaussians. For the wrapped Gaussians, the components of the log-likelihood function (3.1) read as

\[ p_{u_k}(x_{u_k}^i | \vartheta_k) = \mathcal{N}_w(x_{u_k}^i | \mu_k, \Sigma_k), \quad \mu_k \in \mathbb{T}^n, \Sigma_k \in \text{SPD}(n), \]

where \( n = |u_k| \). Unfortunately, the maximizer in the second M-step of the EM algorithm, Algorithm 1, cannot be computed analytically for the wrapped Gaussian distribution. Therefore we adapt the EM by choosing the variable \( Z \) in an appropriate way. Note, that the resulting EM algorithm is similar to an EM algorithm for non-sparse mixtures of wrapped Gaussians, which was already sketched, e.g., in [1].

**E-step:** Let \( X^1, \ldots, X^N \) be i.i.d. random variables. We assign to each \( X^i \) a label \( W^i_k \) with \( W^i_k = 1 \) if \( X^i \) belongs to the \( k \)-th component of the mixture model and \( W^i_k = 0 \) otherwise. Moreover, recall that for a random variable \( Y \sim \mathcal{N}(\mu, \Sigma) \) it holds that \( Y - \lfloor Y \rfloor \sim \mathcal{N}_w(\mu, \Sigma) \). Thus, we assign for each \( X^i \) a random variable \( Y^i \) such that the conditional distribution of \( (Y^i | W^i_k = 1) \) is given by the distribution \( \mathcal{N}(\mu_k, \Sigma_k) \), and it holds \( X^i = Y^i - \lfloor Y^i \rfloor \). Now we use as hidden variables in the EM algorithm the random variables \( Z^i_{k,l} \), where \( Z^i_{k,l} = 1 \) if \( [Y^i] = l \) and \( W^i_k = 1 \) for \( l \in \mathbb{Z}^{u_k} \) and set \( Z^i_{k,l} = 0 \) otherwise. Let \( \mathcal{X} = (x^i_r) \), and \( Z = (z^i_{k,l})_{r,k,i} \). Then, with the appropriate complete weighted log-likelihood function

\[
\ell(\alpha, \mu, \Sigma | X, Z) := \sum_{i=1}^N w_i \sum_{k=1}^K \sum_{l \in \mathbb{Z}^{u_k}} z^i_{k,l} \log(\alpha_k \mathcal{N}(x_{u_k}^i + l | \mu_k, \Sigma_k)),
\]

the \( Q \)-function reads as

\[
Q((\alpha, \mu, \Sigma), (\alpha^{(r)}, \mu^{(r)}, \Sigma^{(r)})) = \mathbb{E}_{(\alpha^{(r)}, \mu^{(r)}, \Sigma^{(r)})} \left( \ell(\alpha, \mu, \Sigma | X, Z) | X = \mathcal{X} \right)
= \sum_{i=1}^N w_i \sum_{k=1}^K \sum_{l \in \mathbb{Z}^{u_k}} \mathbb{E}_{(\alpha^{(r)}, \mu^{(r)}, \Sigma^{(r)})} \left( Z^i_{k,l} | X = \mathcal{X} \right) \log(\alpha_k \mathcal{N}(x_{u_k}^i + l | \mu_k, \Sigma_k))
= \sum_{i=1}^N w_i \sum_{k=1}^K \sum_{l \in \mathbb{Z}^{u_k}} \beta^{(r)}_{i,k,l} \log(\alpha_k \mathcal{N}(x_{u_k}^i + l | \mu_k, \Sigma_k)),
\]

(3.10)

where by the definition of conditional probabilities

\[
\beta^{(r)}_{i,k,l} = \mathbb{E}_{(\alpha^{(r)}, \mu^{(r)}, \Sigma^{(r)})} \left( Z^i_{k,l} | X^i = x^i \right) = P_{(\alpha^{(r)}, \mu^{(r)}, \Sigma^{(r)})}(Z^i_{k,l} = 1 | X^i = x^i)
= \frac{\alpha^{(r)}_k \mathcal{N}(x_{u_k}^i + l | \mu^{(r)}_k, \Sigma^{(r)}_k)}{\sum_{j=1}^K \alpha^{(r)}_j \mathcal{N}_w(x_{u_k}^i | \mu^{(r)}_j, \Sigma^{(r)}_j)}.
\]

**M-step:** Analogous to the EM algorithm for Gaussian mixture models, the maximizer of the \( Q \) function is given by

\[
\alpha^{(r+1)}_k = \frac{1}{N} \sum_{i=1}^N w_i \sum_{l \in \mathbb{Z}^{u_k}} \beta^{(r)}_{i,k,l},
\mu^{(r+1)}_k = \frac{1}{N \alpha^{(r+1)}_k} \sum_{i=1}^N w_i \sum_{l \in \mathbb{Z}^{u_k}} \beta^{(r)}_{i,k,l} (x_{u_k}^i + l),
\Sigma^{(r+1)}_k = \frac{1}{N \alpha^{(r+1)}_k} \sum_{i=1}^N w_i \sum_{l \in \mathbb{Z}^{u_k}} \beta^{(r)}_{i,k,l} (x_{u_k}^i + l - \mu^{(r+1)}_k)(x_{u_k}^i + l - \mu^{(r+1)}_k)^T.
\]
Unfortunately, for every \( i = 1, \ldots, N \) and \( k = 1, \ldots, K \), there are infinitely many \( \beta_{i,k,l}^{(r)} \). However, by definition, the \( \beta_{i,k,l}^{(r)} \) decay exponentially for \( |l| \to \infty \) since, again by definition, the inverse covariance matrix \( \Sigma^{-1} \) of the wrapped normal distribution is positive definite, which implies that \( -1/2 (x + l - \mu)^T \Sigma^{-1} (x + l - \mu) < 0 \), and this tends to \(-\infty\) as \(|l| \to \infty\).

Thus, for numerical purposes it suffices to consider \( l \in \{-l_{\text{max}}, \ldots, l_{\text{max}}\} \) for some \( l_{\text{max}} \in \mathbb{N} \). In other words, we truncate the infinite sum defining the wrapped Gaussian by

\[
p_{w_k}(x_{u_k} | \mu_k, \Sigma_k) \approx \sum_{l \in \{-l_{\text{max}}, \ldots, l_{\text{max}}\}} N(x_{u_k} + l | \mu_k, \Sigma_k).
\]

Nevertheless, the number of coefficients \( \beta_{i,k,l} \) depends exponentially on the dimension \(|u_k|\) of the wrapped normal distributions. Therefore, parameter estimation can only be performed for small \(|u_k|\) such that the evaluation does not become the bottleneck of the computation.

