Supplementary materials for "Mini-batch learning of exponential family finite mixture models"

Hien D. Nguyen[1*], Florence Forbes[2], and Geoffrey J. McLachlan[3]

July 10, 2019

[1]Department of Mathematics and Statistics, La Trobe University, Melbourne, Victoria, Australia.

[2]Universite de Grenoble Alpes, Inria, CNRS, Grenoble INP[†], LJK, 38000 Grenoble, France.

[†]Institute of Engineering Universite de Grenoble Alpes.

[3]School of Mathematics and Physics,University of Queensland, St. Lucia, Brisbane, Australia.

[*]Corresponding author: Hien Nguyen (Email: h.nguyen5@latrobe.edu.edu.au).

Abstract

In these supplementary materials, we include various additional references and remarks as well as mathematical results that provide additional detail and context to the main article. Additional numerical results are also presented in order to demonstrate the capabilities of the presented framework.

1 Supplementary Materials for Section 1

1.1 Finite mixtures of exponential family distributions

Finite mixtures (or mixture models) of exponential family distributions appear commonly in the literature. Results regarding mixtures of exponential family distributions can be found in Hasselblad (1969), Lindsay (1983), Redner & Walker (1984), and Atienza et al. (2007). In particular, discussions regarding normal mixture models and their applications appear in McLachlan & Peel (2000, Ch. 3), Bishop (2006, Sec. 9.2 and 9.3), and Yu & Deng (2015, Ch. 2).

1.2 EM-type stochastic approximation algorithms

There is a rich literature regarding EM-type stochastic approximation algorithms, which includes the works of Titterington (1984), Celeux & Diebolt (1992), Jordan & Jacobs (1994), Celeux et al. (1996), Delyon et al. (1999), Neal & Hinton (1999), Kuhn & Lavielle (2004), and Wang & Zhao (2006). Iterating upon the previous works, Cappé & Moulines (2009) developed a general framework for constructing online EM-type stochastic approximation for ML estimation of latent data models. The developed framework is powerful and has since been extended to hidden Markov and Gibbs sampling scenarios. See Cappé (2011), Le Corff & Fort (2013a) and Le Corff & Fort (2013b), and Dupuy & Bach (2017), respectively.

1.3 Related recent works

Our exposition shares some commonalities with some of the recent literature. Firstly, Li et al. (2013a), Li et al. (2013b), and Li et al. (2014) considered the construction of online algorithms for ML estimation of normal mixtures, based on the results of Cappé & Moulines (2009). The algorithms from these articles can be seen as variants of the presented algorithms, where batches of size one are sampled without replacement, and whereupon the number of epoch (number of sweeps of the data; cf. Bengio, 2012) is fixed at one. Although the derived algorithm is correct, no results were presented regarding the convergence properties of the algorithm. Our provided convergence results can be modified to prove the convergence of the aforementioned online EM algorithms.

Secondly, Saint-Jean & Nielsen (2015) considered a class of so-called online k-ML estimator algorithms for exponential family distributions. Upon inspection, the k-ML estimators that they considered, proposed in Nielsen (2012), is the same as the classification EM algorithms that were proposed by Ganesalingam & McLachlan (1980), Celeux & Govaert (1992), and Celeux & Govaert (1993). The suggested algorithms are therefore similar to those that were proposed in Same et al. (2007). Although motivated by the construction of Cappé & Moulines (2009), the algorithms of Saint-Jean & Nielsen (2015) do not satisfy the regularity assumptions that permit the use of the convergence results. Thus, no theoretical guarantees are established for the online k-ML estimator algorithms.

2 Supplementary Materials for Section 2

2.1 The Kullback-Leibler divergence

Let $F_{\boldsymbol{\theta}}$ be the probability measure corresponding to the hypothesized PDF $f\left(\cdot;\boldsymbol{\theta}\right)$. Suppose that F_0 and $F_{\boldsymbol{\theta}}$ are both dominated by some common measure ν (cf. Pollard, 2002, Ch. 3), and let f_0 be the PDF corresponding to the measure F_0. Since both f_0 and $f\left(\cdot;\boldsymbol{\theta}\right)$ exist, we can define the

Kullback-Leibler (KL; Kullback & Leibler, 1951) divergence between f_0 and $f\left(\cdot;\boldsymbol{\theta}\right)$ as

$$K\left(f_0, f\left(\cdot;\boldsymbol{\theta}\right)\right) = \mathbb{E}_{F_0}\left[\log\frac{f_0\left(\boldsymbol{Y}\right)}{f\left(\boldsymbol{Y};\boldsymbol{\theta}\right)}\right].$$

If the KL divergence exists, then we may instead define the sets $\mathbb{W}_\Gamma$ and $\mathbb{M}_\Theta$ (from Main Text Section 2.1) as

$$\mathbb{W}_\Gamma = \left\{K\left(f_0, f\left(\cdot;\boldsymbol{\theta}\right)\right) : \boldsymbol{\theta} = \bar{\boldsymbol{\theta}}\left(\boldsymbol{s}\right), \, \boldsymbol{s}\in\Gamma\right\}$$

and

$$\mathbb{M}_\Theta = \left\{\hat{\boldsymbol{\theta}}\in\Theta : \nabla_{\boldsymbol{\theta}}\, K\left(f_0, f\left(\cdot;\boldsymbol{\theta}\right)\right)\big|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = \mathbf{0}\right\}.$$

2.2 Justification for the $N > 1$ case of the minibatch algorithm

Let F_{Emp}^N denote the probability measure corresponding to the DGP of N independent random samples from F_{Emp}^N. Since a uniformly random sample with replacement can be considered as an IID sample from this empirical measure, for $N > 1$, algorithm

$$\boldsymbol{s}^{(r)} = \boldsymbol{s}^{(r-1)} + \gamma_r\left[N^{-1}\sum_{i=1}^{N}\bar{\boldsymbol{s}}\left(\boldsymbol{Y}_i^r;\boldsymbol{\theta}^{(r-1)}\right) - \boldsymbol{s}^{(r-1)}\right], \text{ and } \boldsymbol{\theta}^{(r)} = \bar{\boldsymbol{\theta}}\left(\boldsymbol{s}^{(r)}\right) \qquad (1)$$

can be considered as solving for a root of the system

$$\frac{1}{n}\sum_{i=1}^{n}\nabla_{\boldsymbol{\theta}}\log\prod_{j=1}^{N}f\left(\cdot;\boldsymbol{\theta}\right)\Bigg|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = \mathbf{0},$$

where

$$\frac{1}{n}\sum_{i=1}^{n}\log\prod_{j=1}^{N}f\left(\cdot;\boldsymbol{\theta}\right) = \frac{N}{n}\sum_{i=1}^{n}\log f\left(\boldsymbol{y}_i;\boldsymbol{\theta}\right).$$

