Maximum Smoothed Likelihood for Multivariate Mixtures

by

M. Levine
Purdue University

D.R. Hunter
Pennsylvania State University

D. Chauveau
University of Orleans

Technical Report #10-02

Department of Statistics
Purdue University

April 2010

◇

Maximum Smoothed Likelihood for Multivariate Mixtures

Michael Levine, David R. Hunter*, Didier Chauveau

April 15, 2010

Abstract

We introduce an algorithm for estimating the parameters in a finite mixture of completely unspecified multivariate components in at least three dimensions under the assumption of conditionally independent coordinate dimensions. We prove that this algorithm, based on a majorization-minimization idea, possesses a desirable descent property just as any EM algorithm does. We discuss the similarities between our algorithm and a related one—the so-called nonlinearly smoothed EM, or NEMS, algorithm for the non-mixture setting. We also demonstrate via simulation studies that the new algorithm gives very similar results to another algorithm that does not satisfy any descent algorithm, thus validating the latter algorithm, which can be simpler to program. We provide code for implementing the new algorithm in a publicly-available R package.

Keywords: EM algorithms, MM algorithms, NEMS, nonparametric mixtures

1 Introduction

Suppose the r-dimensional vectors $\mathbf{X}_1, \ldots, \mathbf{X}_n$ are a simple random sample from a finite mixture density of m components $f_1, \ldots, f_m$, with $m > 1$ and known in advance. It is assumed throughout this manuscript that each one of these densities f_j is equal with probability 1 to the product of its marginal densities:

$$f_j(\mathbf{x}) = \prod_{k=1}^{r} f_{jk}(x_k) \tag{1}$$

This, in turn, means that, conditional on knowing the particular subpopulation the observation $\mathbf{X}_j$ came from, its coordinates are independent. This

*Corresponding author. Address: Department of Statistics, Pennsylvania State University, University Park, PA 16802, USA. dhunter@stat.psu.edu

conditional independence assumption has appeared in a growing body of literature on non- and semi-parametric multivariate mixture models; see Benaglia et al. (2009b) for a discussion of the relevant literature.

We let $\boldsymbol{\theta}$ denote the vector of parameters, including the mixing proportions $\lambda_1, \ldots, \lambda_m$ and the univariate densities f_{jk}. Here and throughout the article, j is the component index and k indexes the coordinate. Consequently, $1 \leq j \leq m$ and $1 \leq k \leq r$. Therefore, under the assumption of conditional independence, the mixture density evaluated at the point $\mathbf{x}_i = (x_{i1}, \ldots, x_{ir})^\top$ can be represented as

$$g_{\boldsymbol{\theta}}(\mathbf{x}_i) = \sum_{j=1}^{m} \lambda_j \prod_{k=1}^{r} f_{jk}(x_{ik}), \tag{2}$$

where $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_m)$ must satisfy

$$\sum_{j=1}^{m} \lambda_j = 1 \quad \text{and each } \lambda_j \geq 0. \tag{3}$$

The question of identifiability of the parameters in Equation (2) is of central theoretical importance. By identifiability, we refer to the question of when $g_{\boldsymbol{\theta}}$ uniquely determines $\boldsymbol{\lambda}$ and each of the f_{jk}, at least up to so-called "label-switching" and any changes to the densities f_{jk} that occur on a set of Lebesgue measure zero that do not therefore change the distributions F_{jk} (here, "label-switching" refers to permuting the order of the summands, 1 through m, in equation (2)). Hall and Zhou (2003) established that when $m = 2$, identifiability of parameters generally follows in $r \geq 3$ dimensions but not in fewer than three. However, a general result for more than two components proved elusive, though several articles on the topic appeared in the literature (e.g., Hall et al., 2005; Kasahara and Shimotsu, 2008). Then, Allman et al. (2009) finally proved an elegant and powerful result using a theorem of Kruskal (1977), establishing the identifiability of the parameters in (2) whenever $r \geq 3$, regardless of m, under weak conditions. To wit, the identifiability follows as long as for each k, the density functions $f_{1k}, \ldots, f_{mk}$ are linearly independent (except possibly on a set of Lebesgue measure zero, of course).

An EM-like algorithm designed to estimate $\boldsymbol{\theta}$ in model (2) was introduced in Benaglia et al. (2009b). That algorithm can handle any number of mixture components and any number of vector coordinates of the multivariate observations, unlike other existing algorithms. It also yields considerably smaller mean integrated squared errors than an alternative algorithm (Hall et al., 2005) in a simulation study. However, despite its empirical success, this algorithm lacks any sort of theoretical justification; indeed, it can only

be called "EM-like" because it resembles an EM algorithm in certain aspects of its formulation. The current article corrects this shortcoming by introducing a smoothed loglikelihood function and formulating an iterative algorithm with a provable monotonicity property that happens to produce results in that resemble those of Benaglia et al. (2009b) in practice.

The association of EM algorithms with mixture models has a long history; indeed, the original "EM algorithm" article—i.e., the article in which the initials "EM" were coined, not the first appearance of such an algorithm—describes a finite mixture model as one of several EM examples (Dempster et al., 1977). Not long after, a sizable literature on "nonparametric mixtures" appeared (see, e.g., Lindsay, 1995), but here, "nonparametric mixtures" was used in a different sense: The mixing distribution, rather than the component densities, were assumed to be unspecified (by contrast, the current article and most of the literature we have cited so far assume the mixing distribution to have finite support with a fixed cardinality, but the component densities are unspecified). For this distinct concept of nonparametric mixture models, Vardi et al. (1985) introduced an EM algorithm for maximum likelihood estimation of the mixing distribution.

The Vardi et al. (1985) algorithm has an elegant convergence theory associated with it, but unfortunately it does not deal with ill-posedness of the problem. To overcome this difficulty, Silverman et al. (1990) proposed the EMS (Smoothed EM) algorithm that smoothes the result of each step of the classical EM algorithm. The practical performance of this algorithm is excellent but, unfortunately, understanding its quantitative properties has turned out to be a difficult issue. Eggermont (1992) first proposed the idea of using a regularization approach to modify the EMS algorithm in such a way that the resulting algorithm is easier to investigate theoretically. Eggermont and LaRiccia (1995) showed that the resulting NEMS (Nonlinear EMS) algorithm is, indeed, an EM algorithm itself and that its convergence theory is very similar to that of the original EM algorithm introduced in the mixing density estimation context by Shepp and Vardi (1982). Eggermont (1999) showed that it also possesses a descent property and that the corresponding maximum likelihood problem has a unique solution. Finally, it is known that the practical performance of the NEMS algorithm is at least as good as that of the EMS algorithm; again, see Eggermont (1999) for details.

In a sense, the current article unites the two different historical meanings of "nonparametric mixture model" by introducing an NEMS-inspired algorithm that does in fact possess a descent property and that converges to a local maximizer of a likelihood-like quantity for the finite mixture model (2) in which the component densities are completely unspecified. As far as we know, our article is a novel adaptation of the regularization approach to the context of nonparametric finite mixture models. The likelihood-like function

used is similar to the one introduced in Eggermont and LaRiccia (1995) in the case of an unspecified continuous *mixing* distribution: It is, essentially, a penalized Kullback-Leibler distance between the target (mixture) density function and the iteratively reweighted sum of smoothed component density function estimates. However, for its optimization, we will rely on a computational tool called an MM algorithm, which may be viewed as yet another generalization of an EM algorithm.