The whole algorithm is given in Algorithm 2 in the appendix.

**EM algorithm for diagonal wrapped Gaussians.** Using in the mixture models only diagonal wrapped normal distributions as in Example 2.3 ii), we get rid of the exponential dependence of the algorithm on the dimension. This was already observed in [1, 51]. We have to maximize the log-likelihood function

\[
\mathcal{L}(\alpha, \mu, \Sigma | \mathcal{X}) := \sum_{i=1}^{N} w_i \log \left( \sum_{k=1}^{K} \alpha_k \sum_{l \in \mathbb{Z}^{u_k}} \prod_{j \in u_k} N(x_j^i + l_j | \mu_{j,k}, \sigma_{j,k}^2) \right).
\]

**E-step:** This step remains basically the same. However, we will see that we can sum up over appropriate values of \( \beta_{i,k,l}^{(r)} \) to get the values \( \gamma_{i,k,m,j}^{(r)} \). These values can finally be computed efficiently with a complexity that depends polynomially on the dimension \( d \); see equation (3.11).

**M-step:** We rewrite the \( Q \)-function in (3.10) as

\[
Q((\alpha, \mu, \Sigma), (\alpha^{(r)}, \mu^{(r)}, \Sigma^{(r)})) = \sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{l \in \mathbb{Z}^{u_k}} w_i \beta_{i,k,l}^{(r)} \log(\alpha_k) + \sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{l \in \mathbb{Z}^{u_k}} \sum_{j \in u_k} w_i \beta_{i,k,l}^{(r)} \log(N(x_j^i + l_j | \mu_{j,k}, \sigma_{j,k}^2))
\]

\[
= \sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{l \in \mathbb{Z}^{u_k}} w_i \beta_{i,k,l}^{(r)} \log(\alpha_k) + \sum_{i=1}^{N} \sum_{k=1}^{K} \sum_{j \in u_k} \sum_{m \in \mathbb{Z}} w_i \gamma_{i,k,m,j}^{(r)} \log(N(x_j^i + m | \mu_{j,k}, \sigma_{j,k}^2))
\]

where \( \gamma_{i,k,m,j}^{(r)} := \sum_{l \in \mathbb{Z}^{u_k}} l = m \beta_{i,k,l}^{(r)} \). Then, analogous to Gaussian mixture models, maximizing the \( Q \)-function gives that

\[
\alpha_k^{(r+1)} = \frac{1}{N} \sum_{i=1}^{N} w_i \sum_{l \in \mathbb{Z}^{u_k}} \beta_{i,k,l}^{(r)} = \frac{1}{N} \sum_{i=1}^{N} w_i \sum_{m \in \mathbb{Z}} \gamma_{i,k,m,j}^{(r)}, \quad \text{for any } j \in u_k,
\]


\[
\mu_{j,k}^{(r+1)} = \frac{1}{N\alpha_k^{(r+1)}} \sum_{i=1}^{N} u_i \sum_{m \in \mathbb{Z}} \gamma_{i,k,m,j}^{(r)}(x_j^i + m),
\]

\[
(\sigma_{j,k}^{(r+1)})^2 = \frac{1}{N\alpha_k^{(r+1)}} \sum_{i=1}^{N} u_i \sum_{m \in \mathbb{Z}} \gamma_{i,k,m,j}^{(r)}(x_j^i + m - \mu_{j,k}^{(r+1)})^2.
\]

Further, we can rewrite \(\gamma_{i,k,m,j}^{(r)}\) as

\[
\gamma_{i,k,m,j}^{(r)} = \frac{\alpha_k^{(r)} \sum_{l \in \mathbb{Z} / \mu_k, l_j = m} \prod_{s \in u_k} N(x_s^i + l_s | \mu_{s,k}^{(r)}), (\sigma_{s,k}^{(r)})^2)}{\sum_{l=1}^{K} \alpha_t^{(r)} \prod_{s \in u_t} N_w(x_s^i | \mu_{s,t}^{(r)}), (\sigma_{s,t}^{(r)})^2)}
\]

\[
= \frac{\alpha_k^{(r)} \sum_{l \in \mathbb{Z} / \mu_k, l_j = m} \prod_{s \in u_k} N(x_s^i + l_s | \mu_{s,k}^{(r)}), (\sigma_{s,k}^{(r)})^2)}{\sum_{l=1}^{K} \alpha_t^{(r)} \prod_{s \in u_t} N_w(x_s^i | \mu_{s,t}^{(r)}), (\sigma_{s,t}^{(r)})^2)}
\]

\[
= \frac{\alpha_k^{(r)} N(x_j^i + m | \mu_{j,k}^{(r)}, (\sigma_{j,k}^{(r)})^2)}{\sum_{l=1}^{K} \alpha_t^{(r)} \prod_{s \in u_t} N_w(x_s^i | \mu_{s,t}^{(r)}), (\sigma_{s,t}^{(r)})^2)}
\]

As in the previous algorithm, for any \(i, k, j, m\), there are infinitely many \(\gamma_{i,k,m,j}^{(r)}\).

However, with the same justifications as above, the \(\gamma_{i,k,m,j}^{(r)}\) decay exponentially for \(|m| \to \infty\) such that it suffices to consider \(m \in \{-m_{\text{max}}, m_{\text{max}}\}\). While this approximation led to an exponential dependence of the complexity of the previous EM on \(\max_{k=1,\ldots,K} |u_k|\), the complexity of the EM algorithm depends only polynomially on \(\max_{k=1,\ldots,K} |u_k|\).

The whole algorithm is given in Algorithm 3 in the appendix.

Convergence considerations. Finally, we return to the coupled proximum-EM algorithm in (3.4) and (3.5). In the following theorem, we restrict our attention to the mixture model with wrapped Gaussians as components, but the statements apply to the other two models in Example 2.3 as well.

Theorem 3.4. Let \((\alpha^{(r)}, \mu^{(r)}, \Sigma^{(r)})_r\) be generated by (3.4)–(3.5) with one EM step as in Algorithm 2. Then, the following holds true.