Thus, when $N > 1$, the algorithm defined by (1) also solves for root $\hat{\boldsymbol{\theta}}$ of the log-likelihood function, scaled by a multiplicative factor of N, which is equivalent to solving for an element in the set

$$\mathbb{M}_{\Theta}^{\text{Emp}} = \left\{ \hat{\boldsymbol{\theta}} \in \Theta : \sum_{i=1}^{n} \nabla_{\boldsymbol{\theta}} \log f\left(\boldsymbol{y}_i ; \boldsymbol{\theta}\right)|_{\boldsymbol{\theta}=\hat{\boldsymbol{\theta}}} = \mathbf{0} \right\}.$$

3 Supplementary Materials for Section 3

3.1 Convergence proof of the mini-batch EM algorithm for normal mixture models

We firstly recall the notation:

$$\left[\bar{\boldsymbol{s}}\left(\boldsymbol{y};\boldsymbol{\theta}\right)\right]^{\top} = \left(\tau_1\left(\boldsymbol{y};\boldsymbol{\theta}\right), \tau_1\left(\boldsymbol{y};\boldsymbol{\theta}\right)\boldsymbol{y}, \tau_1\left(\boldsymbol{y};\boldsymbol{\theta}\right)\boldsymbol{y}\boldsymbol{y}^{\top}, \dots, \tau_g\left(\boldsymbol{y};\boldsymbol{\theta}\right), \tau_g\left(\boldsymbol{y};\boldsymbol{\theta}\right)\boldsymbol{y}, \tau_g\left(\boldsymbol{y};\boldsymbol{\theta}\right)\boldsymbol{y}\boldsymbol{y}^{\top}\right), \qquad (2)$$

$$\tau_z\left(\boldsymbol{y};\boldsymbol{\theta}\right) = \frac{\pi_z \varphi\left(\boldsymbol{y};\boldsymbol{\mu}_z,\boldsymbol{\Sigma}_z\right)}{\sum_{\zeta=1}^{g} \pi_\zeta \varphi\left(\boldsymbol{y};\boldsymbol{\mu}_\zeta,\boldsymbol{\Sigma}_\zeta\right)}, \qquad (3)$$

and

$$\bar{\pi}_z = t_{1z}\left(\boldsymbol{y};\boldsymbol{\theta}\right), \; \bar{\boldsymbol{\mu}}_z = \frac{\boldsymbol{t}_{2z}\left(\boldsymbol{y};\boldsymbol{\theta}\right)}{t_{1z}\left(\boldsymbol{y};\boldsymbol{\theta}\right)}, \text{ and } \bar{\boldsymbol{\Sigma}}_z = \frac{\mathbf{T}_{3z}\left(\boldsymbol{y};\boldsymbol{\theta}\right)}{t_{1z}\left(\boldsymbol{y};\boldsymbol{\theta}\right)} - \frac{\boldsymbol{t}_{2z}\left(\boldsymbol{y};\boldsymbol{\theta}\right)\left[\boldsymbol{t}_{2z}\left(\boldsymbol{y};\boldsymbol{\theta}\right)\right]^{\top}}{\left[t_{1z}\left(\boldsymbol{y};\boldsymbol{\theta}\right)\right]^2}, \qquad (4)$$

from the main text.

We prove the convergence of the mini-batch EM algorithm for normal mixture models by by verifying Assumptions A1–A3, B1–B3, and C1–C3.

By Proposition 2, we have the automatic verification of A1. A2 is also verified by the forms of the elements of (2). We must also make a note that here $\mathbb{Y} = \mathbb{R}^d$ and upon writing

$$\boldsymbol{\theta}^\top = \left(\pi_1, \ldots, \pi_g, \boldsymbol{\mu}_1^\top, \ldots, \boldsymbol{\mu}_g^\top, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_g\right),$$

for convenience, we write the parameter space Θ in the form,

$$\Theta = \mathbb{D}_{g-1} \times \prod_{i=1}^{g} \mathbb{R}^d \times \prod_{i=1}^{g} \mathbb{H}_d,$$

where $\mathbb{D}_{g-1}$ is the open unit simplex of form

$$\mathbb{D}_{g-1} = \left\{ (\pi_1, \ldots, \pi_g) \in \mathbb{R}^g : \sum_{z=1}^{g} \pi_z = 1, \text{ and } \pi_z > 0, \text{ for each } z \in [g] \right\}, \tag{5}$$

and $\mathbb{H}_d$ denotes the space of positive definite symmetric matrices in $\mathbb{R}^{d \times d}$.

In order to begin validating A3, we must first define a set $\mathbb{S}$ that is suitable for the assumption. Rearranging (2), so that $\left[\bar{\boldsymbol{s}}\left(\boldsymbol{y}; \boldsymbol{\theta}\right)\right]^\top$ equals

$$\left(\tau_1\left(\boldsymbol{y}; \boldsymbol{\theta}\right), \ldots, \tau_g\left(\boldsymbol{y}; \boldsymbol{\theta}\right), \tau_1\left(\boldsymbol{y}; \boldsymbol{\theta}\right)\boldsymbol{y}^\top, \ldots, \tau_g\left(\boldsymbol{y}; \boldsymbol{\theta}\right)\boldsymbol{y}^\top, \tau_1\left(\boldsymbol{y}; \boldsymbol{\theta}\right)\boldsymbol{y}\boldsymbol{y}^\top, \ldots, \tau_g\left(\boldsymbol{y}; \boldsymbol{\theta}\right)\boldsymbol{y}\boldsymbol{y}^\top\right).$$

We can now propose that $\mathbb{S} = \Theta$ is an appropriate choice. We begin by verifying A3 (i). It is known that the unit simplex is convex. Furthermore, let $\boldsymbol{s}_\pi \in \mathbb{D}_{g-1}$ and note that

$$\left(\tau_1\left(\boldsymbol{y}; \boldsymbol{\theta}\right), \ldots, \tau_g\left(\boldsymbol{y}; \boldsymbol{\theta}\right)\right) \in \mathbb{D}_{g-1},$$

by construction this is observable by inspecting the function form of (3). Thus, the convex sum

$$(1-\gamma)\,\boldsymbol{s}_{\pi}+\gamma\left(\tau_1\left(\boldsymbol{y};\boldsymbol{\theta}\right),\dots,\tau_g\left(\boldsymbol{y};\boldsymbol{\theta}\right)\right)\in\mathbb{D}_{g-1}.$$

Next, since $\tau_z\left(\boldsymbol{y};\boldsymbol{\theta}\right)\in(0,1)$ is well-defined, for each $z\in[g]$, it is trivial to verify that

$$(1-\gamma)\,\boldsymbol{s}_{\boldsymbol{\mu}}+\gamma\begin{bmatrix}\tau_1\left(\boldsymbol{y};\boldsymbol{\theta}\right)\boldsymbol{y}\\ \vdots\\ \tau_g\left(\boldsymbol{y};\boldsymbol{\theta}\right)\boldsymbol{y}\end{bmatrix}\in\prod_{i=1}^{g}\mathbb{R}^d,$$

where $\boldsymbol{s}_{\boldsymbol{\mu}}\in\prod_{i=1}^{g}\mathbb{R}^d$. For each $z\in[g]$, assume that $\boldsymbol{s}_{\boldsymbol{\Sigma}}\in\mathbb{H}^d$. It is not possible to guarantee that $\tau_z\left(\boldsymbol{y};\boldsymbol{\theta}\right)\boldsymbol{y}\boldsymbol{y}^\top\in\mathbb{H}^d$. However, it is true that the outer product of a column vector is positive semidefinite and symmetric. Then, by the fact that the sum of a positive definite and symmetric matrix with a positive semidefinite and symmetric matrix is in $\mathbb{H}^d$ (cf. Seber, 2008, Thm. 10.58), have the final required result that, for each $z\in[g]$,

$$(1-\gamma)\,\boldsymbol{s}_{\boldsymbol{\Sigma}}+\gamma\tau_z\left(\boldsymbol{y};\boldsymbol{\theta}\right)\boldsymbol{y}\boldsymbol{y}^\top\in\mathbb{H}^d.$$

