Title

Semi-Parametric Mixture Models Through Log-Concave Density Estimation

Author

Zhou, Yangmei

Publication Date

2019

UNIVERSITY OF CALIFORNIA
RIVERSIDE

Semi-Parametric Mixture Models Through Log-Concave Density Estimation

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Applied Statistics

by

Yangmei Zhou

September 2019

Dissertation Committee:

Prof. Weixin Yao, Chairperson
Prof. Subir Ghosh
Prof. Esra Kurum

The Dissertation of Yangmei Zhou is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

I would like to take this opportunity to express my sincere gratitude toward my advisor, Prof. Weixin Yao, for his detailed guidance, encouragement and support through the journey of my Ph.D. research, which is full of obstacles and challenges and yet so exciting and rewarding.

I am so grateful to my dissertation committee member, Prof. Subir Ghosh and Prof. Esra Kurum, for their continuous support and encouragement.

I would also like to thank all the professors and staffs for their guidance and assistance, thank my friends and colleagues I have worked with in the statistics department.

To my beloved family.

ABSTRACT OF THE DISSERTATION

Semi-Parametric Mixture Models Through Log-Concave Density Estimation

by

Yangmei Zhou

Doctor of Philosophy, Graduate Program in Applied Statistics
University of California, Riverside, September 2019
Prof. Weixin Yao, Chairperson

This dissertation consists of two parts. The first part considers a semi-parametric two-component mixture model with one component completely known. Assuming the density of the unknown component to be log-concave, which contains a very broad family of densities, we develop a semi-parametric maximum likelihood estimator and propose an EM algorithm to compute it. Our new estimation method finds the mixing proportion and the distribution of the unknown component simultaneously. We establish the identifiability of the proposed semi-parametric mixture model and prove the existence and consistency of the proposed estimators. We further compare our estimator with several existing estimators through simulation studies and apply our method to two real data sets from biological sciences and astronomy.

The second part of this dissertation considers the model $g(x) = (1-p)f_0(x;\boldsymbol{\theta}) + pf(x)$, where $\boldsymbol{\theta}$ represents the unknown parameters of a known distribution f_0 , and f represents the distribution of possible outliers. We propose two innovative algorithms to estimate $\boldsymbol{\theta}$ nonparametrically. The first method is called Minimum Search, which is based on identifiability of the mixture model. A strong sufficient condition is proposed for the model to be identifiable and a weaker condition is given for the model to be locally identifiable. The second estimator is the maximum likelihood estimator, which is obtained by EM algorithm assuming f is log-concave. Extensive simulation studies show that our methods give very promising performances.

Contents

List of Figures

List of Tables

Chapter 1

Maximum Likelihood Estimation of a Semiparametric Two-component Mixture Model using Log-concave Density Estimation

1.1 Introduction

In this chapter, we consider the following two-component mixture model,

$$g(x) = (1-p)f_0(x) + pf(x), \tag{1.1}$$

where the probability density function (pdf) $f_0(x)$ is known, whereas the mixing proportion $p \in [0,1]$ and the pdf f are unknown. Model (2.1) is motivated by studies in biological sciences to cluster differentially expressed genes in microarray data, see [2]. Typically for microarray data, we build a test statistic, say T_i, for each gene i. Under the null hypothesis, which presumes no difference in expression levels under two or more conditions, T_i is assumed to have a known distribution (in general Student's t or Fisher). Under the alternative hypothesis, the distribution is unknown. Thus, the distribution of the test statistic can be described by model (2.1) where p is the proportion of non-null statistics. The estimation of p and the pdf f can tell us the probability P_i that gene i is differentially expressed given $T_i = t_i$:

$$P_i = \frac{pf(t_i)}{(1-p)f_0(t_i) + pf(t_i)}.$$

[2] considered model (2.1) assuming f to be symmetric. They obtained some identifiability results under moment and symmetry conditions.

[30] considered another special case,

$$g(x) = (1-p)\phi_\sigma(x) + pf(x)$$

where $f_0 = \phi_\sigma$ is a normal density with mean 0 and unknown standard deviation σ. This model was inspired by sequential clustering [29], which finds candidates for centers of clusters first, then carries out a local search to find the objects that belong to those clusters, and finally selects the best cluster. The algorithm repeats after the best cluster is being removed. [30] proposed an EM-type estimator and a maximizing π-type estimator for their model which can be easily extended to models where f_0 is not normal.

A slightly different model is considered by [35]:

$$g(x) = (1 - p)f_0(x; \xi) + pf(x - \mu),$$

where ξ is a possibly unknown parameter, and μ is a non-null location parameter for f. They proposed a new effective estimator based on the minimum profile Hellinger distance (MPHD). They established the existence and uniqueness of their estimator and also proved its consistency under some regularity conditions. Their method does not require f to be symmetric and thus can be applied to more general models. For some other alternative estimators, see, for example [21, 16].

In this chapter, we propose to estimate (2.1) using a new approach by imposing a fairly general log-concave shape constraint on f, i.e. $\log(f) \in \Phi$; here Φ denotes the family of concave functions ϕ on R which are upper semicontinuous and coercive in the sense that $\phi(x) \to -\infty$, as $|x| \to \infty$. Note that $\log(f)$ needs to be coercive in order for f to be a density function. The family of log-concave densities [7, 8] is very broad and contain many commonly used parametric families of distributions, such as normal distribution, exponential distribution, logistic distribution, etc. We propose to estimate the new model by maximizing a semiparametric mixture likelihood. Compared to the kernel density estimation of f used by many existing methods [2, 35, 16], the new method does not require the choice of one or more bandwidths [26]. We establish the identifiability of the proposed semiparametric mixture model and prove the existence and consistency of the proposed estimators. We further compare our estimator with several existing estimators through simulation studies and apply our method to two real data sets from biological sciences and astronomy.

The rest of the chapter is organized as follows. In Section 2.2 we discuss some identifiability issues for model (2.1). Section 1.3 introduces our maximum likelihood estimator and a detailed EM type algorithm. Existence and consistency properties of our estimator are established. Section 2.5 demonstrates the finite sample performance of our proposed estimator by comparing with many other existing algorithms. Two real data applications are given in Section 1.5. Section 2.6 gives a brief discussion. The Appendix in Section 2.7 contains the detailed proofs.

1.2 Identifiability

Note that the model (2.1) is non-identifiable without any constraint on the density f, see e.g., [2], and [21]. However, a parametric model for f might create biased or even misleading statistical inference when the model assumption is incorrect. In this chapter, we impose a general log-concave shape constraint on $f(x)$, i.e. $f(x) = e^{\phi(x)}$, where $\phi(x)$ is a concave function. Log-concave densities attracted lots of attention in the recent years since it is very flexible and can be estimated by nonparametric maximum likelihood estimator without requiring the choice of any tuning parameter. For more details, see [5], [8], [34], [9] and the review of the recent progress in log-concave density estimation by [26].

We first provide a lemma which can be easily proved by extending the result of Lemma 4 of [21].

Lemma 1.2.1. *The model (2.1) is identifiable if there exists a such that*

$$\lim_{x \to a^+} \frac{f(x)}{f_0(x)} = 0 \ or \ \lim_{x \to a^-} \frac{f(x)}{f_0(x)} = 0.$$

Remark 1.2.1. *Proposition 1.2.1 also holds if $a = \pm\infty$, and this result is more general (requiring weaker condition) than the result of Proposition 3(i) of [2].*

Remark 1.2.2. *Proposition 1.2.1 guarantees that model (2.1) is identifiable if the support of f is strictly contained in the support of f_0 and the two supports have different Legesgue measure.*

If $\log(f)$ is assumed to be log-concave, we can have the following result with the proof provided in Section 2.7.

Proposition 1.2.1. *Assume $f_0 > 0$ and $log(f) \in \Phi$. Model (2.1) is identifiable if either of the following two conditions are satisfied*

1. *$\phi(x) - log f_0(x) \to -\infty$ as $x \to +\infty$ or $x \to -\infty$.*

2. *$|\log f_0(x)| = O(|x|^k)$, for some $0 < k < 1$.*

Next we provide some examples to demonstrate how to use the above results to establish the identifiability of the model (2.1).

Example 1.2.1. *If $f_0(x)$ is the density of a t distribution with ν degrees of freedom, and f is log-concave, then model (2.1) is identifiable.*

Proof. Since $f_0(x) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\Gamma(\frac{\nu}{2})}(1 + \frac{x^2}{\nu})^{-\frac{\nu+1}{2}}$, we have,

$$\log(f_0(x)) = \log(\Gamma(\frac{\nu+1}{2})) - \frac{1}{2}\log(\nu\pi) - \log(\Gamma(\frac{\nu}{2})) - \frac{\nu+1}{2}\log(1 + \frac{x^2}{\nu}).$$

Thus, for any $0 < k < 1$, $\log(f_0(x))/x^k \to 0$, as $x \to +\infty$. Based on Proposition 1.2.1, we can conclude that model (2.1) is identifiable when $\log(f) \in \Phi$. $\square$

Remark 1.2.3. *Similarly, one can check that when f_0 is the pdf of an F distribution, log-normal distribution, or Pareto distribution, then model (2.1) is identifiable under the condition that $log(f) \in \Phi$.*

Remark 1.2.4. *Example 1.2.1 ensures Model 7 from Section 2.5 is identifiable.*

Example 1.2.2. *Suppose $f_0(x)$ is the density of a normal distribution with mean μ and variance σ^2, then model (2.1) is identifiable if $\lim_{x\to+\infty} \frac{\phi(x)}{x^2} < -\frac{1}{2\sigma^2}$, or $\lim_{x\to-\infty} \frac{\phi(x)}{x^2} < -\frac{1}{2\sigma^2}$, or the condition of Remark 1.2.2 holds.*

Proof. Suppose $\lim_{x\to+\infty} \frac{\phi(x)}{x^2} < -\frac{1}{2\sigma^2}$, or $\lim_{x\to-\infty} \frac{\phi(x)}{x^2} < -\frac{1}{2\sigma^2}$. Since

$$\begin{aligned}
\phi(x) - \log f_0(x) &= \phi(x) + \log(\sqrt{2\pi}\sigma) + \frac{1}{2\sigma^2}(x-\mu)^2 \\
&= x^2(\frac{\phi(x)}{x^2} + \frac{1}{x^2}\log(\sqrt{2\pi}\sigma) + \frac{1}{2\sigma^2}(1-\frac{\mu}{x})^2) \\
&\to -\infty, \text{ as } x \to +\infty \text{ or } x \to -\infty.
\end{aligned}$$

Hence $\dfrac{f(x)}{f_0(x)} \to 0$ as $x \to +\infty$ or $x \to -\infty$, and Proposition 1.2.1 asserts the identifiability of model (2.1). $\square$

Remark 1.2.5. *Under the constraints set by Example 1.2.2, Model 1, 4, and 5 from Section 2.5 are identifiable.*

Example 1.2.3. *Suppose $f_0(x)$ is the density of an exponential distritution with rate λ, then model (2.1) is identifiable if $\lim_{x\to+\infty} \frac{\phi(x)}{x} < -\lambda$, or the condition of Remark 1.2.2* *holds.*

Proof. Suppose $\lim_{x\to+\infty}\frac{\phi(x)}{x} < -\lambda$. Since,

$$\begin{aligned}\phi(x) - \log f_0(x) &= \phi(x) - \log\lambda + \lambda x \\ &= x(\frac{\phi(x)}{x} - \frac{\log\lambda}{x} + \lambda) \\ &\to -\infty, \text{ as } x \to +\infty.\end{aligned}$$

Hence $\dfrac{f(x)}{f_0(x)} \to 0$ as $x \to +\infty$, and again Proposition 1.2.1 ensures the identifiability of model (2.1). □

Remark 1.2.6. *Under the constraints set by Example 1.2.3, Model 3 from Section 2.5 is identifiable.*

1.3 Maximum Likelihood Estimation

Suppose we have a random sample of n i.i.d. observations $X_1, X_2, \cdots, X_n$ from the density $g(x) = (1-p)f_0(x) + pf(x)$, $p \in [0,1]$, and $f = e^{\phi}$ is a log-concave density, i.e., $\phi \in \Phi$. For any distribution Q on $\mathcal{R}$, we define,

$$L(p, \phi; Q) = \int \log((1-p)f_0 + pe^{\phi})dQ.$$

Then, with the empirical distribution $Q_n = \dfrac{1}{n}\sum_{i=1}^{n}\delta_{X_i}$, where δ_{X_i} is the degenerate distribution function at $\{X_i\}$, we propose to estimate p and ϕ by maximizing the following *semiparametric log likelihood*,

$$L(p, \phi; Q_n) = \frac{1}{n} \sum_{i=1}^{n} \log((1-p) f_0(X_i) + p e^{\phi(X_i)}), \tag{1.2}$$

subject to the condition that $\int e^{\phi(x)} dx = 1$. The log-likelihood (1.2) is semiparametric since it contains both the parameter p and the nonparametric component ϕ.

1.3.1 Algorithm

Maximizing the semiparametric log likelihood (1.2) is not trivial. To this end, we propose an EM algorithm [6] to maximize $L(p, \phi; Q_n)$.

Algorithm 1.3.1. *Staring from initial values $p^{(0)}$ and $f^{(0)}$, iterating the following E step and M step until convergence.*

E step *Given $p^{(k)}$ and $f^{(k)}$, find the classification probabilities*

$$\omega_i^{(k+1)} = \frac{(1-p^{(k)}) f_0(x_i)}{(1-p^{(k)}) f_0(x_i) + p^{(k)} f^{(k)}(x_i)}, i = 1, \ldots, n.$$

M step *Given $\omega_i^{(k+1)}$, update the parameter p and the nonparametric concave function ϕ,*

$$\begin{aligned} p^{(k+1)} &= \frac{1}{n} \sum_{i=1}^{n} (1 - \omega_i^{(k+1)}), \\ \phi^{(k+1)} &= \underset{\phi \in \Phi,\ \int e^{\phi(x)} dx = 1}{\arg\max} \sum_{i=1}^{n} (1 - \omega_i^{(k+1)}) \phi(x_i), \\ f^{(k+1)} &= e^{\phi^{(k+1)}}. \end{aligned}$$

In the M step, we find $\phi^{(k+1)}$ using an active set algorithm, which is described in [7] and implemened in the R package *logcondens* by [25]. Throughout this chapter, we use "EM_logconcave" to represent the above algorithm. The following result establishes the monotone properties of our EM_logconcave algorithm.

Proposition 1.3.1. *Let* $\ell^{(k)} = \sum_{i=1}^{n} \log((1-p^{(k)})f_0(x_i) + p^{(k)}e^{\phi^{(k)}(x_i)})$*, where* $p^{(k)}$ *and* $\phi^{(k)}$ *are kth update in Algorithm 1.3.1, then*

$$\ell^{(k+1)} \geq \ell^{(k)},$$

for any $k \geq 0$*.*

1.3.2 Theoretical Properties

For the existence of a maximizer of $L(p, \phi; Q)$ for a general distribution Q, we follow the approach of [9]. We define the convex support of Q as,

$$\mathrm{csupp}(Q) = \bigcap \{C : C \subseteq \mathcal{R} \text{ closed and convex, } Q(C) = 1\}.$$

Theorem 1.3.1. *For fixed* f_0*, assume* $\mathrm{supp}\{f_0\} \subseteq \mathrm{csupp}(Q)$*, and there exists some integer* $k \geq 1$*, such that,*

$$\int |x|^k Q(dx) < \infty \quad \text{and} \quad \mathrm{interior}(\mathrm{csupp}(Q)) \neq \emptyset.$$

For some fixed $m(x) = c_0 e^{c_1 |x|^k}, c_0, c_1 > 0$*. Let* $\tilde{\Phi} = \{\phi \in \Phi : \int e^{\phi(x)}dx = 1$ *and* $f_0(x) \leq$

$m(x)e^{\phi(x)}\}$. *Then*

$$L(Q) = \sup_{p\in[0,1],\ \phi\in\tilde{\Phi}} L(p, \phi, Q)$$

is real and there exists

$$(p_0, \phi_0) \in \operatorname*{argmax}_{p\in[0,1],\phi\in\tilde{\Phi}} L(p, \phi; Q).$$

Moreover,

$$\text{interior}(\text{csupp}(Q)) \subseteq \text{dom}(\phi_0) = \{x \in \mathcal{R} : \phi_0(x) > -\infty\} \subseteq \text{csupp}(Q).$$

The proof of Theorem 2.4.3 is given in the Appendix (Section 2.7).

Example 1.3.1. *Assume f_0 represents the standard normal density. Consider all the log-concave normal densities with mean μ and standard deviation σ. Suppose μ and σ are bounded. Then, for integer $k = 2$, there exist $c_0, c_1 > 0$, such that $\tilde{\Phi}$ contains all such normal pdfs with mean μ and standard deviation σ. In addition, Theorem 2.4.3 implies that the maximum of $L(p, \phi; Q)$ exists over $p \in [0, 1]$ and $\phi \in \tilde{\Phi}$.*

In general, the maximizer of $L(p, \phi; Q)$ is not unique. However, if Q has density $g_0(x) = (1 - p_0)f_0(x) + p_0e^{\phi_0(x)}$, where $g_0(x)$ is identifiable, then $L(p_0, \phi_0; Q) = \int \log(g_0(x))g_0(x)dx$, and this (p_0, ϕ_0) is the unique maximizer. This is because as noted by [9], if we have (p_1, ϕ_1), such that $L(Q) = L(p_0, \phi_0; Q) = L(p_1, \phi_1; Q)$, let $g_1(x) = (1 - p_1)f_0(x) + p_1e^{\phi_1(x)}$, then,

$$\int \log(g_0(x)/g_1(x))g_0(x)dx = 0.$$

Note the above integral is exactly the Kullback-Leibler divergence which is positive and equals 0 iff $g_0 = g_1$ almost everywhere. Thus $(p_0, \phi_0) = (p_1, \phi_1)$ except that ϕ_0 and ϕ_1 may differ on a set of Lebesgue measure zero.

Next we establish the consistency of our maximum likelihood estimator. First, we introduce some notations,

$$\begin{aligned} \mathcal{Q}^k &= \{Q \text{ on } \mathcal{R} : \int |x|^k Q(dx) < \infty\}, \\ \mathcal{Q}_0 &= \{Q \text{ on } \mathcal{R} : \text{interior}(\text{csupp}(Q)) \neq \emptyset\}. \end{aligned}$$

In the remainder of this section, we consider the convergence of distributions under Mallows' distance D_1 [17]. Specifically, for two distributions Q, $Q' \in \mathcal{Q}^k$,

$$D_k(Q, Q') = \inf_{\substack{X,\ X' \\ X \sim Q,\ X' \sim Q'}} \{E|X - X'|^k\}^{1/k}.$$

It is known that $\lim_{n\to\infty} D_k(Q_n, Q) \to 0$ is equivalent to $Q_n \to_w Q$ and $\int |x|^k Q_n(dx) \to \int |x|^k Q(dx)$ [1, 17]. Here $Q_n \to_w Q$ means weak convergence, or convergence in distribution.

Now we are ready to state our main consistency theorem.

Theorem 1.3.2. *Assume, (a).* $\text{supp}\{f_0\} \subseteq \text{csupp}(Q)$*; (b). for some fixed integer $k \geq 1$, the unknown density f satisfies the following condition: $\exists\ m(x) = c_0 e^{c_1 |x|^k}$, where $c_i > 0$, $i = 0, 1$, such that, $f_0(x) \leq m(x) f(x) = m(x) e^{\phi(x)}$. Let $\{Q_n\}$ be a sequence of distributions in $\mathcal{Q}_0 \bigcap \mathcal{Q}^k$ such that $\lim_{n\to\infty} D_k(Q_n, Q) = 0$ for some $Q \in \mathcal{Q}_0 \bigcap \mathcal{Q}^k$. Suppose f_0 is upper* *semi-continuous and* $\dfrac{log(f_0)}{1+|x|}$ *is bounded. Then*

$$\lim_{n\to\infty} L(Q_n) = L(Q).$$

Assume there exist maximizers (p_n, ϕ_n) *of* $L(p, \phi; Q_n)$*, and a unique maximizer* (p^*, ϕ^*) *of* $L(p, \phi; Q)$*, where* $p_n, p^* \in [0, 1], \phi_n, \phi^* \in \tilde{\Phi}$*. Let* $f_n = \exp(\phi_n)$*,* $f^* = \exp(\phi^*)$*, then*

$$\begin{aligned} \lim_{n\to\infty} p_n &= p^*, \\ \lim_{n\to\infty,\ x\to y} f_n(x) &= f^*(y), \quad \forall y \in \mathcal{R} \setminus \partial\{f^* > 0\}, \\ \operatorname*{limsup}_{n\to\infty,\ x\to y} f_n(x) &\leq f^*(y), \quad \forall y \in \partial\{f^* > 0\}, \\ \lim_{n\to\infty} \int |f_n(x) - f^*(x)|dx &= 0, \end{aligned}$$

here $\partial\{f^* > 0\}$ *represents the boundary of the set* $\{f^* > 0\}$*.*

Practically, Q_n will be the empirical distribution function which automatically satisfies the above assumption. Based on the above theorem, we can know that the proposed semiparametric maximum likelihood estimators of p and f are consistent.

Remark 1.3.1. *Theorem 2.4.3 and Theorem 2.4.4 still holds if we consider the distribution* Q *to be defined on* $\mathcal{R}^d$*, with* $d = 1, 2, 3, \cdots$ *.*

1.4 Simulation

In this section, we investigate the finite sample performance of our algorithm and compare it to the estimator proposed by [21] ($\hat{\alpha}_0^{0.1k_n}$ from their chapter), the Symmetrization estimator by [2], the EM-type estimator and Maximizing-π type estimator by [30], and the Minimum profile Hellinger distance estimator by [35].

In order to test our method under different settings, we simulate $K = 200$ samples of n i.i.d. random variables with the common distribution given by the following seven models:

- Model 1: $g(x) = (1-p)N(\mu = 0, \sigma = 2) + pN(\mu = 3, \sigma = 1)$,
- Model 2: $g(x) = (1-p) \cdot \text{unif}(0,1) + p \cdot \text{beta}(\alpha = 1, \beta = 5)$,
- Model 3: $g(x) = (1-p) \cdot \exp(\lambda = 1) + p \cdot (\exp(\lambda = 1) + 2)$,
- Model 4: $g(x) = (1-p)N(0,1) + p(\chi^2(3) + 2)$,
- Model 5: $g(x) = (1-p)N(0,1) + p \cdot (\exp(\lambda = 0.5) + 3)$,
- Model 6: $g(x) = (1-p)N(0,1) + p \cdot (t(d.f. = 5) + 3)$,
- Model 7: $g(x) = (1-p) \cdot t_{df=5} + p \cdot \text{logistic}(\text{location} = 5, \text{scale} = 0.5)$.

For each sample we estimate p, the mean μ of the unknown component f and the classification error. For our algorithm and the algorithm by [35], final estimators $\hat{p}$ and $\hat{f}$ are always produced, thus the estimated probability $\hat{w}_i$ that the i-th observation is from the known component $f_0(x)$, given $X_i = x_i$, can be calculated by

$$\hat{w}_i = \frac{(1-\hat{p})f_0(x_i)}{(1-\hat{p})f_0(x_i) + \hat{p}\hat{f}(x_i)}.$$

For other methods, $\hat{f}$ may not always be given directly. Suggested by [30], we estimate $\hat{w}_i$

by the following,

$$\hat{w_i} = \frac{2(1-\hat{p})f_0(x_i)}{(1-\hat{p})f_0(x_i) + \hat{h}(x_i)},$$

where $\hat{h}$ is the kernel density estimator of g with Gaussian kernel and Silverman's "rule of thumb" bandwidth [28]. Note that the algorithm proposed by [21] actually can estimate f when f is non-increasing. But we find that the algorithm works best when f_0 and f have the same support and it often produces unreliable estimates when the two supports differ from each other. Thus, we do not use $\hat{f}$ to estimate $\hat{w_i}$ for [21]'s algorithm even when the true f does decrease on its support, instead, we follow [30]'s recommendation to get $\hat{w_i}$.

The algorithms by [35] and [2] give a final mean estimator $\hat{\mu}$ directly. For other methods, after we get $\hat{w_i}$, we estimate μ by the following weighted sum,

$$\hat{\mu} = \frac{\sum_{i=1}^{n}(1-\hat{w_i})X_i}{\sum_{i=1}^{n}(1-\hat{w_i})}.$$

Last, we report the classification error (CE) based on $\hat{w_i}$ as the mean squared error between $\hat{w_i}$ and the true w_i, i.e.,

$$\text{CE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{w_i} - w_i)^2,$$

where $w_i = 1$ if x_i is from the known component $f_0(x)$ and 0 if x_i is from the unknown component $f(x)$.