1. Assume that \(\alpha_k^{(r)} = 0\). Then, we have that \(\alpha_k^{(r)} = 0\) for any \(r \geq r_0\). In particular, the number \(\|\alpha^{(r)}\|_0\) of non-zero elements in \(\alpha\) is monotonically decreasing.
2. There exists some \(\lambda > 0\) such that for all \(\lambda > \lambda\) the sequence \(\{L_\lambda(\alpha^{(r)}), \mu^{(r)}, \Sigma^{(r)}\}_r\) is monotonically decreasing.
3. Assume that \(\gamma < K - 1\). Then, there exists \(r_0 \in \mathbb{N}\) such that for any \(r \geq r_0\) either \(\alpha_k^{(r)} = 0\) or \(\alpha_k^{(r)} \geq \sqrt{2\pi(K_0 - 1)} / K_0\), \(k = 1, \ldots, K\), where \(K_0 = \min_{r \in \mathbb{N}} \|\alpha^{(r)}\|_0\).

Proof.

1. Assume that \(\alpha_k^{(r)} = 0\). This implies that in Algorithm 2 it holds that \(\beta_{i,k,l}^{(r)} = 0\) for all \(i\) and \(l\), and consequently it holds that \(\alpha_k^{(r+1/2)} = 0\). By part ii) of Lemma 3.1 we obtain that also \(\alpha_k^{(r+1)} = 0\), which by induction proves part i).
2. By the first part of the proof, we conclude that \(R = \{r \in \mathbb{N} : \|\alpha^{(r+1)}\|_0 < \|\alpha^{(r)}\|_0\}\) is finite. Furthermore, by the definition of an EM step it holds that \(\|\alpha^{(r+1)}\|_0 = \|\alpha^{(r)}\|_0\).
for any \( r \in \mathbb{N} \). This in combination with Proposition 3.3 yields that

\[
(3.12) \quad \mathcal{L}_\lambda(\alpha^{(r+1/2)}, \mu^{(r+1)}, \Sigma^{(r+1)}) \leq \mathcal{L}_\lambda(\alpha^{(r)}, \mu^{(r)}, \Sigma^{(r)})
\]

for any \( \lambda > 0 \). Thus, we have for \( r \not\in R \) that \( \| \alpha^{(r+1/2)} \|_0 = \| \alpha^{(r+1)} \|_0 \), which by definition of the proximal operator and of \( h \) in (3.3) yields that \( \alpha^{(r+1/2)} = \alpha^{(r+1)} \).

Together with (3.12), for any \( r \not\in R \) and \( \lambda > 0 \), we obtain that

\[
(3.13) \quad \mathcal{L}_\lambda(\alpha^{(r+1)}, \mu^{(r+1)}, \Sigma^{(r+1)}) \leq \mathcal{L}_\lambda(\alpha^{(r)}, \mu^{(r)}, \Sigma^{(r)}).
\]

For \( r \in R \), we have \( \| \alpha^{(r+1)} \|_0 < \| \alpha^{(r+1/2)} \|_0 \). Consequently, we obtain

\[
\mathcal{L}_\lambda(\alpha^{(r+1/2)}, \mu^{(r+1)}, \Sigma^{(r+1)}) - \mathcal{L}_\lambda(\alpha^{(r+1)}, \mu^{(r+1)}, \Sigma^{(r+1)}) = \mathcal{L}(\alpha^{(r+1/2)}, \mu^{(r+1)}, \Sigma^{(r+1)}|\lambda_\mathcal{X}) - \mathcal{L}(\alpha^{(r+1)}, \mu^{(r+1)}, \Sigma^{(r+1)}|\lambda_\mathcal{X})
\]

\[
+ \lambda(\| \alpha^{(r+1/2)} \|_0 - \| \alpha^{(r+1)} \|_0).
\]

This is greater or equal to zero, if

\[
\lambda \geq \lambda_r := \frac{\mathcal{L}(\alpha^{(r+1)}, \mu^{(r+1)}, \Sigma^{(r+1)}|\lambda_\mathcal{X}) - \mathcal{L}(\alpha^{(r+1/2)}, \mu^{(r+1)}, \Sigma^{(r+1)}|\lambda_\mathcal{X})}{\| \alpha^{(r+1/2)} \|_0 - \| \alpha^{(r+1)} \|_0}.
\]

Together with (3.12), for \( r \not\in R \) and \( \lambda \geq \lambda_r \), we obtain that

\[
(3.14) \quad \mathcal{L}_\lambda(\alpha^{(r+1)}, \mu^{(r+1)}, \Sigma^{(r+1)}) \leq \mathcal{L}_\lambda(\alpha^{(r)}, \mu^{(r)}, \Sigma^{(r)}).
\]

Finally, we set \( \tilde{\lambda} := \max_{r \in R} \lambda_r \), which is finite since \( R \) is finite. Combined with (3.13) and (3.14) this yields the thesis.

iii) As in the previous part of the proof, the set \( R := \{ r \in \mathbb{N} : \| \alpha^{(r+1)} \|_0 < \| \alpha^{(r)} \|_0 \} \) is finite, and for \( r \not\in R \) it holds that \( \alpha^{(r+1)} = \alpha^{(r+1/2)} \).

Now, let \( r_0 := \max_{r \in R} r + 1 \), and assume that there exists \( r \geq r_0 \) with \( \alpha_k^{(r)} \in \left( 0, \sqrt{\frac{2\gamma(K_0 - 1)}{K_0}} \right) \). Then, it holds that \( \| \alpha^{(r)} \|_0 = K_0 \) and

\[
\alpha^{(r)} = \text{prox}_{\gamma h}(\alpha^{(r-1/2)}) = \text{prox}_{\gamma h}(\alpha^{(r)}).
\]

Moreover, define \( \tilde{\alpha} \in \Delta_K \) by \( \tilde{\alpha}_k = 0 \) and

\[
\tilde{\alpha}_j = \begin{cases} 
0, & \text{if } \alpha_j^{(r)} = 0, \\
\alpha_j^{(r)} + \frac{\alpha_k^{(r)}}{K_0 - 1}, & \text{otherwise.}
\end{cases}
\]

This yields

\[
\frac{1}{2\gamma} \| \tilde{\alpha} - \alpha^{(r)} \|^2 + \| \tilde{\alpha} \|_0 = \frac{1}{2\gamma} \left( (\alpha_k^{(r)})^2 + (K_0 - 1) \left( \frac{\alpha_k^{(r)}}{K_0 - 1} \right)^2 \right) + \| \alpha^{(r)} \|_0 - 1
\]

\[
= \frac{1}{2\gamma} \frac{K_0}{K_0 - 1} \left( (\alpha_k^{(r)})^2 + \| \alpha^{(r)} \|_0 - 1 < \| \alpha^{(r)} \|_0,
\]

which is a contradiction to (3.15), and the proof is completed. \( \square \)
3.3. Model selection. The model and optimization algorithm in the previous section assumed that the $u_k$, $k = 1, \ldots, K$, are known. In the following, we propose a heuristic for selecting the $u_k$.