Finally, in order to complete the validation of A3, we must consider part (ii). For the mixture of normal distributions, the verification of (ii) is standard and can be conducted by combining the results of Anderson & Olkin (1985) and McLachlan & Peel (2000, Ch. 3), for instance.

We proceed to verify B1–B3. Upon inspection of the forms of Θ, $\boldsymbol{\phi}$, and ψ, we obtain the automatic verification of B1. Similarly, B2 is verified automatically upon inspection of the function forms of (4). Next, recall that for the mini-batch algorithm with any N, $F_0=F_{\mathrm{Emp}}^N$ and we write the left-hand side of

$$\sup_{\boldsymbol{s}\in\mathbb{K}} \mathbb{E}_{F_0} \left|\bar{\boldsymbol{s}}\left(\boldsymbol{Y};\bar{\boldsymbol{\theta}}\left(\boldsymbol{s}\right)\right)\right|^p < \infty,$$

as

$$\sup_{\boldsymbol{s}\in\mathbb{K}} F_{\text{Emp}}^{N} \left|\bar{\boldsymbol{s}}\left(N^{-1}\sum_{j=1}^{N}\boldsymbol{Y}_j^r;\bar{\boldsymbol{\theta}}\left(\boldsymbol{s}\right)\right)\right|^p, \tag{6}$$

where $\bar{\boldsymbol{s}}\left(\cdot;\bar{\boldsymbol{\theta}}\left(\boldsymbol{s}\right)\right)$ is now a function of the IID random sample $\left\{\boldsymbol{Y}_j^r\right\}_{j=1}^{N}$, drawn from the DGP characterized by f_{Emp}. We observe that $\bar{\boldsymbol{s}}$ is continuous with respect to $\boldsymbol{\theta}$, from (2), and $\bar{\boldsymbol{\theta}}$ is continuous with respect to $\boldsymbol{s}$, from (4). Furthermore, for any $p > 0$, $|\cdot|^p$ is continuous. Furthermore, since f_{Emp} is discretely and compactly supported, the expectation in (6) is a finite sum of continuous functions with respect to $\boldsymbol{s}$ and is thus continuous in $\boldsymbol{s} \in \mathbb{K}$. Finally, since $\mathbb{K}$ is compact, the celebrated Weierstrass extreme value theorem yields the boundedness of (6) and thus the verification of B3.

Finally, we are left with C1–C3. C1 can be verified by choice of an appropriate learning rate sequence, and is thus trivially verifiable. And as discussed earlier, other than an appropriate choice of $\boldsymbol{s}^{(0)}$, the remainder of C2 cannot easily be inspected, and hence truncation be required, instead. We note that C3 is also difficult to verify, in its primitive form, however it is implied by the following sensible assumption.

D1 The Hessian matrix of $\sum_{i=1}^{n} \log f\left(\boldsymbol{y}_i;\boldsymbol{\theta}\right)$, evaluated at any $\boldsymbol{\theta}_0 \in \mathbb{M}_{\Theta}^{\text{Emp}}$, is non-singular with respect to $\boldsymbol{\theta} \in \Theta$.

This is generally satisfied for all but pathological samples $\{\boldsymbol{y}_i\}_{i=1}^{n}$. Using Corollaries we can state Proposition 3 from the main text, regarding the finite mixture of normal distributions.

3.2 Details regarding the truncated mini-batch EM algorithm for normal mixture models

To see that

$$\mathbb{K}_m = \mathbb{D}_{g-1}^m \times \prod_{i=1}^{g} \mathbb{B}_m^d \times \prod_{i=1}^{g} \mathbb{H}_d^m,$$

satisfies the conditions of

$$\mathbb{K}_m \subset \text{interior}\left(\mathbb{K}_{m+1}\right), \text{ and } \bigcup_{m=0}^{\infty} \mathbb{K}_m = \mathbb{S},$$

we observe that each $\mathbb{D}_{g-1}^m$ is simply a polygon embedded in the interior of the next polygon $\mathbb{D}_{g-1}^{m+1}$. Furthermore each $\mathbb{K}_m$ is compact by virtue of the closure condition $\pi_z \geq \left(c_1 + m\right)^{-1}$ and $\cup_{m=1}^{\infty} \mathbb{D}_{g-1}^m = \mathbb{D}_{g-1}$, since $\left(c_1 + m\right)^{-1} \to 0$, as m increases. Next, $\mathbb{B}_m^d$ is a closed box in $\mathbb{R}^d$ and is thus compact. Since the boundaries of the boxes are expanding, we have the embedding property. Lastly, since the boundaries expand infinitely, we have $\cup_{m=1}^{\infty} \mathbb{B}_m^d = \mathbb{R}^d$. Lastly, $\mathbb{H}_d^m$ has interior embedded in $\mathbb{H}_d^{m+1}$ by virtue that any matrix $\mathbf{H} \in \mathbb{H}_d^m$ must satisfy $\lambda_1\left(\mathbf{H}\right) > \left(c_3 + m + 1\right)^{-1}$ and $\lambda_d\left(\mathbf{H}\right) < c_3 + m + 1$. Next, $\mathbb{H}_d^m$ is compact since each $\mathbf{H} \in \mathbb{H}_d^m$ can be written as $\mathbf{H} = \mathbf{U}^\top \mathbf{D} \mathbf{U}$, where $\mathbf{U}$ is an orthonormal matrix in $\mathbb{R}^{d \times d}$, which forms compact set (cf. Horn & Johnson, 2013, Sec. 2.1) and $\mathbf{D}$ is a diagonal matrix with elements equal to the eigenvalues of $\mathbf{H}$. Since the eigenvalues are bounded above and below, the Heine-Borel theorem suffices to demonstrate the compactness of the set of possible matrices $\mathbf{D}$, and subsequently $\mathbb{H}_d^m$. Finally, since the space $\mathbb{H}_d$ can contain matrices with arbitrarily large eigenvalues that must be bounded below by zero, we have $\cup_{m=1}^{\infty} \mathbb{H}_d^m = \mathbb{H}_d$, since $\left(c_3 + m\right)^{-1} \to 0$ and $c_3 + m \to \infty$, as m increases. The result is finally established by noting that the product space of compact sets is compact.