For model 1, Table 1 reports the bias and MSE of the estimates of p, the bias and MSE of the estimates of μ, and the mean of the classification error (MCE) for different methods over $K = 200$ repetitions when $p = 0.2$, $p = 0.5$, and $p = 0.8$, with sample size $n = 1000$. Similar reports of other models can be found in Tables 2.2 — 2.7. Simulation results for sample sizes $n = 250$ and $n = 500$ are reported in the Appendix (Section 2.7). We report the results of [2]'s algorithm only for model 1, 2, 6 and 7, because this method fails when $f(x)$ is not symmetric.

Table 1.1: Bias (MSE) of estimates of p/μ and mean of the classification error for model 1 when $n = 1000$.

	EM_logconcave	Patra	Bordes	Song EM	Song max π	Xiang
$p = 0.2$						
p	0.002(0.0004)	0.009(0.0007)	0.001(0.0009)	0.08(0.0066)	0.087(0.0122)	0.006(0.0006)
μ	0.063(0.0180)	-0.152(0.0650)	-0.021(0.0426)	0.116(0.0446)	-0.675(0.5680)	0.116(0.0396)
MCE	0.0960	0.1056	0.1052	0.1102	0.1052	0.0973
$p = 0.5$						
p	-0.002(0.0004)	-0.025(0.0011)	0.001(0.0006)	-0.132(0.0177)	0.106(0.0149)	0.007(0.0006)
μ	0.018(0.0042)	0.051(0.0073)	0.000(0.0046)	0.185(0.0375)	-0.322(0.1392)	0.013(0.0056)
MCE	0.1094	0.1219	0.1198	0.1352	0.1206	0.1104
$p = 0.8$						
p	0.001(0.0002)	-0.252(0.0020)	0.001(0.0003)	-0.107(0.0118)	0.063(0.0047)	0.009(0.0003)
μ	0.005(0.0013)	0.066(0.0057)	0.000(0.0016)	0.118(0.0153)	-0.128(0.0220)	-0.002(0.0021)
MCE	0.0645	0.0739	0.0694	0.0834	0.0721	0.0664

Table 1.2: Bias (MSE) of estimates of p/μ and mean of the classification error for model 2 when $n = 1000$.

	EM_logconcave	Patra	Bordes	Song EM	Song max π	Xiang
$p = 0.2$						
p	-0.008(0.0014)	-0.023(0.0015)	-0.015(0.0012)	-0.15(0.0228)	0.382(0.1496)	0.017(0.0019)
μ	-0.018(0.0015)	0.027(0.0016)	-0.029(0.0017)	-0.014(0.0007)	0.199(0.0401)	-0.007(0.0010)
MCE	0.1270	0.1520	0.1511	0.1676	0.1847	0.1339
$p = 0.5$						
p	0.001(0.0007)	-0.046(0.0030)	-0.040(0.0024)	-0.248(0.0811)	0.228(0.0548)	-0.047(0.0035)
μ	-0.003(0.0001)	-0.011(0.0002)	-0.032(0.0011)	-0.038(0.0015)	0.077(0.0064)	-0.022(0.0012)
MCE	0.1609	0.1990	0.1974	0.2638	0.1887	0.1753
$p = 0.8$						
p	-0.001(0.0004)	-0.074(0.0059)	-0.070(0.0055)	-0.311(0.0974)	0.099(0.0105)	-0.060(0.0043)
μ	-0.001(0.00004)	-0.019(0.0004)	-0.033(0.0011)	-0.040(0.0016)	0.025(0.0007)	-0.030(0.0014)
MCE	0.1000	0.1264	0.1261	0.2103	0.1129	0.1142

Table 1.3: Bias (MSE) of estimates of p/μ and mean of the classification error for model 3 when $n = 1000$.

	EM_logconcave	Patra	Bordes	Song EM	Song max π	Xiang
$p = 0.2$						
p	0.001(0.0002)	-0.001(0.0006)	NA	-0.060(0.0038)	0.410(0.1698)	0.024(0.0011)
μ	0.006(0.0082)	-0.039(0.0152)	NA	0.048(0.0149)	-1.140(1.3094)	-0.105(0.0184)
MCE	0.0709	0.0851	NA	0.0879	0.1568	0.0790
$p = 0.5$						
p	0.000(0.0003)	-0.013(0.0006)	NA	-0.073(0.0057)	0.259(0.0681)	0.042(0.0028)
μ	0.003(0.0021)	-0.011(0.0030)	NA	0.018(0.0030)	-0.502(0.2578)	-0.091(0.0157)
MCE	0.0595	0.0767	NA	0.0790	0.1166	0.0732
$p = 0.8$						
p	0.001(0.0002)	-0.228(0.0010)	NA	-0.231(0.0012)	0.104(0.0112)	0.071(0.0060)
μ	-0.001(0.0013)	-0.002(0.0014)	NA	-0.002(0.0014)	-0.159(0.0283)	-0.104(0.0224)
MCE	0.0260	0.0325	NA	0.0322	0.0526	0.0617

Table 1.4: Bias (MSE) of estimates of p/μ and mean of the classification error for model 4 when $n = 1000$.

	EM_logconcave	Patra	Bordes	Song EM	Song max π	Xiang
$p = 0.2$						
p	-0.000(0.0002)	0.005(0.0005)	NA	0.006(0.0003)	0.106(0.0160)	0.056(0.0041)
μ	-0.023(0.0321)	-0.286(0.1299)	NA	-0.304(0.1353)	-1.066(1.4174)	-0.738(1.0279)
MCE	0.0112	0.0139	NA	0.0137	0.0205	0.0215
$p = 0.5$						
p	0.000(0.0002)	-0.009(0.0004)	NA	0.014(0.0005)	0.067(0.0057)	0.049(0.0030)
μ	-0.005(0.0074)	-0.148(0.0332)	NA	-0.185(0.0459)	-0.333(0.1439)	-0.676(0.6616)
MCE	0.0110	0.0157	NA	0.0163	0.0160	0.0207
$p = 0.8$						
p	-0.001(0.0001)	-0.023(0.0007)	NA	0.006(0.0002)	0.038(0.0019)	0.066(0.0048)
μ	-0.002(0.0052)	-0.025(0.0077)	NA	-0.054(0.0098)	-0.147(0.0352)	-0.718(0.5396)
MCE	0.0045	0.0078	NA	0.0089	0.0108	0.0319

Table 1.5: Bias (MSE) of estimates of p/μ and mean of the classification error for model 5 when $n = 1000$.

	EM_logconcave	Patra	Bordes	Song EM	Song max π	Xiang
$p = 0.2$						
p	0.001(0.0002)	0.006(0.0006)	NA	0.018(0.0005)	0.110(0.0154)	0.037(0.0018)
μ	0.004(0.0193)	-0.399(0.2021)	NA	-0.427(0.2126)	-1.129(1.4965)	-0.923(0.9073)
MCE	0.0012	0.0044	NA	0.0045	0.0105	0.0092
$p = 0.5$						
p	0.000(0.0003)	-0.009(0.0005)	NA	0.023(0.0008)	0.131(0.0208)	0.061(0.0044)
μ	-0.001(0.0079)	-0.173(0.0384)	NA	-0.213(0.0538)	-0.681(0.5620)	-0.650(0.4645)
MCE	0.0007	0.0079	NA	0.0090	0.0202	0.0189
$p = 0.8$						
p	0.001(0.0001)	-0.223(0.0007)	NA	0.009(0.0002)	0.079(0.0068)	0.081(0.0071)
μ	0.000(0.0047)	-0.027(0.0061)	NA	-0.051(0.0077)	-0.313(0.1123)	-0.473(0.2482)
MCE	0.0003	0.0022	NA	0.0030	0.0190	0.0426

Table 1.6: Bias (MSE) of estimates of p/μ and mean of the classification error for model 6 when $n = 1000$.

	EM_logconcave	Patra	Bordes	Song EM	Song max π	Xiang
$p = 0.2$						
p	-0.010(0.0003)	-0.007(0.0006)	0.083(0.0012)	-0.022(0.0006)	0.077(0.0080)	0.001(0.0003)
μ	0.171(0.0455)	0.029(0.0262)	-0.075(0.0843)	0.083(0.0257)	-0.414(0.2369)	0.001(0.0177)
MCE	0.0440	0.0450	0.0455	0.0457	0.0468	0.0435
$p = 0.5$						
p	0.001(0.0003)	-0.031(0.0015)	-0.002(0.0006)	-0.053(0.0031)	0.038(0.0028)	-0.002(0.0631)
μ	-0.010(0.0115)	0.148(0.0264)	-0.003(0.0066)	0.182(0.0375)	-0.020(0.0185)	0.018(0.0066)
MCE	0.0094	0.0672	0.0658	0.0680	0.0656	0.0631
$p = 0.8$						
p	-0.001(0.0001)	-0.059(0.0037)	-0.002(0.0004)	-0.063(0.0043)	0.008(0.0006)	0.001(0.0034)
μ	-0.004(0.0072)	0.169(0.0307)	-0.001(0.0024)	0.174(0.0321)	0.055(0.0073)	-0.003(0.0034)
MCE	0.0046	0.0637	0.0570	0.0643	0.0567	0.0545

Table 1.7: Bias (MSE) of estimates of p/μ and mean of the classification error for model 7 when $n = 1000$.

	EM_logconcave	Patra	Bordes	Song EM	Song max π	Xiang
$p = 0.2$						
p	-0.001(0.0002)	0.003(0.0004)	-0.002(0.0003)	0.011(0.0004)	0.2(0.0529)	0.0095(0.0004)
μ	0.007(0.006)	-0.423(0.2122)	0(0.0071)	-0.468(0.2455)	-1.799(3.7131)	-0.0116(0.0057)
MCE	0.0117	0.0139	0.0137	0.014	0.0403	0.0138
$p = 0.5$						
p	-0.003(0.0003)	-0.012(0.0005)	0(0.0003)	0.014(5e-04)	0.131(0.0217)	0.0134(0.0004)
μ	0.017(0.0023)	-0.148(0.0255)	-0.001(0.0022)	-0.192(0.0413)	-0.683(0.5779)	-0.0041(0.0017)
MCE	0.0129	0.0145	0.0146	0.0151	0.0315	0.0144
$p = 0.8$						
p	-0.0060(0.0002)	-0.024(0.0008)	0(0.0002)	0.002(0.0002)	0.061(0.0046)	0.0097(0.0002)
μ	0.02(0.0015)	-0.014(0.0015)	0(0.0012)	-0.044(0.0035)	-0.24(0.0695)	0.0046(0.0013)
MCE	0.0103	0.0094	0.01	0.0101	0.0197	0.0108

The simulation results demonstrate that our method has the overall best performance among all methods. In addition, our method is even more favorable when the sample size n gets larger. Overall, all the estimates of μ get better when p gets larger, which is expected because we are getting more points from the unknown component. [2]'s method does not work well when f is not symmetric due to the fact that their algorithm is based on the symmetry of f. [35]'s method has excellent performance when f is symmetric, because their algorithm incorporates the symmetry property of f in these cases. When f is not symmetric, our algorithm is far superior. [21]'s method works better when p is small, but it only estimates f when it is decreasing.

To better display our simulation results, we also plot the MSE of point estimates of p and μ vs. different models for all the methods we mentioned above when $p = 0.2$ and $n = 1000$, except for the method by [2] as their method fails to estimate p and μ for half of the models we discussed here. Figure 1.1 shows that the curve representing our method always lies at the bottom for all seven models considered, which demonstrates the effectiveness of our new method.

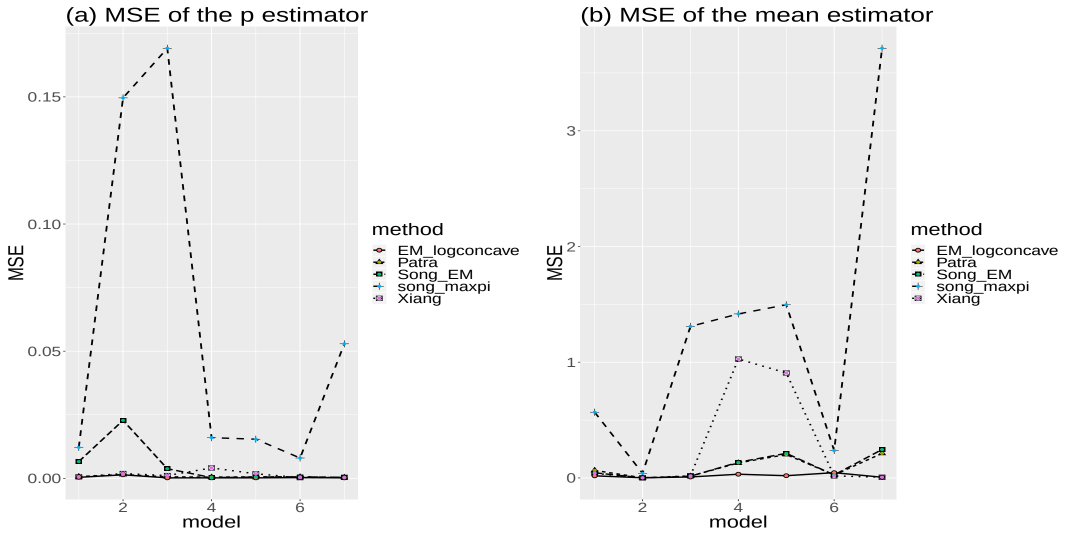

Figure 1.1: (a): MSE of the estimates of p when $p = 0.2,\ n = 1000$; (b): MSE of the estimates of μ when $p = 0.2,\ n = 1000$.

1.5 Real Data Application

1.5.1 Prostate Data

In this section we consider the prostate data consisting of genetic expression levels related to prostate cancer patients of [10]. The data set is a 6033 × 102 matrix, with entries x_{ij} = expression level for gene i on patient j, $i = 1, \cdots, n$, $j = 1, \cdots, m$, here, $n = 6033$, $m = 102$. Among the $m = 102$ patients, $m_1 = 50$ of them are normal control subjects (corresponding to $j = 1, \cdots, m_1$) and $m_2 = 52$ of them are prostate cancer patients (corresponding to $j = m_1 + 1, \cdots, m_2$). The goal of the study is to discover the potential genes that are differentially expressed between normal control and prostate cancer patients.

Two-sample t-test is performed to test the significance of each gene i by,

$$t_i = (\bar{x}_i(1) - \bar{x}_i(2))/s_i,$$

where $\bar{x}_i(1) = (\sum_{j=1}^{m_1} x_{ij})/m_1$, $\bar{x}_i(2) = (\sum_{j=m_1+1}^{m_2} x_{ij})/m_2$, $s_i^2 = (1/m_1 + 1/m_2) \left\{ \sum_{j=1}^{m_1} (x_{ij} - \bar{x}_i(1))^2 + \sum_{j=m_1+1}^{m_2} (x_{ij} - \bar{x}_i(2))^2 \right\} /(m-2)$. These two-sided t-tests produce $n = 6033$ p-values, and the distribution of these p-values under the null hypothesis (i.e., the gene is not differentially expressed) has a uniform density, while under the alternative hypothesis (i.e., the gene is differentially expressed) has a non-increasing density.

The estimation of p is reported in Table 1.8. We can see that the estimate by [2] and the Maximizing-π type estimate by [30] give a relatively big estimate. The estimate procedure by [2] assumes the density function under the alternative hypothesis to be symmetric, while in our example this density is non-increasing, which violates the symmetric assumption. (If we apply [2]'s method to the original t statistics directly, the estimate of p is $\hat{p} = 0.0072$.) It is known that the Maximizing-π type estimator by [30] tends to overestimate the p value, which can also be seen in Table 1.8. We also want to point out that several approaches have been proposed by [10] to estimate p as well, the estimator based on central matching method gives $\hat{p} = 0.020$ (please see [10] and [11] for detailed description of those estimators), and Table 1.8 shows that our estimator gives a closest value to Efron's result.

Table 1.8: Estimates of p for the prostate cancer data.

EM_logconcave	Patra	Bordes	Song EM	Song max π	Xiang
0.0173	0.0817	0.1975	0.0076	0.6132	0.1915

Figure 1.2 plots the estimated density $\hat{f}$ based on our method and the method by [21]. It can be seen that our estimate of the density $\hat{f}$ tends to have a much smaller support compared to the one given by [21]. Note that small p-values indicate the support for the alternative hypothesis. Therefore, it makes sense that the support of f for this prostate data may be much smaller than $(0, 1)$.

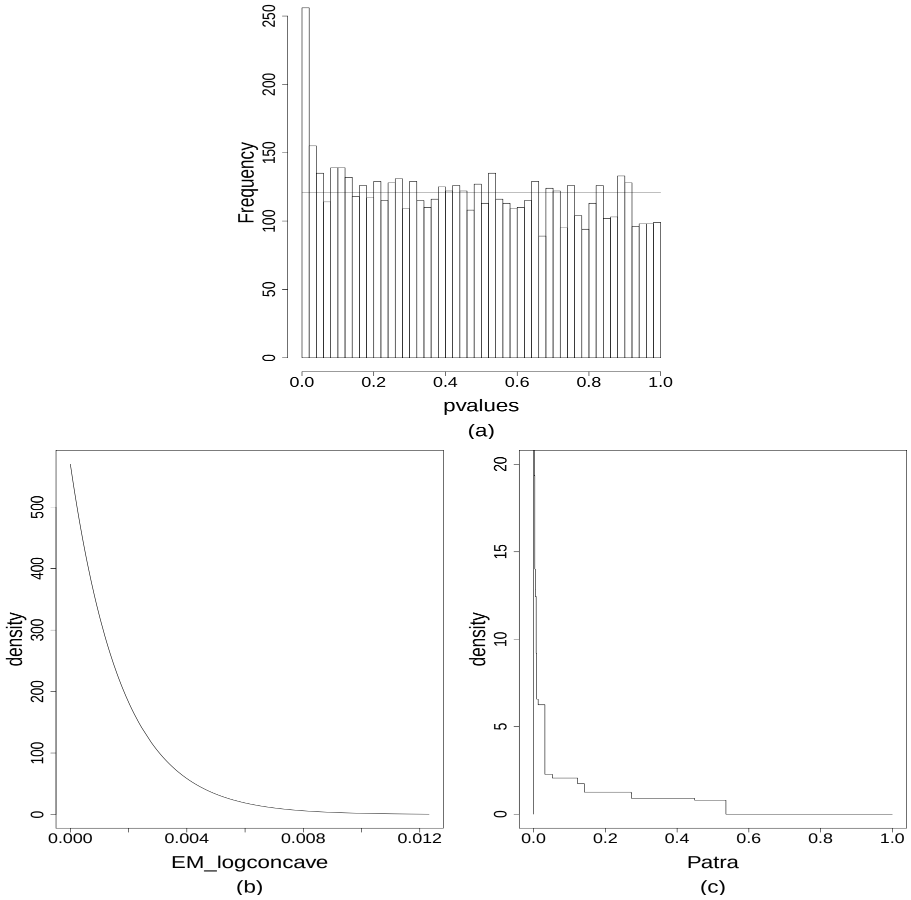

Figure 1.2: Plots for the prostate data: (a) Histogram of the p-values. The horizontal line represents the Uniform(0,1) distribution. (b) Plot of the estimated density $\hat{f}$ by our maximum likelihood estimation via EM algorithm. (c) Plot of the estimated density $\hat{f}$ by the method of [21].

1.5.2 Carina Data

Carina is one of the seven dwarf spheroidal (dSph) satellite galaxies of Milkey Way. Here we consider the data consisting of radial velocities (RV) of $n = 1266$ stars from Carina galaxy. The data is obtained by Magellan and MMT telescopes ([33]). The stars of Milkey

Way contribute contamination to this data set. We assume the distribution f_0 of RV from stars of Milkey Way is known and follows the Besancon Milky Way model ([23]). We would like to analyze this data set to better understand the inhomogeneous distribution of the RV of stars in Carina galaxy.

The estimation of p is reported in Table 1.9. We see that the estimation by [30]'s Maximizing-π type estimator gives a relatively big estimate. Other estimates are relatively close.

Table 1.9: Estimates of p for the Carina data.

EM_logconcave	Patra	Bordes	Song EM	Song max π	Xiang
0.354	0.364	0.363	0.370	0.687	0.385

In Figure 1.3, we plot the histogram of the RV data overlaid with our estimated two components of the mixture density. Based on the plot, we can see that our estimation approximates the data fairly well. The component corresponding to the stars of Carina looks very symmetric, and in fact astronomers usually assume the distribution to be Gaussian, which makes the density estimation proposed by [21] fail.

1.6 Discussion

In this chapter we study the two-component mixture model with one component completely known. A semiparametric maximum likelihood estimator is developed via EM

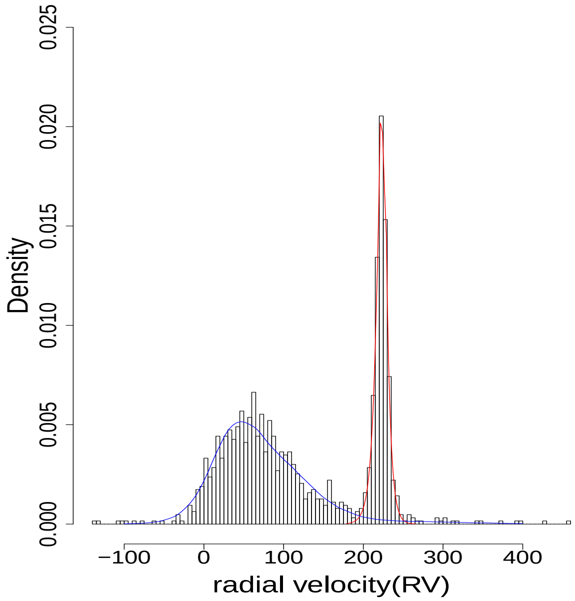

Figure 1.3: Histogram of RV data overlaid with the estimated two components from our EM log-concave algorithm

algorithm and log-concave approximation. Unlike most existing estimation procedures, our new method finds the mixing proportion and the distribution of the unknown component simultaneously without any selection of a tuning parameter and the proposed EM algorithm satisfies the non-decreasing property of the traditional EM algorithm. We establish the existence of consistency of the proposed estimator. Simulation results show that our method is more favorable than many other competing estimation methods.

In this chapter, we assume that the first component is completely known, it would be our interest to apply our method to a more general model where the component f_0 also contains some unknown parameter and extend our method to the regression setting.

1.7 Appendix

1.7.1 Theoretical Proof

Proof of Lemma 1.2.1. According to [21], if we let G, F_0, and F be the cumulative distribution functions of g, f_0 and f respectively, define $p_0 = \inf\{\gamma \in (0,1] : [G-(1-\gamma)F_0]/\gamma$ is a CDF$\}$, then

$$p_0 = p\{1 - \text{essinf}\frac{f}{f_0}\},$$

where essinf$(h) = \sup\{t \in R : \mathfrak{m}\{x : h(x) < t\} = 0\}$, and here $\mathfrak{m}$ represents the Lebesgue measure. Now if essinf$\frac{f}{f_0} > 0$, there must exist some $t > 0$, such that, $\mathfrak{m}\{x : \frac{f(x)}{f_0(x)} < t\} = 0$, i.e., $\frac{f(x)}{f_0(x)} \geq t$ almost everywhere, which contradicts to the fact that $\lim_{x\to a^+} \frac{f(x)}{f_0(x)} = 0$ or $\lim_{x\to a^-} \frac{f(x)}{f_0(x)} = 0$. Hence we can conclude that essinf$\frac{f}{f_0} = 0$, and consequently $p_0 = p$, which means if we can write $g(x) = (1-p)f_0(x) + pf(x)$, this p is fixed and equals p_0. Consequently $f(x) = (g(x) - (1-p)f_0(x))/p$ is fixed as well, and our model (2.1) is identifiable. □

Proof of Proposition 1.2.1. Since $f(x) = e^{\phi(x)}$ is a log-concave density, there exist constants a and $b > 0$, such that $\phi(x) \leq a - b|x|$ (see [4]), which implies

$$\phi(x) - \text{log}f_0(x) \leq a - b|x| - \text{log}f_0(x).$$

Now if $|\text{log}f_0(x)| = O(|x|^k)$, for some $0 < k < 1$, apparently,

$$-b|x| - \text{log}f_0(x) = |x|^k(-b|x|^{1-k} - \text{log}f_0(x)/|x|^k) \to -\infty, \text{ as } x \to +\infty \text{ or } x \to -\infty.$$

Hence $\phi(x) - \log f_0(x) \to -\infty$ as $x \to +\infty$ or $x \to -\infty$, which shows $\lim_{x\to+\infty} \frac{f(x)}{f_0(x)} = 0$. Thus, model (2.1) is identifiable from Proposition 1.2.1. □

Proof of Theorem 2.4.3. Suppose $\int |x|^k Q(dx) < \infty$, interior(csupp($Q$)) $\neq \emptyset$, $\int e^{\phi(x)}dx = 1$ and $f_0(x) \leq m(x)e^{\phi(x)}$. For any concave function ϕ satisfying the above conditions, there exist (a_0, b_0), such that $\phi(x) \leq a_0 - b_0|x|$, thus for any $p \neq 0$, $L(p, -b_0|x| - \log(\int e^{-b_0|x|}dx); Q) \geq \log\frac{p}{\int e^{-b_0|x|}dx} - b_0 \int |x|Q(dx) > -\infty$, thus we have $L(Q) > -\infty$. When maximizing $L(p, \phi; Q)$ over all $\phi \in \tilde{\Phi}$, we may restrict our attention to functions ϕ such that $\text{dom}(\phi) = \{x \in \mathcal{R} : \phi(x) > -\infty\} \subseteq \text{csupp}(Q)$. For if $\text{dom}(\phi) \nsubseteq \text{csupp}(Q)$, replacing $\phi(x)$ with $-\infty$ for all $x \notin \text{csupp}(Q)$, then the value of $L(p, \phi - \log(\int e^{\phi(x)}dx); Q)$ would be greater or equal to the original $L(p, \phi; Q)$. Note that since $\text{supp}(f_0) = \text{csupp}(Q)$, the new concave function $\phi' = \phi - \log(\int e^{\phi(x)}dx)$ still satisfies the conditions above, i.e., $\int e^{\phi'(x)}dx = 1$, $f_0(x) \leq m(x)e^{\phi'(x)}$ and $\text{dom}(\phi') = \{x \in \mathcal{R} : \phi(x) > -\infty\} \subseteq \text{csupp}(Q)$. We denote $\Phi(Q)$ to be the family of all $\phi \in \tilde{\Phi}$ with $\text{dom}(\phi) = \{x \in \mathcal{R} : \phi(x) > -\infty\} \subseteq \text{csupp}(Q)$.