The underlying assumption is that the distribution of the samples $(x^i)$, can be represented by a sparse mixture model

$$p(x) = \sum_{k=1}^{K} p(x_{uk}|\theta_k)$$

with a small number of variable interactions, i.e., $|u_k| \leq d_s$, $k = 1, \ldots, K$, for some small $d_s \in \{1, \ldots, d\}$. Here, we assume that the number $d_s$ is known a priori. Furthermore, we assume that the number $K$ of required components in the sparse mixture model is small.

For our heuristic, we start with $K = 1$ and $u_1 = \emptyset$, and then we extend our model iteratively by repeating the following steps $d_s$ times.

1. For every $k = 1, \ldots, K$, we first compute the probability that the sample $x^i$ belongs to component $k$ of the sparse mixture model. This probability is given by

$$\beta_{i,k} = \frac{\alpha_k p(x^i_{uk}|\theta_k)}{\sum_{j=1}^{K} \alpha_j p(x^i_{uj}|\theta_j)}.$$

For $m \in [d] \setminus u_k$, we want to test if we can fit the weighted samples $(x^i)_i$ with importance weights $(w_i\beta_{i,k})$, better by a density function $p(x^i_{uk} \cup (m)|\mu_k, \Sigma_k)$ than by the density function $p(x_{uk}|\mu_k, \Sigma_k)$. If the distribution $p(x_{uk}|\mu_k, \Sigma_k)$ fits the samples perfectly, then we have that for any $m \in [d] \setminus u_k$, the samples $(x^i_{m})_i$ with importance weights $(w_i\beta_{i,k})$, are uniformly distributed and independent from $(x^i_{uk})_i$ with weights $(w_i\beta_{i,k})_i$. Consequently, we apply two tests. First, we apply a Kolmogorov-Smirnov test described in Appendix B for the hypothesis

$$H_0: \text{the samples } (x^i_{m})_i \text{ with weights } (w_i\beta_{i,k})_i \text{ are uniformly distributed}$$

against the alternative

$$H_1: \text{the samples } (x^i_{m})_i \text{ with weights } (w_i\beta_{i,k})_i \text{ are not uniformly distributed.}$$

The hypothesis $H_0$ is accepted if the Kolmogorov-Smirnov test statistic (see equation (B.2)) fulfills $\text{KS}((w_i\beta_{i,k}), (x^i_{m})_i) < c_1$ for some a priori fixed $c_1 \in \mathbb{R}_{>0}$. Since it is difficult to test for independence, our second test is based on the correlation. Here, we test the hypothesis

$$\tilde{H}_0: (w_i\beta_{i,k}, x^i_{m})_i \text{ and } (w_i\beta_{i,k}, x^i_{uk})_i \text{ are uncorrelated}$$

against the alternative

$$\tilde{H}_1: (w_i\beta_{i,k}, x^i_{m})_i \text{ and } (w_i\beta_{i,k}, x^i_{uk})_i \text{ are correlated.}$$

The hypothesis $\tilde{H}_0$ is accepted, if the correlation coefficient satisfies

$$|\text{Corr}((w_i\beta_{i,k}, x^i_{m})_i, (w_i\beta_{i,k}, x^i_{j})_i)| < c_2$$

for all $j \in u_k$ and some a priori fixed $c_2 \in \mathbb{R}_{>0}$. Now, we set

$$U_k := \{u_k\} \cup \{u_k \cup \{m\} : m \in [d] \setminus u_k \text{ with } H_0 \text{ is rejected or } \tilde{H}_0 \text{ is rejected}\}.$$
and define a new sparse mixture model with $\tilde{K} = \sum_{k=1}^{K} |U_k|$ components, where the new $u_k$ are given by the elements of the $U_k$, $k = 1, \ldots, K$. For wrapped normal distributions, we initialize the parameters of $u_k \cup \{m\}$ by the following procedure: first, we estimate the parameters $(\hat{\mu}, \hat{\sigma}^2)$ of a univariate wrapped normal distribution based on the samples $(x_{im})$ with importance weights $(w_i, \beta_i, k)$. Then, we initialize the component with indices $u_k \cup \{m\}$ by the parameters of the distribution of a random variable $X \sim N_{w}(\mu_k, \Sigma_k) \times N_{w}(\hat{\mu}, \hat{\sigma}^2)$, where $(\mu_k, \Sigma_k)$ are the old parameters corresponding to $u_k$.

2. As a second step, we estimate the parameters $(\alpha, \vartheta)$ of the new sparse mixture model using the iterations (3.4) and (3.5).

3. Finally, we reduce the number of components of the sparse mixture model by
   i) removing all components $k$ with weight $\alpha_k = 0$ and
   ii) replacing the components $k$ and $l$ with $u_k = u_l$ by one component $u_k$ with weight $\alpha_k + \alpha_l$, $\mu_k$, and $\Sigma_k$ if the corresponding distributions are similar. As a similarity measure, we use here the Kullback-Leibler divergence, which can be approximated by the Monte Carlo method as
   $$KL(p, q) \approx \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} \log(p(s_i)) - \log(q(s_i)),$$
   where the $s_i$ are sampled from the probability distribution corresponding to the density $p$.

4. Numerical results. In this section, we demonstrate the performance of our algorithm by four numerical examples. The implementation is done in Tensorflow and Python. The code is available online².

In the first two sections, the non-negative density function $f : [0, 1]^d \rightarrow \mathbb{R}$ with $\|f\|_{L^1} = 1$ is given as ground truth, and we can sample from the corresponding distribution. More precisely, we consider the following functions:

1. two mixture models,
2. the sum of the tensor products of splines, which was also considered in [4], and
3. the normalized Friedman-1 function, which was also examined in [5, 6, 44].

The samples in the third section are created in a special way, and the underlying density function is unknown.