4 Further simulation studies regarding mini-batch algorithms for finite mixtures of normal distributions

We recall from the main text that all computations are conducted in the `R` programming environment, although much of the bespoke programs are programmed in `C` and integrated in `R` via the `Rcpp` and `RcppArmadillo` packages of (Eddelbuettel, 2013). Furthermore, timings of programs were conducted on a MacBook Pro with a 2.2 GHz Intel Core i7 processor, 16 GB of 1600 MHz DDR3 RAM, and a 500 GB SSD hard drive. We note that all of the code used to conduct the simulations and computations for this manuscript can be accessed from `https://github.com/hiendn/StoEMMIX`.

Additionally, as in the main text, we set the learning rate sequence $\{\gamma_r\}_{r=1}^{\infty}$ to $\gamma_r = (1 - 10^{-10}) \times r^{6/10}$, following the recommendations of Cappé & Moulines (2009). We also set the allowed data access budget for each of our assessed algorithms to 10 epochs, or epoch equivalents.

Like in Section 4 of the main text, we assess the standard EM algorithm, as applied through the `mclust` package for `R` (Scrucca et al., 2016), and apply the mini-batch EM algorithms with various settings. These include using batch sizes of sizes $N = n/10$ and $N = n/5$, using Polyak averaging, as well as using truncation, where the truncation parameters are set to $c_1, c_2, c_3 = 1000$. For each estimation problem, we use the randomization method of McLachlan & Peel (2000, Sec. 3.9.3) to generate a single initial parameter vector $\boldsymbol{\theta}^{(0)}$ that is used for all of the assessed algorithms.

In all of our simulation studies, we use the `simdataset` function from the `MixSim` package (Melnykov et al., 2012) for simulation and the `proc.time` function for timing.

4.1 Flea data

The Flea data, from the `tourr` package of Wickham et al. (2011) and contains six real valued measurements from 74 flea-beetles. Of the 74 flea-beetles, 21 are of the Concinna species, 31 are of the Heikertingeri species, and 22 are of the Heptapotamica species. A scatterplot matrix of the data is provided in Figure 1.

Like the Iris data, we fit a single multivariate normal distribution to each of the subpopulation of observations (i.e., we estimate a mean vector and covariance matrix, for each species). Then, using the three mean vectors and covariance matrices, we construct a template $g = 3$ component normal mixture model with mixing proportions equal to the representation proportion of each species (i.e. $\pi_1 = 21/74$ $\pi_2 = 31/74$, and $\pi_3 = 22/74$). This template distribution is then used to generate synthetic data sets of any size n.

As with the Iris1 and Iris2 experiments, we simulate $n = 10^6$ and $n = 10^7$ observations to generate experimental instances for the Flea1 and Flea2 experiments, respectively. To measure the performances of the assessed methods, we again measure the computation times, log-likelihoods, squared errors (SE) and adjusted-Rand indices (ARI). The experiments are repeated Rep = 100 times each in order to observe the relative performances of the different algorithms, on average.

4.2 ELKI scenario

The ELKI scenario is a synthetic data scenario, motivated by an example proposed on `https://github.com/elki-project/elki` as part of the ELKI project of Schubert et al. (2015). The data is an unbalanced and separable $d = 2$ dimensional $g = 3$ component normal mixture scenario.

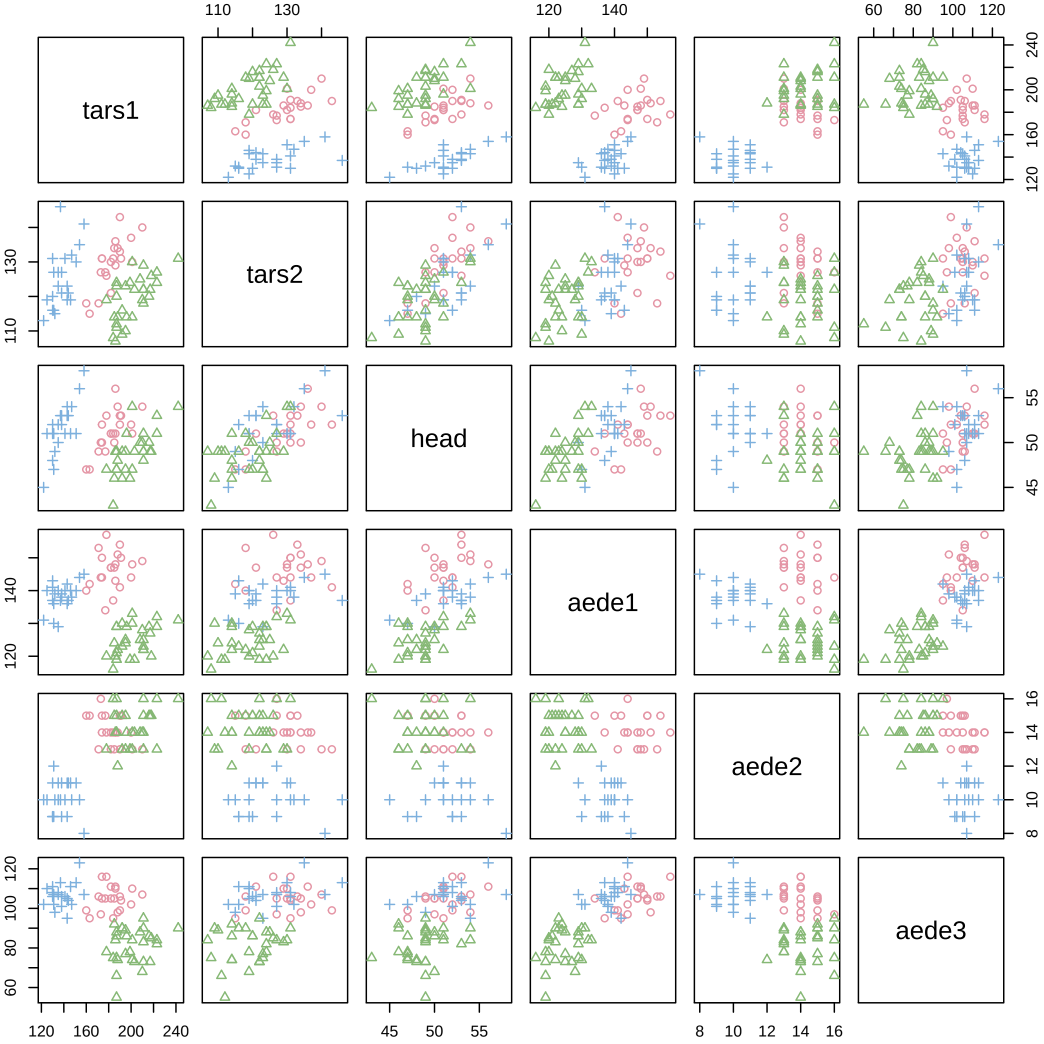


Figure 1: Scatterplot matrix of the six variables of the Flea data set. Here, each of the symbols represents a different species of flea-beetle. That is, circle indicates Concinna, triangle indicates Heikertingeri, and plus signs indicate Heptapotamica.