Now we show that $L(Q) < \infty$. Suppose that $\phi \in \Phi(Q)$ is such that $M = \max_{x \in R^d}\phi(x) > 0$. Let $D_t = \{\phi \geq t\}$, hence D_t is closed and convex. For any $\alpha > 0$, we have the following estimate,

$$\begin{aligned}
L(p, \phi; Q) &= \int \log((1-p)f_0 + pe^{\phi})dQ \\
&\leq \int \log((1-p)m(x) + p)Q(dx) + \int \phi dQ \\
&\leq \int \log(m(x)+1)Q(dx) - \alpha M Q(R^d \setminus D_{-\alpha M}) + MQ(D_{-\alpha M}) \\
&= \int \log(m(x)+1)Q(dx) - (\alpha+1)M(\frac{\alpha}{\alpha+1} - Q(D_{-\alpha M})).
\end{aligned}$$

Note that $\int \log(m(x)+1)Q(dx)$ exists since $\int |x|^k Q(dx) < \infty$. By Lemma 4.1 of [9], for any fixed α,

$$\begin{aligned} Leb(D_{-\alpha M}) &\leq (1+\alpha)^d M^d e^{-M} / \int_0^{(1+\alpha)M} t^d e^{-t} dt \\ &= (1+\alpha)^d M^d e^{-M} / (d! + o(1)) \to 0, \text{ as } M \to \infty. \end{aligned}$$

Lemma 2.1 of [9] says that for sufficiently large α and sufficiently small $\delta > 0$, there exist some sufficiently small $\epsilon > 0$, such that,

$$\sup\{Q(C) : C \subseteq \mathcal{R} \text{ closed and convex, Leb}(C) \leq \delta\} < \frac{\alpha}{\alpha+1} - \epsilon,$$

which implies that $L(p, \phi; Q) \to -\infty$, as $M \to \infty$. Since for any $\phi \in \Phi(Q)$, we also have $L(p, \phi; Q) \leq \int \log((1-p)m(x) + p)dQ + M$, $\Rightarrow L(Q) < \infty$ and there exist constants M_0 and M_*, such that

$L(Q) = \sup\limits_{\substack{p \in [0,1],\ \phi \in \Phi(Q) \\ M_0 \leq \max(\phi(x)) \leq M_*}} L(p, \phi; Q).$

Now that we know $L(Q)$ is real, we are ready to prove the existence of a maximizer (p_0, ϕ_0) of $L(Q)$. Let (p_n, ϕ_n) be a sequence such that $p_n \in [0,1]$, $\phi_n \in \Phi(Q)$, $M_n = \max(\phi_n(x)) \in [M_0, M_*]$, and $-\infty < L(p_n, \phi_n; Q) \uparrow L(Q)$ as $n \to \infty$. Here we assume $\{p_n\}$ is a convergent sequence, say $p_n \to p_0 \in [0,1]$, as $n \to \infty$. If $\{p_n\}$ is not convergent, since it is bounded, it must have a convergent subsequence $\{p_{n_k}\}$, and the sequence $\{p_{n_k}, \phi_{n_k}\}$ would satisfy all those properties above and we can just simply replace the original sequence with this subsequence.

Next, we show that,

$$\inf_{n\geq 1} \phi_n(x_0) > -\infty, \ \forall x_0 \in \text{interior}(\text{csupp}(Q)). \tag{1.3}$$

For any $x_0 \in \text{interior}(\text{csupp}(Q))$, if $\phi_n(x_0) < M_n$, then x_0 can not be an interior point of $\{\phi_n \geq \phi_n(x_0)\}$, hence,

$$\begin{aligned}
L(p_n, \phi_n; Q) &= \int \log((1-p_n)f_0 + p_n e^{\phi_n})dQ \\
&\leq \int \log(m(x)+1)Q(dx) + \int \phi_n dQ \\
&\leq \int \log(m(x)+1)Q(dx) + \phi_n(x_0) + (M_n - \phi_n(x_0))Q(\phi_n \geq \phi_n(x_0)) \\
&\leq \int \log(m(x)+1)Q(dx) + \phi_n(x_0)(1 - h(Q, x_0)) + \max(M_n, 0),
\end{aligned}$$

where $h(Q, x) = \sup\{Q(C) : C \subseteq R^d \text{ closed and convex}, \ x \notin \text{interior}(C)\} < 1$ by Lemma 2.13 of [9]. And the above inequalities still hold even if $\phi_n(x_0) = M_n$. Thus we have,

$$\begin{aligned}
\phi_n(x_0) &\geq \frac{L(p_n, \phi_n; Q) - \int \log(m(x)+1)Q(dx) - \max(M_n, 0)}{1 - h(Q, x_0)} \\
\Rightarrow \inf_{n\geq 1} \phi_n(x_0) &\geq \frac{L(p_1, \phi_1; Q) - \int \log(m(x)+1)Q(dx) - \max(M_*, 0)}{1 - h(Q, x_0)} > -\infty,
\end{aligned}$$

which establishes (1.3). Since $\phi_n \leq M_*$, together with (1.3), Lemma 3.3 of [27] implies that there exist constants a and $b > 0$ such that,

$$\phi_n(x) \leq a - b|x|, \ \forall n \geq 1, x \in \mathcal{R}. \tag{1.4}$$

Let $C = \{x \in \mathcal{R} :\liminf_{n\to\infty} \phi_n(x) > -\infty\} \supseteq$ interior(csupp(Q)) and $\bar{\phi}(x) = a - b|x|$, using Lemma 4.2 of [9], together with (1.3) and (2.8) we can conclude that there exist $\phi_0 \in \Phi^d$ and a subsequence ϕ_{n_k} such that $C \subseteq \text{dom}(\phi_0) \subseteq \text{csupp}(Q)$ and,

$$\begin{aligned}\limsup_{k\to\infty} \phi_{n_k}(x) &\leq \phi_0(x) \leq a - b|x|, \ \forall x \in \mathcal{R},\\ \lim_{k\to\infty} \phi_{n_k}(x) &= \phi_0(x) > -\infty, \ \forall x \in \text{interior}(\text{csupp}(Q)).\end{aligned}$$

Since $\text{dom}(\phi_{n_k}) \subseteq \text{csupp}(Q)$, we have ϕ_{n_k} converges to ϕ_0 almost everywhere as the Lebesgue measure of the boundary of csupp(Q) is zero, then we can conclude $\int e^{\phi_0(x)}dx = 1$ by dominated convergence. Thus, $\phi_0 \in \Phi(Q)$. Next, we apply Fatou's Lemma to the nonnegative functions $x \mapsto \int \log(m(x) + 1)Q(dx) + a - b|x| - \log((1 - p_{n_k})f_0 + p_{n_k}e^{\phi_{n_k}})$, and we get,

$$\limsup_{k\to\infty} L(p_{n_k}, \phi_{n_k}; Q) \leq L(p_0, \phi_0; Q).$$

Hence,

$$L(Q) \geq L(p_0, \phi_0; Q) \geq \limsup_{k\to\infty} L(p_{n_k}, \phi_{n_k}; Q) = L(Q),$$

which shows (p_0, ϕ_0) is the maximizer that we are looking for.

□

Proof of Theorem 2.4.4. Since $\lim_{n\to\infty} D_k(Q_n, Q) \to 0$, hence

$$Q_n \to_w Q \text{ and} \int |x|^k Q_n(dx) \to \int |x|^k Q(dx), \text{ as } n \to \infty.$$

Suppose $\limsup_{n\to\infty} L(Q_n) = \lambda \in [-\infty, \infty]$, thus there exist a subsequence $\{Q_{n_k}\}$, such that $L(Q_{n_k}) \to \lambda$. If we let $h(x) = -b_0|x| - \log(\int e^{-b_0|x|}dx)$ as we did in the proof of Theorem 2.4.3, then, $h \in \tilde{\Phi}$, and for any $p > 0$,

$$\begin{aligned}
\lambda &\geq \operatorname*{limsup}_{k\to\infty} L(p, h; Q_{n_k}) = \operatorname*{limsup}_{k\to\infty} \int \log((1-p)f_0 + pe^h)dQ_{n_k} \\
&\geq \log p - b_0 \int |x|Q(dx) - \log(\int e^{-b_0|x|}dx) > -\infty.
\end{aligned}$$

Note that in the above inequalities, we used the fact that $\lim_{n\to\infty} \int |x|Q_n(dx) = \int |x|Q(dx)$ by Lemma 4.6 of [27].

Let $M_n = \max_{x\in R^d}\phi_n(x)$. Since $\lim_{n\to\infty} \int \log(m(x)+1)Q_n(dx) = \int \log(m(x)+1)Q(dx)$ by Lemma 4.6 of [27], similar to the proof of Theorem 2.4.3, one can show that for n sufficiently large, we have $L(p_n, \phi_n; Q_n) \to -\infty$, if $M_n \to \infty$ as $n \to \infty$, and $L(p_n, \phi_n; Q_n) \leq \int \log(m(x)+1)Q(dx) + M_n$, provided that

$$\operatorname*{limsup}_{n\to\infty} Q_n(C_n) < 1, \text{ for any } \{C_n : C_n \subseteq \mathcal{R} \text{ closed and convex, } \lim_{n\to\infty} \text{Leb}(C_n) = 0\}. \quad (1.5)$$

Hence there exist some suitable constants M_0 and M_*, such that $M_0 < M_{n_k} < M_*$ for k sufficiently large and thus $\lambda < \infty$.

Here we explain how (1.5) is derived. As in the proof of Lemma 2.1 of Schuhmacher, [27], there exist a simplex $\tilde{\Delta} = \text{conv}(\tilde{x}_0, \cdots, \tilde{x}_d)$ with positive Lebesgue measure and open sets U_0, U_1, $\cdots$, U_d with $Q(U_j) \geq \eta > 0$, for $0 \leq j \leq d$, here $\eta = \min_{0\leq j\leq d} Q(U_j) > 0$. For any convex and closed set C with $C \cap U_j \neq \emptyset$ for all j, we have $\tilde{\Delta} \subseteq C$. By Theorem 4.4.4 of [3], $\operatorname*{liminf}_{n\to\infty} Q_n(U_j) \geq Q(U_j) \geq \eta$ for all j. Thus if $\lim_{n\to\infty} \text{Leb}(C_n) = 0$, then for any n sufficiently large, $\text{Leb}(C_n) < \text{Leb}(\tilde{\Delta}), \Rightarrow \tilde{\Delta} \nsubseteq C_n, \Rightarrow$ there exist some j, such that

$C_n \cap U_j = \emptyset, \Rightarrow Q_n(C_n) \leq 1 - Q_n(U_j) \leq 1 - \min_{1\leq j\leq d} Q_n(U_j)$. Since,

$$\begin{aligned} Q_n(U_j) &= \liminf_{n\to\infty} Q_n(U_j) + Q_n(U_j) - \liminf_{n\to\infty} Q_n(U_j) \\ &\geq \eta + \inf_{k\geq n} Q_k(U_j) - \liminf_{n\to\infty} Q_n(U_j) = \eta + o(1), \end{aligned}$$

thus $\min_{1\leq j\leq d} Q_n(U_j) \geq \eta + o(1)$, which shows that $Q_n(C_n) \leq 1 - \eta + o(1)$, and hence (1.5) is established.

Now that we know M_{n_k} is bounded for k sufficiently large, and $L(p_{n_k}, \phi_{n_k}; Q_{n_k}) \to \lambda \in R$ as $k \to \infty$, we may assume $\{p_{n_k}\}$, is a convergent sequence, say $p_{n_k} \to p_* \in [0,1]$, as $k \to \infty$. For if $\{p_{n_k}\}$ is not convergent, since it is bounded, it must have a convergent subsequence $\{p_{n_{k_l}}\}$, and the sequence $\{p_{n_{k_l}}, \phi_{n_{k_l}}\}$ would satisfy all those properties above and we can just simply replace the original sequence with this subsequence.

Again, as in the proof of Theorem 2.4.3, for any $x_0 \in$ interior(csupp(Q)), we have,

$$\phi_{n_k}(x_0) \geq \frac{L(p_{n_k}, \phi_{n_k}; Q_{n_k}) - \int \log(m(x)+1)Q(dx) - \max(M_{n_k}, 0)}{1 - h(Q_{n_k}, x_0)}.$$

As Lemma 2.13 of [27] states that $\limsup_{n\to\infty} h(Q_{n_k}, x) \leq h(Q, x)$ for any $x \in \mathcal{R}$, we have,

$$\liminf_{k\to\infty} (\phi_{n_k}(x_0)) \geq \frac{\lambda - \int \log(m(x)+1)Q(dx) - \max(M_*, 0)}{1 - h(Q, x_0)} > -\infty.$$

Hence, for k large enough,

$$\inf_{l\geq k} \phi_{n_l}(x_0) > -\infty, \ \forall x_0 \in \text{interior}(\text{csupp}(Q)). \tag{1.6}$$

Again, we can deduce from (1.6) and the boundedness of M_{n_k} that there exist constants a and $b > 0$ such that,

$$\phi_{n_k}(x) \leq a - b|x|, \ \forall k \text{ sufficiently large}, \ x \in R^d. \tag{1.7}$$

Similar as before we conclude that there exist $\phi_* \in \Phi^d$ and a subsequence $\{\phi_{n_{k_l}}\}$ such that interior(csupp(Q)) $\subseteq$ dom(ϕ_*) $\subseteq$ csupp(Q) and,

$$\begin{aligned}
\operatorname*{limsup}_{l\to\infty, x\to y} \phi_{n_{k_l}}(x) &\leq \phi_*(y) \leq a - b|y|, \ \forall y \in \mathcal{R}, \\
\lim_{l\to\infty, x\to y} \phi_{n_{k_l}}(x) &= \phi_*(y) > -\infty, \ \forall y \in \text{interior}(\text{csupp}(Q)).
\end{aligned}$$

Then $\int e^{\phi_*(x)}dx = 1$ by dominated convergence, which implies that $\phi_* \in \tilde{\Phi}$.

By Skorohod's theorem, there exist a probability space $(\Omega, \mathcal{F}, P)$ and random variables $X_{n_{k_l}} \sim Q_{n_{k_l}}$, $X \sim Q$, such that $\lim_{l\to\infty} X_n = X$ almost surely. Let $H_{n_{k_l}} = \int \log(m(x)+1)Q(dx) + a - b||X_{n_{k_l}}|| - \log\{(1-p_{n_{k_l}})f_0(X_{n_{k_l}}) + p_{n_{k_l}}\exp(\phi_{n_{k_l}}(X_{n_{k_l}}))\}$. By

Fatou's Lemma, we have,

$$\begin{aligned}
\lambda &= \lim_{l\to\infty} L(p_{n_{k_l}}, \phi_{n_{k_l}}; Q_{n_{k_l}}) = \lim_{l\to\infty} \int \log((1-p_{n_{k_l}})f_0 + p_{n_{k_l}} e^{\phi_{n_{k_l}}}) dQ_{n_{k_l}} \\
&= \lim_{l\to\infty} \{\int \log(m(x)+1)Q(dx) + \int (a - b||x||)Q_{n_{k_l}}(dx) - E(H_{n_{k_l}})\} \\
&= \int \log(m(x)+1)Q(dx) + a - b\int ||x||Q(dx) - \liminf_{l\to\infty} E(H_{n_{k_l}}) \\
&\le \int \log(m(x)+1)Q(dx) + a - b\int ||x||Q(dx) - E(\liminf_{l\to\infty} (H_{n_{k_l}})) \\
&\le E\{\limsup_{l\to\infty} \log((1-p_{n_{k_l}})f_0(X_{n_{k_l}}) + p_{n_{k_l}} \exp(\phi_{n_{k_l}}(X_{n_{k_l}})))\} \\
&\le E(\log((1-p_*)f_0(X) + p_*\exp(\phi_*(X)))) \\
&\le L(Q).
\end{aligned}$$

In order to show that $\lambda \ge L(Q)$, we use the approximations $\phi^* \le \phi^{*(\epsilon)} \le \phi^{*(1)}$, $0 < \epsilon \le 1$ from Lemma 4.4 of [9], since $\phi^{*(\epsilon)} \in \Phi$ is Lipschitz continuous, one can show that $\frac{|\phi^{*(\epsilon)}|}{1+||x||}$ is bounded, and hence by Lemma 4.6 of [9], we have,

$$\begin{aligned}
\lambda &= \lim_{k\to\infty} L(p_{n_k}, \phi_{n_k}; Q_{n_k}) \\
&\ge \lim_{k\to\infty} L(p^*, \phi^{*(\epsilon)} - \log(\int e^{\phi^{*(\epsilon)}(x)}dx), Q_{n_k}) \\
&= L(p^*, \phi^{*(\epsilon)} - \log(\int e^{\phi^{*(\epsilon)}(x)}dx), Q) \\
&= \int \log\{(1-p^*)f_0 \int e^{\phi^{*(\epsilon)}(x)}dx + p^* e^{\phi^{*(\epsilon)}}\}dQ - \log(\int e^{\phi^{*(\epsilon)}(x)}dx) \\
&\to \int \log((1-p^*)f_0 + p^* e^{\phi^*})dQ = L(p^*, \phi^*; Q), \text{ as } \epsilon \to 0.
\end{aligned}$$

The last step above is by applying dominated convergence on $e^{\phi^{*(\epsilon)}}$ and monotone convergence on $(1-p^*)f_0\int e^{\phi^{*(1)}(x)}dx + p^*e^{\phi^{*(1)}} - (1-p^*)f_0\int e^{\phi^{*(\epsilon)}(x)}dx - p^*e^{\phi^{*(\epsilon)}}$. Thus we have shown that $\lambda = L(Q)$, and $(p^*, \phi^*) = (p_*, \phi_*)$ is the unique maximizer.

With exactly the same argument, we can show that $\operatorname*{liminf}_{n\to\infty} L(Q_n) = L(Q)$ as well, and hence $L(Q_n) \to L(Q)$, as $n \to \infty$.

Also, if we let $f^* = \exp \circ \phi^*$, $f_n = \exp \circ \phi_n$, we have shown that,

$$\begin{aligned}
\lim_{l\to\infty} p_{n_{k_l}} &= p^*, \\
\operatorname*{limsup}_{l\to\infty, x\to y} f_{n_{k_l}}(x) &\leq f^*(y),\ \forall y \in \partial\{f^* > 0\}, \\
\lim_{l\to\infty, x\to y} f_{n_{k_l}}(x) &= f^*(y),\ \forall y \in R^d \setminus \partial\{f^* > 0\}.
\end{aligned}$$

In particular, $\{f_{n_{k_l}}\}$ converges to f^* almost everywhere w.r.t. Lebesgue measure and hence $\int |f_{n_{k_l}}(x) - f^*(x)|dx \to 0$, as $l \to \infty$ by dominated convergence. Our proof actually shows that for any subsequence of $\{Q_n\}$, we can further find a subsequence with the above convergence properties. That means the original sequence must satisfy those properties as well, otherwise we would arrive at contradictions and that completes the proof.

□

1.7.2 More Simulation Result

Table 1.10: Bias(MSE) of estimates of p/μ and mean of the classification error for model 1 when $n = 250$.

	EM_logconcave	Patra	Bordes	Song EM	Song max π	Xiang
$p = 0.2$						
p	0.008(0.0019)	-0.001(0.0021)	0.018(0.0046)	-0.071(0.0057)	0.106(0.0160)	0.021(0.0056)
μ	0.057(0.0738)	-0.396(0.3286)	-0.166(0.2109)	-0.108(0.2243)	-0.846(0.9437)	0.243(0.1864)
MCE	0.1029	0.1094	0.1097	0.1138	0.1104	0.1058
$p = 0.5$						
p	0.000(0.0023)	-0.041(0.0036)	0.005(0.0025)	-0.130(0.0185)	0.100(0.0153)	0.014(0.0026)
μ	0.017(0.0225)	0.023(0.0167)	-0.021(0.0198)	0.143(0.0393)	-0.344(0.1635)	0.032(0.0208)
MCE	0.1151	0.1259	0.1232	0.1379	0.1248	0.1138
$p = 0.8$						
p	-0.001(0.0011)	-0.070(0.0057)	-0.001(0.0014)	-0.104(0.0123)	0.056(0.0040)	0.016(0.0014)
μ	0.003(0.0072)	0.059(0.0102)	-0.001(0.0085)	0.097(0.0158)	-0.147(0.0323)	-0.008(0.0079)
MCE	0.0670	0.0781	0.0722	0.0835	0.0752	0.0703

Table 1.11: Bias(MSE) of estimates of p/μ and mean of the classification error for model 1 when $n = 500$.

	EM_logconcave	Patra	Bordes	Song EM	Song max π	Xiang
$p = 0.2$						
p	0.000(0.0008)	-0.008(0.0014)	0.003(0.0021)	-0.077(0.0063)	0.086(0.0102)	0.011(0.0013)
μ	0.059(0.0348)	-0.258(0.1430)	-0.054(0.1013)	0.001(0.0734)	-0.738(0.6525)	0.175(0.0843)
MCE	0.0972	0.1070	0.1060	0.1106	0.1051	0.0990
$p = 0.5$						
p	-0.003(0.0009)	-0.031(0.0021)	0.000(0.0012)	-0.132(0.0181)	0.107(0.0158)	0.011(0.0011)
μ	0.019(0.0097)	0.035(0.0109)	-0.003(0.0098)	0.169(0.0363)	-0.346(0.1605)	0.020(0.0097)
MCE	0.1111	0.1239	0.1209	0.1359	0.1226	0.1120
$p = 0.8$						
p	0.003(0.0006)	-0.053(0.0033)	0.001(0.0007)	-0.104(0.0117)	0.056(0.0040)	0.014(0.0006)
μ	-0.001(0.0032)	0.065(0.0073)	0.000(0.0041)	0.110(0.0155)	-0.121(0.0220)	-0.007(0.0040)
MCE	0.0644	0.0758	0.0693	0.0822	0.0711	0.0685

Table 1.12: Bias(MSE) of estimates of p/μ and mean of the classification aerror for model 2 when $n = 250$.

	EM_logconcave	Patra	Bordes	Song EM	Song max π	Xiang
$p = 0.2$						
p	0.004(0.0051)	-0.021(0.0033)	-0.006(0.0036)	-0.156(0.0248)	0.371(0.1443)	0.056(0.0087)
μ	-0.022(0.0038)	0.061(0.0073)	-0.019(0.0029)	0.013(0.0032)	0.197(0.0401)	-0.011(0.0026)
MCE	0.1368	0.1554	0.1568	0.1746	0.1858	0.1437
$p = 0.5$						
p	0.004(0.0043)	-0.064(0.0071)	-0.037(0.0041)	-0.300(0.0916)	0.230(0.0576)	-0.013(0.0041)
μ	-0.005(0.0007)	-0.004(0.0004)	-0.032(0.0013)	-0.034(0.0013)	0.080(0.0070)	-0.014(0.0011)
MCE	0.1678	0.2110	0.2031	0.2778	0.1964	0.1764
$p = 0.8$						
p	0.009(0.0019)	-0.105(0.0124)	-0.065(0.0057)	-0.312(0.1001)	0.081(0.0078)	-0.049(0.0056)
μ	0.000(0.0002)	-0.020(0.0005)	-0.033(0.0012)	-0.039(0.0016)	0.010(0.0006)	-0.018(0.0010)
MCE	0.1030	0.1384	0.1266	0.2137	0.1124	0.1140

Table 1.13: Bias(MSE) of estimates of p/μ and mean of the classification error for model 2 when $n = 500$.