Since the reconstruction quality possibly depends on the random choice of the samples, we repeat this procedure 10 times. In the first example, we can directly sample from the distribution, while we use rejection sampling for the two other ones; see, e.g., [2, 49]. This works as follows.

Let $M \geq \sup_{x \in [0, 1]} f(x)$. Now, we generate a candidate by drawing $x$ from the uniform distribution on $U_{[0, 1]^d}$ and $z$ from $U_{[0, 1]}$. Then, we accept $x$ as a sample if $z < f(x)/M$ and reject $x$ otherwise. It can be shown that the samples $x^i, i = 1, \ldots, N$, generated by this procedure correspond to the distribution given by the density $f$; see, e.g., [49, pp. 49].

To evaluate our reconstruction results, we compare
   - the value of the log-likelihood function of the original function $f$ with that of the estimated one $\hat{p}$, and

²https://github.com/johertrich/Sparse_Mixture_Models
the relative $L_q$-errors, $q = 1, 2$,

$$e_{L_q}(\hat{p}, f) = \frac{\|f - \hat{p}\|_{L_q}}{\|f\|_{L_q}}.$$ 

Since we cannot calculate the high-dimensional integral $e_{L_q}$ directly, we approximate it via Monte-Carlo integration, that is, we draw $N_{MC} = 100000$ samples $s_1, \ldots, s_{N_{MC}}$ from the uniform distribution in $[0, 1]^d$ and approximate the $L_q$-norms by

$$\|f\|_{L_q}^q \approx \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} f(s_i)^q, \quad \|f - \hat{p}\|_{L_q}^q \approx \frac{1}{N_{MC}} \sum_{i=1}^{N_{MC}} (f(s_i) - \hat{p}(s_i))^q.$$

4.1. Samples from mixture models. For $d = 9$, we consider the ground truth density function

$$f(x) := \sum_{k=1}^{6} \alpha_k p_{u_k}(x_{u_k} | \mu_k, \Sigma_k),$$

(4.1)

$$p_{u_k}(x_{u_k} | \mu_k, \Sigma_k) := \sum_{l \in \mathbb{Z}^{u_k}} N(x_{u_k} + l | \mu_k, \Sigma_k)$$

with

$$(u_1, \ldots, u_6) := (\{0,1\}, \{2,3\}, \{4,5,6\}, \{6,7\}, \{8,9\}, \{2\}),$$

$$\alpha := (0.2, 0.2, 0.2, 0.2, 0.1, 0.1),$$

$$\mu := \frac{1}{2} \left( (1,1)^T, (1,1)^T, (1,1)^T, (1,1)^T, (1,1)^T, 1 \right),$$

and the following settings of covariance matrices:

a) Diagonal matrices: for $k = 1, \ldots, 6$,

$$\Sigma_k = \sigma^2 I_{|u_k|}, \quad \sigma^2 = 0.01.$$ 

b) Non-diagonal matrices: for $k = 1, 2, 4, 5$,

$$\Sigma_k := \sigma^2 \begin{bmatrix} 1 & c_k \\
 c_k & 1 \end{bmatrix}, \quad c_1 = c_2 = 0.5, \ c_4 = -0.6, \ c_5 = 0.1,$$

and

$$\Sigma_3 := \sigma^2 \begin{bmatrix} 1 & 0.3 & 0.2 \\
 0.3 & 1 & 0.1 \\
 0.2 & 0.1 & 1 \end{bmatrix}, \quad \Sigma_6 = \sigma^2, \quad \sigma^2 = 0.01.$$ 

We perform all computations for $N = 10000$ and $\tilde{N} = 50000$ samples. We iterate the heuristic from Section 3.3 for $d_s = 3$ times. Then, the average relative $L_q$-errors, $q = 1, 2$, as well as the log-likelihood values are given for the three settings from Example 2.3 in Table 4.1. Figure 4.1 displays a diagram with the weights of the recovered couplings $u_k$. More precisely, the value of the bar with label $u$ is given by the sum of all $\alpha_k$, where $u_k = u$ in the reconstruction.

We observe that the couplings $u_k$ are reconstructed exactly. Furthermore, the log-likelihood value for the sparse mixture model with full wrapped Gaussian covariance matrices
is in all cases larger than for the ground truth function. Thus, the approximation error is due to the estimation error of the maximum likelihood estimator and therefore due to the lack of information contained in the samples and not due to the approximation method. Moreover, the sparse mixture model with full wrapped Gaussian covariances admits in all cases a larger log-likelihood value than those with the diagonal wrapped Gaussians. This should not be surprising since the sparse mixture model with the wrapped Gaussian covariance matrices contains the other setting.

### 4.2. Samples from functions

Next, we approximate the functions \( f_i, i = 1, 2 \), where \( f_1: [0, 1]^9 \rightarrow \mathbb{R} \) and \( f_2: [0, 1]^{10} \rightarrow \mathbb{R} \) are given by

\[
\begin{align*}
\text{(4.2): } f_1(x) &= B_2(x_1)B_4(x_3)B_6(x_5) + B_2(x_2)B_4(x_5)B_6(x_6) + B_2(x_4)B_4(x_7)B_6(x_9), \\
\text{(4.3): } f_2(x) &= 10 \sin(\pi x_1 x_2) + 20(x_3 + 0.5)^2 + 10x_4 + 5x_5.
\end{align*}
\]
Table 4.2
Approximation of the \( f_i \) in (4.2) and (4.3) by the three sparse mixture models in Example 2.3 for \( N = 10000 \) samples. Average value of the log-likelihood function and the relative \( L_q \)-errors, \( q = 1, 2 \). Top: spline function \( f_1 \), bottom: Friedmann-1 function \( f_2 \).