Specifically, the ELKI scenario generates data from a distribution with PDF

$$f\left(\boldsymbol{y};\boldsymbol{\theta}\right)=\sum_{z=1}^{g}\pi_{z}\varphi\left(\boldsymbol{y};\boldsymbol{\mu}_{z},\boldsymbol{\Sigma}_{z}\right), \tag{7}$$

where $\pi_1 = 5/10$, $\pi_2 = 3/10$, $\pi_3 = 2/10$, $\boldsymbol{\mu}_1^\top = (0.3, 0.3)$, $\boldsymbol{\mu}_2^\top = (0.85, 0.35)$, $\boldsymbol{\mu}_3^\top = (0.45, 0.85)$,

$$\boldsymbol{\Sigma}_1=\begin{bmatrix}0.09^2 & 0\\ 0 & 0.09^2\end{bmatrix}, \boldsymbol{\Sigma}_2=\begin{bmatrix}0.05^2 & 0\\ 0 & 0.1^2\end{bmatrix}, \text{ and } \boldsymbol{\Sigma}_3=\begin{bmatrix}0.035^2 & 0\\ 0 & 0.035^2\end{bmatrix}.$$

Here $\boldsymbol{\theta}$ contains the parameter components π_z, $\boldsymbol{\mu}_z$, and $\boldsymbol{\Sigma}_z$ ($z \in [3]$) and φ is the normal density function.

A visualization of $n = 1000$ observations from the ELKI scenario is provided in Figure (2). ELKI1 and ELKI2 indicate test experimental instances generated by simulating $n = 10^6$ and $n = 10^7$ observations from the ELKI scenario, respectively.

4.3 Results

Figures 3 and 4 contain box plots that summarize the results of Flea1 and Flea2, respectively. Similarly, Figures 5 and 6 contain box plots that summarize the results of ELKI1 and ELKI2, respectively.

As with the Iris and Wreath data, we note that Polyak averaging requires no additional amount of computational effort and thus timing of Polyak averaging variants of the mini-batch algorithms are not provided separately. From the timing data, we observe that, as before, the standard EM algorithm is uniformly faster than the mini-batch algorithms. This is to be expected due to the additional number of iteration loops and within-loop steps required to process some fixed

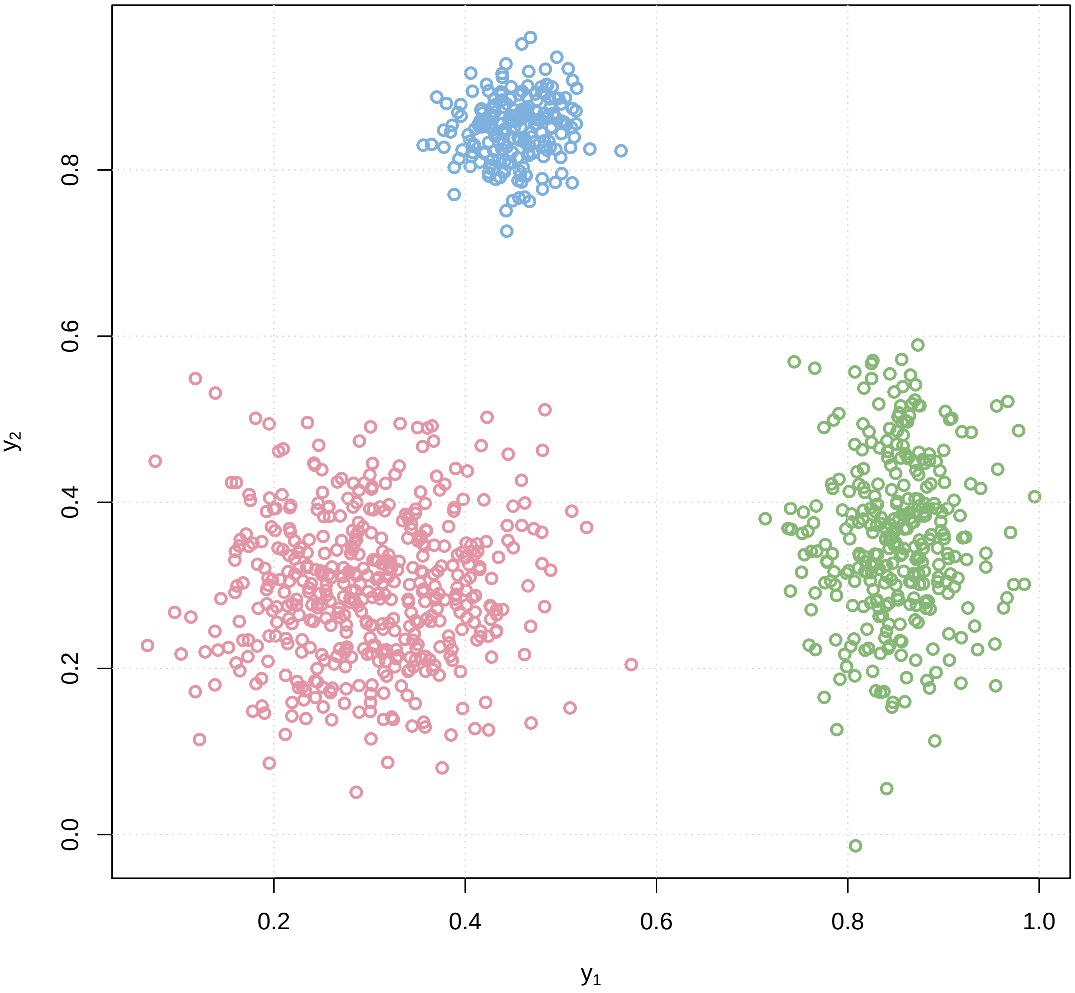
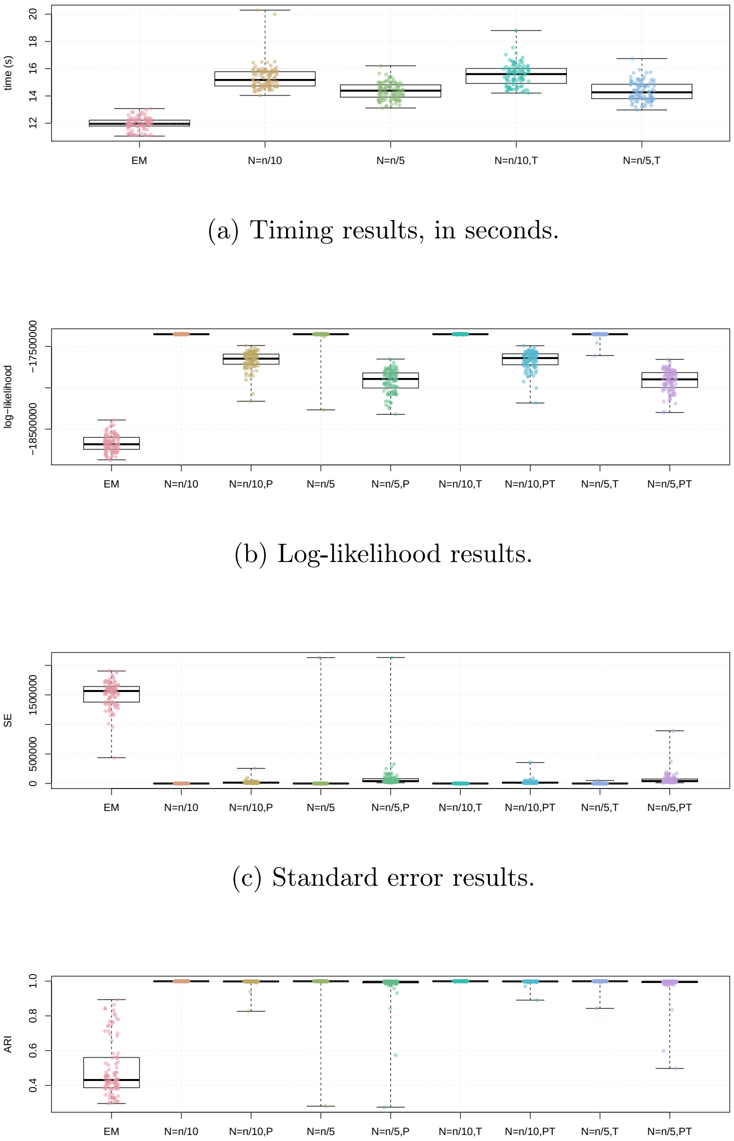