2 Smoothing the log-density

Let us assume that Ω is a compact subset of R^r and define the linear vector function space

$$\mathcal{F} = \{\mathbf{f} = (f_1, \ldots, f_m)^\top : 0 < f_j \in L_1(\Omega), \log f_j \in L_1(\Omega), j = 1, \ldots, m\}.$$

The assumption of compact support may appear somewhat limiting from a theoretical point of view, but it is not problematic from a practical point of view; it plays no role in the implementation of the algorithm we propose here, for instance.

Take $K(\cdot)$ to denote some kernel density function on the real line. With a slight abuse of notation, let us define the product kernel function $K(\mathbf{u}) = \prod_{k=1}^r K(u_k)$ and its rescaled version $K_h(\mathbf{u}) = h^{-r} \prod_{k=1}^r K(h^{-1}u_k)$. Furthermore, we define a smoothing operator $\mathcal{S}$ for any function $f \in L_1(\Omega)$ by

$$\mathcal{S}f(\mathbf{x}) \;=\; \int_\Omega K_h(\mathbf{x} - \mathbf{u}) f(\mathbf{u})\, d\mathbf{u}.$$

Furthermore, we extend $\mathcal{S}$ to $\mathcal{F}$ by defining $\mathcal{S}\mathbf{f} = (\mathcal{S}f_1, \ldots, \mathcal{S}f_m)^\top$. We also define a nonlinear smoothing operator $\mathcal{N}$ as

$$\mathcal{N}f(\mathbf{x}) \;=\; \exp\{(\mathcal{S}\log f)(\mathbf{x})\} \;=\; \exp \int_\Omega K_h(\mathbf{x} - \mathbf{u}) \log f(\mathbf{u})\, d\mathbf{u}.$$

This operator is strictly concave, and it is also multiplicative in the sense that $\mathcal{N}f_j = \prod_k \mathcal{N}f_{jk}$ for f_j defined as in (1). The concavity is proven as Lemma 3.1(iii) of Eggermont (1999); we do not repeat the proof here. The idea of smoothing the logarithm of the density function goes back to Silverman (1982), where a penalty based on the second derivative of the density logarithm is discussed.

To simplify notation, we introduce the finite mixture operator

$$\mathcal{M}_{\boldsymbol{\lambda}}\mathbf{f}(\mathbf{x}) \stackrel{\text{def}}{=} \sum_{j=1}^m \lambda_j f_j(\mathbf{x}),$$

whence we also obtain $\mathcal{M}_{\boldsymbol{\lambda}}\mathbf{f}(\mathbf{x}) = g_{\boldsymbol{\theta}}(\mathbf{x})$ and

$$\mathcal{M}_{\boldsymbol{\lambda}}\mathcal{N}\mathbf{f}(\mathbf{x}) \stackrel{\text{def}}{=} \sum_{j=1}^{m} \lambda_j \mathcal{N} f_j(\mathbf{x}).$$

Let $g(\mathbf{x})$ now represent a known target density function. We begin by defining the following functional of $\boldsymbol{\theta}$ (and, implicitly, g):

$$\ell(\boldsymbol{\theta}) = \int_{\Omega} g(\mathbf{x}) \log \frac{g(\mathbf{x})}{[\mathcal{M}_{\boldsymbol{\lambda}}\mathcal{N}\mathbf{f}](\mathbf{x})}\, d\mathbf{x}. \tag{4}$$

Note that we will suppress the subscripted Ω on the integral sign from now on. Our goal in Section 3 will be to find a minimizer of $\ell(\boldsymbol{\theta})$ subject to the assumptions that each f_{jk} is a univariate density function and $\boldsymbol{\lambda}$ satisfies (3).
Remark: An immediate consequence of Equation (4) is that $\ell(\boldsymbol{\theta})$ can be viewed as a penalized Kullback-Leibler distance between $g(\mathbf{x})$ and $(\mathcal{M}_{\boldsymbol{\lambda}}N\mathbf{f})(\mathbf{x})$. Indeed, if we define

$$D(a \,|\, b) \;=\; \int \left[a(\mathbf{x}) \log \frac{a(\mathbf{x})}{b(\mathbf{x})} + b(\mathbf{x}) - a(\mathbf{x}) \right] d\mathbf{x} \tag{5}$$

as usual, it follows that

$$\ell(\boldsymbol{\theta}) \;=\; D(g \,|\, \mathcal{M}_{\boldsymbol{\lambda}}\mathcal{N}\mathbf{f}) + \int g(\mathbf{x})\, d\mathbf{x} - \sum_{j=1}^{m} \lambda_j \int \mathcal{N} f_j(\mathbf{x})\, d\mathbf{x}, \tag{6}$$

where $-\lambda_j \int \mathcal{N} f_j(\mathbf{x})\, d\mathbf{x}$ is a penalization term (cf. Eggermont, 1999, equation (1.12) and the discussion immediately following).

3 An MM algorithm

Our goal is to define an iterative algorithm that possesses a descent property with respect to the functional $\ell(\mathbf{f}, \boldsymbol{\lambda})$; that is, we wish to ensure that the value of $\ell(\mathbf{f}, \boldsymbol{\lambda})$ cannot increase from one iteration to the next. (Here, we write $\ell(\mathbf{f}, \boldsymbol{\lambda})$ instead of $\ell(\boldsymbol{\theta})$ so we may discuss $\mathbf{f}$ and $\boldsymbol{\lambda}$ separately.) Suppose that we were to define an iteration operator G, to be applied to the vector $\mathbf{f} = (f_1, \ldots, f_m)^\top$, as

$$G f_j(\mathbf{x}) \;=\; \alpha_j \int K_h(\mathbf{x} - \mathbf{u}) \frac{g(\mathbf{u}) \mathcal{N} f_j(\mathbf{u})}{\mathcal{M}_{\boldsymbol{\lambda}}\mathcal{N}\mathbf{f}(\mathbf{u})}\, d\mathbf{u}$$

for each j, where α_j is a proportionality constant that makes $G f_j(\cdot)$ integrate to one. Using an argument analogous to the proof of Lemma 1, we could

show that

$$\begin{aligned}\ell(\mathbf{f},\boldsymbol{\lambda}) - \ell(G\mathbf{f},\boldsymbol{\lambda}) &\geq \sum_{j=1}^{m} \frac{\lambda_j}{\alpha_j} \int Gf_j(\mathbf{x}) \log \frac{Gf_j(\mathbf{x})}{f_j(\mathbf{x})}\, d\mathbf{x} \\ &= \sum_{j=1}^{m} \frac{\lambda_j}{\alpha_j} D(Gf_j \,|\, f_j) \geq 0.\end{aligned}$$

Thus, the above definition evidently results in an algorithm that satisfies the descent property. Unfortunately, however, it does not preserve the essential conditional independence assumption (1). We must therefore use a slightly different approach.

Let $(\mathbf{f}^0, \boldsymbol{\lambda}^0)$ denote the current parameter values in an iterative algorithm. Our strategy for minimizing $\ell(\mathbf{f},\boldsymbol{\lambda})$ is based on a "majorization-minimization" (MM) algorithm, in which we define a functional $b^0(\mathbf{f},\boldsymbol{\lambda})$ that, when shifted by a constant, *majorizes* $\ell(\mathbf{f},\boldsymbol{\lambda})$—i.e.,

$$b^0(\mathbf{f},\boldsymbol{\lambda}) + C^0 \geq \ell(\mathbf{f},\boldsymbol{\lambda}), \text{ with equality when } (\mathbf{f},\boldsymbol{\lambda}) = (\mathbf{f}^0,\boldsymbol{\lambda}^0). \tag{7}$$

Note that the superscript on b^0 indicates that the definition of $b^0(\mathbf{f},\boldsymbol{\lambda})$ will in general depend on the parameter values $(\mathbf{f}^0,\boldsymbol{\lambda}^0)$, which are considered fixed constants in this context. A general introduction to MM algorithms is given by Hunter and Lange (2004); in brief, by defining a majorizing function, we may minimize the majorizer instead of the original function.