	EM_logconcave	Patra	Bordes	Song EM	Song max π	Xiang
$p = 0.2$						
p	-0.001(0.0028)	-0.021(0.0020)	-0.007(0.0022)	-0.154(0.0238)	0.379(0.1496)	0.035(0.0047)
μ	-0.019(0.0024)	0.041(0.0030)	-0.025(0.0022)	-0.006(0.0008)	0.197(0.0393)	-0.009(0.0018)
MCE	0.1294	0.1544	0.1520	0.1697	0.1862	0.1369
$p = 0.5$						
p	0.002(0.0022)	-0.053(0.0045)	-0.039(0.0032)	-0.292(0.0860)	0.234(0.0583)	-0.034(0.0035)
μ	-0.004(0.0003)	-0.008(0.0002)	-0.033(0.0013)	-0.037(0.0014)	0.080(0.0069)	-0.014(0.0009)
MCE	0.1638	0.2031	0.2011	0.2723	0.1940	0.1753
$p = 0.8$						
p	0.003(0.0010)	-0.086(0.0081)	-0.070(0.0058)	-0.312(0.0990)	0.097(0.0102)	-0.048(0.0038)
μ	-0.001(0.0001)	-0.020(0.0005)	-0.034(0.0012)	-0.040(0.0016)	0.024(0.0007)	-0.022(0.0010)
MCE	0.1001	0.1307	0.1263	0.2119	0.1129	0.1139

Table 1.14: Bias(MSE) of estimates of p/μ and mean of the classification error for model 3 when $n = 250$.

	EM_logconcave	Patra	Bordes	Song EM	Song max π	Xiang
$p = 0.2$						
p	0.005(0.0011)	0.009(0.0021)	NA	-0.050(0.0034)	0.431(0.1889)	0.048(0.0042)
μ	0.026(0.0400)	-0.041(0.0581)	NA	0.048(0.0591)	-1.115(1.2677)	-0.139(0.0493)
MCE	0.0737	0.0869	NA	0.0895	0.1718	0.0842
$p = 0.5$						
p	0.002(0.0013)	-0.019(0.0019)	NA	-0.069(0.0065)	0.269(0.0742)	0.081(0.0103)
μ	-0.001(0.0096)	-0.001(0.0126)	NA	0.013(0.0120)	-0.495(0.2618)	-0.174(0.0590)
MCE	0.0623	0.0806	NA	0.0839	0.1271	0.0860
$p = 0.8$						
p	0.003(0.0007)	-0.046(0.0029)	NA	-0.225(0.0017)	0.107(0.0122)	0.087(0.0096)
μ	0.001(0.0052)	0.003(0.0061)	NA	-0.004(0.0057)	-0.157(0.0316)	-0.150(0.0446)
MCE	0.0274	0.0351	NA	0.0354	0.0589	0.0734

Table 1.15: Bias(MSE) of estimates of p/μ and mean of the classification error for model 3 when $n = 500$.

	EM_logconcave	Patra	Bordes	Song EM	Song max π	Xiang
$p = 0.2$						
p	0.004(0.0005)	0.004(0.0011)	NA	-0.055(0.0034)	0.415(0.1746)	0.038(0.0024)
μ	0.014(0.0204)	-0.039(0.0316)	NA	0.047(0.0315)	-1.119(1.2657)	-0.131(0.0357)
MCE	0.0722	0.0862	NA	0.0855	0.1610	0.0819
$p = 0.5$						
p	0.001(0.0005)	-0.016(0.0010)	NA	-0.070(0.0057)	0.260(0.0692)	0.060(0.0059)
μ	0.004(0.0047)	-0.007(0.0061)	NA	0.017(0.0064)	-0.489(0.2475)	-0.126(0.0338)
MCE	0.0604	0.0787	NA	0.0811	0.1189	0.0790
$p = 0.8$						
p	0.002(0.0003)	-0.036(0.0017)	NA	-0.029(0.0013)	0.106(0.0115)	0.080(0.0078)
μ	0.001(0.0027)	0.001(0.0026)	NA	-0.002(0.0030)	-0.159(0.0294)	-0.117(0.0284)
MCE	0.0270	0.0334	NA	0.0341	0.0557	0.0677

Table 1.16: Bias(MSE) of estimates of p/μ and mean of the classification error for model 4 when $n = 250$.

	EM_logconcave	Patra	Bordes	Song EM	Song max π	Xiang
$p = 0.2$						
p	-0.001(0.0009)	0.019(0.0021)	NA	0.019(0.0013)	0.129(0.0198)	0.113(0.0179)
μ	0.016(0.1454)	-0.574(0.5652)	NA	-0.569(0.5268)	-1.296(2.0126)	-0.728(1.0325)
MCE	0.0128	0.0168	NA	0.0168	0.0247	0.0438
$p = 0.5$						
p	0.003(0.0009)	-0.014(0.0014)	NA	0.022(0.0014)	0.085(0.0089)	0.086(0.0101)
μ	0.025(0.0493)	-0.192(0.0898)	NA	-0.200(0.0878)	-0.379(0.2136)	-0.605(0.6314)
MCE	0.0096	0.0184	NA	0.0188	0.0204	0.0378
$p = 0.8$						
p	0.000(0.0006)	-0.039(0.0022)	NA	0.013(0.0007)	0.044(0.0026)	0.077(0.0076)
μ	0.009(0.0279)	-0.024(0.0247)	NA	-0.073(0.0332)	-0.157(0.0572)	-0.621(0.4861)
MCE	0.0044	0.0093	NA	0.0115	0.0149	0.0468

Table 1.17: Bias(MSE) of estimates of p/μ and mean of the classification error for model 4 when $n = 500$.

	EM_logconcave	Patra	Bordes	Song EM	Song max π	Xiang
$p = 0.2$						
p	-0.002(0.0003)	0.009(0.0009)	NA	0.010(0.0005)	0.108(0.0141)	0.074(0.0073)
μ	-0.008(0.0721)	-0.410(0.2935)	NA	-0.404(0.2668)	-1.109(1.4363)	-0.803(1.1190)
MCE	0.0116	0.0151	NA	0.0147	0.0201	0.0271
$p = 0.5$						
p	0.000(0.0005)	-0.011(0.0008)	NA	0.017(0.0009)	0.074(0.0066)	0.069(0.0062)
μ	-0.011(0.0244)	-0.169(0.0525)	NA	-0.220(0.0720)	-0.374(0.1801)	-0.607(0.6089)
MCE	0.0095	0.0167	NA	0.0178	0.0178	0.0288
$p = 0.8$						
p	0.002(0.0004)	-0.031(0.0014)	NA	0.011(0.0005)	0.042(0.0023)	0.068(0.0054)
μ	-0.007(0.0137)	-0.034(0.0151)	NA	-0.072(0.0190)	-0.159(0.0447)	-0.699(0.5384)
MCE	0.0048	0.0082	NA	0.0103	0.0126	0.0360

Table 1.18: Bias(MSE) of estimates of p/μ and mean of the classification error for model 5 when $n = 250$.

	EM_logconcave	Patra	Bordes	Song EM	Song max π	Xiang
$p = 0.2$						
p	-0.001(0.0007)	0.020(0.0022)	NA	0.030(0.0018)	0.142(0.0240)	0.047(0.0036)
μ	-0.026(0.0943)	-0.679(0.6177)	NA	-0.716(0.6592)	-1.449(2.3985)	-0.895(0.9545)
MCE	0.0017	0.0090	NA	0.0087	0.0185	0.0127
$p = 0.5$						
p	0.001(0.0010)	-0.015(0.0014)	NA	0.031(0.0019)	0.116(0.0162)	0.074(0.0075)
μ	0.009(0.0288)	-0.208(0.0734)	NA	-0.267(0.1017)	-0.613(0.4915)	-0.680(0.5574)
MCE	0.0011	0.0114	NA	0.0136	0.0202	0.0248
$p = 0.8$						
p	0.000(0.0006)	-0.042(0.0025)	NA	0.015(0.0008)	0.069(0.0056)	0.089(0.0096)
μ	0.006(0.0216)	-0.036(0.0232)	NA	-0.089(0.0305)	-0.296(0.1188)	-0.488(0.3336)
MCE	0.0004	0.0037	NA	0.0055	0.0188	0.0544

Table 1.19: Bias(MSE) of estimates of p/μ and mean of the classification error for model 5 when $n = 500$.

	EM_logconcave	Patra	Bordes	Song EM	Song max π	Xiang
$p = 0.2$						
p	-0.001(0.0003)	0.012(0.0011)	NA	0.023(0.0009)	0.119(0.0176)	0.042(0.0025)
μ	-0.007(0.0438)	-0.500(0.3260)	NA	-0.553(0.3655)	1.237(1.7273)	-0.928(0.9531)
MCE	0.0014	0.0058	NA	0.0061	0.0132	0.0107
$p = 0.5$						
p	-0.001(0.0006)	-0.010(0.0007)	NA	0.025(0.0012)	0.120(0.0179)	0.077(0.0073)
μ	0.014(0.0170)	-0.194(0.0565)	NA	-0.233(0.0722)	-0.630(0.4822)	-0.726(0.5840)
MCE	0.0008	0.0094	NA	0.0111	0.0188	0.0240
$p = 0.8$						
p	0.001(0.0003)	-0.031(0.0013)	NA	0.013(0.0004)	0.078(0.0067)	0.088(0.0087)
μ	0.006(0.0094)	-0.020(0.0105)	NA	-0.065(0.0146)	-0.324(0.1188)	-0.506(0.3185)
MCE	0.0003	0.0026	NA	0.0039	0.0199	0.0501

Table 1.20: Bias(MSE) of estimates of p/μ and mean of the classification error for model 6 when $n = 250$.

	EM_logconcave	Patra	Bordes	Song EM	Song max π	Xiang
$p = 0.2$						
p	-0.006(0.0015)	0.003(0.0021)	0.020(0.0036)	-0.012(0.0009)	0.109(0.0147)	0.027(0.0023)
μ	0.127(0.1230)	-0.200(0.1172)	-0.685(0.8830)	-0.118(0.0777)	-0.654(0.5415)	-0.067(0.0877)
MCE	0.0464	0.0468	0.1738	0.0470	0.0509	0.0473
$p = 0.5$						
p	-0.015(0.0028)	-0.044(0.0037)	0.000(0.0024)	-0.045(0.0031)	0.057(0.0054)	0.031(0.0061)
μ	0.073(0.0525)	0.123(0.0325)	-0.009(0.0298)	0.127(0.0329)	-0.099(0.0405)	-0.068(0.0676)
MCE	0.0688	0.0682	0.0676	0.0688	0.0683	0.0738
$p = 0.8$						
p	0.006(0.0016)	-0.077(0.0067)	-0.003(0.0016)	-0.059(0.0044)	0.011(0.0012)	0.006(0.0013)
μ	-0.024(0.0205)	0.173(0.0369)	0.004(0.0104)	0.155(0.0320)	0.032(0.0125)	-0.002(0.0105)
MCE	0.0591	0.0660	0.0595	0.0647	0.0588	0.0572

Table 1.21: Bias(MSE) of estimates of p/μ and mean of the classification error for model 6 when $n = 500$.

	EM_logconcave	Patra	Bordes	Song EM	Song max π	Xiang
$p = 0.2$						
p	-0.008(0.0006)	-0.001(0.0008)	0.016(0.0021)	-0.017(0.0007)	0.095(0.0119)	0.011(0.0007)
μ	0.132(0.0565)	-0.066(0.0452)	-0.161(0.1613)	0.000(0.0312)	-0.538(0.3834)	-0.025(0.0370)
MCE	0.0435	0.0457	0.0451	0.0450	0.0479	0.0439
$p = 0.5$						
p	-0.011(0.0014)	-0.033(0.0019)	-0.001(0.0010)	-0.049(0.0030)	0.043(0.0032)	0.011(0.0017)
μ	0.069(0.0304)	0.137(0.0272)	-0.004(0.0109)	0.161(0.0339)	-0.038(0.0183)	-0.023(0.0203)
MCE	0.0655	0.0676	0.0666	0.0684	0.0665	0.0664
$p = 0.8$						
p	0.004(0.0005)	-0.065(0.0047)	-0.002(0.0008)	-0.062(0.0043)	0.010(0.0011)	0.002(0.0007)
μ	-0.021(0.0069)	0.169(0.0325)	-0.002(0.0054)	0.165(0.0311)	0.043(0.0102)	0.002(0.0060)
MCE	0.0541	0.0646	0.0576	0.0642	0.0574	0.0546

Table 1.22: Bias(MSE) of estimates of p/μ and mean of the classification error for model 7 when $n = 250$.

	EM_logconcave	Patra	Bordes	Song EM	Song max π	Xiang
$p = 0.2$						
p	-0.004(0.0007)	0.016(0.0021)	-0.002(0.0011)	0.025(0.0015)	0.163(0.033)	0.0273(0.0018)
μ	0.011(0.0301)	-0.782(0.7361)	-0.043(0.0697)	-0.822(0.78)	-1.695(3.1805)	-0.0248(0.0225)
MCE	0.0133	0.0184	0.0174	0.0183	0.0338	0.0169
$p = 0.5$						
p	0(0.001)	-0.015(0.0017)	0.002(0.0011)	0.028(0.0018)	0.115(0.016)	0.0263(0.0018)
μ	0.013(0.008)	-0.242(0.0743)	-0.008(0.0089)	-0.303(0.1082)	-0.63(0.4636)	-0.0261(0.0083)
MCE	0.0132	0.0173	0.0176	0.0189	0.0284	0.0181
$p = 0.8$						
p	-0.0080(0.0008)	-0.037(0.002)	-0.003(0.0008)	0.01(0.0009)	0.059(0.0043)	0.0175(0.0010)
μ	0.022(0.0044)	-0.03(0.0061)	0.002(0.0038)	-0.102(0.0159)	-0.264(0.0828)	-0.0080(0.0036)
MCE	0.0104	0.0106	0.0118	0.0126	0.0221	0.0144

Table 1.23: Bias(MSE) of estimates of p/μ and mean of the classification error for model 7 when $n = 500$.

	EM_logconcave	Patra	Bordes	Song EM	Song max π	Xiang
$p = 0.2$						
p	0.000(0.0003)	0.012(0.0011)	0.000(0.0005)	0.020(0.0008)	0.173(0.038)	0.0196(0.0010)
μ	0.009(0.0102)	-0.566(0.3799)	-0.01(0.0237)	-0.625(0.4484)	-1.662(3.1148)	-0.0403(0.0127)
MCE	0.0129	0.0157	0.0158	0.0163	0.0338	0.0156
$p = 0.5$						
p	-0.001(0.0005)	-0.011(0.0009)	0.002(0.0006)	0.022(0.001)	0.121(0.0189)	0.0155(0.0007)
μ	0.021(0.0038)	-0.188(0.043)	-0.002(0.0032)	-0.243(0.0675)	-0.637(0.5017)	-0.0083(0.0038)
MCE	0.0128	0.0155	0.0159	0.0168	0.0301	0.0159
$p = 0.8$						
p	-0.005(0.0004)	-0.031(0.0014)	0.000(0.0004)	0.007(0.0004)	0.061(0.0046)	0.0129(0.0004)
μ	0.02(0.0021)	-0.023(0.003)	-0.001(0.0017)	-0.071(0.0076)	-0.249(0.0748)	-0.0034(0.0017)
MCE	0.0101	0.0096	0.0106	0.0109	0.0204	0.0119

1.7.3 Source code

EM_logconcave algorithm, R code

```
library(logcondens)##package for univariate log-concave density estimation
library(ks)##package for Kernal density estimation
########################################
##custom kmeans with one center fixed##
########################################
kmeans1<-function(x,center_fix){
  n<-length(x)
  c1<-center_fix
  c2<-mean(x)
  cluster<-numeric(n)##assign points closer to the fixed center as cluster
      ↪ 1
  for(i in 1:n){
    if(abs(x[i]-c1)<abs(x[i]-c2)){
      cluster[i]<-1
    }else{
      cluster[i]<-2
    }
  }
  c2<-mean(x[which(cluster==2)])
  clusterold<-rep(1,n)
  while(sum(cluster!=clusterold)!=0){
    clusterold<-cluster
    for(i in 1:n){
      if(abs(x[i]-c1)<abs(x[i]-c2)){
        cluster[i]<-1
      }else{
        cluster[i]<-2
      }
    }
    c2<-mean(x[which(cluster==2)])
  }
  res<-list(centers=c(c1,c2),cluster=cluster,size=c(sum(cluster==1),sum(cluster==2)))
  return(res)
  }


#################################################
## EM algorithm+log concave density estimation ##
#################################################
##a:density from the known component; ini_pi: true mixing proportion; ini_b: true density from the unknown component.
mle_logcon<-function(x,a,center_known,ini_pi,ini_b){
```

```
##using estimated initial value from kmeans

true<-0

n<-length(x)

fit1<-kmeans1(x,center_fix=center_known)

pi<-(fit1$size[1])/n

fit2<-activeSetLogCon(x=x[which(fit1$cluster==2)])

b<-evaluateLogConDens(xs=x,res=fit2)[,3]

l<-sum(log(pi*a+(1-pi)*b))

lold<-l-1

while((l-lold)/abs(lold)>10^-4){

  lold<-l

  p<-pi*a/(pi*a+(1-pi)*b)##updated probabilities

  pi<-sum(p)/n##updated mixing proportion

  weight=(1-p)/sum(1-p)

  x1<-cbind(x,weight)

  x1<-x1[x1[,2]>10^-4,]##delete points with weight<=10^-4

  x1<-x1[order(x1[,1]),]

  fit2<-activeSetLogCon(x=x1[,1],w=x1[,2])

  b<-evaluateLogConDens(xs=x,res=fit2)[,3]

  l<-sum(log(pi*a+(1-pi)*b))

}
```

```
mu_loc<-sum((fit2$x)*(fit2$w))/sum(fit2$w)##use weighted sum to calculate
    ↪  the mean for the unknown component
res<-list(p=p,pi=pi,mu=mu_loc,x=fit2$x,phi=fit2$phi,usetrueini=true)

##use the true initial value
pi1<-ini_pi
b1<-ini_b
l1<-sum(log(pi1*a+(1-pi1)*b1))
lold1<-l1-1
while((l1-lold1)/abs(lold1)>10^-4){
  lold1<-l1
  p<-pi1*a/(pi1*a+(1-pi1)*b1)##updated probabilities
  pi1<-sum(p)/n##updated mixing proportion
  weight=(1-p)/sum(1-p)
  x1<-cbind(x,weight)
  x1<-x1[x1[,2]>10^-4,]##delete points with weight<=10^-4
  x1<-x1[order(x1[,1]),]
  fit2<-activeSetLogCon(x=x1[,1],w=x1[,2])
  b1<-evaluateLogConDens(xs=x,res=fit2)[,3]
  l1<-sum(log(pi1*a+(1-pi1)*b1))
}
##select the maximum likelihood fit
```

```
  if((l1>l)&&(abs((l1-l)/l)>0.0002)){

    true<-1

    mu_loc<-sum((fit2$x)*(fit2$w))/sum(fit2$w)

    res<-list(p=p,pi=pi1,mu=mu_loc,x=fit2$x,phi=fit2$phi,usetrueini=true)

  }

  return(res)

}
```

Chapter 2

Robust Maximum Likelihood Estimation Based on Semiparametric Mixture Models

2.1 Introduction

Maximum likelihood estimators are widely used since they have many desirable properties such as consistency and efficiency. However, most of these estimators are very sensitive to outliers and might provide biased or even misleading results when the data are contaminated. Many robust estimators have been proposed to overcome this issue; see for example, [15], [14], [12], [13], [18], [32], [24]. However, most of the above robust estimators focus on the robust estimation of a location parameter and/or require the choice of a tuning parameter, with the exceptions of [13] and [18], which proposed a trimmed likelihood estimation method and weighted likelihood estimation method, respectively.

In this article, we propose a new class of robust maximum likelihood estimator by fitting a semiparametric mixture model to the contaminated data,

$$g(x) = (1-p)f_0(x;\boldsymbol{\theta}) + pf(x), \tag{2.1}$$

where f_0 is a known assumed density function with unknown parameters $\boldsymbol{\theta} \in \Theta$, $p \in [0,1]$ is the proportion of possible contaminated data/outliers, and $f(x)$ represents the unknown density for the contaminated component. The above contaminated mixture model is commonly used in the literature of robust statistics to describe the situation when there is violation/departure of the assumed model. Our goal is to find a robust estimation of $\boldsymbol{\theta}$ despite possible contamination from the unknown density f. By estimating the semiparametric mixture model (2.1) directly, we can not only estimate the parameter $\boldsymbol{\theta}$ robuslty but also recover the density of the contaminated component. In addition, based on the new model, we can also assign a probability of each observation being an outlier. We propose two methods to estimate the semiparametric mixture model (2.1). The first estimator is an extension of the method proposed by [21] which assumes the first component is completely known without unknown parameter $\boldsymbol{\theta}$. For the second estimator, we assume that f is log-concave and then estimate the model (2.1) by maximizing the corresponding the semiparametric maximum likelihood over the unknown parameter $\boldsymbol{\theta}$ and the log-concave density f. One nice of feature of using log-concave density for f is that it can be estimated by nonparametric likelihood estimator without requiring any tuning parameter. For more details of log-concave densities, please refer to [5], [8], [34], [9] and the review of the recent progress in log-concave density estimation by [26]. We further investigate the identifiability conditions of the proposed semiparametric mixture models and propose two innovative algorithms to estimate $\boldsymbol{\theta}$ without assuming a parametric form for the contaminated density $f(x)$. Extensive simulation studies demonstrate that our methods provide comparable performance to traditional MLE whether the data are clean and much better performance when the data contain outliers.

The rest of the paper is organized as follows. Section 2.2 discusses the identifiability problem of our model. Our two algorithms are proposed in Section 2.3. Basic theoretical properties are described in Section 2.4. In Section 2.5, we present our extensive simulation results. We conclude this article with a brief discussion in Section 2.6.

2.2 Identifiability

We first investigate the identifiability conditions of model (2.1). Without any constraints, the model (2.1) is non-identifiable. For example,

$$g(x) = (1-p)f_0(x;\boldsymbol{\theta}) + pf(x) = (1-p-\gamma)f_0(x;\boldsymbol{\theta}) + (p+\gamma)(\frac{\gamma}{p+\gamma}f_0(x;\boldsymbol{\theta}) + \frac{p}{p+\gamma}f(x)),$$

for any $0 < \gamma < (1-p)$.

When $f(x)$ represents the density of outliers, it is reasonable to assume that $f(x)$ achieves small densities in situations where $f_0(x;\boldsymbol{\theta})$ is large. If we restrict $f(x)$ to be 0 on a fixed set, say A, with non-zero measure, then we have the following identifiable result.

Theorem 2.2.1. *Assume $f_0(x;\boldsymbol{\theta})$ is analytic w.r.t. x on $\mathcal{R}$, then model (2.1) is identifiable* *if*

$$f(x) = 0, \forall x \in A, \tag{2.2}$$

where $\mu(A) > 0$ and $\mu(\cdot)$ is a Lebesgue measure in $\mathcal{R}$.

Remark 2.2.1. *A function h is said to be real analytic on $\mathcal{R}$ if it is infinitely differentiable and the Taylor series at any point x_0 converges to $h(x)$ for x in a neighborhood of x_0. For example, the normal density is real analytic.*

In Theorem 2.2, the main identifiability condition of the model (2.1) *is that there exists an interval A from which the observations are not outliers with certainty. Such assumption is reasonable in most of the applications. For example, A can be a very small interval around the median, say* 45% *percentile to* 55% *percentile of the data.*

Remark 2.2.2. *In fact, the condition we give in Theorem 2.2.1 is quite strong, and the result still holds even if the parametric form of f_0 is unknown,. This can be proved with similar arguments as proof of Theorem 2.2.1 (Section 2.7.2).*

Next we establish a local identifiability condition of the model (2.1).

Theorem 2.2.2. *Assume $f_0(x;\boldsymbol{\theta})$ is analytic w.r.t. x on $\mathcal{R}$, differentiable w.r.t. $\boldsymbol{\theta}$ and $\frac{\partial f_0(x;\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}$ is bounded. Then, the model (2.1) is identifiable over a sufficiently small neighborhood of $\boldsymbol{\theta_0}$ if there exists $\alpha \in (0,1)$ and a Lipschitz continuous function $C(\boldsymbol{\theta})$ such that $\int_{\{x: f_0(x;\boldsymbol{\theta}) \geq C(\boldsymbol{\theta})\}} f_0(x;\boldsymbol{\theta})dx = \alpha$ and*

$$f(x) = 0 \text{ when } f_0(x;\boldsymbol{\theta}) \geq C(\boldsymbol{\theta}). \tag{2.3}$$

The main assumption of the local identifiability is that there is zero chance for the outliers to appear in the area where $f_0(x;\boldsymbol{\theta})$ is large.