Figure 2: Scatter of $n = 1000$ observations generated from the ELKI scenario.



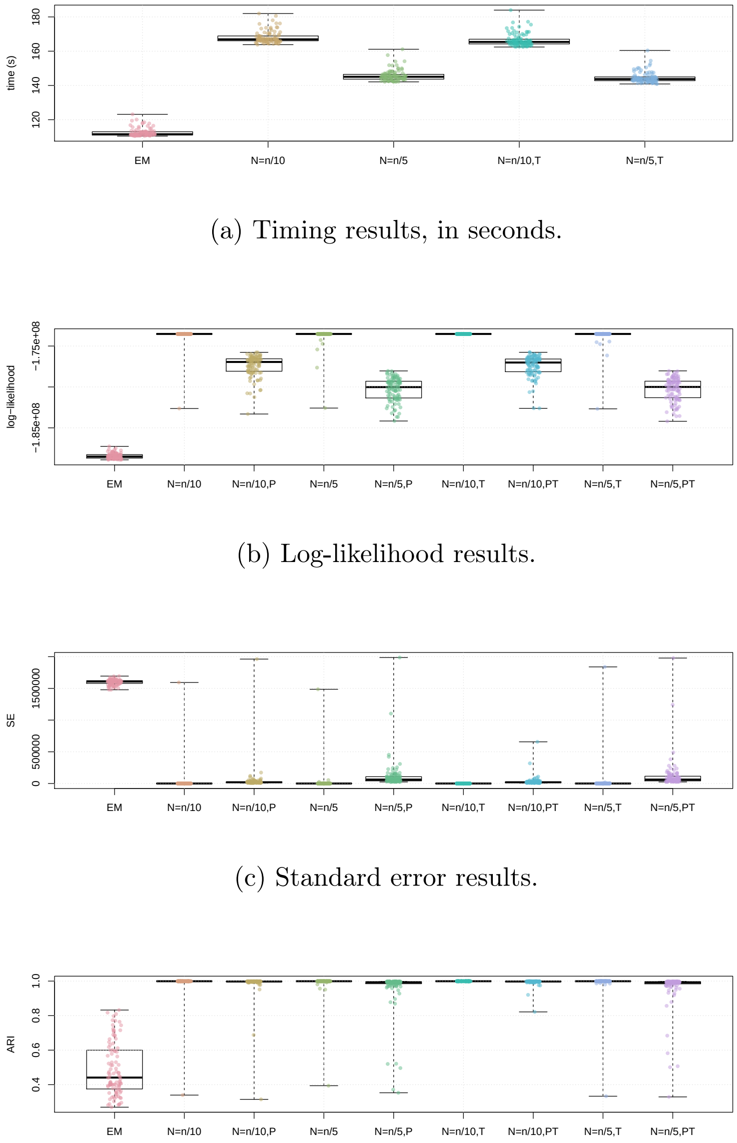
(a) Timing results, in seconds.

(b) Log-likelihood results.

(c) Standard error results.

(d) Adjusted-Rand index results.

Figure 3: Results from Rep = 100 replications of the Flea1 simulation experiment. The 'EM' box plot summarizes the performance of the standard EM algorithm. The other plots are labelled by which variant of the mini-batch EM algorithm is summarized. The value of the batch size N is indicated (either $N = n/10$ or $N = n/5$), and a 'P' or a 'T' designates that Polyak averaging or truncation was used, respectively.



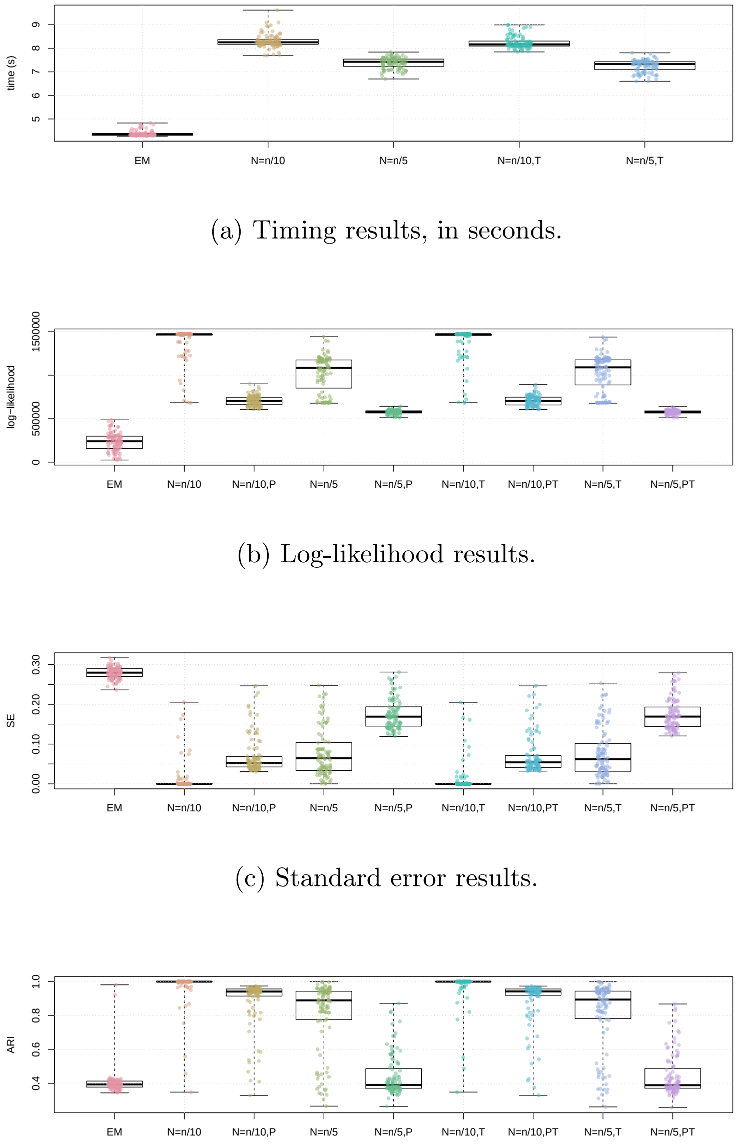
(a) Timing results, in seconds.