For $j = 1, \ldots, m$, let

$$w_j^0(\mathbf{x}) \stackrel{\text{def}}{=} \frac{\lambda_j^0 \mathcal{N} f_j^0(\mathbf{x})}{\mathcal{M}_{\boldsymbol{\lambda}^0} \mathcal{N} \mathbf{f}^0(\mathbf{x})}. \tag{8}$$

Note in particular that the "weight" functions w_j^0 satisfy $\sum_j w_j^0(\mathbf{x}) = 1$. We now claim that

$$b^0(\mathbf{f},\boldsymbol{\lambda}) \stackrel{\text{def}}{=} -\int g(\mathbf{x}) \sum_{j=1}^{m} w_j^0(\mathbf{x}) \log\left[\lambda_j \mathcal{N} f_j(\mathbf{x})\right]\, d\mathbf{x} \tag{9}$$

gives the majorizing functional we seek:

Lemma 1. $\ell(\mathbf{f},\boldsymbol{\lambda}) - \ell(\mathbf{f}^0,\boldsymbol{\lambda}^0) \leq b^0(\mathbf{f},\boldsymbol{\lambda}) - b^0(\mathbf{f}^0,\boldsymbol{\lambda}^0)$.

Proof:

$$\begin{aligned}
\ell(\mathbf{f},\boldsymbol{\lambda}) - \ell(\mathbf{f}^0,\boldsymbol{\lambda}^0) &= -\int g(\mathbf{x}) \log \frac{\mathcal{M}_{\boldsymbol{\lambda}}\mathcal{N}\mathbf{f}(\mathbf{x})}{\mathcal{M}_{\boldsymbol{\lambda}^0}\mathcal{N}\mathbf{f}^0(\mathbf{x})}\, d\mathbf{x} \\
&= -\int g(\mathbf{x}) \log \frac{\sum_{j=1}^m \lambda_j \mathcal{N} f_j(\mathbf{x})}{\mathcal{M}_{\boldsymbol{\lambda}^0}\mathcal{N}\mathbf{f}^0(\mathbf{x})}\, d\mathbf{x} \\
&= -\int g(\mathbf{x}) \log \sum_{j=1}^m \frac{\lambda_j^0 \mathcal{N} f_j^0(\mathbf{x})}{\mathcal{M}_{\boldsymbol{\lambda}^0}\mathcal{N}\mathbf{f}^0(\mathbf{x})} \frac{\lambda_j \mathcal{N} f_j(\mathbf{x})}{\lambda_j^0 \mathcal{N} f_j^0(\mathbf{x})}\, d\mathbf{x} \\
&= -\int g(\mathbf{x}) \log \sum_{j=1}^m w_j^0(\mathbf{x}) \frac{\lambda_j \mathcal{N} f_j(\mathbf{x})}{\lambda_j^0 \mathcal{N} f_j^0(\mathbf{x})}\, d\mathbf{x} \\
&\leq -\int g(\mathbf{x}) \sum_{j=1}^m w_j^0(\mathbf{x}) \log \frac{\lambda_j \mathcal{N} f_j(\mathbf{x})}{\lambda_j^0 \mathcal{N} f_j^0(\mathbf{x})}\, d\mathbf{x} \\
&= b^0(\mathbf{f},\boldsymbol{\lambda}) - b^0(\mathbf{f}^0,\boldsymbol{\lambda}^0),
\end{aligned}$$

where the inequality follows directly from the convexity of the negative logarithm function, since $\sum_j w_j^0(\mathbf{x}) = 1$. □

Lemma 1 verifies the majorization claim (7), where we take the constant C^0 to be $\ell(\mathbf{f}^0,\boldsymbol{\lambda}^0) - b^0(\mathbf{f}^0,\boldsymbol{\lambda}^0)$.

Rewriting (9), we obtain

$$\begin{aligned}
b^0(\mathbf{f},\boldsymbol{\lambda}) &= -\sum_{j=1}^m \sum_{k=1}^r \iint K_h(x_k - u) g(\mathbf{x}) w_j^0(\mathbf{x}) \log f_{jk}(u)\, du\, d\mathbf{x} \\
&\quad -\sum_{j=1}^m \log \lambda_j \int g(\mathbf{x}) w_j^0(\mathbf{x})\, d\mathbf{x}. \qquad (10)
\end{aligned}$$

Note that above (and henceforth), u denotes a scalar, whereas $\mathbf{x}$ is an r-dimensional vector. Also note that $b^0(\mathbf{f},\boldsymbol{\lambda})$ separates the parameters from each other, in the sense that it is the sum of separate functions of the individual f_{jk} and λ_j.

Subject to the constraint $\sum_j \lambda_j = 1$, it is not hard to minimize $b^0(\mathbf{f},\boldsymbol{\lambda})$ with respect to the $\boldsymbol{\lambda}$ parameter: For each j, the minimizer is

$$\hat{\lambda}_j = \frac{\int g(\mathbf{x}) w_j^0(\mathbf{x})\, d\mathbf{x}}{\sum_{j=1}^m \int g(\mathbf{x}) w_j^0(\mathbf{x})\, d\mathbf{x}} = \int g(\mathbf{x}) w_j^0(\mathbf{x})\, d\mathbf{x}. \qquad (11)$$

Next, let us focus on only the part of (10) involving f_{jk} by defining

$$b_{jk}^0(f_{jk}) \stackrel{\text{def}}{=} -\iint K_h(x_k - u) g(\mathbf{x}) w_j^0(\mathbf{x}) \log f_{jk}(t)\, du\, d\mathbf{x}. \qquad (12)$$

Lemma 2. *For $j = 1, \ldots, m$ and $k = 1, \ldots, r$, define*

$$\hat{f}_{jk}(u) \;=\; \alpha_{jk} \int K_h(x_k - u) g(\mathbf{x}) w_j^0(\mathbf{x})\, d\mathbf{x}, \tag{13}$$

where α_{jk} is a constant chosen so that $\int \hat{f}_{jk}(u)\, dt = 1$. Then $\hat{f}_{jk}$ is the unique (up to changes on a set of Lebesgue measure zero) density function minimizing $b_{jk}^0(\cdot)$.

Proof: Fubini's Theorem yields

$$b_{jk}^0(f_{jk}) \;=\; \frac{1}{\alpha_{jk}} D(\hat{f}_{jk} \,|\, f_{jk}) - \frac{1}{\alpha_{jk}} \int \hat{f}_{jk}(u) \log \hat{f}_{jk}(u)\, du,$$

where the second term on the right hand side does not depend on f_{jk}. The result follows immediately. □

Let us now combine the preceding results. From Lemma 1, we conclude that

$$\ell(\hat{\mathbf{f}}, \hat{\boldsymbol{\lambda}}) - \ell(\mathbf{f}^0, \boldsymbol{\lambda}^0) \leq b^0(\hat{\mathbf{f}}, \hat{\boldsymbol{\lambda}}) - b^0(\mathbf{f}^0, \boldsymbol{\lambda}^0). \tag{14}$$

Furthermore, we know from Lemma 2 and Equation (11) that each individual piece of the $b^0(\cdot)$ function of Equation (10) is minimized by the corresponding piece of $(\hat{\mathbf{f}}, \hat{\boldsymbol{\lambda}})$. We conclude that the right side of Inequality (14) is bounded above by zero, which proves the descent property summarized by the following theorem.