2.3 Proposed Algorithms

2.3.1 Minimum search

Introduction to [21]'s Algorithm

In [21]'s paper, they considered the model

$$G(x) = (1-p)F_0(x) + F(x),$$

where the CDF F_0 is completely known, p and F ($F \neq F_0$) are unknown. They define

$$p_0 = \inf\{\gamma \in (0,1] : \frac{G-(1-\gamma)F_0}{\gamma} \text{ is a CDF}\}.$$

Intuitively, this definition defines the smallest proportion p_0 such that the "signal" distribution F does not include any background information from the known distribution F_0.

For an i.i.d. random sample $\{X_i\}_{i=1}^n$ from G, let $\mathbb{G}_n$ be the empirical CDF of the random sample. For any $\gamma \in (0,1]$, they define the naive estimator of F to be

$$\hat{F}_n^\gamma = \frac{\mathbb{G}_n - (1-\gamma)F_0}{\gamma}.$$

In order to improve this estimator to be non-decreasing, they propose to minimize

$$\int \{W(x) - \hat{F}_n^\gamma(x)\}^2 d\mathbb{G}_n(x) = \frac{1}{n}\sum_{i=1}\{W(X_i) - \hat{F}_n^\gamma(X_i)\}$$

over all CDFs W, and use $\check{F}_n^\gamma$ to denote the minimizer.

Finally, the estimator of p_0 is defined by

$$\hat{p}_0^{c_n} = \inf\{\gamma \in (0,1] : \gamma d_n(\hat{F}_n^\gamma, \check{F}_n^\gamma) \leq \frac{c_n}{\sqrt{n}}\} = \inf\{\gamma \in (0,1] : d_n(\mathbb{G}_n, (1-\gamma)F_0 + \gamma\check{F}_n^\gamma) \leq \frac{c_n}{\sqrt{n}}\},$$

where d_n represents the $L_2(\mathbb{G}_n)$ distance, $c_n = 0.1\text{loglog}n$ following the recommendation from simulation results of [21].

Proposed estimator: $p_{\min}$

Let G, $F_0(\cdot, \boldsymbol{\theta})$ and F be the corresponding cumulative distribution functions of g, $f_0(\cdot, \boldsymbol{\theta})$ and f, respectively, then, model (2.1) can be written as

$$G(x) = (1-p)F_0(x; \boldsymbol{\theta}) + pF(x).$$

Therefore,

$$F(x) = \frac{G - (1-p)F_0(\cdot\, ; \boldsymbol{\xi})}{p}.$$

Inspired by [21], we define,

$$p_{\min} = \inf_{\boldsymbol{\xi}} \inf\{\gamma \in (0,1] : \frac{G - (1-\gamma)F_0(\cdot\, ; \boldsymbol{\xi})}{\gamma} \text{ is a c.d.f.}\}$$

Apparently, if our model is identifiable, then $p_{\min} = p$. We propose to estimate $p_{\min}$ by a minimum search:

$$\hat{p}_{\min} = \inf_{\boldsymbol{\xi}} \hat{\alpha}(\boldsymbol{\xi})_0^{c_n}, \tag{2.4}$$

where $\hat{\alpha}(\boldsymbol{\xi})_0^{c_n}$ represents the $\hat{\alpha}_0^{c_n}$ estimator in [21] with their $F_0(x)$ replaced by our $F_0(x;\boldsymbol{\xi})$.

In (2.4), the minimizer $\boldsymbol{\xi}$ is the proposed estimator of $\boldsymbol{\theta}$:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\xi}} \hat{\alpha}(\boldsymbol{\xi})_0^{c_n}.$$

2.3.2 EM log-concave method

Suppose we have a random sample of n i.i.d. observations $(X_1, X_2, \cdots, X_n)$ from the density $g(x) = (1-p)f_0(x;\boldsymbol{\theta}) + pf(x)$, $p \in [0,1]$, $f = e^{\phi}$ is a log concave density. Here, we assume $\phi : \mathcal{R} \to [-\infty, \infty)$ is upper semicontinuous and coercive, i.e. $\phi(x) \to -\infty$, as $|x| \to \infty$, and we use Φ^d to denote the family of such functions on $\mathcal{R}^d$. Then, with the empirical distribution $Q_n = \dfrac{1}{n}\displaystyle\sum_{i=1}^{n} \delta_{X_i}$, where δ_{X_i} is the degenerate distribution function at $\{X_i\}$, the log likelihood of our random sample can be written as:

$$L(p, \boldsymbol{\theta}, \phi, Q_n) = \int \log(g) dQ_n = \frac{1}{n}\sum_{i=1}^{n} \log \left\{(1-p)f_0(X_i;\boldsymbol{\theta}) + pe^{\phi(X_i)}\right\}, \qquad (2.5)$$

subject to the condition that $\int e^{\phi(x)}dx = 1$. We propose to estimate p, $\boldsymbol{\theta}$ and ϕ by maximizing $L(p, \boldsymbol{\theta}, \phi, Q_n)$. One advantage of assuming a log-concave density for f is that such semiparametric maximum likelihood estimate exists without requiring any tuning parameter.

The computation algorithm

Maximizing the log likelihood (2.5) is not easy. To this end, we propose an EM algorithm to simplify the computation.

Algorithm 2.3.1. *Starting with initial values $p^{(0)}$, $\boldsymbol{\theta}^{(0)}$ and $f^{(0)}$, iterate the following two steps until convergence:*

"E step"*: Given $p^{(k)}$, $\boldsymbol{\theta}^{(k)}$ and $f^{(k)}$,*

$$\omega_i^{(k+1)} = \frac{(1-p^{(k)})f_0(x_i;\boldsymbol{\theta}^{(k)})}{(1-p^{(k)})f_0(x_i;\boldsymbol{\theta}^{(k)}) + p^{(k)}f^{(k)}(x_i)}.$$

"M step"*: Update the estimates of p, $\boldsymbol{\theta}$ and f,*

$$\begin{aligned}
p^{(k+1)} &= \frac{1}{n}\sum_{i=1}^{n}(1-\omega_i^{(k+1)}),\\
\boldsymbol{\theta}^{(k+1)} &= arg\ \underset{\boldsymbol{\theta}}{max}\sum_{i=1}^{n}\omega_i^{(k+1)}\log\{f_0(x_i;\boldsymbol{\theta})\},\\
\phi^{(k+1)} &= arg\ \underset{\phi\in\Phi}{max}\sum_{i=1}^{n}(1-\omega_i^{(k+1)})\phi(x_i),\\
f^{(k+1)} &= e^{\phi^{(k+1)}}.
\end{aligned}$$

In M step, the ϕ is updated by the active set algorithm proposed by [7] and implemented in the package logcondens by [25] in R ([22]).

Throughout this paper, we use "EM logconcave2" to denote this procedure where "2" indicates the two-component mixture model. Please see the Appendix (Section 2.7) for details of implementation of this algorithm.

2.4 Theoretical Properties

2.4.1 Consistency of $(\hat{p}_{\mathbf{min}})_n$ and $\hat{\boldsymbol{\theta}}_n$

[21] showed the following consistency theorem for their model:

Theorem 2.4.1. *If $c_n = o(\sqrt{n})$ and $c_n \to \infty$, then, $\hat{p}_0^{c_n} \overset{P}{\to} p_0$.*

In our setting, if model (2.1) is identifiable, $c_n = o(\sqrt{n})$ and $c_n \to \infty$, then,

$$\hat{p}(\boldsymbol{\xi})_0^{c_n} \overset{P}{\to} \begin{cases} 1, & \boldsymbol{\xi} \neq \boldsymbol{\theta}, \\ p, & \boldsymbol{\xi} = \boldsymbol{\theta}. \end{cases}$$

If we assume the convergence is uniform, then we have the following consistency theorem for our estimator $(\hat{p}_{min})_n$ and $\hat{\boldsymbol{\theta}}_n$:

Theorem 2.4.2. *Suppose*

$$\hat{p}(\boldsymbol{\xi})_0^{c_n} \to \begin{cases} 1, & \boldsymbol{\xi} \neq \boldsymbol{\theta}, \\ p, & \boldsymbol{\xi} = \boldsymbol{\theta}. \end{cases}$$

uniformly, then,

$$(\hat{p}_{min})_n \to p, \text{ and } \hat{\boldsymbol{\theta}}_n \to \boldsymbol{\theta}.$$

2.4.2 Existence and consistency of our maximum likelihood estimator

In this section, we establish the existence of the maximizer of (2.5) and prove the consistency of the proposed semiparametric maximum likelihood estimator. For any distribution Q on $\mathcal{R}^d$, we define,

$$\begin{aligned} L(p, \boldsymbol{\theta}, \phi, Q) &= \int \log\{(1-p)f_0(\cdot, \boldsymbol{\theta}) + pe^{\phi}\}dQ, \\ L(Q) &= \sup_{\substack{p\in[0,1], \boldsymbol{\theta}\in\boldsymbol{\Theta} \\ \phi\in\Phi^d,\ \int e^{\phi(x)}dx=1}} L(p, \boldsymbol{\theta}, \phi, Q). \end{aligned}$$

Define the convex support of Q as,

$$\text{csupp}(Q) = \bigcap\{C : C \subseteq R^d \text{ closed and convex, } Q(C) = 1\}.$$

We first provide the existence result of the maximizer of (2.5) in the following theorem.

Theorem 2.4.3. *Assume* $\text{supp}\{f_0(x, \boldsymbol{\theta})\} \subseteq \text{csupp}(Q)$ *for any* $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ *and* $\boldsymbol{\Theta}$ *is compact. Suppose there exists some integer* $k \geq 1$*, such that,*

$$\int ||x||^k Q(dx) \quad < \quad \infty \quad \text{and} \quad \text{interior}(\text{csupp}(Q)) \neq \emptyset.$$

For some fixed $m(x) = c_0 e^{c_1||x||^k}, c_0, c_1 > 0$*, let* $\tilde{\Phi}^d = \{\phi \in \Phi^d : \int e^{\phi(x)}dx = 1$ *and* $f_0(x, \boldsymbol{\theta}) \leq m(x)e^{\phi(x)}, \forall \boldsymbol{\theta} \in \Theta\}$*. Then*

$$L(Q) = \underset{p\in[0,1], \boldsymbol{\theta}\in\boldsymbol{\Theta}, \phi\in\tilde{\Phi}^d}{sup} L(p, \boldsymbol{\theta}, \phi, Q)$$

is real. In that case, there exists,

$$(p_0, \boldsymbol{\theta}_0, \phi_0) \in \underset{p\in[0,1], \boldsymbol{\theta}\in\boldsymbol{\Theta}, \phi\in\tilde{\Phi}^d}{\text{argmax}} L(p, \boldsymbol{\theta}, \phi, Q).$$

Moreover,

$$\text{interior}(\text{csupp}(Q)) \subseteq \text{dom}(\phi_0) = \{x \in \mathcal{R}^d : \phi_0(x) > -\infty\} \subseteq \text{csupp}(Q).$$

Next we establish the consistency of our maximum likelihood estimator. Let

$$\begin{aligned}
\mathcal{Q}^k &= \{Q \text{ on } \mathcal{R}^d : \int ||x||^k Q(dx) < \infty\}, \\
\mathcal{Q}_0 &= \{Q \text{ on } \mathcal{R}^d : \text{interior}(\text{csupp}(Q)) \neq \emptyset\}.
\end{aligned}$$

In what follows, we consider the convergence of distributions under Mallows' distance D_1 [17]. Specifically, for two distributions Q, $Q' \in \mathcal{Q}^k$,

$$D_k(Q, Q') = \inf_{\substack{X,\ X' \\ X \sim Q,\ X' \sim Q'}} \{E||X - X'||^k\}^{1/k}.$$

It is known that $\lim_{n\to\infty} D_k(Q_n, Q) \to 0$ is equivalent to $Q_n \to_w Q$ and $\int ||x||^k Q_n(dx) \to \int ||x||^k Q(dx)$ [1, 17]. Here $Q_n \to_w Q$ means the weak convergence, or convergence in distribution.

Theorem 2.4.4. *Assume, (a).* $\text{supp}\{f_0\} \subseteq \text{csupp}(Q)$*; (b). there exist some integer* $k \geq 1$, $c_i \geq 0$, $i = 0, 1$, *and* $m(x) = c_0 e^{c_1 ||x||^k}$, *such that,* $f_0(x, \boldsymbol{\theta}) \leq m(x) f(x) = m(x) e^{\phi(x)}, \forall \boldsymbol{\theta} \in \Theta$. *Let* $\{Q_n\}$ *be a sequence of distributions in* $\mathcal{Q}_0 \bigcap \mathcal{Q}^k$ *such that* $\lim_{n\to\infty} D_k(Q_n, Q) = 0$ *for some* $Q \in \mathcal{Q}_0 \bigcap \mathcal{Q}^k$. *Suppose* f_0 *is upper semi-continuous and* $\dfrac{log(f_0)}{1 + ||x||}$ *is bounded. Then*

$$\lim_{n\to\infty} L(Q_n) = L(Q).$$

Assume there exist maximizers $(p_n, \boldsymbol{\theta}_n, \phi_n)$ *of* $L(p, \boldsymbol{\theta}, \phi, Q_n)$*, and a unique maximizer* $(p^*, \boldsymbol{\theta}^*, \phi^*)$ *of* $L(p, \boldsymbol{\theta}, \phi, Q)$*, where* $p_n, p^* \in [0,1], \boldsymbol{\theta}_n, \boldsymbol{\theta}^* \in \boldsymbol{\Theta}, \phi_n, \phi^* \in \tilde{\Phi}^d$*. Let* $f_n = exp(\phi_n)$*,* $f^* = exp(\phi^*)$*, then*

$$\begin{aligned}
\lim_{n\to\infty} p_n &= p^*, \\
\lim_{n\to\infty} \boldsymbol{\theta}_n &= \boldsymbol{\theta}^*, \\
\lim_{n\to\infty,\ x\to y} f_n(x) &= f^*(y), \quad \forall y \in R^d \setminus \partial\{f^* > 0\}, \\
\operatorname*{limsup}_{n\to\infty,\ x\to y} f_n(x) &\leq f^*(y), \quad \forall y \in \partial\{f^* > 0\}, \\
\lim_{n\to\infty} \int |f_n(x) - f^*(x)| dx &= 0.
\end{aligned}$$

2.5 Simulation

To demonstrate the performances of our proposed algorithms, we generate a finite random sample $(X_1, X_2, \cdots, X_n)$ from the following seven models:

- Model 1: $g(x) = (1-p)N(\mu_0 = 0, \sigma_0 = 1)$;
- Model 2: $g(x) = (1-p)\text{exp}(\lambda_0 = 2) + pN(\mu_1 = 3, \sigma_1 = 0.5)$;
- Model 3: $g(x) = (1-p)\text{exp}(\lambda_0 = 2) + p(\text{exp}(\lambda_1 = 2) + 3)$;
- Model 4: $g(x) = (1-p)\text{gamma}(\text{shape}_0 = 2, \text{scale}_0 = 0.5) + p(F(d_1 = 100, d_2 = 100) + 5)$;
- Model 5: $g(x) = (1-p)\text{Weibull}(\text{shape}_0 = 2, \text{scale}_0 = 1) + p(\text{beta}(0.5, 0.5) + 2)$;

- Model 6: $g(x) = p_0 N(\mu_0 = 0, \sigma_0 = 1) + p_1 U(10, 11) + p_2 U(-5, -4)$;
- Model 7: $g(x) = p_0 \text{logistic}(\mu_0 = 0, s_0 = 1) + p_1 U(-11, -10) + p_2 Pareto(m_2 = 5, s_2 = 5)$.

Model 1 has no outliers and is used to test how our robust maximum likelihood estimates perform when there are no outliers. Model 2, 3, 4, and 5 are two-component mixtures.

For each model, we generate a random sample of size $n = 250$ or $n = 500$, with the proportion of outliers to be $p = 0.2$, or $p = p_1 + p_2 = 0.2$. For model 6 and 7, $p_1 = 0.05$, $p_2 = 0.15$. For each sample, we calculate the estimated proportion of outliers ($\hat{p}$), the parameter of the known component ($\hat{\boldsymbol{\theta}}$), the mean of the unknown component ($\hat{\mu}_1$), the L_2 distance ($\hat{d}_{L_2}$) and Kullback Leibler divergence ($\hat{d}_{KL}$) between $f_0(x, \boldsymbol{\theta})$ and $f_0(x, \hat{\boldsymbol{\theta}})$. To calculate $\hat{d}_{KL}$, for each $\hat{\boldsymbol{\theta}}$, we generate a random sample of size $m = 10000$ under the distribution $f_0(x, \boldsymbol{\theta})$, and estimate the distance by

$$\hat{d}_{KL} = \frac{1}{m} \sum_{i=1}^{m} \log \frac{f_0(x_i, \boldsymbol{\theta})}{f_0(x_i, \hat{\boldsymbol{\theta}})}.$$

Over $K = 200$ repetitions, we report the bias and MSE of the estimates of p, $\boldsymbol{\theta}$, and μ_1, and the mean of estimates of d_{L_2} and d_{KL}.

For comparison, we report the parameter estimation results from MLE (maximum likelihood estimation without outlier detection), oracle (the MLE after deleting all outliers) and TLE (trimmed likelihood estimator implemented using FAST-TLE by [20]) methods. For the TLE method, the trimming parameter is selected to be 0.1, 0.2 and 0.3, we denote these methods by TLE (0.1), TLE (0.2) and TLE (0.3), respectively. Table 2.1—2.7 reports the results for sample size $n = 500$. The results for sample size $n = 250$ are included in the Appendix (Section 2.7).

From the simulation results, we can see that when our data are heavily contaminated (with 20% outliers), the MLE estimates are very problematic and thus sensitive to outliers. On the other hand, the two algorithms we proposed, $p_{\min}$ and EM logconcave methods, both have very promising performances. In general, they also outperform the TLE method.

When the generated data does not contain outliers (Model 1), $p_{\min}$ method works better than two-component and three-component EM log-concave methods. When simulated data is generated from Model 2—5, both two-component and three-component EM log-concave methods work well. But when the data is generated from a three-component mixture, Model 6 and 7, the two-component EM log-concave method fails in these situations as expected. Instead, three-component EM log-concave method still works very well.

In general, the maximum likelihood criterion and the minimum of the CDF distance criterion work very similar. The minimum p value criterion works better for Model 2—5 when we use three-component EM log-concave method, but is less favorable if we use two-component EM log-concave algorithm. For Model 6 and 7, again this criterion is less favorable. In general, we recommend the MLE and minimum CDF distance criteria.

Table 2.1: Bias (MSE) of estimates of $p/\boldsymbol{\theta}$ and mean of d_{L2} and d_{KL} for the model 1 when $n = 500$.

Model 1 (no outlier): $g(x) = N(\mu_0 = 0, \sigma_0 = 1)$						
method	$p_{\min}$	MLE	oracle	TLE(0.1)	TLE(0.2)	TLE(0.3)
p	0.08(0.0073)					
μ_0	-0.012(0.0031)	-0.002(0.0016)	-0.002(0.0016)	0.002(0.0036)	0.004(0.0058)	0.008(0.0095)
σ_0	0.004(0.0021)	0.002(0.001)	0.002(0.001)	-0.209(0.0444)	-0.338(0.1148)	-0.444(0.198)
d_{L2}	0.0259	0.0184	0.0184	0.1171	0.2107	0.3086
d_{KL}	0.0038	0.0018	0.0018	0.07	0.239	0.5574
method	EM logcon2-1	EM logcon2-2	EM logcon2-3	EM logcon3-1	EM logcon3-2	EM logcon3-3
p	0.04(0.0023)	0.008(4e-04)	0.021(0.0013)	0.099(0.0116)	0.036(0.0031)	0.064(0.0075)
μ_0	-0.002(0.0098)	0(0.0031)	-0.002(0.0062)	0.005(0.0081)	0.003(0.0058)	0.007(0.0083)
σ_0	-0.056(0.0053)	-0.011(0.0021)	-0.03(0.0035)	-0.135(0.0238)	-0.049(0.0074)	-0.081(0.0144)
d_{L2}	0.0468	0.0248	0.0344	0.0813	0.0426	0.059
d_{KL}	0.0124	0.0041	0.008	0.0413	0.0132	0.0268

Table 2.2: Bias (MSE) of estimates of $p/\boldsymbol{\theta}/\mu_1$ and mean of d_{L2} and d_{KL} for the model 2 when $n = 500$.

Model 2: $g(x) = 0.8\exp(\lambda_0 = 2) + 0.2N(\mu_1 = 3, \sigma_1 = 0.5)$						
method	$p_{\min}$	MLE	oracle	TLE(0.1)	TLE(0.2)	TLE(0.3)
p	0.042(0.0027)					
λ_0	-0.094(0.0402)	-0.996(0.9948)	-0.001(0.0093)	-0.626(0.4013)	0.046(0.0364)	0.862(0.7883)
μ_1	-0.218(0.0585)					
d_{L2}	0.0591	0.4066	0.0269	0.2415	0.054	0.2751
d_{KL}	0.0055	0.1929	0.0011	0.065	0.0043	0.0757
method	EM logcon2-1	EM logcon2-2	EM logcon2-3	EM logcon3-1	EM logcon3-2	EM logcon3-3
p	0.001(4e-04)	-0.012(6e-04)	0.001(5e-04)	0.001(5e-04)	-0.003(4e-04)	0.005(6e-04)
λ_0	0.017(0.0214)	-0.084(0.0335)	0.014(0.027)	0.014(0.0218)	-0.015(0.0189)	0.034(0.0278)
μ_1	-0.012(0.007)	0.043(0.008)	0.043(0.008)			
d_{L2}	0.058	0.0585	0.0696	0.0401	0.0382	0.0448
d_{KL}	0.0026	0.0045	0.0033	0.0026	0.0024	0.0033

Table 2.3: Bias (MSE) of estimates of $p/\boldsymbol{\theta}/\mu_1$ and mean of d_{L2} and d_{KL} for the model 3 when $n = 500$.

Model 3: $g(x) = 0.8\exp(\lambda_0 = 2) + 0.2(\exp(\lambda_1 = 2) + 3)$						
method	$p_{\min}$	MLE	oracle	TLE(0.1)	TLE(0.2)	TLE(0.3)
p	0.045(0.0029)					
λ_0	-0.063(0.0328)	-1.089(1.1882)	0.005(0.0109)	-0.727(0.5378)	0.017(0.0432)	0.86(0.7794)
μ_1	-0.307(0.1077)					
d_{L2}	0.0538	0.4515	0.029	0.2848	0.0603	0.2746
d_{KL}	0.0042	0.2435	0.0013	0.091	0.0055	0.0748
method	EM logcon2-1	EM logcon2-2	EM logcon2-3	EM logcon3-1	EM logcon3-2	EM logcon3-3
p	-0.001(3e-04)	-0.006(4e-04)	-0.001(3e-04)	0(3e-04)	-0.002(3e-04)	0.001(4e-04)
λ_0	-0.008(0.0135)	-0.061(0.0279)	-0.003(0.0182)	0.004(0.0123)	-0.019(0.0154)	0.007(0.0165)
μ_1	0.003(0.0025)	0.012(0.0037)	0.012(0.0037)			
d_{L2}	0.0322	0.0478	0.0381	0.0309	0.0355	0.0362
d_{KL}	0.0017	0.0038	0.0023	0.0015	0.0019	0.002

Table 2.4: Bias (MSE) of estimates of $p/\boldsymbol{\theta}/\mu_1$ and mean of d_{L2} and d_{KL} for the model 4 when $n = 500$.

Model 4: $g(x) = 0.8\text{gamma}(\text{shape}_0 = 2, \text{scale}_0 = 0.5) + 0.2(F(d_1 = 100, d_2 = 100) + 5)$						
method	$p_{\min}$	MLE	oracle	TLE(0.1)	TLE(0.2)	TLE(0.3)
p	0.037(0.002)					
$shape_0$	-0.042(0.0473)	-0.102(0.0664)	0.004(0.0173)	-1.142(1.3058)	-0.123(0.3)	1.332(1.8739)
$scale_0$	0.035(0.0055)	0.06(0.0109)	0.001(0.0013)	1.299(1.7065)	0.11(0.0708)	-0.25(0.0631)
μ_1	-0.409(0.2006)					
d_{L2}	0.0504	0.3284	0.0338	0.3605	0.1144	0.2301
d_{KL}	0.0067	0.3071	0.0027	0.238	0.0327	0.1588
method	EM logcon2-1	EM logcon2-2	EM logcon2-3	EM logcon3-1	EM logcon3-2	EM logcon3-3
p	0.000(3e-04)	-0.002(3e-04)	-0.001(3e-04)	0.005(4e-04)	0(3e-04)	0.003(3e-04)
$shape_0$	0.008(0.0184)	-0.046(0.0255)	-0.019(0.0209)	0.051(0.0345)	0.005(0.0186)	0.027(0.0271)
$scale_0$	0(0.0014)	0.023(0.0037)	0.011(0.0024)	-0.007(0.0018)	0.001(0.0014)	-0.002(0.0019)
μ_1	0.002(4e-04)	0.007(5e-04)	0.007(5e-04)			
d_{L2}	0.0345	0.0402	0.0372	0.0399	0.0345	0.0376
d_{KL}	0.0028	0.0042	0.0034	0.0039	0.0029	0.0036

Table 2.5: Bias (MSE) of estimates of $p/\boldsymbol{\theta}/\mu_1$ and mean of d_{L2} and d_{KL} for the model 5 when $n = 500$.