(b) Log-likelihood results.

(c) Standard error results.

(d) Adjusted-Rand index results.

Figure 4: Results from Rep = 100 replications of the Flea2 simulation experiment. The 'EM' box plot summarizes the performance of the standard EM algorithm. The other plots are labelled by which variant of the mini-batch EM algorithm is summarized. The value of the batch size N is indicated (either $N = n/10$ or $N = n/5$), and a 'P' or a 'T' designates that Polyak averaging or truncation was used, respectively.



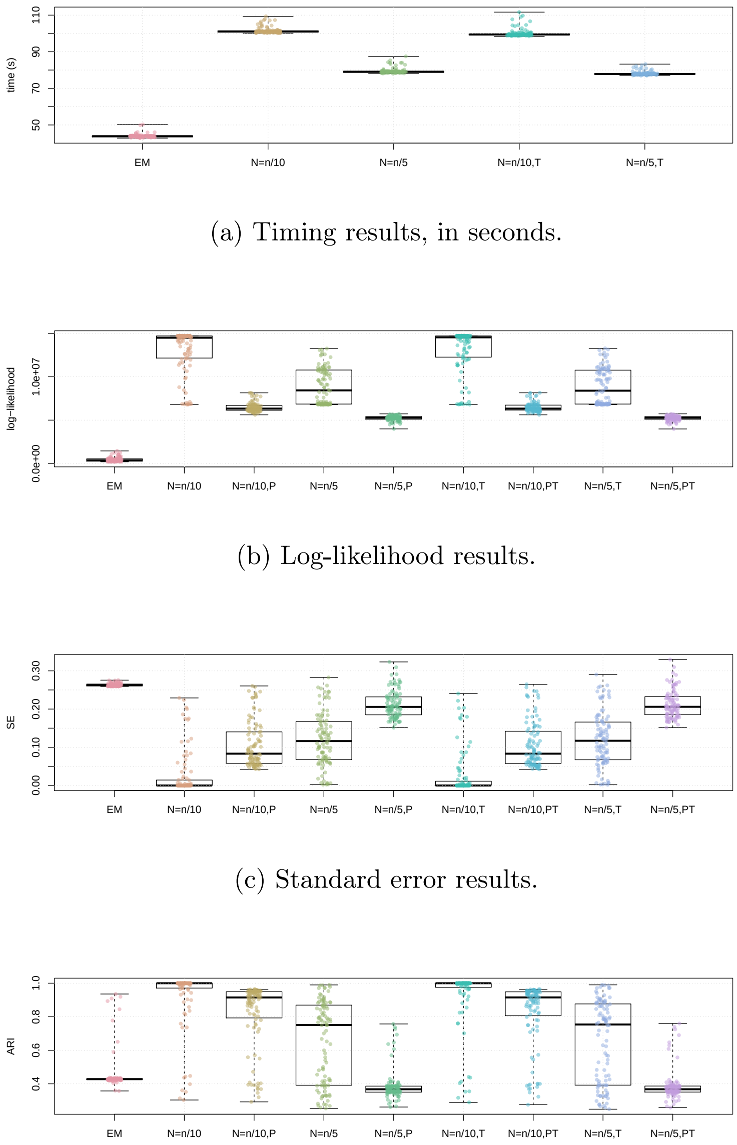
(a) Timing results, in seconds.

(b) Log-likelihood results.

(c) Standard error results.

(d) Adjusted-Rand index results.

Figure 5: Results from Rep = 100 replications of the ELKI1 simulation experiment. The 'EM' box plot summarizes the performance of the standard EM algorithm. The other plots are labelled by which variant of the mini-batch EM algorithm is summarized. The value of the batch size N is indicated (either $N = n/10$ or $N = n/5$), and a 'P' or a 'T' designates that Polyak averaging or truncation was used, respectively.



(a) Timing results, in seconds.

(b) Log-likelihood results.

(c) Standard error results.

(d) Adjusted-Rand index results.

Figure 6: Results from Rep = 100 replications of the ELKI2 simulation experiment. The 'EM' box plot summarizes the performance of the standard EM algorithm. The other plots are labelled by which variant of the mini-batch EM algorithm is summarized. The value of the batch size N is indicated (either $N = n/10$ or $N = n/5$), and a 'P' or a 'T' designates that Polyak averaging or truncation was used, respectively.

epoch number of computations, when comparing the mini-batch algorithms to the standard EM algorithm.

The Flea1 and Flea2 experiments performed similarly to the Iris1 and Iris2 experiments. We observe that the mini-batch EM algorithms uniformly outperform the standard EM algorithm in terms of the log-likelihood, SE, and ARI. In all three measurements, we observe that $N = n/10$ performed better than $N = n/5$, and also that there were no differences between truncated versions of the mini-batch algorithms and equivalent variants without truncation. As with the Iris data, Polyak averaging appears to reduce the performance of the mini-batch algorithms, for a given level of N, with respect to each of the three measurements.

The ELKI scenario demonstrated a level of performance that appeared to be between the Iris data and the Wreath data. We observe that the mini-batch EM algorithms tended outperform the standard EM algorithm in terms of the log-likelihood and SE, albeit by a lesser amount than the Iris data. In all three measurements, we observe that $N = n/10$ performed better than $N = n/5$, and also that there were no differences between truncated versions of the mini-batch algorithms and equivalent variants without truncation. Again, Polyak averaging appears to reduce the performance of the mini-batch algorithms, for a given level of N, with respect to each of the three measurements. The most interesting feature of the ELKI experiments is that for $N = n/5$ and with Polyak averaging, the SE and ARI could be equal to or be out performed by the standard EM algorithm, even when the log-likelihood was better. This is an interesting result and confirms our intuition that separability of the data is a determining feature for the relative performances between the standard EM algorithm and the mini-batch variants.

5 Mini-batch algorithms for non-normal exponential family mixtures

5.1 Mixtures of exponential distributions

We say that the random variable Y is generated from a g component mixture of exponential distributions if it has a PDF of the form

$$f\left(y;\boldsymbol{\theta}\right)=\sum_{z=1}^{g}\pi_{z}f_{\text{Exp}}\left(y;\lambda_{z}\right),\tag{8}$$

where

$$f_{\text{Exp}}\left(y;\lambda\right)=\lambda\exp\left(-\lambda y\right)$$

is the exponential PDF, and $\boldsymbol{\theta}$ contains the parameter elements π_z and λ_z, for $z\in[g]$.