<table>
<thead>
<tr>
<th>Method</th>
<th>( \mathcal{L}(x^1, \ldots, x^N) )</th>
<th>( \mathcal{L}_{\hat{p}}(x^1, \ldots, x^N) )</th>
<th>( \epsilon_{L^1}(\hat{p}, f) )</th>
<th>( \epsilon_{L^2}(\hat{p}, f) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>full</td>
<td>7009.8 ± 59.4</td>
<td>7026.7 ± 79.2</td>
<td>0.0971 ± 0.0053</td>
<td>0.0963 ± 0.0056</td>
</tr>
<tr>
<td>diagonal</td>
<td>7009.8 ± 59.4</td>
<td>7001.9 ± 52.0</td>
<td>0.0828 ± 0.0030</td>
<td>0.0828 ± 0.0037</td>
</tr>
<tr>
<td>von Mises</td>
<td>7009.8 ± 59.4</td>
<td>6932.4 ± 56.4</td>
<td>0.1292 ± 0.0034</td>
<td>0.1272 ± 0.0041</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>( \mathcal{L}(x^1, \ldots, x^N) )</th>
<th>( \mathcal{L}_{\hat{p}}(x^1, \ldots, x^N) )</th>
<th>( \epsilon_{L^1}(\hat{p}, f) )</th>
<th>( \epsilon_{L^2}(\hat{p}, f) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>full</td>
<td>630.0 ± 43.9</td>
<td>566.1 ± 43.0</td>
<td>0.1150 ± 0.0131</td>
<td>0.1419 ± 0.0142</td>
</tr>
<tr>
<td>diagonal</td>
<td>630.0 ± 43.9</td>
<td>558.6 ± 53.8</td>
<td>0.1175 ± 0.0135</td>
<td>0.1444 ± 0.0139</td>
</tr>
<tr>
<td>von Mises</td>
<td>630.0 ± 43.9</td>
<td>549.8 ± 56.4</td>
<td>0.1220 ± 0.0124</td>
<td>0.1490 ± 0.0128</td>
</tr>
</tbody>
</table>

![Fig. 4.2. Weights of the recovered couplings \( u_k \) for the approximation of the \( f_i \) in (4.2) and (4.3) by the three sparse mixture models in Example 2.3 for \( N = 10000 \) samples. Top: spline function \( f_1 \), bottom: Friedmann-1 function \( f_2 \).](image)

Here \( B_2, B_4, \) and \( B_6 \) are the \( L_2 \)-normalized \( B \)-splines of order 2, 4, and 6 supported on \( [0, 1] \). Note, that the spline function \( f_1 \) was also used in [4] for numerical evaluations. The function \( f_2 \) is the so-called Friedmann-1 function. We use \( d_s = 3 \) iterations within the heuristic of Section 3.3 for \( f_1 \) and \( d_s = 2 \) iterations for \( f_2 \). The results are given in Table 4.2. Figure 4.2 displays a diagram with the weights of the recovered couplings \( u_k \). We observe that all recovered couplings \( u_k \) with a significant weight match the definitions of the functions \( f_i \), \( i = 1, 2 \).

The estimated parameters achieve a slightly worse log-likelihood value than the original function \( f \). Thus, the original function fits the samples slightly better than the estimated sparse mixture model. This should not be surprising since the function \( f/f_{\|f\|_{L_1}} \) is not contained in the class of sparse mixture models. Nevertheless, for the spline function \( f_1 \) the relative \( L_1 \) and \( L_2 \)-errors are still comparable with the results from [4].
4.3. A synthetic image example. In this example, we consider gray-valued images consisting of $7 \times 10$ pixels that are sampled from five different classes. Each class contains noisy piecewise constant images with one straight edge in a fixed position. In the following we learn a sparse mixture model for the semi-supervised classification of images into one of these classes. Here we assume that not the whole image is given but only the orientations of some of the gradients within the images.

Image generation. We generate an image $y = (y_{i,j})_{i,j} \in \mathbb{R}^{7 \times 10}$ using the following procedure:

- **Class 1:** we draw $a$ from $\mathcal{N}(0.1, 0.05^2)$ and $b$ from $\mathcal{N}(0.9, 0.1^2)$ and set $y_{i,j} = a$ for $j = 1, 2$ and $y_{i,j} = b$ otherwise.
- **Class 2:** we draw $a$ from $\mathcal{N}(0.9, 0.1^2)$ and $b$ from $\mathcal{N}(0.1, 0.05^2)$ and set $y_{i,j} = a$ for $j = 1, \ldots, 4$ and $y_{i,j} = b$ otherwise.
- **Class 3:** we draw $a$ from $\mathcal{N}(0.2, 0.025^2)$ and $b$ from $\mathcal{N}(0.6, 0.05^2)$ and set $y_{i,j} = a$ for $j = 1, \ldots, 6$ and $y_{i,j} = b$ otherwise.
- **Class 4:** we draw $a$ from $\mathcal{N}(0.7, 0.1^2)$ and $b$ from $\mathcal{N}(0.1, 0.05^2)$ and set $y_{i,j} = a$ for $j = 1, \ldots, 8$ and $y_{i,j} = b$ otherwise.
- **Class 5:** we draw $a$ from $\mathcal{N}(0.2, 0.1^2)$ and $b$ from $\mathcal{N}(0.9, 0.025^2)$ and set $y_{i,j} = a$ for $i = 1, \ldots, 4$ and $y_{i,j} = b$ otherwise.

Finally, we add Gaussian white noise with standard deviation $0.2$ to each of the images.

Figure 4.3 illustrates one sample from each class.

Gradient orientations and data generation. We assume that not the full images are given but only the orientations of some of the gradients within the images. For an image $y = (y_{i,j})_{i,j} \in \mathbb{R}^{m \times n}$, the central discrete gradient at position $(i, j) \in \{2, \ldots, m-1\} \times \{2, \ldots, n-1\}$ is defined as the vector $(y_{i+1,j} - y_{i-1,j}, y_{i,j+1} - y_{i,j-1})$. Consequently, the orientation of the gradient at position $(i, j)$ is given by

$$\omega_{i,j} := \frac{1}{2\pi} \arctan^* \left( \frac{y_{i+1,j} - y_{i-1,j}}{y_{i,j+1} - y_{i,j-1}} \right),$$

where $\arctan^*$ again denotes the quadrant-specific inverse of the tangent as defined in (3.9).

Finally, we assume that not all of the gradient orientations are given but only the $\omega_{i,j}$ with $(i,j) \in \mathcal{I}$, where

$$\mathcal{I} := \{(2, 2), (2, 3), (2, 6), (2, 7), (4, 4), (4, 5), (4, 8), (4, 9), (6, 2), (6, 3), (6, 6), (6, 7)\}.$$
ETNA
Kent State University and
Johann Radon Institute (RICAM)

SPARSE MIXTURE MODELS INSPIRED BY ANOVA

FIG. 4.4. The black marked pixels represent the positions \((i, j) \in \mathcal{I}\).

FIG. 4.5. Histograms of the components of \((\omega_{ij})_{(i,j) \in \mathcal{I}}\) for 10000 samples from class 2.