Model 5: $g(x) = 0.8\text{Weibull}(\text{shape}_0 = 2, \text{scale}_0 = 1) + 0.2(\text{beta}(0.5, 0.5) + 2)$

method	$p_{\min}$	MLE	oracle	TLE(0.1)	TLE(0.2)	TLE(0.3)
p	0.034(0.002)					
$shape_0$	-0.085(0.0303)	-1.702(3.2258)	0.006(0.0065)	-0.206(0.0463)	0.053(0.0114)	0.507(0.2785)
$scale_0$	0.052(0.0066)	0.671(1.4677)	-0.001(6e-04)	0.164(0.0284)	0.002(0.0015)	-0.102(0.0116)
μ_1	-0.158(0.0435)					
d_{L2}	0.0758	0.271	0.0337	0.1548	0.0475	0.2132
d_{KL}	0.0189	0.1675	0.0024	0.0556	0.0056	0.148
method	EM logcon2-1	EM logcon2-2	EM logcon2-3	EM logcon3-1	EM logcon3-2	EM logcon3-3
p	-0.006(0.001)	-0.034(0.0026)	0.003(0.001)	0.015(0.0017)	0.008(0.0011)	0.02(0.0014)
$shape_0$	-0.011(0.013)	-0.087(0.0221)	0.003(0.0159)	0.037(0.0198)	0.001(0.0154)	0.035(0.018)
$scale_0$	0.012(0.0028)	0.058(0.0077)	0.001(0.0026)	0(0.0042)	0.001(0.0034)	-0.012(0.0028)
μ_1	0.022(0.009)	0.09(0.0205)	0.09(0.0205)			
d_{L2}	0.0494	0.0775	0.0575	0.0643	0.0592	0.0613
d_{KL}	0.0079	0.0185	0.0099	0.0126	0.0106	0.011

Table 2.6: Bias (MSE) of estimates of $p/\boldsymbol{\theta}$ and mean of d_{L2} and d_{KL} for the model 6 when $n = 500$.

Model 6: $g(x) = 0.8N(\mu_0 = 0, \sigma_0 = 1) + 0.05U(10, 11) + 0.15U(-5, -4)$

method	$p_{\min}$	MLE	oracle	TLE(0.1)	TLE(0.2)	TLE(0.3)
p	0.026(0.0015)					
μ_0	-0.004(0.0037)	-0.159(0.0441)	-0.002(0.0025)	-0.48(0.2411)	-0.037(0.0074)	-0.004(0.0057)
σ_0	0.054(0.006)	8.237(69.0208)	0.001(0.0013)	0.652(0.4356)	0.035(0.0172)	-0.244(0.0615)
d_{L2}	0.0359	0.4714	0.0221	0.2273	0.0512	0.1421
d_{KL}	0.0069	1.7218	0.0026	0.2285	0.0163	0.1079
method	EM logcon2-1	EM logcon2-2	EM logcon2-3	EM logcon3-1	EM logcon3-2	EM logcon3-3
p	-0.151(0.0228)	-0.158(0.0251)	-0.157(0.0246)	-0.001(3e-04)	-0.008(4e-04)	-0.001(3e-04)
μ_0	-0.711(0.5136)	-0.628(0.4012)	-0.645(0.4233)	-0.002(0.0027)	-0.037(0.0061)	-0.005(0.0027)
σ_0	0.879(0.7773)	1.087(1.2264)	1.046(1.1271)	0.001(0.0019)	0.064(0.0107)	0.005(0.0021)
d_{L2}	0.2786	0.291	0.2884	0.023	0.0413	0.0235
d_{KL}	0.3441	0.3962	0.3855	0.0031	0.0106	0.0033

Table 2.7: Bias (MSE) of estimates of $p/\boldsymbol{\theta}$ and mean of d_{L2} and d_{KL} for the model 7 when $n = 500$.

Model 7: $g(x) = 0.8\text{logistic}(\mu_0 = 0, s_0 = 1) + 0.05U(-11, -10) + 0.15Pareto(m_2 = 5, s_2 = 5)$						
method	$p_{\min}$	MLE	oracle	TLE(0.1)	TLE(0.2)	TLE(0.3)
p	0.034(0.0018)					
μ_0	0.01(0.0159)	0.466(0.2347)	-0.002(0.0069)	0.676(0.4791)	0.091(0.0297)	-0.011(0.0113)
s_0	0.052(0.0073)	0.92(0.8553)	0.002(0.0017)	0.337(0.118)	-0.018(0.0072)	-0.295(0.0889)
d_{L2}	0.0305	0.1852	0.0182	0.1306	0.0369	0.1288
d_{KL}	0.0069	0.2507	0.0025	0.1043	0.0101	0.1079
method	EM logcon2-1	EM logcon2-2	EM logcon2-3	EM logcon3-1	EM logcon3-2	EM logcon3-3
p	-0.148(0.0221)	-0.097(0.0103)	-0.087(0.0085)	0.001(3e-04)	-0.036(0.0017)	-0.001(4e-04)
μ_0	0.673(0.4777)	0.082(0.1062)	0.032(0.1038)	-0.004(0.0074)	0.134(0.0303)	0.002(0.0087)
s_0	0.569(0.33)	0.556(0.3262)	0.524(0.2881)	-0.001(0.0022)	0.138(0.025)	0.012(0.003)
d_{L2}	0.1552	0.1344	0.1291	0.0197	0.0488	0.0214
d_{KL}	0.1608	0.1286	0.1183	0.0029	0.0177	0.0035

2.6 Discussion

In this paper, we propose a novel semiparametric mixture model to provide robust maximum likelihood estimation. To incorporate the contaminated data, we assume a log-concave density for the possible outlier component and thus the whole data can be considered from a semiparametric mixture model. We propose to estimate the semiparametric mixture model by maximizing the corresponding semiparametric likelihood. We prove the existence and consistency of the proposed semiparametric maximum likelihood estimator. One main advantage of the proposed semiparametric maximum likelihood estimator is that it does not require choose a tuning parameter unlike many other kernel density estimators. Based on the new model, we can also assign a probability of each observation being an outlier. The simulation results demonstrate that our proposed algorithms perform very well across a variety of contaminated densities (from which the outliers are generated from) even when the contaminated densities are not log-concave.

An interesting possible future research is to extend the proposed robust maximum likelihood estimator to the regression context. For the regression model $y = \boldsymbol{x}^T\boldsymbol{\beta} + \epsilon$, in order to incorporate the possible outliers and get a robust regression estimate of β, we can model the density ϵ by the proposed semiparametric mixture model with f_0 being a normal density with mean 0. In addition, it will be also interesting to extend the proposed method to provide robust maximum likelihood estimator for multivariate data.

2.7 Appendix

2.7.1 EM log-concave algorithm

Detailed implementation of the two-component EM log-concave algorithm

We start the algorithm from multiple initial values and propose the following three criteria to select the final best model:

- EM logconcave2-1: use the maximum likelihood criterion;
- EM logconcave2-2: use the minimum p value criterion;
- EM logconcave2-3: use the minimum of the CDF distance between the estimated CDF corresponding to the two-component mixture model with the empirical CDF.

Specifically, the CDF distance is defined by

$$\frac{1}{n}\sum_{i=1}^{n}(\hat{G}(x_i)-\frac{i}{n})^2,$$

where $\hat{G}$ is the estimated CDF of the proposed model.

Suppose $f_0(x;\boldsymbol{\theta})$ represents the normal density, the initial probabilities $\omega_i^{(0)}$ of our algorithm are generated as follows: We randomly select $\hat{\mu}_0$ from $U(X_{(0.25)}, X_{(0.75)})$ of the random sample $(X_1, X_2, \cdots, X_n)$, where $X_{(p)}$ denotes the empirical quantile corresponding to the probability p. The initial $\hat{\sigma}_0$ is randomly selected from $U(0.1s, s)$, where s is the standard deviation of the random sample. The initial f_0 is estimated to be $\hat{f}_0(x) = \frac{1}{\hat{\sigma}_0}\phi(\frac{x-\hat{\mu}_0}{\hat{\sigma}_0})$, where ϕ represents the standard normal density. Let $\alpha \sim U(0, 0.3)$, we find one sided $100\alpha\%$ points with lowest $\hat{f}_0$ values, assign $w_i^{(0)}$ to be 0 on those points and 1 otherwise, i.e., we assume those points belong to the component f initially. Here we take a random toss $Z \sim Bernoulli(0.5)$. If $Z = 1$, we assign the right hand side points to f; if $Z = 0$, we assign the left hand side points to f.

For the general f_0, the initial generation of the probabilities $\omega_i^{(0)}$ is very similar: we first generate random initial parameter $\hat{\boldsymbol{\theta}}_0$, then randomly select one sided lowest $100\alpha\%$ points with respect to $\hat{f}_0 = f_0(x;\hat{\boldsymbol{\theta}}_0)$, and assign these points to the unknown outlier component.

After we estimate p, $\boldsymbol{\theta}$ and f, we want to make sure if x is from the highest 80% density points of f_0, the probability for x to be outliers is relatively low. For example, when

f_0 represents the normal density, we define

$$ratio = \max_{x \in (\hat{\mu}+Z_{0.1}\hat{\sigma}, \hat{\mu}+Z_{0.9}\hat{\sigma})} (\frac{\hat{p}\hat{f}(x)}{(1-\hat{p})f_0(x,\hat{\boldsymbol{\theta}}) + \hat{p}\hat{f}(x)}).$$

The fitted model is discarded if ratio > 0.2. More general, the ratio is defined to be the maximum value over all x within the highest 80% density points of $f_0(x, \hat{\boldsymbol{\theta}})$.

In order to apply our procedure more efficiently, we adopt the FAST-MCD strategy proposed by [24] which consists of a two-step procedure: a trial step followed by a refinement step. The pseudocodes are as follows:

- Start from any random initial values of $\{\omega_i^{(0)}\}_{i=1}^n$. Run 5 steps of EM logconcave2 iteration, the fitted model is selected if $ratio < 0.2$. Repeat this until we have selected 50 fitted models or until we have used 500 initials.
- Select the top 10 fitted models with respect to the likelihood, the minimum of $\hat{p}$ value or the minimum of CDF distance, and run the EM algorithm until convergence.
- Take the best fitted model according to the likelihood, the minimum of $\hat{p}$ value or the minimum of CDF distance.

Three-component EM logconcave

If the outliers lie both sides of f_0, then the single log-concave density will not be adequate to approximate f. For such situation, we further propose the following three-component semiparametric mixture model,

$$g(x) = p_0 f_0(x; \boldsymbol{\theta}) + p_1 f_1(x) + p_2 f_2(x), \tag{2.6}$$

where the contaminated density f is modeled by a two component mixture of log-concave density functions. We propose to estimate p_i, $\boldsymbol{\theta}$ and f_i, $i = 1, 2$ by using the following EM algorithm.

Algorithm 2.7.1. *Starting with initial values $p^{(0)}$, $p_1^{(0)}$, $\boldsymbol{\theta}^{(0)}$, $f_1^{(0)}$ and $f_2^{(0)}$, iterate the following two steps until convergence:*

"E step": *Given $p_0^{(k)}$, $p_1^{(k)}$, $p_2^{(k)}$, $\boldsymbol{\theta}^{(k)}$, $f_1^{(k)}$ and $f_2^{(k)}$,*

$$\begin{aligned}
\omega_{0i}^{(k+1)} &= \frac{p_0^{(k)} f_0(x_i; \boldsymbol{\theta}^{(k)})}{p_0^{(k)} f_0(x_i; \boldsymbol{\theta}^{(k)}) + p_1^{(k)} f_1^{(k)}(x_i) + p_2^{(k)} f_2^{(k)}(x_i)}, \\
\omega_{1i}^{(k+1)} &= \frac{p_1^{(k)} f_1^{(k)}(x_i; \boldsymbol{\theta}^{(k)})}{p_0^{(k)} f_0(x_i; \boldsymbol{\theta}^{(k)}) + p_1^{(k)} f_1^{(k)}(x_i) + p_2^{(k)} f_2^{(k)}(x_i)}, \\
\omega_{2i}^{(k+1)} &= 1 - \omega_{0i}^{(k+1)} - \omega_{1i}^{(k+1)}.
\end{aligned}$$

“M step”*:*

$$
\begin{aligned}
p_0^{(k+1)} &= \frac{1}{n}\sum_{i=1}^{n}\omega_{0i}^{(k+1)},\\
p_1^{(k+1)} &= \frac{1}{n}\sum_{i=1}^{n}\omega_{1i}^{(k+1)},\\
p_2^{(k+1)} &= 1 - p_0^{(k+1)} - p_1^{(k+1)},\\
\boldsymbol{\theta}^{(k+1)} &= \underset{\boldsymbol{\theta}}{argmax}\sum_{i=1}^{n}\omega_{0i}^{(k+1)} log(f_0(x_i;\boldsymbol{\theta})),\\
\phi_1^{(k+1)} &= \underset{\phi\in\Phi^1}{argmax}\sum_{i=1}^{n}\omega_{1i}^{(k+1)}\phi(x_i),\\
f_1^{(k+1)} &= e^{\phi_1^{(k+1)}},\\
\phi_2^{(k+1)} &= \underset{\phi\in\Phi^1}{argmax}\sum_{i=1}^{n}\omega_{2i}^{(k+1)}\phi(x_i),\\
f_2^{(k+1)} &= e^{\phi_2^{(k+1)}}.
\end{aligned}
$$

Similar to the two-component EM logconcave algorithm, we use “EM logconcave3” to represent three-component EM logconcave algorithm, and denote

- EM logconcave3-1: use maximum likelihood criterion;
- EM logconcave3-2: use minimum of p value criterion;
- EM logconcave3-3: use minimum of CDF distance criterion.

Initial generations are also very similar to the two-component EM logconcave algorithm: first we generate random initial parameter $\hat{\boldsymbol{\theta}}_0$, then randomly select two sided lowest $100\alpha\%$ ($\alpha \sim U(0, 0.3)$) points with respect to $\hat{f}_0 = f_0(x; \hat{\boldsymbol{\theta}}_0)$, and assign these points to the unknown component f_1 and f_2.

After we estimate p, $\boldsymbol{\theta}$, f_1 and f_2, we further restrict that if x is from the highest 80% density points of f_0, the probability for x to be from the two unknown outlier components is relatively low. For example, when f_0 represents the normal density, we define

$$ratio1 = \max_{x \in (\hat{\mu}+Z_{0.1}\hat{\sigma}, \hat{\mu}+Z_{0.9}\hat{\sigma})} \left(\frac{\hat{p}_1\hat{f}_1(x)}{\hat{p}_0 f_0(x;\hat{\boldsymbol{\theta}}) + \hat{p}_1\hat{f}_1(x) + \hat{p}_2\hat{f}_2(x)}\right)$$

$$ratio2 = \max_{x \in (\hat{\mu}+Z_{0.1}\hat{\sigma}, \hat{\mu}+Z_{0.9}\hat{\sigma})} \left(\frac{\hat{p}_2\hat{f}_2(x)}{\hat{p}_0 f_0(x;\hat{\boldsymbol{\theta}}) + \hat{p}_1\hat{f}_1(x) + \hat{p}_2\hat{f}_2(x)}\right)$$

The pseudocode of our three-component EM logconcave algorithm can be described as follows:

- Start from any randomized initial values of $\{\omega_{ji}^{(0)}\}_{i=1}^{n}, j = 0, 1, 2$. Run 5 steps of EM logconcave3 iteration, select the ones with $ratio1 < 0.2$ and $ratio2 < 0.2$. Repeat this until we have selected 50 fitted models or until we have used 500 initials.

- Select the top 10 fitted models with respect to the likelihood, the minimum of $\hat{p}$ value or the minimum of CDF distance, and run EM algorithm until convergence.

- Take the best fitted model according to the likelihood, the minimum of $\hat{p}$ value or the minimum of CDF distance.

2.7.2 Sketch of proofs

Proof of Theorem 2.4.2. The condition implies that $\forall \varepsilon > 0$, $\exists N$, such that if $n > N$,

$$|\hat{p}(\boldsymbol{\xi})_0^{c_n} - 1| < \varepsilon, \quad \forall \boldsymbol{\xi} \neq \boldsymbol{\theta}$$

$$|\hat{p}(\boldsymbol{\theta})_0^{c_n} - p| < \varepsilon.$$

For any $\varepsilon < \frac{1-p}{2}$, we have $\hat{p}(\boldsymbol{\theta})_0^{c_n} < p + \varepsilon < \frac{1+p}{2} < 1 - \varepsilon < \hat{p}(\boldsymbol{\xi})_0^{c_n}$. In this case, $(\hat{p}_{\min})_n = \underset{\boldsymbol{\xi}}{inf}\ \hat{p}(\boldsymbol{\xi})_0^{c_n} = \hat{p}(\boldsymbol{\theta})_0^{c_n}$. Thus, for $n > N$,

$$\begin{aligned} |(\hat{p}_{\min})_n - p| &= |\hat{p}(\boldsymbol{\theta})_0^{c_n} - p| < \varepsilon, \\ \hat{\boldsymbol{\theta}}_n &= \arg\min_{\boldsymbol{\xi}} \hat{p}(\boldsymbol{\xi})_0^{c_n} = \boldsymbol{\theta}. \end{aligned}$$

Hence, we have

$$(\hat{p}_{min})_n \to p, \text{ and } \hat{\boldsymbol{\theta}}_n \to \boldsymbol{\theta}.$$

□

Proof of Theorem 2.2.1. Suppose, $g(x) = (1-p_1)f_0(x;\boldsymbol{\theta}_1)+p_1 f_1(x) = (1-p_2)f_0(x;\boldsymbol{\theta}_2)+ p_2 f_2(x)$. Since $f(x) = 0$ on A, then $\forall x \in A$,

$$g(x) = (1-p_1)f_0(x;\boldsymbol{\theta}_1) = (1-p_2)f_0(x;\boldsymbol{\theta}_2). \tag{2.7}$$

If both $f_0(x;\boldsymbol{\theta}_1)$ and $f_0(x;\boldsymbol{\theta}_2)$ are analytic w.r.t. x on R, then the Identity Theorem guarantees that (2.7) actually holds on $\mathcal{R}$. Since density functions integrate to one, we have $p_1 = p_2$, and $\boldsymbol{\theta}_1 = \boldsymbol{\theta}_2$. Throughout this paper, we assume $f_0(x;\boldsymbol{\theta})$ is identifiable w.r.t $\boldsymbol{\theta}$. And it follows that

$$f_1(x) = \frac{1}{p_1}(g(x) - (1-p_1)f_0(x;\boldsymbol{\theta}_1)) = \frac{1}{p_2}(g(x) - (1-p_2)f_0(x;\boldsymbol{\theta}_2)) = f_2(x).$$

Hence, the model (2.1) is identifiable. □

Proof of Theorem 2.2.2. Without loss of generality, we assume $\boldsymbol{\theta}$ represents a single parameter. If $\boldsymbol{\theta}$ represents more than one parameter, the proof is very similar.

Suppose $g(x) = (1-p_1)f_0(x;\theta_0) + p_1 f_1(x) = (1-p_2)f_0(x;\theta_1) + p_2 f_2(x)$.

Since $C(\theta)$ is Lipschitz continuous, hence $\exists\, k_1 > 0$, s.t. $|C(\theta_0) - C(\theta_1)| \le k_1|\theta_0 - \theta_1|$. The boundedness of $\frac{\partial f_0(x;\theta)}{\partial\theta}$ implies $\exists\, k_2 > 0$, s.t. $|\frac{\partial f_0(x;\theta)}{\partial\theta}| \le k_2$.

As $\alpha \in (0,1)$, $\exists\, \delta > 0$, s.t. $A = \{x : f_0(x;\theta_0) > C(\theta_0) + \delta\}$ is a nonempty open set in $\mathcal{R}$. Hence, $f_1(x) = 0$ on A.

By the Mean Value Theorem, $\exists\, \theta^* \in (\theta_0, \theta_1)$, s.t., $f_0(x;\theta_1) - f_0(x;\theta_0) = \frac{\partial f_0}{\partial\theta}(x;\theta^*)(\theta_1 - \theta_0)$. Thus, on the set A,

$$\begin{aligned} f_0(x;\theta_1) &= f_0(x;\theta_0) + \frac{\partial f_0}{\partial\theta}(x;\theta^*)(\theta_1-\theta_0) \\ &\ge C(\theta_0) + \delta - k_2|\theta_1 - \theta_0| \\ &\ge C(\theta_1) - k_1|\theta_1-\theta_0| + \delta - k_2|\theta_1-\theta_0| \\ &\ge C(\theta_1), \text{ if } |\theta_1 - \theta_0| \le \frac{\delta}{k_1+k_2}. \end{aligned}$$

We have shown that if $|\theta_1 - \theta_0| \le \frac{\delta}{k_1+k_2}$, then $f_0(x;\theta_1) \ge C(\theta_1)$ on A, consequently $f_2(x) = 0$ on A. In this case, we have

$$g(x) = (1-p_1)f_0(x;\theta_0) = (1-p_2)f_0(x;\theta_1),\ x \in A.$$

By the Identity Theorem, $(1-p_1)f_0(x;\theta_0) = (1-p_2)f_0(x;\theta_1)$ for $x \in \mathcal{R}$, hence $p_1 = p_2$ and $\theta_0 = \theta_1$ as long as $f_0(x;\theta)$ is identifiable w.r.t. θ. □

Proof outline of Theorem 2.4.3. The proof is very similar to the proof we give in [36], here we give a brief outline of the proof.

1. $L(Q) > -\infty$.

2. When maximizing $L(p, \boldsymbol{\theta}, \phi, Q)$ over all $\phi \in \tilde{\Phi}^d$, we may restrict our attention to functions $\phi \in \Phi(Q) \subset \tilde{\Phi}^d$ such that $\text{dom}(\phi) = \{x \in R^d : \phi(x) > -\infty\} \subseteq \text{csupp}(Q)$.

3. $L(Q) < \infty$ and there exist constants M_0 and M_*, such that, $L(Q) = \sup\limits_{\substack{p\in[0,1],\theta\in\Theta,\phi\in\Phi(Q)\\ M_0\le\max(\phi(x))\le M_*}}$ $L(p, \boldsymbol{\theta}, \phi, Q)$.

4. Let $(p_n, \boldsymbol{\theta}_n, \phi_n)$ be a sequence such that $p_n \in [0, 1]$, $p_n \to p_0$, $\boldsymbol{\theta}_n \in \Theta$, $\boldsymbol{\theta}_n \to \boldsymbol{\theta}_0$, $\phi_n \in \Phi(Q)$, $M_n = \max(\phi_n(x)) \in [M_0, M_*]$, and $-\infty < L(p_n, \phi_n, Q) \uparrow L(Q)$ as $n \to \infty$. Then, there exist constants a and $b > 0$ such that,

$$\phi_n(x) \le a - b||x||, \ \forall n \ge 1, x \in \mathcal{R}^d.$$

5. There exist $\phi_0 \in \Phi^d$ and a subsequence ϕ_{n_k} such that,

$$\begin{aligned} \operatorname*{limsup}_{k\to\infty} \phi_{n_k}(x) &\le \phi_0(x) \le a - b||x||, \ \forall x \in \mathcal{R}^d, \\ \lim_{k\to\infty} \phi_{n_k}(x) &= \phi_0(x) > -\infty, \ \forall x \in \text{interior}(\text{csupp}(Q)). \end{aligned}$$

6. Fatou's Lemma concludes that

$$L(Q) = L(p_0, \boldsymbol{\theta}_0, \phi_0, Q).$$

□

Proof outline of Theorem 2.4.4. Again we give a brief outline of the proof similar to [36].

1. Suppose $\limsup_{n\to\infty} L(Q_n) = \lambda \in [-\infty, \infty]$, and $L(Q_{n_k}) \to \lambda$. First we show that $\lambda > -\infty$.

2. Let $M_{n_k} = \max_{x\in R^d} \phi_{n_k}(x)$. There exist constants M_0 and M_*, such that $M_0 < M_{n_k} < M_*$ for k sufficiently large and thus $\lambda < \infty$.

3. We may assume that $p_{n_k} \to p_* \in [0, 1]$, $\boldsymbol{\theta}_{n_k} \to \boldsymbol{\theta}_*$. If not, just replace these sequences with the convergent subsequences.