Theorem 1. *Define $\hat{\boldsymbol{\lambda}}$ as in Equation (11) and $\hat{\mathbf{f}}$ as in Lemma 2. Then*

$$\ell(\hat{\mathbf{f}}, \hat{\boldsymbol{\lambda}}) \leq \ell(\mathbf{f}^0, \boldsymbol{\lambda}^0).$$

4 Inference for the parameters

We now assume that we are given a simple random sample $\mathbf{x}_1, \ldots, \mathbf{x}_n$ distributed according to the $g_{\boldsymbol{\theta}}(\mathbf{x})$ density defined in Equation (2). Letting $\tilde{G}_n(\cdot)$ denote the empirical distribution function of the sample and ignoring the term $\int g_{\boldsymbol{\theta}}(\mathbf{x}) \log g_{\boldsymbol{\theta}}(\mathbf{x})\, d\mathbf{x}$ that does not involve any parameters, a discrete version of (4) is

$$\ell_n(\mathbf{f}, \boldsymbol{\lambda}) \;\stackrel{\text{def}}{=}\; \int \log \frac{1}{[\mathcal{M}_{\boldsymbol{\lambda}} N \mathbf{f}](\mathbf{x})}\, d\tilde{G}_n(\mathbf{x}) \;=\; -\sum_{i=1}^{n} \log[\mathcal{M}_{\boldsymbol{\lambda}} N \mathbf{f}](\mathbf{x}_i).$$

Note that $\ell_n(\mathbf{f}, \boldsymbol{\lambda})$ resembles a penalized loglikelihood function except for the presence of the nonlinear smoothing operator $\mathcal{N}$ and the fact that with the negative sign preceding the sum, our goal is minimization rather than maximization of $\ell_n(\cdot)$.

Using an argument nearly identical to the one leading to equation (13), we may show that the following algorithm results in an EM algorithm in which the value of $\ell_n(\cdot)$ decreases at each iteration:

Given initial values $(\mathbf{f}^0, \boldsymbol{\lambda}^0)$, iterate the following three steps for $t = 0, 1, \ldots$:

- **E-step:** Define, for each i and j,

$$w_{ij}^t = \frac{\lambda_j^t \mathcal{N} f_j^t(\mathbf{x}_i)}{\mathcal{M}_{\boldsymbol{\lambda}^t} \mathcal{N} \mathbf{f}^t(\mathbf{x}_i)} = \frac{\lambda_j^t \mathcal{N} f_j^t(\mathbf{x}_i)}{\sum_{a=1}^m \lambda_a \mathcal{N} f_a^t(\mathbf{x}_i)}. \tag{15}$$

- **M-step, part 1:** Set

$$\lambda_j^{t+1} = \frac{1}{n} \sum_{i=1}^n w_{ij}^t \tag{16}$$

  for $j = 1, \ldots, m$.

- **M-step, part 2:** For each j and k, let

$$f_{jk}^{t+1}(u) = \frac{\sum_{i=1}^n w_{ij}^t K_h(u - x_{ik})}{\sum_{i=1}^n w_{ij}^t} = \frac{1}{nh\lambda_j^{t+1}} \sum_{i=1}^n w_{ij}^t K\left(\frac{u - x_{ik}}{h}\right). \tag{17}$$

Note that equations (15), (16), and (17) are merely the discrete versions of equations (8), (11), and (13), respectively. With regard to the convergence properties of the algorithm we have defined here, we prove in the Appendix that, if we hold $\boldsymbol{\lambda}$ fixed and repeatedly iterate equation (13), then the sequence of $\mathbf{f}$ functions converges to a global minimizer of $\ell(\mathbf{f}, \boldsymbol{\lambda})$ *for that value of* $\boldsymbol{\lambda}$.

5 Implementation and numerical examples

Here, we test our algorithm on several examples from the recent literature on nonparametric finite mixtures. In order to do this, it is first necessary to introduce a slight extension of the basic model (2) in order to allow for "block structure" in which some blocks of the coordinates might be identically distributed in addition to being independent.

5.1 Blocks of identically distributed coordinates

As in Benaglia et al. (2009b), we can extend the model of conditional independence to a more general model: We will allow that the coordinates of $\mathbf{X}_i$ are conditionally independent and that there exist *blocks* of coordinates

that are also identically distributed. If we let b_k denote the block index of the kth coordinate, where $1 \leq b_k \leq B$ and B is the total number of such blocks, then equation (2) is replaced by

$$g_{\boldsymbol{\theta}}(\mathbf{x}_i) = \sum_{j=1}^{m} \lambda_j \prod_{k=1}^{r} f_{jb_k}(x_{ik}). \tag{18}$$

These blocks may all be of size one ($b_k = k$, $k = 1, \ldots r$), which coincides with equation (2), or there may exist only a single block ($b_k = B = 1$, $k = 1, \ldots, r$), which is the conditional i.i.d. case. The non-linear smoothing operator $\mathcal{N}$ applied to f_j is simply $\mathcal{N} f_j = \prod_{k=1}^{r} \mathcal{N} f_{jb_k}$, and definitions of $\mathcal{M}_{\boldsymbol{\lambda}}\mathbf{f}$ and $\mathcal{M}_{\boldsymbol{\lambda}}\mathcal{N}\mathbf{f}$ are unchanged.

The algorithm of section 4 can easily be adapted for handling the block structure. In fact, both the E-step (15) and the first part of the M-step (16) remain unchanged. The second part of the M-step (17) becomes

- **M-step, part 2:** For each component j and block $\ell \in \{1, \ldots, B\}$, let

$$\begin{aligned} f_{j\ell}^{t+1}(u) &= \frac{\sum_{k=1}^{r} \sum_{i=1}^{n} w_{ij}^{t} I_{\{b_k=\ell\}} K_h\left(u - x_{ik}\right)}{\sum_{k=1}^{r} \sum_{i=1}^{n} w_{ij}^{t} I_{\{b_k=\ell\}}} \\ &= \frac{1}{nh\lambda_j^{t+1} C_\ell} \sum_{k=1}^{r} \sum_{i=1}^{n} w_{ij}^{t} I_{\{b_k=\ell\}} K\left(\frac{u - x_{ik}}{h}\right), \end{aligned} \tag{19}$$

where $C_\ell = \sum_{k=1}^{r} I_{\{b_k=\ell\}}$ is the number of coordinates in the ℓth block.

This algorithm is implemented in the latest version of the publicly-available R (R Development Core Team, 2008) package called mixtools (Young et al., 2009; Benaglia et al., 2009c).

5.2 Simulated Examples

This simulation study compares the nonparametric "EM-like" algorithm from Benaglia et al. (2009b), which we refer to as npEM here, with the new algorithm using the same examples for which Hall et al. (2005) tested their estimation technique based on inverting the mixture model. The three simulated models, described below, are trivariate two-component mixtures ($m = 2, r = 3$) with independent but not identically distributed repeated measures, i.e., $b_k = k$ for $k = 1, 2, 3$. We ran $S = 300$ replications of $n = 500$ observations each and computed the errors in terms of the square root of the Mean Integrated Squared Error (MISE) for the densities, where

$$\text{MISE}_{jk} = \frac{1}{S} \sum_{s=1}^{S} \int \left(\hat{f}_{jk}^{(s)}(u) - f_{jk}(u)\right)^2 du, \quad j = 1, 2 \text{ and } k = 1, 2, 3;$$

and the integral is computed numerically (using an appropriate function defined in mixtools). Each density $\hat{f}_{jk}^{(s)}$ is computed using the weighted kernel density estimate (17) together with the final values of the posterior probabilities p_{ij}^t after convergence of the algorithm.