Figure 4.4 visualizes the pixels corresponding to the positions \((i, j) \in \mathcal{I}\).

Now, we generate \(N_{\text{train}} = 10000\) training samples and \(N_{\text{test}} = 1000\) test samples, where each sample \(x \in \mathbb{T}^{12}\) is generated by the following steps: first, we randomly choose a class \(c \in \{1, \ldots, 5\}\) accordingly to some fixed (but unknown for the classification) probabilities \(\alpha\). Second, we generate an image \(y\) from a class \(c\) as described above and compute the gradient orientations \(\omega_{ij}\). Finally, we define our sample \(x \in \mathbb{T}^{12}\) as \(x := (\omega_{ij})_{(i,j) \in \mathcal{I}}\).

We visualize the components of \((\omega_{ij})_{(i,j) \in \mathcal{I}}\) by histograms of 10000 samples from class 2 in Figure 4.5.

REMARK 4.1. Let us briefly comment why the above sampling generation fits into our model setting. If the \(Y_{i,j}\) are i.i.d. Gaussian distributed (noise on constant areas), then the components of their centered gradients are also i.i.d. Gaussian distributed for \((i, j) \in \mathcal{I}\). Note that the special set \(\mathcal{I}\) is needed to ensure independence of the random variables in the gradient. Finally, it follows from the transformation theorem that the random variables where these \(\omega_{i,j}\) are sampled from, are uniformly distributed; see, e.g., [18]. Thus, in our samples \(x\), most of the components will arise from a uniformly distributed random variable (constant areas), and only few ones (phases of the gradients at edges) have to be approximated by a Gaussian-like mixture.

Semi-supervised classification. In the following, assume that we have given \(N_{\text{train}}\) unlabeled training samples \(x^1, \ldots, x^{N_{\text{train}}}\) (i.e., it is unknown which sample belongs to which class). Additionally, we have given \(3\) labeled samples \(x^{1,c}, x^{2,c}, x^{3,c}\) from each class \(c = 1, \ldots, 5\). Based on this, we would like to classify the samples from the test set into the five classes. For this, we perform three steps.
First, we estimate the components $u_k$ and the parameters $(\alpha, \mu, \Sigma)$ of a sparse mixture model as described in Section 3, where we use $d_s = 4$ steps within the heuristic from Section 3.3.

In general, this mixture model will have $K > 5$ classes, and we have to assign the components to the different classes. For this we assume that we have additionally given three labeled samples $x^{1,c}, \ldots, x^{i,c}$ from each class $c = 1, \ldots, 5$. Then, we assign $k$ to class $c(k) \in \{1, \ldots, 5\}$ by

$$c(k) := \arg \max_{c=1, \ldots, 5} \sum_{i=1}^{3} \log(p_{u_k}(x^{i,c} | \vartheta_k)).$$

Finally, we predict for a data point $x$ from the test set the appropriate class by

$$\hat{c} := \arg \max_{c=1, \ldots, 5} \sum_{k \in \{K\}, c_k = c} \alpha_k p_{u_k}(x | \vartheta_k).$$

Using this procedure, we achieve an accuracy of 93.6% on the test set.

Acknowledgments. Funding by the BMBF 01|S20053B project $\text{SA} \ell$E and by the German Research Foundation (DFG) within the project STE 571/16-1 SUPREMATIM is gratefully acknowledged.

Appendix A. EM algorithm for distributions in Example 2.3. We summarize the EM algorithms for the mixture models with components from Example 2.3 i)–iii) in this order.

Algorithm 2 EM algorithm for MMs with wrapped normal distribution.

Input: $x = (x^1, \ldots, x^N) \in \mathbb{R}^{d,N}$, initialization $\alpha^{(0)}, \vartheta^{(0)} = (\mu^{(0)}, \Sigma^{(0)}).$

for $r = 0, 1, \ldots$ do

E-Step: for $k = 1, \ldots, K$, $i = 1, \ldots, N$, and $l \in \mathbb{Z}^{|u|}$, compute

$$\beta_{i,k,l}^{(r)} = \frac{\alpha_k^{(r)} N(x_{uk} + l | \mu_k^{(r)}, \Sigma_k^{(r)})}{\sum_{j=1}^{K} \alpha_j^{(r)} p(x_{uj} | \mu_j^{(r)}, \Sigma_j^{(r)})}.$$

M-Step: for $k = 1, \ldots, K$, compute

$$\alpha_k^{(r+1)} = \frac{1}{N} \sum_{i=1}^{N} w_i \sum_{l \in \mathbb{Z}^{|u|}} \beta_{i,k,l}^{(r)},$$

$$\mu_k^{(r+1)} = \frac{1}{N \alpha_k^{(r+1)}} \sum_{i=1}^{N} w_i \sum_{l \in \mathbb{Z}^{|u|}} \beta_{i,k,l}^{(r)} (x_{uk} + l),$$

$$\Sigma_k^{(r+1)} = \frac{1}{N \alpha_k^{(r+1)}} \sum_{i=1}^{N} w_i \sum_{l \in \mathbb{Z}^{|u|}} \beta_{i,k,l}^{(r)} (x_{uk} + l - \mu_k^{(r+1)})(x_{uk} + l - \mu_k^{(r+1)})^T.$$

end for
Algorithm 3 EM algorithm for MMs with diagonal wrapped normal distribution.

Input: \( x = (x^1, \ldots, x^N) \in \mathbb{R}^{d,N} \), initialization \( \alpha^{(0)}, \varphi^{(0)} = (\mu^{(0)}, (\sigma^{(0)})^2) \).

for \( r = 0, 1, \ldots \) do

E-Step: for \( k = 1, \ldots, K, i = 1, \ldots, N, j \in u_k \), and \( m \in \mathbb{Z} \), compute
\[
\gamma_{i,k,m,j}^{(r)} = \frac{\alpha_k^{(r)} \mathcal{N}(x_i^j + m|\mu_{j,k}^{(r)}, (\sigma_{j,k}^{(r)})^2) \prod_{s \in u_k \setminus \{j\}} \mathcal{N}_w(x_i^j|\mu_{s,k}^{(r)}, (\sigma_{s,k}^{(r)})^2)}{\sum_{t=1}^K \alpha_t^{(r)} \prod_{s \in u_t} \mathcal{N}_w(x_i^j|\mu_{s,t}^{(r)}, (\sigma_{s,t}^{(r)})^2)}
\]