4. Show that there exist constants a and $b > 0$ such that,

$$\phi_{n_k}(x) \leq a - b||x||, \ \forall k \text{ sufficiently large, } x \in \mathcal{R}^d. \tag{2.8}$$

5. There exist $\phi_* \in \Phi^d$ and a subsequence $\{\phi_{n_{k_l}}\}$ such that,

$$\begin{aligned} \limsup_{l\to\infty, x\to y} \phi_{n_{k_l}}(x) &\leq \phi_*(y) \leq a - b||y||, \ \forall y \in \mathcal{R}^d, \\ \lim_{l\to\infty, x\to y} \phi_{n_{k_l}}(x) &= \phi_*(y) > -\infty, \ \forall y \in \text{interior}(\text{csupp}(Q)). \end{aligned}$$

6. By Skorohod's theorem and Fatous Lemma we can show that $\lambda \leq L(Q)$.

7. By Lemma 4.4 and Lemma 4.6 of [9] we can show that $\lambda \geq L(Q)$. Hence $\lambda = L(Q)$.

8. With similar arguments, we can show that $\liminf_{n\to\infty} L(Q_n) = L(Q)$ as well, and hence

$L(Q_n) \to L(Q)$, as $n \to \infty$. The proof also establishes that,

$$\begin{aligned}
\lim_{n\to\infty} p_n &= p^*, \\
\lim_{n\to\infty} \boldsymbol{\theta}_n &= \boldsymbol{\theta}^*, \\
\lim_{n\to\infty,\ x\to y} f_n(x) &= f^*(y), \quad \forall y \in R^d \setminus \partial\{f^* > 0\}, \\
\text{limsup}_{n\to\infty,\ x\to y} f_n(x) &\leq f^*(y), \quad \forall y \in \partial\{f^* > 0\}, \\
\lim_{n\to\infty} \int |f_n(x) - f^*(x)|dx &= 0.
\end{aligned}$$

where $f_n = exp(\phi_n)$, $f^* = exp(\phi^*)$.

□

2.7.3 More simulation results

Table 2.8: Bias(MSE) of estimates of $p/\boldsymbol{\theta}$ and mean of d_{L2} and d_{KL} when $p = 0$, $n = 250$, $K = 200$.

model 1(no outlier): $g(x) = N(\mu_0 = 0, \sigma_0 = 1)$, $\boldsymbol{\theta_0} = (\mu_0, \sigma_0)$						
method	$p_{\min}$	MLE	oracle	TLE0.1	TLE0.2	TLE0.3
p	0.116(0.0153)					
μ_0	-0.014(0.008)	-0.002(0.0037)	-0.002(0.0037)	0.008(0.0082)	0.007(0.0123)	0.004(0.0199)
σ_0	0.006(0.0039)	0.003(0.0021)	0.003(0.0021)	-0.213(0.047)	-0.342(0.1181)	-0.448(0.202)
d_{L2}	0.0375	0.0269	0.0269	0.3175	0.2178	0.3175
d_{KL}	0.0082	0.0038	0.0038	0.6	0.2599	0.595
method	EM logcon2-1	EM logcon2-2	EM logcon2-3	EM logcon3-1	EM logcon3-2	EM logcon3-3
p	0.061(0.0056)	0.023(0.0018)	0.042(0.0037)	0.123(0.0177)	0.057(0.0061)	0.091(0.013)
μ_0	-0.001(0.0229)	-0.011(0.0093)	-0.009(0.0154)	-0.006(0.0131)	-0.015(0.0106)	-0.011(0.0144)
σ_0	-0.085(0.0117)	-0.033(0.0058)	-0.059(0.0087)	-0.178(0.0392)	-0.083(0.0145)	-0.128(0.0278)
d_{L2}	0.072	0.0444	0.0574	0.1101	0.0633	0.0886
d_{KL}	0.031	0.0128	0.0214	0.0743	0.0273	0.0563

Table 2.9: Bias(MSE) of estimates of $p/\boldsymbol{\theta}/\mu_1$ and mean of d_{L2} and d_{KL} when $p = 0.2$, $n = 250,\ K = 200$.

model 2: $g(x) = (1-p)\exp(\lambda_0 = 2) + pN(\mu_1 = 3, \sigma_1 = 0.5),\ \theta_0 = \lambda_0$

method	$p_{\min}$	MLE	oracle	TLE(0.1)	TLE(0.2)	TLE(0.3)
p	0.066(0.0059)					
λ_0	-0.116(0.0843)	-0.995(0.9944)	0.005(0.0217)	-0.624(0.4061)	0.037(0.0533)	0.841(0.7777)
μ_1	-0.289(0.1041)					
d_{L2}	0.0868	0.4062	0.041	0.2413	0.066	0.2684
d_{KL}	0.0118	0.193	0.0026	0.0661	0.0064	0.0741
method	EM logcon2-1	EM logcon2-2	EM logcon2-3	EM logcon3-1	EM logcon3-2	EM logcon3-3
p	0.003(0.001)	-0.013(0.001)	0.005(0.0013)	0.004(0.001)	-0.002(8e-04)	0.01(0.0015)
λ_0	0.041(0.0553)	-0.086(0.0611)	0.044(0.0653)	0.036(0.0514)	-0.001(0.0444)	0.079(0.0717)
μ_1	-0.02(0.0154)	0.046(0.0119)	0.046(0.0119)			
d_{L2}	0.0619	0.0712	0.0667	0.058	0.0585	0.0696
d_{KL}	0.0065	0.0083	0.0076	0.0059	0.0054	0.008

Table 2.10: Bias(MSE) of estimates of $p/\boldsymbol{\theta}/\mu_1$ and mean of d_{L2} and d_{KL} when $p = 0.2$, $n = 250,\ K = 200$.

model 3: $g(x) = (1-p)\exp(\lambda_0 = 2) + p(\exp(\lambda_1 = 2) + 3),\ \theta_0 = \lambda_0$

method	$p_{\min}$	MLE	oracle	TLE(0.1)	TLE(0.2)	TLE(0.3)
p	0.069(0.0062)					
λ_0	-0.1(0.057)	-1.084(1.1806)	0.008(0.021)	-0.725(0.5468)	0.004(0.0838)	0.855(0.8184)
μ_1	-0.383(0.1733)					
d_{L2}	0.0697	0.4495	0.0398	0.2847	0.0831	0.2717
d_{KL}	0.0076	0.2419	0.0025	0.094	0.0106	0.0772
method	EM logcon2-1	EM logcon2-2	EM logcon2-3	EM logcon3-1	EM logcon3-2	EM logcon3-3
p	-0.001(7e-04)	-0.007(7e-04)	-0.001(7e-04)	0.001(6e-04)	-0.003(7e-04)	0.001(8e-04)
λ_0	0.005(0.0258)	-0.062(0.0375)	0.004(0.0268)	0.023(0.0249)	-0.008(0.0259)	0.024(0.0345)
μ_1	0.003(0.006)	0.014(0.0077)	0.014(0.0077)			
d_{L2}	0.0445	0.0559	0.0459	0.0432	0.0452	0.0473
d_{KL}	0.0033	0.0051	0.0034	0.003	0.0032	0.0039

Table 2.11: Bias(MSE) of estimates of $p/\boldsymbol{\theta}/\mu_1$ and mean of d_{L2} and d_{KL} when $p = 0.2$, $n = 250,\ K = 200$.

model 4: $g(x) = (1-p)\text{gamma}(\text{shape}_0 = 2, \text{scale}_0 = 0.5) + p(F(d_1 = 100, d_2 = 100) + 5)$, $\boldsymbol{\theta_0} = (\text{shape}_0, \text{scale}_0)$

method	$p_{\min}$	MLE	oracle	TLE(0.1)	TLE(0.2)	TLE(0.3)
p	0.053(0.0042)					
$shape_0$	-0.085(0.0758)	-0.102(0.0664)	0.015(0.0364)	-1.121(1.2608)	-0.059(0.4025)	1.354(2.0406)
$scale_0$	0.055(0.011)	0.06(0.0109)	0.001(0.0029)	1.26(1.6315)	0.131(0.1234)	-0.248(0.0628)
μ_1	-0.551(0.3588)					
d_{L2}	0.067	0.3275	0.0482	0.3518	0.1321	0.232
d_{KL}	0.0118	0.3048	0.0056	0.2316	0.0455	0.1667
method	EM logcon2-1	EM logcon2-2	EM logcon2-3	EM logcon3-1	EM logcon3-2	EM logcon3-3
p	-0.001(7e-04)	-0.005(6e-04)	-0.002(7e-04)	0.01(9e-04)	-0.001(7e-04)	0.005(8e-04)
$shape_0$	0.02(0.0378)	-0.059(0.0526)	-0.007(0.0407)	0.133(0.1036)	0.021(0.0409)	0.078(0.0776)
$scale_0$	-0.001(0.0031)	0.037(0.0094)	0.011(0.0043)	-0.018(0.0043)	-0.001(0.0032)	-0.008(0.0039)
μ_1	-0.001(8e-04)	0.006(0.001)	0.006(0.001)			
d_{L2}	0.0488	0.0575	0.0505	0.0619	0.0491	0.0566
d_{KL}	0.0057	0.009	0.0063	0.0099	0.0058	0.0082

Table 2.12: Bias(MSE) of estimates of $p/\boldsymbol{\theta}/\mu_1$ and mean of d_{L2} and d_{KL} when $p = 0.2$, $n = 250,\ K = 200$.

model 5: $g(x) = (1-p)\text{Weibull}(\text{shape}_0 = 2, \text{scale}_0 = 1) + p(\text{beta}(0.5, 0.5) + 2)$, $\boldsymbol{\theta} = (\text{shape}_0, \text{scale}_0)$

method	$p_{\min}$	MLE	oracle	TLE(0.1)	TLE(0.2)	TLE(0.3)
p	0.042(0.0032)					
$shape_0$	-0.156(0.0664)	-1.702(3.2258)	0.009(0.0124)	-0.193(0.0454)	0.073(0.0267)	0.52(0.3176)
$scale_0$	0.089(0.0178)	0.671(1.4677)	0.002(0.0013)	0.165(0.0304)	0.004(0.0031)	-0.096(0.0116)
μ_1	-0.282(0.133)					
d_{L2}	0.1183	0.2705	0.0474	0.1556	0.0694	0.2171
d_{KL}	0.0414	0.1661	0.0048	0.0566	0.0131	0.1571
method	EM logcon2-1	EM logcon2-2	EM logcon2-3	EM logcon3-1	EM logcon3-2	EM logcon3-3
p	-0.006(0.0022)	-0.044(0.0037)	0.001(0.0021)	0.016(0.0028)	0.003(0.0021)	0.021(0.0026)
$shape_0$	0.004(0.0244)	-0.1(0.0304)	0.009(0.0263)	0.08(0.0461)	0.004(0.0272)	0.063(0.041)
$scale_0$	0.013(0.005)	0.075(0.012)	0.005(0.0049)	0.017(0.0079)	0.01(0.0055)	-0.007(0.0047)
μ_1	0.017(0.0167)	0.11(0.0299)	0.108(0.0299)			
d_{L2}	0.0692	0.1003	0.078	0.0947	0.0785	0.0825
d_{KL}	0.0153	0.0274	0.0187	0.0276	0.0197	0.0237

Table 2.13: Bias(MSE) of estimates of $p/\boldsymbol{\theta}$ and mean of d_{L2} and d_{KL} when $p_0 = 0.8$, $p_1 = 0.05$, $p_2 = 0.15$, $n = 250$, $K = 200$.

model 6: $g(x) = (1-p)N(\mu_0 = 0, \sigma_0 = 1) + p_1 U(10, 11) + p_2 U(-5, -4)$, $\boldsymbol{\theta} = (\mu_0, \sigma_0)$

method	$p_{\min}$	MLE	oracle	TLE(0.1)	TLE(0.2)	TLE(0.3)
p	0.043(0.0034)					
μ_0	-0.003(0.0084)	-0.141(0.0583)	-0.002(0.005)	-0.486(0.254)	-0.05(0.0124)	-0.004(0.0103)
σ_0	0.082(0.0155)	8.299(71.2154)	0.006(0.0028)	0.653(0.4408)	0.047(0.0228)	-0.244(0.0625)
d_{L2}	0.0555	0.471	0.0318	0.2278	0.0602	0.1459
d_{KL}	0.0159	1.7233	0.0052	0.2307	0.0221	0.116
method	EM logcon2-1	EM logcon2-2	EM logcon2-3	EM logcon3-1	EM logcon3-2	EM logcon3-3
p	-0.149(0.0225)	-0.155(0.0243)	-0.154(0.0239)	-0.003(6e-04)	-0.016(0.001)	-0.003(8e-04)
μ_0	-0.697(0.5057)	-0.633(0.4172)	-0.646(0.4352)	-0.007(0.0058)	-0.075(0.0165)	-0.012(0.0056)
σ_0	0.875(0.7742)	1.034(1.1244)	1.002(1.0439)	0.01(0.005)	0.122(0.0333)	0.02(0.0063)
d_{L2}	0.2767	0.286	0.2841	0.0349	0.067	0.0375
d_{KL}	0.3417	0.3814	0.3731	0.0071	0.0271	0.0081

Table 2.14: Bias(MSE) of estimates of $p/\boldsymbol{\theta}$ and mean of d_{L2} and d_{KL} when $p_0 = 0.8$, $p_1 = 0.05$, $p_2 = 0.15$, $n = 250$, $K = 200$.

model 7: $g(x) = (1-p)\text{logistic}(\mu_0 = 0, s_0 = 1) + p_1 U(-11, -10) + p_2 Pareto(m_2 = 5, s_2 = 5)$, $\boldsymbol{\theta} = (\mu_0, s_0)$

method	$p_{\min}$	MLE	oracle	TLE(0.1)	TLE(0.2)	TLE(0.3)
p	0.045(0.0032)					
μ_0	0.017(0.0255)	0.474(0.2635)	0.007(0.0139)	0.662(0.4779)	0.101(0.0418)	0.007(0.021)
s_0	0.078(0.0179)	0.891(0.8101)	-0.003(0.0038)	0.319(0.1094)	-0.034(0.0116)	-0.301(0.0941)
d_{L2}	0.0426	0.1826	0.0265	0.1274	0.0471	0.1351
d_{KL}	0.0139	0.2431	0.005	0.1008	0.0163	0.1208
method	EM logcon2-1	EM logcon2-2	EM logcon2-3	EM logcon3-1	EM logcon3-2	EM logcon3-3
p	-0.133(0.0193)	-0.095(0.0102)	-0.083(0.0083)	-0.003(8e-04)	-0.049(0.0031)	-0.006(0.001)
μ_0	0.565(0.4284)	0.126(0.1169)	0.071(0.1164)	0.004(0.0181)	0.181(0.0647)	0.012(0.0201)
s_0	0.509(0.2797)	0.495(0.273)	0.454(0.2282)	0.001(0.0072)	0.168(0.0422)	0.017(0.0093)
d_{L2}	0.1423	0.1241	0.1176	0.0307	0.0618	0.0335
d_{KL}	0.1404	0.1121	0.0999	0.0077	0.029	0.0091