The first example is a normal model, for which the individual densities $f_{j\ell}$ are the pdf's of $\mathcal{N}(\mu_{j\ell}, 1)$, with component means $\boldsymbol{\mu}_1 = (0, 0, 0)$ and $\boldsymbol{\mu}_2 = (3, 4, 5)$. The second example uses double exponential distributions with densities $f_{j\ell}(t) = \exp\{-|t - \mu_{j\ell}|\}/2$, where $\boldsymbol{\mu}_1 = (0, 0, 0)$ and $\boldsymbol{\mu}_2 = (3, 3, 3)$. In the third example, the first component has a central $t(10)$ distribution and thus $\boldsymbol{\mu}_1 = (0, 0, 0)$, whereas the second component's coordinates are noncentral $t(10)$ distributions with noncentrality parameters 3, 4, and 5. Thus, the mean of the third component is $\boldsymbol{\mu}_2 = (3, 4, 5) \times 1.0837$. Note that both algorithms assume *only* the general model of conditional independence, with $b_k = k$ for all k.

Since it has already been shown in Benaglia et al. (2009b) that the npEM dramatically outperforms the inversion method of Hall et al. (2005) for the three test cases, Figure 1 only compares the npEM against the new algorithm, which is labeled "smoothed" in the figure. This figure shows that the two algorithms provide nearly identical efficiency (in terms of MISE) and that there is no clear winner for the models and the various settings (λ_1) considered.

5.3 The water-level experiment

We consider in this section a dataset from an experiment involving $n = 405$ children aged 11 to 16 years subjected to a water-level task as initially described by Thomas et al. (1993). In this experiment, each child is presented with eight rectangular vessels on a sheet of paper, each tilted to one of $r = 8$ clock-hour orientations: in order of presentation to the subjects, these orientations are 11, 4, 2, 7, 10, 5, 1, and 8 o'clock. The children's task was to draw a line representing the surface of still liquid in the closed, tilted vessel in each picture. Each such line describes two points of intersection with the sides of the vessel; the acute angle, in degrees, formed between the horizontal and the line passing through these two points was measured for each response. The sign of each such measurement was taken to be the sign of the slope of the line. The water-level dataset is available in the mixtools package (Young et al., 2009; Benaglia et al., 2009c). This dataset has been analyzed previously by Hettmansperger and Thomas (2000) and Elmore et al. (2004), who assume that the $r = 8$ coordinates are all conditionally identically distributed and then bin the data to produce multinomial vectors (these authors call this a "cutpoint" approach).

However, because of the experimental methodology used to collect the

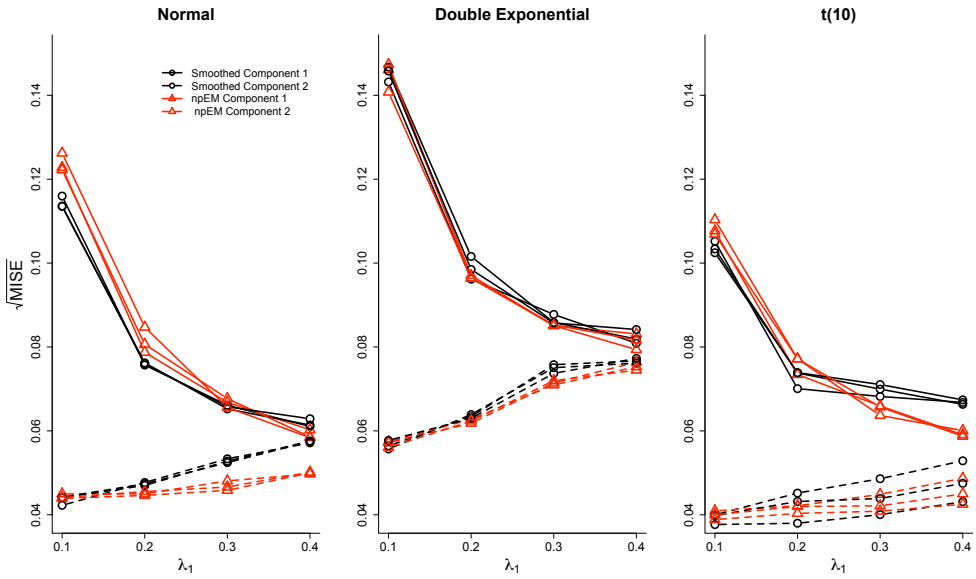

Figure 1: *Square roots of Mean Integrated Squared Errors (MISE) as a function of* λ_1*, the proportion of component 1, for all* f_{jk}*,* $j = 1, 2$ *and* $k = 1, 2, 3$*, for the three benchmark models from Hall et al. (2005). Some points are coincident with others and are therefore not visible.*

data, it seems reasonable to weaken the assumption that each orientation's measurements are identically distributed; instead, we only assume that opposite clock-face orientations lead to conditionally independent and identically distributed responses, so that the eight coordinates may be organized into four blocks of two each, where the densities within each block are identical, which is model (18).

Benaglia et al. (2009b) apply the npEM algorithm to model (18) with $B = 4$ and blocks of coordinates defined by

$$\mathbf{b} = (b_1, \ldots, b_8) = (4, 3, 2, 1, 3, 4, 1, 2),$$

which means, e.g., that $b_4 = b_7 = 1$, i.e., block 1 relates to coordinates 4 and 7, corresponding to clock orientations 1:00 and 7:00.

Figure 2 compares, for $m = 3$ components, the npEM solution with the solution given by the new algorithm (Section 5.1), which we refer to as the "smoothed" algorithm. The two solutions are evidently quite similar; each appears to detect one component of children who understand the task (the component peaked around the correct answer of zero degrees), another group who appear to complete the task correctly when the vessel is near vertical but who do not do as well in the "sideways" orientations of blocks 2 and 3, and a third group who appear to draw the line perpendicular to the sides