M-Step: for \( k = 1, \ldots, K \), compute
\[
\alpha_k^{(r+1)} = \frac{1}{N} \sum_{i=1}^N w_i \sum_{m \in \mathbb{Z}} \gamma_{i,k,m,j}^{(r)} \text{ for any } j \in u_k
\]
\[
\mu_{j,k}^{(r+1)} = \frac{1}{N \alpha_k^{(r+1)}} \sum_{i=1}^N w_i \sum_{m \in \mathbb{Z}} \gamma_{i,k,m,j}^{(r)} (x_i^j + m),
\]
\[
(\sigma_{j,k}^{(r+1)})^2 = \frac{1}{N \alpha_k^{(r+1)}} \sum_{i=1}^N w_i \sum_{m \in \mathbb{Z}} \gamma_{i,k,m,j}^{(r)} (x_i^j + m - \mu_{j,k}^{(r+1)})^2.
\]
end for

Algorithm 4 EM algorithm for MMs with products of von Mises distributions.

Input: \( x = (x^1, \ldots, x^N) \in \mathbb{R}^{d,N} \), initialization \( \alpha^{(0)}, \varphi^{(0)} = (\mu_{j,k}^{(0)}, \kappa_{j,k}^{(0)}) \).

for \( r = 0, 1, \ldots \) do

E-Step: for \( k = 1, \ldots, K, i = 1, \ldots, N \), compute
\[
\beta_{i,k}^{(r)} = \frac{\alpha_k^{(r)} \prod_{j \in u_k} \mathcal{PM}(x_i^j|\mu_{j,k}^{(r)}, \kappa_{j,k}^{(r)})}{\sum_{t=1}^K \alpha_t^{(r)} \prod_{j \in u_t} \mathcal{PM}(x_i^j|\mu_{j,t}^{(r)}, \kappa_{j,t}^{(r)})}
\]

M-Step: for \( k = 1, \ldots, K \) and \( j \in u_k \), compute
\[
\alpha_k^{(r+1)} = \frac{1}{N} \sum_{i=1}^N w_i \beta_{i,k}^{(r)}, \quad \mu_{j,k}^{(r+1)} = \arctan \left( \frac{S_{j,k}^{(r)}}{C_{j,k}^{(r)}} \right), \quad \kappa_{j,k}^{(r+1)} = A^{-1}(R_{j,k}^{(r)}),
\]
where
\[
C_{j,k}^{(r)} = \sum_{i=1}^N w_i \beta_{i,k}^{(r)} \cos(2\pi x_i^j), \quad S_{j,k}^{(r)} = \sum_{i=1}^N w_i \beta_{i,k}^{(r)} \sin(2\pi x_i^j),
\]
\[
R_{j,k}^{(r)} = \frac{1}{N \alpha_k} \sqrt{(S_{j,k}^{(r)})^2 + (C_{j,k}^{(r)})^2}.
\]
end for

Appendix B. The weighted Kolmogorov-Smirnov test. We briefly review the weighted Kolmogorov-Smirnov (KS) test. The following definition and facts about the KS test can be found in [45]. Given univariate samples \( x^1, \ldots, x^N \in [0, 1] \) and a probability distribution
defined by its cumulative distribution function \( f : [0, 1] \rightarrow [0, 1] \), we test the hypothesis

\[ H_0 : (x^i)_i \text{ belong to the distribution } f \]

against the alternative

\[ H_1 : (x^i)_i \text{ belong not to the distribution } f. \]

We define the empirical cumulative density function \( f_N : [0, 1] \rightarrow [0, 1] \) of the samples \((x_i^i)_i\) by

\[ f_N = \frac{1}{N} \sum_{i=1}^{N} 1_{[x_i^i, 1]}, \]

Then, the hypothesis \( H_0 \) is accepted, if the test statistic

\[ \text{KS}((x_i^i)_i) := \sqrt{N} \| f_N - f \|_{L^\infty} \]

is smaller or equal than some constant \( c \), which is fixed a priori and controls the significance level of the test. It is shown that, for \( X_1, \ldots, X_N \) being i.i.d. random variables with cumulative distribution function \( f \), the KS test statistic \( \text{KS}((X_i^i)_i) \) converges in distribution to the Kolmogorov distribution as \( N \rightarrow \infty \). The test can be extended for weighted samples \((w_i, x_i^i)_i \in \mathbb{R}_{>0} \times [0, 1] \) by replacing the empirical cumulative distribution function by

\[ f_N = \frac{1}{\sum_{i=1}^{N} w_i} \sum_{i=1}^{N} w_i 1_{[x_i^i, 1]}, \]

Furthermore, one has to replace \( \sqrt{N} \) in (B.1). Here, as suggested in [45], we replace \( N \) by \( \frac{\left( \sum_{i=1}^{N} w_i \right)^2}{\sum_{i=1}^{N} w_i^2} \). Thus, the weighted KS test statistic reads as

\[ \text{KS}((w_i, x_i^i)_i) = \sqrt{\frac{\left( \sum_{i=1}^{N} w_i \right)^2}{\sum_{i=1}^{N} w_i^2}} \| f_N - f \|_{L^\infty}. \]

**Remark B.1.** Note that for two cumulative distribution functions \( f \) and \( g \), the term \( d(f, g) = \| f - g \|_{L^\infty} \) defines a metric on the probability measures on \([0, 1]\). In particular, for fixed weights \( w_i \), the weighted KS test statistic can be interpreted as the distance of the measure induced by \( f \) to the measure \( \frac{1}{\sum_{i=1}^{N} w_i} \sum_{i=1}^{N} w_i \delta_{x_i^i} \), where \( \delta_x \) is the Dirac-measure in \( x \).

**Remark B.2.** For the uniform distribution, the weighted KS test statistic can be easily computed. Assume that the \( x_i^i \) are sorted, i.e., \( x^1 \leq \cdots \leq x^N \), and define \( s_i = \frac{\sum_{j=1}^{i} w_j}{\sum_{j=1}^{N} w_j} \). Then, the statistic in (B.2) is given by

\[ \text{KS}((w_i, x_i^i)_i) = \sqrt{\frac{\left( \sum_{i=1}^{N} w_i \right)^2}{\sum_{i=1}^{N} w_i^2}} \max_{i=1, \ldots, N} \{ \max(s_i - x_i^i, x_i^i - s_{i-1}) \}. \]

**References**