2.7.4 Source code

R code for two-component EM log-concave method

```
#####################################################################

##EM_logcon2: 2-comp EM algorithm from initial probabilities ini_w0##

#####################################################################

EM_logcon2<-function(x,ini_w0,knowndist,iteration){

  n<-length(x)

  w0<-ini_w0

  w1<-1-w0

  w1[which(w1<10^-3)]<-0

  w0<-1-w1

  lold<-(-10^5)

  l<-(lold+100)

  ite<-0

  while((abs(l-lold)/abs(lold)>10^-6)&&(ite<iteration)){

    ite<-ite+1

    lold<-l

    p0<-sum(w0)/n##update mixing proportion

    p1<-(1-p0)

    ##assume the proportion for the known component>0.5

    if(p0<=0.5){

      w0<-w1
      w1<-1-w0

      w1[which(w1<10^-3)]<-0

      w0<-1-w1

      p0<-sum(w0)/n

      p1<-(1-p0)

    }

    ##assume the proportion of the unknown component>=0.02

    if((p1)<0.02){

      p0<-1

      p1<-0

      w0<-rep(1,n)

      w1<-rep(0,n)

      if(knowndist=="normal"){

        mu0<-mean(x)

        sigma0<-sqrt(mean((x-mu0)^2))

        ##f0: density estimation of the 1st known component

        f0<-dnorm(x,mean=mu0,sd=sigma0)

        theta0<-c(mu0,sigma0)

      }

      if(knowndist=="exponential"){

        lambda0<-n/sum(x)

        f0<-dexp(x,rate=lambda0)
        theta0<-lambda0
        mu0<-1/theta0
      }
      if(knowndist=="gamma"){
        scale0<-var(x)/mean(x)
        shape0<-mean(x)/scale0
        gammalik<-function(theta){
          result<-(-sum(log(dgamma(x,shape=theta[1],scale=theta[2]))))
          return(result)
        }
        theta0<-optim(par=c(shape0,scale0),fn=gammalik, method="L-BFGS-B",
            ↪ lower=c(shape0-0.9*shape0,scale0-0.9*scale0),upper=c(shape0
            ↪ +0.9*shape0,scale0+0.9*scale0))$par
        f0<-dgamma(x,shape=theta0[1],scale=theta0[2])
      }
      if(knowndist=="logistic"){
        mu0<-mean(x)
        scale0<-sqrt(var(x)*3/pi^2)
        gammalik<-function(theta){
          result<-(-sum(log(dlogis(x,location=theta[1],scale=theta[2]))))
          return(result)
        }
        theta0<-optim(par=c(mu0,scale0),fn=gammalik, method="L-BFGS-B",
            ↪ lower=c(mu0-2,scale0-0.9*scale0),upper=c(mu0+2,scale0+0.9*
            ↪ scale0))$par
        f0<-dlogis(x,location=theta0[1],scale=theta0[2])
      }
      if(knowndist=="weibull"){
        tempf<-function(k){
          result<-gamma(1+2/k)-(var(x)/mean(x)^2+1)*gamma(1+1/k)^2
          return(result)
        }
        shape0<-max(0.1,multiroot(f=tempf,start=1)$root)
        scale0<-max(0.1,mean(x)/gamma(1+1/shape0))
        weibulllik<-function(theta){
          result<-(-sum(log(dweibull(x,shape=theta[1],scale=theta[2]))))
          return(result)
        }
        theta0<-nlm(weibulllik, p = c(shape0,scale0), hessian=TRUE)$estimate
        f0<-dweibull(x,shape=theta0[1],scale=theta0[2])
      }
      l<-sum(log(f0))
      mu1<-"NA"
      fit1<-"NA"
```

```
  break
}
if(knowndist=="normal"){
  mu0<-sum(w0*x)/sum(w0)
  sigma0<-sqrt(sum(w0*(x-mu0)^2)/sum(w0))
  f0<-dnorm(x,mean=mu0,sd=sigma0)
  theta0<-c(mu0,sigma0)
}
if(knowndist=="exponential"){
  lambda0<-sum(w0)/sum(w0*x)
  f0<-dexp(x,rate=lambda0)
  theta0<-lambda0
  mu0<-1/theta0
}
if(knowndist=="gamma"){
  mu0<-sum(w0*x)/sum(w0)
  var0<-sum(w0*(x-mu0)^2)/sum(w0)
  scale0<-var0/mu0
  shape0<-mu0/scale0
    gammalik<-function(theta){
      result<-(-sum(w0*log(dgamma(x,shape=theta[1],scale=theta[2]))))
      return(result)
  }
 theta0<-optim(par=c(shape0,scale0),fn=gammalik, method="L-BFGS-B",
   ↪ lower=c(shape0-0.9*shape0,scale0-0.9*scale0),upper=c(shape0
   ↪ +0.9*shape0,scale0+0.9*scale0))$par
 f0<-dgamma(x,shape=theta0[1],scale=theta0[2])
}
if(knowndist=="logistic"){
 mu0<-sum(w0*x)/sum(w0)
 var0<-sum(w0*(x-mu0)^2)/sum(w0)
 scale0<-sqrt(var0*3/pi^2)
 gammalik<-function(theta){
  result<-(-sum(w0*log(dlogis(x,location=theta[1],scale=theta[2]))))
  return(result)
 }
 theta0<-optim(par=c(mu0,scale0),fn=gammalik, method="L-BFGS-B", lower=
   ↪ c(mu0-2,scale0-0.9*scale0),upper=c(mu0+2,scale0+0.9*scale0))
   ↪ $par
 f0<-dlogis(x,location=theta0[1],scale=theta0[2])
}
if(knowndist=="weibull"){
 mu0<-sum(w0*x)/sum(w0)
 var0<-sum(w0*(x-mu0)^2)/sum(w0)
```

```
  tempf<-function(k){
    result<-gamma(1+2/k)-(var0/mu0^2+1)*gamma(1+1/k)^2
    return(result)
  }
  shape0<-max(0.1,multiroot(f=tempf,start=1)$root)
  scale0<-max(0.1,mu0/gamma(1+1/shape0))
  weibulllik<-function(theta){
    result<-(-sum(w0*log(dweibull(x,shape=theta[1],scale=theta[2]))))
    return(result)
  }
  theta0<-nlm(weibulllik, p = c(shape0,scale0), hessian=TRUE)$estimate
  f0<-dweibull(x,shape=theta0[1],scale=theta0[2])
}
x1<-cbind(x,w1)
x1<-x1[x1[,2]>0,]

x1<-x1[order(x1[,1]),]
fit1<-activeSetLogCon(x=x1[,1],w=x1[,2])
mu1<-sum((fit1$x)*(fit1$w))/sum(fit1$w)
##f1: density estimation of the 2nd component
f1<-evaluateLogConDens(xs=x,res=fit1)[,3]
```

```
 w0<-p0*f0/(p0*f0+p1*f1)##update probabilities
 ##if for some point, both f0 and f1 are 0, then we try to determine
   ↪ which component is it more close to.
 temp_index<-which(w0=="NaN")
 if(length(temp_index)>0){
  for(i in 1:length(temp_index)){
   if(abs(x[temp_index[i]]-mu0)>abs(x[temp_index[i]]-mu1)){
    w0[temp_index[i]]<-0
   }else{
    w0[temp_index[i]]<-1
   }
  }
 }
 w1<-(1-w0)
 w1[which(w1<10^-3)]<-0
 w0<-1-w1
 l<-sum(log(p0*f0+p1*f1))
}

res<-list(ite=ite,p=c(p0,p1),w=cbind(w0,w1),theta=theta0,mu=mu1, L=l,fit=
  ↪ fit1)
return(res)
```

```
}

#######################################################
##maximum likelihood estimation from random initials##
#######################################################
mle_logcon2<-function(data,knowndist,ite1, ite2){
  data.1<-data
  n1<-length(data.1)
  ini_num<-50
  fit<-list()
  likelihood<-numeric(ini_num)
  pvalue<-numeric(ini_num)
  cdf_dist<-numeric(ini_num)
  ratio<-numeric(ini_num)##maximum ratio of pf(x)/((1-p)f0(x)+pf(x)) at
      ↪ maximal central part of f0(x)
  ite<-0
  for(j in 1:500){
    ##randomly generate initial parameters for the known component
    if(knowndist=="normal"){
      mu0<-runif(1,min=quantile(data.1,probs=0.25),max=quantile(data.1,probs
          ↪ =0.75))
      sigma0<-runif(1,0.1*sd(data.1),sd(data.1))
  f0<-dnorm(data.1,mean=mu0,sd=sigma0)
}
if(knowndist=="exponential"){
  mu0<-runif(1,min=quantile(data.1,probs=0.25),max=quantile(data.1,probs
      ↪ =0.75))
  lambda0<-1/max(mu0,0.1)
  f0<-dexp(data.1,rate=lambda0)
}
if(knowndist=="gamma"){
  mu0<-runif(1,min=quantile(data.1,probs=0.25),max=quantile(data.1,probs
      ↪ =0.75))
  mu0<-max(mu0,0.1)
  sigma0<-runif(1,0.1*sd(data.1),sd(data.1))
  scale0<-sigma0^2/mu0
  shape0<-mu0/scale0
  f0<-dgamma(data.1,shape=shape0,scale=scale0)
}
if(knowndist=="logistic"){
  mu0<-runif(1,min=quantile(data.1,probs=0.25),max=quantile(data.1,probs
      ↪ =0.75))
  sigma0<-runif(1,0.1*sd(data.1),sd(data.1))
  scale0<-sqrt(sigma0^2*3/pi^2)
  f0<-dlogis(data.1,location=mu0,scale=scale0)
}
if(knowndist=="weibull"){
  mu0<-runif(1,min=quantile(data.1,probs=0.25),max=quantile(data.1,probs
      ↪ =0.75))
  sigma0<-runif(1,0.1*sd(data.1),sd(data.1))
  tempf<-function(k){
    result<-gamma(1+2/k)-(sigma0^2/mu0^2+1)*gamma(1+1/k)^2
    return(result)
  }
  shape0<-max(0.1,multiroot(f=tempf,start=1)$root)
  scale0<-max(0.1,mean(data.1)/gamma(1+1/shape0))
  f0<-dweibull(data.1,shape=shape0,scale=scale0)
}


##randomly select one-sided outlier points
perc_outlier<-runif(n=1,min=0,max=0.3)
temp_sign<-rbinom(n=1,size=1,prob=0.5)
if(temp_sign==0){
  f0_1<-f0
  f0_1[which(data.1>mu0)]<-1
  w1<-numeric(n1)
  w1[order(f0_1)[1:round(perc_outlier*n1)]]<-1
}else{
  f0_2<-f0
  f0_2[which(data.1<mu0)]<-1
  w1<-numeric(n1)
  w1[order(f0_2)[1:round(perc_outlier*n1)]]<-1
}

w0<-1-w1
fit_temp<-EM_logcon2(x=data.1, ini_w0=w0,knowndist=knowndist,iteration=
    ↪ ite1)
ratio_temp<-0
if(fit_temp$p[2]>0){
  if(knowndist=="normal"){
    index_temp<-intersect(which(data.1>fit_temp$theta[1]+qnorm(0.1)*
        ↪ fit_temp$theta[2]),which(data.1<fit_temp$theta[1]+qnorm(0.9)*
        ↪ fit_temp$theta[2]))
  }
  if(knowndist=="exponential"){
    index_temp<-intersect(which(data.1>0),which(data.1<qexp(0.8,rate=
        ↪ fit_temp$theta[1])))
  }
```

```
if(knowndist=="gamma"){
  index_temp<-intersect(which(data.1>qgamma(0.1,shape=fit_temp$theta
      ↪ [1],scale=fit_temp$theta[2])),which(data.1<qgamma(0.9,shape=
      ↪ fit_temp$theta[1],scale=fit_temp$theta[2])))
}
if(knowndist=="logistic"){
  index_temp<-intersect(which(data.1>qlogis(0.1,location=
      ↪ fit_temp$theta[1],scale=fit_temp$theta[2])),which(data.1<
      ↪ qlogis(0.9,location=fit_temp$theta[1],scale=fit_temp$theta
      ↪ [2])))
}
if(knowndist=="weibull"){
  if(fit_temp$theta[1]<=1){
    index_temp<-intersect(which(data.1>0),which(data.1<qweibull(0.8,
        ↪ shape=fit_temp$theta[1],scale=fit_temp$theta[2])))
  }else{
    index_temp<-intersect(which(data.1>qweibull(0.1,shape=
        ↪ fit_temp$theta[1],scale=fit_temp$theta[2])),which(data.1<
        ↪ qweibull(0.9,shape=fit_temp$theta[1],scale=fit_temp$theta
        ↪ [2])))
  }
}
```

```
  ratio_temp<-max(fit_temp$w[index_temp,2])
}

##select the fitted models with ratio_temp<0.2
if(ratio_temp<0.2){
  ite<-ite+1
  fit[[ite]]<-fit_temp
  ratio[ite]<-ratio_temp
  likelihood[ite]<-fit[[ite]]$L
  pvalue[ite]<-fit[[ite]]$p[2]
  if(fit_temp$p[2]>0){
    cdf_unknown<-evaluateLogConDens(xs=sort(data.1),res=fit_temp$fit,
       ↪ which=3)[,4]
  }else{
    cdf_unknown<-0
  }
  if(knowndist=="normal"){
    cdf_known<-pnorm(sort(data.1),mean=fit_temp$theta[1],sd=
       ↪ fit_temp$theta[2])
  }
  if(knowndist=="exponential"){
    cdf_known<-pexp(sort(data.1),rate=fit_temp$theta[1])
```

```
   }
   if(knowndist=="gamma"){
    cdf_known<-pgamma(sort(data.1),shape=fit_temp$theta[1],scale=
      ↪ fit_temp$theta[2])
   }
   if(knowndist=="logistic"){
    cdf_known<-plogis(sort(data.1),location=fit_temp$theta[1],scale=
      ↪ fit_temp$theta[2])
   }
   if(knowndist=="weibull"){
    cdf_known<-pweibull(sort(data.1),shape=fit_temp$theta[1],scale=
      ↪ fit_temp$theta[2])
   }
   cdf<-fit_temp$p[1]*cdf_known+fit_temp$p[2]*cdf_unknown
   cdf_dist[ite]<-mean((cdf-seq(1/n1,1,1/n1))^2)
   if(ite==ini_num){
    break
   }
  }
 }

##select the top 10 fitted models and run EM algorithm until convergence
```

```
top_num<-10
##criteria: MLE
likelihood<-likelihood[1:ite]
index_lik<-order(-likelihood)[1:min(top_num,ite)]
n_1<-length(index_lik)
fit_1<-list()
likelihood_1<-numeric(n_1)
ratio_1<-numeric(n_1)##maximum ratio of pf(x)/((1-p)f0(x)+pf(x)) at maximal
    ↪  central part of f0(x)
for(j in 1:n_1){
  weight_temp<-fit[[index_lik[j]]]$w
  fit_1[[j]]<-EM_logcon2(x=data.1, ini_w0=weight_temp[,1], knowndist=
      ↪ knowndist,iteration=ite2)
  likelihood_1[j]<-fit_1[[j]]$L
  if(fit_1[[j]]$p[2]>0){
    if(knowndist=="normal"){
      index_temp<-intersect(which(data.1>fit_1[[j]]$theta[1]+qnorm(0.1)*
          ↪ fit_1[[j]]$theta[2]),which(data.1<fit_1[[j]]$theta[1]+qnorm
          ↪ (0.9)*fit_1[[j]]$theta[2]))
    }
    if(knowndist=="exponential"){
```

```
  index_temp<-intersect(which(data.1>0),which(data.1<qexp(0.8,rate=fit_1
      ↪ [[j]]$theta[1])))
}
if(knowndist=="gamma"){
  index_temp<-intersect(which(data.1>qgamma(0.1,shape=fit_1[[j]]$theta
      ↪ [1],scale=fit_1[[j]]$theta[2])),which(data.1<qgamma(0.9,shape=
      ↪ fit_1[[j]]$theta[1],scale=fit_1[[j]]$theta[2])))
}
if(knowndist=="logistic"){
  index_temp<-intersect(which(data.1>qlogis(0.1,location=fit_1[[j]]
      ↪ $theta[1],scale=fit_1[[j]]$theta[2])),which(data.1<qlogis(0.9,
      ↪ location=fit_1[[j]]$theta[1],scale=fit_1[[j]]$theta[2])))
}
if(knowndist=="weibull"){
  if(fit_1[[j]]$theta[1]<=1){
    index_temp<-intersect(which(data.1>0),which(data.1<qweibull(0.8,
        ↪ shape=fit_1[[j]]$theta[1],scale=fit_1[[j]]$theta[2])))
  }else{
    index_temp<-intersect(which(data.1>qweibull(0.1,shape=fit_1[[j]]
        ↪ $theta[1],scale=fit_1[[j]]$theta[2])),which(data.1<qweibull
        ↪ (0.9,shape=fit_1[[j]]$theta[1],scale=fit_1[[j]]$theta[2])))
  }
```

```
    }
    ratio_1[j]<-max(fit_1[[j]]$w[index_temp,2])
  }
}
index_1<-which(ratio_1<0.2)
ite_temp<-0
while(length(index_1)==0){
  ite_temp<-ite_temp+1
  if(ite<=(top_num*ite_temp)){
    stop("error: no possible fitted models are found for MLE criteria")
  }else{
    index_lik<-order(-likelihood)[(top_num*ite_temp+1):min(top_num*(
        ↪ ite_temp+1),ite)]
    n_1<-length(index_lik)
    fit_1<-list()
    likelihood_1<-numeric(n_1)
    ratio_1<-numeric(n_1)##maximum ratio of pf(x)/((1-p)f0(x)+pf(x)) at
        ↪ maximal central part of f0(x)
    for(j in 1:n_1){
      weight_temp<-fit[[index_lik[j]]]$w
      fit_1[[j]]<-EM_logcon2(x=data.1, ini_w0=weight_temp[,1], knowndist=
          ↪ knowndist,iteration=ite2)
```

```
likelihood_1[j]<-fit_1[[j]]$L
if(fit_1[[j]]$p[2]>0){
  if(knowndist=="normal"){
    index_temp<-intersect(which(data.1>fit_1[[j]]$theta[1]+qnorm(0.1)*
        ↪ fit_1[[j]]$theta[2]),which(data.1<fit_1[[j]]$theta[1]+qnorm
        ↪ (0.9)*fit_1[[j]]$theta[2]))
  }
  if(knowndist=="exponential"){
    index_temp<-intersect(which(data.1>0),which(data.1<qexp(0.8,rate=
        ↪ fit_1[[j]]$theta[1])))
  }
  if(knowndist=="gamma"){
    index_temp<-intersect(which(data.1>qgamma(0.1,shape=fit_1[[j]]
        ↪ $theta[1],scale=fit_1[[j]]$theta[2])),which(data.1<qgamma
        ↪ (0.9,shape=fit_1[[j]]$theta[1],scale=fit_1[[j]]$theta[2])))
  }
  if(knowndist=="logistic"){
    index_temp<-intersect(which(data.1>qlogis(0.1,location=fit_1[[j]]
        ↪ $theta[1],scale=fit_1[[j]]$theta[2])),which(data.1<qlogis
        ↪ (0.9,location=fit_1[[j]]$theta[1],scale=fit_1[[j]]$theta
        ↪ [2])))
  }
```

```
        if(knowndist=="weibull"){
          if(fit_1[[j]]$theta[1]<=1){
            index_temp<-intersect(which(data.1>0),which(data.1<qweibull(0.8,
              ↪ shape=fit_1[[j]]$theta[1],scale=fit_1[[j]]$theta[2])))
          }else{
            index_temp<-intersect(which(data.1>qweibull(0.1,shape=fit_1[[j]]
              ↪ $theta[1],scale=fit_1[[j]]$theta[2])),which(data.1<
              ↪ qweibull(0.9,shape=fit_1[[j]]$theta[1],scale=fit_1[[j]]
              ↪ $theta[2])))
          }
        }
        ratio_1[j]<-max(fit_1[[j]]$w[index_temp,2])
      }
    }
    index_1<-which(ratio_1<0.2)
  }
}
indexa<-order(-likelihood_1[index_1])
fit_mle<-fit_1[[index_1[indexa]]]


##criteria: min(p)
cdf_dist<-cdf_dist[1:ite]
```

```
index_minprop<-order(cdf_dist)[1:min(top_num,ite)]
n_2<-length(index_minprop)
fit_2<-list()
ratio_2<-numeric(n_2)##maximum ratio of pf(x)/((1-p)f0(x)+pf(x)) at maximal
   ↪  central part of f0(x)
cdf_dist_2<-numeric(n_2)
prop_2<-numeric(n_2)
for(j in 1:n_2){
  weight_temp<-fit[[index_minprop[j]]]$w
  fit_2[[j]]<-EM_logcon2(x=data.1, ini_w0=weight_temp[,1], knowndist=
     ↪ knowndist,iteration=ite2)
  prop_2[j]<-fit_2[[j]]$p[2]
  if(fit_2[[j]]$p[2]>0){
    if(knowndist=="normal"){
      index_temp<-intersect(which(data.1>fit_2[[j]]$theta[1]+qnorm(0.1)*
         ↪ fit_2[[j]]$theta[2]),which(data.1<fit_2[[j]]$theta[1]+qnorm
         ↪ (0.9)*fit_2[[j]]$theta[2]))
    }
    if(knowndist=="exponential"){
      index_temp<-intersect(which(data.1>0),which(data.1<qexp(0.8,rate=fit_2
         ↪ [[j]]$theta[1])))
    }
```

```
 if(knowndist=="gamma"){
  index_temp<-intersect(which(data.1>qgamma(0.1,shape=fit_2[[j]]$theta
    ↪ [1],scale=fit_2[[j]]$theta[2])),which(data.1<qgamma(0.9,shape=
    ↪ fit_2[[j]]$theta[1],scale=fit_2[[j]]$theta[2])))
 }
 if(knowndist=="logistic"){
  index_temp<-intersect(which(data.1>qlogis(0.1,location=fit_2[[j]]
    ↪ $theta[1],scale=fit_2[[j]]$theta[2])),which(data.1<qlogis(0.9,
    ↪ location=fit_2[[j]]$theta[1],scale=fit_2[[j]]$theta[2])))
 }
 if(knowndist=="weibull"){
  if(fit_2[[j]]$theta[1]<=1){
   index_temp<-intersect(which(data.1>0),which(data.1<qweibull(0.8,
     ↪ shape=fit_2[[j]]$theta[1],scale=fit_2[[j]]$theta[2])))
  }else{
   index_temp<-intersect(which(data.1>qweibull(0.1,shape=fit_2[[j]]
     ↪ $theta[1],scale=fit_2[[j]]$theta[2])),which(data.1<qweibull
     ↪ (0.9,shape=fit_2[[j]]$theta[1],scale=fit_2[[j]]$theta[2])))
  }
 }
 ratio_2[j]<-max(fit_2[[j]]$w[index_temp,2])
}
```

```
if(fit_2[[j]]$p[2]>0){
  cdf_unknown<-evaluateLogConDens(xs=sort(data.1),res=fit_2[[j]]$fit,
      ↪ which=3)[,4]
}else{
  cdf_unknown<-0
}
if(knowndist=="normal"){
  cdf_known<-pnorm(sort(data.1),mean=fit_2[[j]]$theta[1],sd=fit_2[[j]]
      ↪ $theta[2])
}
if(knowndist=="exponential"){
  cdf_known<-pexp(sort(data.1),rate=fit_2 [[j]]$theta[1])
}
if(knowndist=="gamma"){
  cdf_known<-pgamma(sort(data.1),shape=fit_2[[j]]$theta[1],scale=fit_2[[j
      ↪ ]]$theta[2])
}
if(knowndist=="logistic"){
  cdf_known<-plogis(sort(data.1),location=fit_2[[j]]$theta[1],scale=fit_2
      ↪ [[j]]$theta[2])
}
if(knowndist=="weibull"){
  cdf_known<-pweibull(sort(data.1),shape=fit_2[[j]]$theta[1],scale=fit_2
    ↪ [[j]]$theta[2])
 }
 cdf<-fit_2[[j]]$p[1]*cdf_known+fit_2[[j]]$p[2]*cdf_unknown
 cdf_dist_2[j]<-mean((cdf-seq(1/n1,1,1/n1))^2)
}
#index_2<-intersect(which(ratio_2<0.2),which(cdf_dist_2<0.001))
index_2<-which(ratio_2<0.2)
ite_temp<-0
while(length(index_2)==0){
 ite_temp<-ite_temp+1
 if(ite<=(top_num*ite_temp)){
  stop("error: no possible fitted models are found for min(p) criteria")
 }else{
  index_minprop<-order(cdf_dist)[(top_num*ite_temp+1):min(top_num*(
    ↪ ite_temp+1),ite)]
  n_2<-length(index_minprop)
  fit_2<-list()
  ratio_2<-numeric(n_2)##maximum ratio of pf(x)/((1-p)f0(x)+pf(x)) at
    ↪ maximal central part of f0(x)
  cdf_dist_2<-numeric(n_2)
  prop_2<-numeric
  for(j in 1:n_2){
    weight_temp<-fit[[index_minprop[j]]]$w
    fit_2[[j]]<-EM_logcon2(x=data.1, ini_w0=weight_temp[,1], knowndist=
        ↪ knowndist,iteration=ite2)
    prop_2[j]<-fit_2[[j]]$p[2]
    if(fit_2[[j]]$p[2]>0){
      if(knowndist=="normal"){
        index_temp<-intersect(which(data.1>fit_2[[j]]$theta[1]+qnorm(0.1)*
            ↪ fit_2[[j]]$theta[2]),which(data.1<fit_2[[j]]$theta[1]+qnorm
            ↪ (0.9)*fit_2[[j]]$theta[2]))
      }
      if(knowndist=="exponential"){
        index_temp<-intersect(which(data.1>0),which(data.1<qexp(0.8,rate=
            ↪ fit_2[[j]]$theta[1])))
      }
      if(knowndist=="gamma"){
        index_temp<-intersect(which(data.1>qgamma(0.1,shape=fit_2[[j]]
            ↪ $theta[1],scale=fit_2[[j]]$theta[2])),which(data.1<qgamma
            ↪ (0.9,shape=fit_2[[j]]$theta[1],scale=fit_2[[j]]$theta[2])))
      }
      if(knowndist=="logistic"){
```

```
    index_temp<-intersect(which(data.1>qlogis(0.1,location=fit_2[[j]]
        ↪ $theta[1],scale=fit_2[[j]]$theta[2])),which(data.1<qlogis
        ↪ (0.9,location=fit_2[[j]]$theta[1],scale=fit_2[[j]]$theta
        ↪ [2])))
  }
  if(knowndist=="weibull"){
    if(fit_2[[j]]$theta[1]<=1){
      index_temp<-intersect(which(data.1>0),which(data.1<qweibull(0.8,
          ↪ shape=fit_2[[j]]$theta[1],scale=fit_2[[j]]$theta[2])))
    }else{
      index_temp<-intersect(which(data.1>qweibull(0.1,shape=fit_2[[j]]
          ↪ $theta[1],scale=fit_2[[j]]$theta[2])),which(data.1<
          ↪ qweibull(0.9,shape=fit_2[[j]]$theta[1],scale=fit_2[[j]]
          ↪ $theta[2])))
    }
  }
  ratio_2[j]<-max(fit_2[[j]]$w[index_temp,2])
}
if(fit_2[[j]]$p[2]>0){
  cdf_unknown<-evaluateLogConDens(xs=sort(data.1),res=fit_2[[j]]$fit,
      ↪ which=3)[,4]
}else{
  cdf_unknown<-0
}
if(knowndist=="normal"){
  cdf_known<-pnorm(sort(data.1),mean=fit_2[[j]]$theta[1],sd=fit_2[[j]]
      ↪ $theta[2])
}
if(knowndist=="exponential"){
  cdf_known<-pexp(sort(data.1),rate=fit_2 [[j]]$theta[1])
}
if(knowndist=="gamma"){
  cdf_known<-pgamma(sort(data.1),shape=fit_2[[j]]$theta[1],scale=fit_2
      ↪ [[j]]$theta[2])
}
if(knowndist=="logistic"){
  cdf_known<-plogis(sort(data.1),location=fit_2[[j]]$theta[1],scale=
      ↪ fit_2[[j]]$theta[2])
}
if(knowndist=="weibull"){
  cdf_known<-pweibull(sort(data.1),shape=fit_2[[j]]$theta[1],scale=
      ↪ fit_2[[j]]$theta[2])
}
cdf<-fit_2[[j]]$p[1]*cdf_known+fit_2[[j]]$p[2]*cdf_unknown
```

```
      cdf_dist_2[j]<-mean((cdf-seq(1/n1,1,1/n1))^2)
    }
    #index_2<-intersect(which(ratio_2<0.2),which(cdf_dist_2<0.001))
    index_2<-which(ratio_2<0.2)
  }
}
indexb<-intersect(order(prop_2[index_2]), which(prop_2<4*min(prop_2)))
fit_minp<-fit_2[[index_2[indexb]]]

##criteria: cdf distance
indexc<-order(cdf_dist_2[index_2])
fit_cdf<-fit_2[[index_2[indexc]]]

res<-list(fit_lik=fit_lik,fit_minp=fit_minp, fit_cdf=fit_cdf)
return(res)
}
```

Bibliography

[1] Peter J Bickel and David A Freedman. Some asymptotic theory for the bootstrap. *The annals of statistics*, pages 1196–1217, 1981.

[2] Laurent Bordes, Céline Delmas, and Pierre Vandekerkhove. Semiparametric estimation of a two-component mixture model where one component is known. *Scandinavian journal of statistics*, 33(4):733–752, 2006.

[3] Kai Lai Chung. *A course in probability theory.* Academic press, 2001.

[4] Madeleine Cule, Richard Samworth, et al. Theoretical properties of the log-concave maximum likelihood estimator of a multidimensional density. *Electronic Journal of Statistics*, 4:254–270, 2010.

[5] Madeleine Cule, Richard Samworth, and Michael Stewart. Maximum likelihood estimation of a multi-dimensional log-concave density. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(5):545–607, 2010.

[6] Arthur P Dempster, Nan M Laird, and Donald B Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

[7] Lutz Dümbgen, André Hüsler, and Kaspar Rufibach. Active set and em algorithms for log-concave densities based on complete and censored data. *arXiv preprint arXiv:0707.4643*, 2007.

[8] Lutz Dümbgen, Kaspar Rufibach, et al. Maximum likelihood estimation of a log-concave density and its distribution function: Basic properties and uniform consistency. *Bernoulli*, 15(1):40–68, 2009.

[9] Lutz Dümbgen, Richard Samworth, and Dominic Schuhmacher. Approximation by log-concave distributions, with applications to regression. *The Annals of Statistics*, pages 702–730, 2011.

[10] Bradley Efron. *Large-scale inference: empirical Bayes methods for estimation, testing, and prediction*, volume 1. Cambridge University Press, 2012.

[11] Bradley Efron et al. Size, power and false discovery rates. *The Annals of Statistics*, 35(4):1351–1377, 2007.

[12] C Field and B Smith. Robust estimation: A weighted maximum likelihood approach. *International Statistical Review/Revue Internationale de Statistique*, pages 405–424, 1994.

[13] Ali S Hadi and Alberto Luceño. Maximum trimmed likelihood estimators: a unified approach, examples, and algorithms. *Computational Statistics & Data Analysis*, 25(3):251–272, 1997.

[14] Frank R Hampel, Elvezio M Ronchetti, Peter J Rousseeuw, and Werner A Stahel. *Robust statistics*. Wiley Online Library, 1986.

[15] PJ Huber. Robust statistics. new york: John wiley and sons. *HuberRobust statistics1981*, 1981.

[16] Yanyuan Ma, Weixin Yao, et al. Flexible estimation of a semiparametric two-component mixture model with one parametric component. *Electronic Journal of Statistics*, 9(1):444–474, 2015.

[17] CL Mallows. A note on asymptotic joint normality. *The Annals of Mathematical Statistics*, pages 508–515, 1972.

[18] Marianthi Markatou, Ayanendranath Basu, and Bruce G Lindsay. Weighted likelihood equations with bootstrap root search. *Journal of the American Statistical Association*, 93(442):740–750, 1998.

[19] Ricardo A Maronna, R Douglas Martin, Victor J Yohai, and Matías Salibián-Barrera. *Robust statistics: theory and methods (with R)*. Wiley, 2018.

[20] Neyko Neykov, Peter Filzmoser, R Dimova, and Plamen Neytchev. Robust fitting of mixtures using the trimmed likelihood estimator. *Computational Statistics & Data Analysis*, 52(1):299–308, 2007.

[21] Rohit Kumar Patra and Bodhisattva Sen. Estimation of a two-component mixture model with applications to multiple testing. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(4):869–893, 2016.

[22] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017.

[23] Annie C Robin, C Reylé, S Derriere, and S Picaud. A synthetic view on structure and evolution of the milky way. *Astronomy & Astrophysics*, 409(2):523–540, 2003.

[24] Peter J Rousseeuw and Annick M Leroy. *Robust regression and outlier detection*, volume 1. Wiley Online Library, 1987.

[25] Kaspar Rufibach and Lutz Duembgen. Logcondens: estimate a log-concave probability density from iid observations. *R package version*, 2(1), 2010.

[26] Richard J Samworth. Recent progress in log-concave density estimation. *arXiv preprint arXiv:1709.03154*, 2017.

[27] Dominic Schuhmacher, André Hüsler, and Lutz Dümbgen. Multivariate log-concave distributions as a nearly parametric model. *Statistics & Risk Modeling with Applications in Finance and Insurance*, 28(3):277–295, 2011.

[28] Bernard W Silverman. *Density estimation for statistics and data analysis.* Chapman and Hall, 1986.

[29] Jongwoo Song and Dan L Nicolae. A sequential clustering algorithm with applications to gene expression data. *Journal of the Korean Statistical Society*, 38(2):175–184, 2009.

[30] Seongjoo Song, Dan L Nicolae, and Jongwoo Song. Estimating the mixing proportion in a semiparametric mixture model. *Computational Statistics & Data Analysis*, 54(10):2276–2283, 2010.

[31] John W Tukey. Mathematics and the picturing of data. In *Proceedings of the International Congress of Mathematicians, Vancouver, 1975*, volume 2, pages 523–531, 1975.

[32] David E Tyler. Robust statistics: Theory and methods, 2008.

[33] Matthew G Walker, Mario Mateo, Edward W Olszewski, Oleg Y Gnedin, Xiao Wang, Bodhisattva Sen, and Michael Woodroofe. Velocity dispersion profiles of seven dwarf spheroidal galaxies. *The Astrophysical Journal Letters*, 667(1):L53, 2007.

[34] Guenther Walther et al. Inference and modeling with log-concave distributions. *Statistical Science*, 24(3):319–327, 2009.

[35] Sijia Xiang, Weixin Yao, and Jingjing Wu. Minimum profile hellinger distance estimation for a semiparametric mixture model. *Canadian Journal of Statistics*, 42(2):246–267, 2014.

[36] Yangmei Zhou and Weixin Yao. Maximum likelihood estimation of a semiparametric two-component mixture model using log-concave approximation. *arXiv preprint arXiv:1903.11200*, 2019.