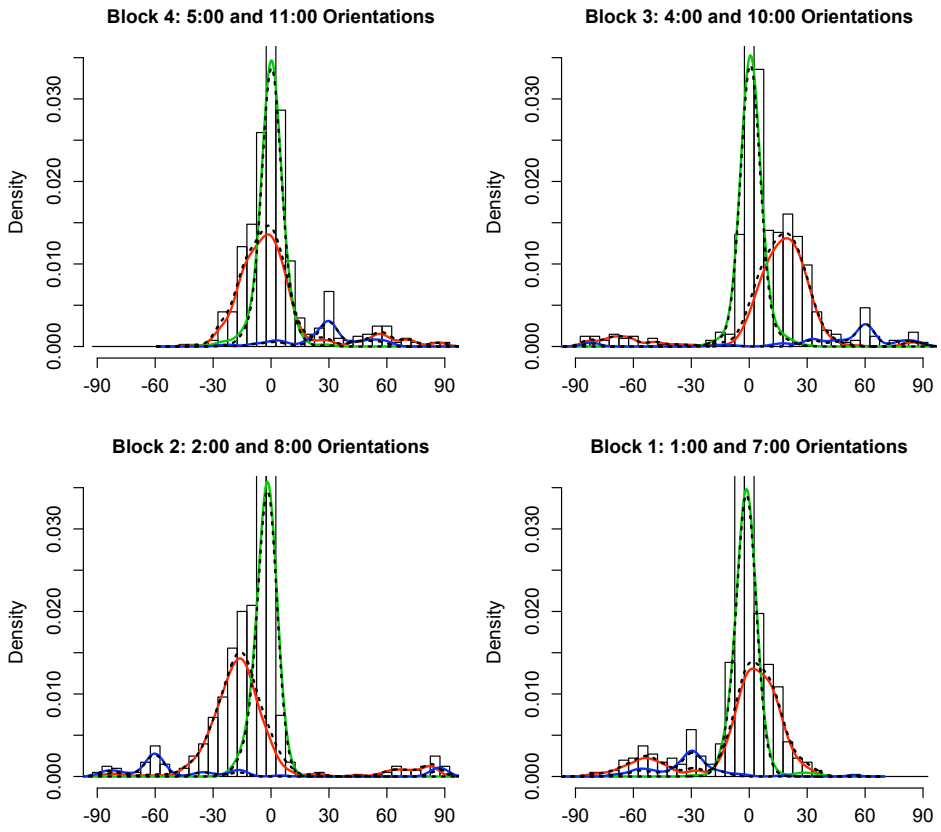

Figure 2: *The water-level data are analyzed using the npEM algorithm (solid colored lines) from Benaglia et al. (2009b) and the new smoothed algorithm (dotted line), assuming model (18) with* $m = 3$ *mixture components.*

of the vessel. The estimated proportions of these three components for the smoothed algorithm (and the corresponding npEM estimates in parentheses) are, respectively, 0.471 (0.492), 0.465 (0.431), and 0.064 (0.077). This observed slight difference between the two algorithms' estimates of these proportions suggests that it might be wise to compute confidence intervals for these parameters.

The confidence intervals for the λ_j, seen in Table 1, were computed using a nonparametric bootstrap approach by repeatedly resampling with replacement from the empirical distribution defined by the n observed r-dimensional vectors and carefully checking for label-switching occurrences in the resulting estimates. Here, "label-switching" refers to permuting the three labels on $\hat{\lambda}_1$, $\hat{\lambda}_2$, and $\hat{\lambda}_3$, which in this example is easy to detect by examining standard deviations of the estimated densities in combination

	λ_1		λ_2		λ_3	
npEM	0.446	0.552	0.337	0.469	0.0542	0.1506
Smoothed	0.420	0.527	0.361	0.496	0.0515	0.1594

Table 1: 95% Confidence Intervals for $\boldsymbol{\lambda}$, based on $10,000$ bootstrap replications, for the Water-level data example.

with the $\hat{\boldsymbol{\lambda}}$ estimates obtained. Boxplots of all the $\hat{\boldsymbol{\lambda}}$ estimates are given in Figure 3. The two algorithms took a comparable number of iterations to converge—69 on average for the smoothed algorithm versus 71 for the npEM, using the same convergence criterion—although each iteration of the smoothed algorithm took slightly longer (roughly 1/3 longer) due to the numerical convolution involved. Overall, despite the small differences in the estimates obtained by the two algorithms, we do not notice a systematic pattern in these differences and the results are really quite close.

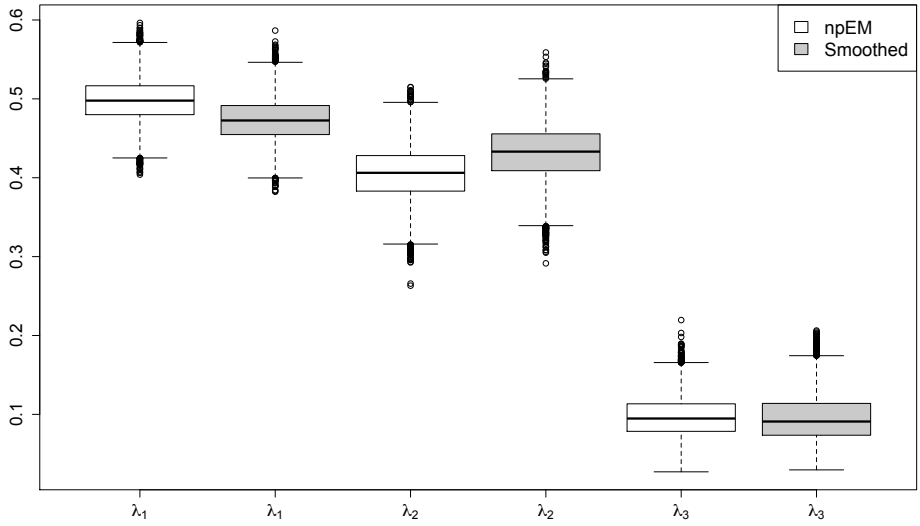

Figure 3: *Comparison of 10,000 nonparametric bootstrap replications of* $\hat{\boldsymbol{\lambda}}$ *for the Water-level data, using both algorithms. Bootstrapped 95% confidence intervals are given in Table 1.*

6 Discussion

The algorithm we propose in this article is an important refinement of the algorithm first proposed in Benaglia et al. (2009b). That earlier algorithm

is, to the best of our knowledge, the first algorithm that can deal easily with Model (2) in its full generality. It also renders itself to fairly easy coding and produces error rates that are considerably lower than those of the inversion method (easily applicable only when $m = 2$) of Hall et al. (2005) for a set of standard test cases. However, it does not appear to minimize any particular objective function and therefore cannot be viewed as a true EM algorithm. The new algorithm introduced in this article has a provable descent property with respect to a loglikelihood-like quantity. That quantity is, essentially, a penalized Kullback-Leibler distance between the true target density and the iteratively reweighted sum of smoothed estimated component densities.

We obtain our new algorithm by combining the so-called regularization approach with the earlier algorithm of Benaglia et al. (2009b). This approach was used by Eggermont and LaRiccia (1995) and Eggermont (1999) in the context of indirect measurements, where it was applied to an earlier EMS (Smoothed EM) algorithm of Silverman et al. (1990) that does not have an easy interpretation. The result of the regularization approach there is a new algorithm, closely related to EMS, called NEMS (Nonlinear EMS). That algorithm has empirical performance almost identical to that of EMS; however, it is a true EM algorithm. In the mixture-model setting, additional computational tools based on a majorization-minimization (MM) theory are necessary in order to produce an algorithm with a descent property.

The MM device used in our algorithm, namely, the convexity of the negative logarithm in the proof of Lemma 1, is exactly the same as used by a classical EM algorithm. However, the question of whether our algorithm represents a true EM algorithm—i.e., whether the right side of Equation (9) is actually the expectation of a loglikelihood function for some idea of "complete data"—is largely academic; one feature of this article is that it demonstrates in a practical case how a direct MM approach can produce the same theoretical advantages as an EM approach.

A future practitioner will have a choice of algorithms when estimating the nonparametric multivariate finite mixture model of Equation (2). The algorithm we propose in the current paper is the wise choice if a descent property and the convergence properties associated with it (e.g., see Lange et al., 2000) are needed. On the other hand, our experimental results validate the use of the earlier algorithm of Benaglia et al. (2009b) if only good empirical performance is desired, since the earlier algorithm appears to produce very similar results to the new, theoretically sound one. Note that our new algorithm is generally the slower of the two since it involves numerical convolutions.

The basic algorithm presented in this article may be generalized in several directions. In addition to the blocking structure introduced in Equation (18), one might posit various location and/or scale models that link

the component densities while still allowing the overall parametric form of the density functions to be unspecified. A thorough discussion of such generalizations is given in Section 4 of Benaglia et al. (2009b), where even a univariate application of these algorithms to the case in which the density functions are assumed symmetric is given. Another possible topic of future research is selection of an appropriate bandwidth h. Complicating this selection is the fact that, when estimating a mixture model, one does not observe individual component densities. At every step of the algorithm, new estimates of these components are computed; thus, bandwidth selection problem in this context may be a problem fundamentally different from the regular bandwidth selection for density estimation purposes. This issue is discussed at length by Benaglia et al. (2009a), who recommend an iterative bandwidth selection algorithm. It does not appear that generalization of our algorithm in any of these many directions would present any serious difficulties. Finally, there is the question of asymptotic convergence rates. Empirical studies in Benaglia et al. (2009b) are suggestive of rates of convergence of the original npEM algorithm, though no theoretical result on this subject is yet known. Now that we have demonstrated that our new algorithm may be used to optimize a particular objective function, it will perhaps be possible to establish such results in the future.

Appendix 1: Some convergence properties

Suppose we fix $\boldsymbol{\lambda}^0$ and consider the function defined by Equation (13) that maps $(\mathbf{f}^0, \boldsymbol{\lambda}^0) \mapsto (\hat{\mathbf{f}}, \boldsymbol{\lambda}^0)$. Iteratively applying this function yields a sequence

$$(\mathbf{f}^0, \boldsymbol{\lambda}^0), (\mathbf{f}^1, \boldsymbol{\lambda}^0), (\mathbf{f}^2, \boldsymbol{\lambda}^0), \ldots . \tag{20}$$

Here, we present a few simple convergence results regarding this sequence. The results of this section have analogues in Section 3 of Eggermont (1999) for a slightly different case.

Throughout this section, we will assume for the sake of simplicity that the kernel $K(\cdot)$ is strictly positive on the whole real line. We define the subset $B \subset \mathcal{F}$ by

$$B = \left\{ \mathcal{S}\boldsymbol{\phi} : 0 \leq \boldsymbol{\phi} \in \mathcal{F} \text{ and } \int_\Omega \phi_j(\mathbf{x})\, d\mathbf{x} = 1 \text{ for all } j \right\}. \tag{21}$$

The idea of defining B in this way is that B will contain the whole sequence $\mathbf{f}^0, \mathbf{f}^1, \mathbf{f}^2, \ldots$ except possibly the initial $\mathbf{f}^0$, where each element in the sequence is defined by applying equation (13) to the preceding element. To verify this claim, we simply note that equation (13) may be rewritten as

$$\hat{f}_{jk}(u) = \mathcal{S}\phi^0_{jk}(u), \tag{22}$$

where

$$\phi_{jk}^0(x_k) \stackrel{\text{def}}{=} \alpha_{jk} \int \cdots \int g(\mathbf{x}) w_j^0(\mathbf{x})\, dx_1 \cdots dx_{k-1} dx_{k+1} \cdots dx_r \tag{23}$$

must integrate to one because of the definition of α_{jk}. Furthermore, the functional $\mathbf{f} \mapsto \ell(\mathbf{f}, \boldsymbol{\lambda})$ is defined on B because for any $(f_1, \ldots, f_m) \in B$, f_j is bounded below by $\inf_{\mathbf{x}\in\Omega} K_h(\mathbf{x}) > 0$ since we have assumed that the original kernel $K(\cdot)$ is positive; thus, $\mathcal{N}\mathbf{f}$ is well-defined for $\mathbf{f} \in B$.

Lemma 3. *The set B is convex and the functional $\ell(\mathbf{f}, \boldsymbol{\lambda})$ of Equation (4) is strictly convex on B for fixed $\boldsymbol{\lambda}$.*

Proof: Let $\mathbf{f}^1$ and $\mathbf{f}^2$ be arbitrary elements of B, and take some $\alpha \in (0, 1)$. The linearity of the $\mathcal{S}$ operator implies that $\alpha\mathbf{f}^1 + (1-\alpha)\mathbf{f}^2 \in B$, which establishes the convexity of B immediately.

Furthermore,

$$\begin{aligned} \ell\left[\alpha\mathbf{f}^1 + (1-\alpha)\mathbf{f}^2\right] &= \int g(\mathbf{x}) \log g(\mathbf{x})\, d\mathbf{x} \\ &\quad - \int g(\mathbf{x}) \log \sum_{j=1}^m \lambda_j \mathcal{N}\left[\alpha f_j^1 + (1-\alpha) f_j^2\right](\mathbf{x})\, d\mathbf{x}. \end{aligned}$$

Focusing on the rightmost term above, we first claim that

$$\mathcal{N}\left[\alpha f_j^1 + (1-\alpha) f_j^2\right](\mathbf{x}) > \alpha \mathcal{N} f^1(\mathbf{x}) + (1-\alpha)\mathcal{N} f^2(\mathbf{x})$$

by the strict concavity of the $\mathcal{N}$ operator [Lemma 3.1(iii) of Eggermont (1999)]. Furthermore, the fact that the logarithm function is concave and strictly increasing implies that

$$\begin{aligned} \ell\left[\alpha\mathbf{f}^1 + (1-\alpha)\mathbf{f}^2, \boldsymbol{\lambda}\right] &< \int g(\mathbf{x}) \log g(\mathbf{x})\, d\mathbf{x} - \alpha \int g(\mathbf{x}) \log \sum_{j=1}^m \lambda_j \{\mathcal{N} f_j^1\}(\mathbf{x})\, d\mathbf{x} \\ &\quad -(1-\alpha) \int g(\mathbf{x}) \log \sum_{j=1}^m \lambda_j \{\mathcal{N} f_j^2\}(\mathbf{x})\, d\mathbf{x} \\ &= \alpha\ell(\mathbf{f}^1, \boldsymbol{\lambda}) + (1-\alpha)\ell(\mathbf{f}^2, \boldsymbol{\lambda}). \end{aligned}$$

□

Remark: We may also, using nearly the same proof as for Lemma 3, show that $\ell(\mathbf{f}, \boldsymbol{\lambda})$ is strictly convex in $\boldsymbol{\lambda}$ for fixed $\mathbf{f}$, though this fact appears less useful. It is not possible to prove that $\ell(\mathbf{f}, \boldsymbol{\lambda})$ is somehow strictly convex in the vector $(\mathbf{f}, \boldsymbol{\lambda})$ jointly—even if we were to define this concept rigorously—since we know that, as in all mixture model settings, permuting the subscripts $1, \ldots, m$ on $(f_1, \lambda_1), \ldots, (f_m, \lambda_m)$ does not change the value

of $\ell(\mathbf{f}, \boldsymbol{\lambda})$, which implies that there cannot in general exist a unique global minimizer of $\ell(\mathbf{f}, \boldsymbol{\lambda})$.

The following lemma establishes a sufficient condition so that the sequence of functions in (20) is guaranteed to have a uniformly convergent subsequence. It turns out that, along with the assumptions made earlier, the only additional assumption we will make is that the kernel density function satisfies a Lipschitz continuity condition:

Lemma 4. *If there exists $L > 0$ such that $|K_h(\mathbf{x}) - K_h(\mathbf{y})| \leq L|\mathbf{x} - \mathbf{y}|$ for any $\mathbf{x}, \mathbf{y} \in \Omega$, then every functional sequence $\mathbf{f}^1, \mathbf{f}^2, \ldots$ defined by (20) has a uniformly convergent subsequence.*

Proof: Since Ω is a compact subset of R^r, we may assume that there exist positive constants $a < A$ such that $a \leq K_h(\cdot) \leq A$ on Ω. Thus, in Equations (22) and (23), we must have $a \leq \hat{f}_{jk} \leq A$ for all j, k. We conclude that the sequence $|\mathbf{f}^1|, |\mathbf{f}^2|, \ldots$ is uniformly bounded.

Furthermore, for arbitrary $\mathbf{x}, \mathbf{y} \in \Omega$ and $\mathbf{f} \in B$,

$$\begin{aligned} |f_j(\mathbf{x}) - f_j(\mathbf{y})| &= |S\phi_j(\mathbf{x}) - S\phi_j(\mathbf{y})| \\ &\leq \int |K_h(\mathbf{x} - \mathbf{u}) - K_h(\mathbf{y} - \mathbf{u})||\phi_j(\mathbf{u})|\, d\mathbf{u} \\ &\leq L|\mathbf{x} - \mathbf{y}| \end{aligned}$$

for all j. We conclude that the sequence $\mathbf{f}^1, \mathbf{f}^2, \ldots$ is uniformly bounded and equicontinuous, so the Arzelà-Ascoli Theorem implies that there is a uniformly convergent subsequence. □

Lemma 5. *The functional $\mathbf{f} \mapsto \ell(\mathbf{f}, \boldsymbol{\lambda})$ is lower semicontinuous on B.*

Proof: Consider a sequence of functions $\{\mathbf{f}_n\} = \{(f_{1,1}, \ldots, f_{m,n})\}' \in B$. Let us denote $\boldsymbol{\psi} = \{\psi_1, \ldots, \psi_m\}' = \liminf_n \mathbf{f}_n$. By Lemma 4, there always exists a subsequence $\mathbf{f}_{n_k} \to \psi$; without loss of generality, assume that this subsequence coincides with the entire sequence $\{\mathbf{f}_n\}$. Since every component function $f_{j,n} \in B$ is bounded away from zero then so is the limit function ψ; therefore, $\log f_{j,n} \to \log \psi$. Consequently, $\mathcal{N} f_{j,n} \to \mathcal{N}\psi_j$ and $\mathcal{M}_\lambda \mathcal{N} \mathbf{f}_n \to \mathcal{M}_\lambda N\psi$. Since the function $\rho(t) = t - \log t - 1 \geq 0$, by Fatou's lemma we have

$$\int g(\mathbf{x})\rho(M_\lambda N\psi(\mathbf{x}))\, d\mathbf{x} \leq \liminf \int g(\mathbf{x})\rho(M_\lambda N\mathbf{f}_n(\mathbf{x}))\, d\mathbf{x}.$$

From the above, the statement of the proposition follows immediately. □

Notice that the functional $\mathbf{f} \mapsto \ell(\mathbf{f}, \boldsymbol{\lambda})$ is uniformly bounded from below on B, which follows from Equation (6) and the fact that $\mathcal{N} f_j(\mathbf{x}) \leq \mathcal{S} f_j(\mathbf{x}) = 1$ by the arithmetic-geometric mean inequality. Thus, the lower semincontinuity combined with strict convexity, as proved above, imply that for any

fixed $\boldsymbol{\lambda}$, the sequence (20) converges to a global maximizer of the functional $\mathbf{f} \mapsto \ell(\mathbf{f}, \boldsymbol{\lambda})$. As a practical matter, this means that we could essentially replace $\ell(\mathbf{f}, \boldsymbol{\lambda})$ by the profile loglikelihood

$$\ell^*(\boldsymbol{\lambda}) \stackrel{\text{def}}{=} \inf_{\mathbf{f} \in B} \ell(\mathbf{f}, \boldsymbol{\lambda})$$

because the minimization on the right-hand side may be accomplished by iterating (13) until convergence. However, dealing with the profile loglikelihood is not the general optimization strategy adopted in Section 4.

References

Allman, E. S., Matias, C., and Rhodes, J. A. (2009). Identifiability of parameters in latent structure models with many observed variables. *Ann. Statist*, 37(6A):3099–3132.

Benaglia, T., Chauveau, D., and Hunter, D. R. (2009a). Bandwidth selection in an EM-like algorithm for nonparametric multivariate mixtures. Technical Report hal-00353297, hal.

Benaglia, T., Chauveau, D., and Hunter, D. R. (2009b). An EM-like algorithm for semi-and non-parametric estimation in multivariate mixtures. *Journal of Computational and Graphical Statistics*, 18(2):505–526.

Benaglia, T., Chauveau, D., Hunter, D. R., and Young, D. (2009c). mixtools: An R package for analyzing finite mixture models. *Journal of Statistical Software*, 32(6):1–29.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38.

Eggermont, P. P. B. (1992). Nonlinear smoothing and the EM algorithm for positive integral equations of the first kind. Unpublished manuscript.

Eggermont, P. P. B. (1999). Nonlinear smoothing and the em algorithm for positive integral equations of the first kind. *Applied Mathematics and Optimization*, 39(1):75–91.

Eggermont, P. P. B. and LaRiccia, V. N. (1995). Maximum smoothed density estimation for inverse problems. *The Annals of Statistics*, 23(1):199–220.

Elmore, R. T., Hettmansperger, T. P., and Thomas, H. (2004). Estimating component cumulative distribution functions in finite mixture models. *Comm. Statist. Theory Methods*, 33(9):2075–2086.

Hall, P., Neeman, A., Pakyari, R., and Elmore, R. T. (2005). Nonparametric inference in multivariate mixtures. *Biometrika*, 92(3):667–678.

Hall, P. and Zhou, X. H. (2003). Nonparametric estimation of component distributions in a multivariate mixture. *Annals of Statistics*, 31:201–224.

Hettmansperger, T. P. and Thomas, H. (2000). Almost nonparametric inference for repeated measures in mixture models. *Journal of the Royal Statistical Society, Series B*, 62(4):811–825.

Hunter, D. R. and Lange, K. (2004). A tutorial on MM algorithms. *The American Statistician*, 58:30–37.

Kasahara, H. and Shimotsu, K. (2008). Nonparametric identification of finite mixture models of dynamic discrete choices. *Econometrica*, 77(1):135–176.

Kruskal, J. B. (1977). Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear algebra and its applications*, 18(2):95–138.

Lange, K., Hunter, D. R., and Yang, I. (2000). Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, 9(1):1–20.

Lindsay, B. G. (1995). *Mixture Models: Theory, Geometry, and Applications*. Institute of Mathematical Statistics.

R Development Core Team (2008). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Shepp, L. A. and Vardi, Y. (1982). Maximum likelihood reconstruction for emission tomography. *IEEE Trans. Med. Imaging*, 1(2):113–122.

Silverman, B. W. (1982). On the estimation of a probability density function by the maximum penalized likelihood method. *The Annals of Statistics*, pages 795–810.

Silverman, B. W., Jones, M. C., Wilson, J. D., and Nychka, D. W. (1990). A smoothed EM algorithm approach to indirect estimation problems, with particular reference to stereology and emission tomography. *Journal of the Royal Statistical Society, Series B*, 52:271–324.

Thomas, H., Lohaus, A., and Brainerd, C. J. (1993). Modeling growth and individual differences in spatial tasks. *Monographs of the Society for Research in Child Development*, 58(9).

Vardi, Y., Shepp, L. A., and Kaufman, L. (1985). A statistical model for positron emission tomography. *Journal of the American Statistical Association*, 80(389):8–20.

Young, D. S., Benaglia, T., Chauveau, D., Elmore, R. T., Hettmansperger, T. P., Hunter, D. R., Thomas, H., and Xuan, F. (2009). mixtools: Tools for mixture models. R package version 0.3.3.