The mixture of exponential distributions have been considered in articles such as Jewell (1982), Feldmann & Whitt (1998), and Jaheen (2005). It is particularly noted in Feldmann & Whitt (1998) that the class of finite mixtures of exponential distributions is dense in the class of monotonically decreasing density functions, making them a type of universal modeling paradigm.

Let

$$\tau_{z}\left(y;\boldsymbol{\theta}\right)=\frac{\pi_{z}f_{\text{Exp}}\left(y;\lambda_{z}\right)}{\sum_{\zeta=1}^{g}\pi_{\zeta}f_{\text{Exp}}\left(y;\lambda_{\zeta}\right)}$$

and define

$$t_{1z}\left(y;\boldsymbol{\theta}\right)=\tau_{z}\left(y;\boldsymbol{\theta}\right),\text{ and }t_{2z}\left(y;\boldsymbol{\theta}\right)=\tau_{z}\left(y;\boldsymbol{\theta}\right)y.$$

Following the results from Section 3 of the main text, we can write the sufficient statistic for the mixture as

$$\left[\bar{\boldsymbol{s}}\left(y;\boldsymbol{\theta}\right)\right]^{\top}=\left(t_{11}\left(y;\boldsymbol{\theta}\right),t_{21}\left(y;\boldsymbol{\theta}\right),\dots,t_{1g}\left(y;\boldsymbol{\theta}\right),t_{2g}\left(y;\boldsymbol{\theta}\right)\right)$$

and the standard EM algorithm update function for a single observation, $\bar{\boldsymbol{\theta}}\left(\boldsymbol{s}\right)$, can be given by

$$\bar{\pi}_z=t_{1z}\left(y;\boldsymbol{\theta}\right)\text{ and }\bar{\lambda}_z=\frac{t_{1z}\left(y;\boldsymbol{\theta}\right)}{t_{2z}\left(y;\boldsymbol{\theta}\right)}.$$

Given an observe sample $\{y_i\}_{i=1}^{n}$ drawn IID from a mixture of exponential distributions, we can then draw subsamples $\{Y_i^r\}_{i=1}^{N}$ ($r\in[R]$; $R\in\mathbb{N}$) of size N, uniformly and with replacement from $\{y_i\}_{i=1}^{n}$, in order to compute the mini-batch updates:

$$\boldsymbol{s}^{(r)}=\boldsymbol{s}^{(r-1)}+\gamma_r\left[N^{-1}\sum_{i=1}^{N}\bar{\boldsymbol{s}}\left(Y_i^r;\boldsymbol{\theta}^{(r-1)}\right)-\boldsymbol{s}^{(r-1)}\right]\text{, and }\boldsymbol{\theta}^{(r)}=\bar{\boldsymbol{\theta}}\left(\boldsymbol{s}^{(r)}\right). \tag{9}$$

Here, we may write $\bar{\boldsymbol{\theta}}\left(\boldsymbol{s}^{(r)}\right)$ as

$$\bar{\pi}_z=N^{-1}\sum_{i=1}^{N}t_{1z}\left(Y_i^r;\boldsymbol{\theta}^{(r)}\right)\text{ and }\bar{\lambda}_z=\frac{\sum_{i=1}^{N}t_{2z}\left(y;\boldsymbol{\theta}^{(r)}\right)}{\sum_{i=1}^{N}t_{1z}\left(y;\boldsymbol{\theta}^{(r)}\right)},$$

which entirely define our mini-batch update, for batches of size N.

5.2 Mixtures of Poisson distributions

We say that the random variable Y is generated from a g component mixture of exponential distributions if it has a PMF of the form

$$f\left(y;\boldsymbol{\theta}\right)=\sum_{z=1}^{g}\pi_{z}f_{\text{Poi}}\left(y;\lambda_{z}\right), \tag{10}$$

where

$$f_{\text{Poi}}\left(y;\lambda\right)=\lambda^{y}\exp\left(-\lambda\right)/y!$$

is the Poisson PMF, and $\boldsymbol{\theta}$ contains the parameter elements π_z and λ_z, for $z\in[g]$.

The an online EM algorithm for mixtures of Poisson distributions was considered in Cappé & Moulines (2009), as a minor example. Further research regarding the mixtures of Poisson distributions appear in Karlis & Xekalaki (1999) and Karlis & Xekalaki (2001).

Like the mixtures of exponential distributions, we let

$$\tau_{z}\left(y;\boldsymbol{\theta}\right)=\frac{\pi_{z}f_{\text{Poi}}\left(y;\lambda_{z}\right)}{\sum_{\zeta=1}^{g}\pi_{\zeta}f_{\text{Poi}}\left(y;\lambda_{\zeta}\right)}$$

and define

$$t_{1z}\left(y;\boldsymbol{\theta}\right)=\tau_{z}\left(y;\boldsymbol{\theta}\right),\text{ and }t_{2z}\left(y;\boldsymbol{\theta}\right)=\tau_{z}\left(y;\boldsymbol{\theta}\right)y.$$

Then, we can write the sufficient statistic for the mixture as

$$\left[\bar{\boldsymbol{s}}\left(y;\boldsymbol{\theta}\right)\right]^{\top}=\left(t_{11}\left(y;\boldsymbol{\theta}\right),t_{21}\left(y;\boldsymbol{\theta}\right),\dots,t_{1g}\left(y;\boldsymbol{\theta}\right),t_{2g}\left(y;\boldsymbol{\theta}\right)\right)$$

and the standard EM algorithm update function for a single observation, $\bar{\boldsymbol{\theta}}\left(\boldsymbol{s}\right)$, can be given by

$$\bar{\pi}_{z}=t_{1z}\left(y;\boldsymbol{\theta}\right)\text{ and }\bar{\lambda}_{z}=\frac{t_{2z}\left(y;\boldsymbol{\theta}\right)}{t_{1z}\left(y;\boldsymbol{\theta}\right)}.$$

Given an observe sample $\left\{y_i\right\}_{i=1}^{n}$ drawn IID from a mixture of Poisson distributions, we can then draw subsamples $\{Y_i^r\}_{i=1}^N$ of size N, uniformly and with replacement from $\{y_i\}_{i=1}^n$, in order to compute the mini-batch updates (9). Here, we may write $\bar{\boldsymbol{\theta}}\left(\boldsymbol{s}^{(r)}\right)$ as

$$\bar{\pi}_z = N^{-1}\sum_{i=1}^{N} t_{1z}\left(Y_i^r;\boldsymbol{\theta}^{(r)}\right) \text{ and } \bar{\lambda}_z = \frac{\sum_{i=1}^{N} t_{2z}\left(y;\boldsymbol{\theta}^{(r)}\right)}{\sum_{i=1}^{N} t_{1z}\left(y;\boldsymbol{\theta}^{(r)}\right)}.$$

5.3 Simulation studies

We use the same setup as in Section 4. That is, all computations are conducted in the `R` programming environment, using bespoke `C` code, integrated in `R` via the `Rcpp` and `RcppArmadillo` packages. Timings of programs were conducted on a MacBook Pro with a 2.2 GHz Intel Core i7 processor, 16 GB of 1600 MHz DDR3 RAM, and a 500 GB SSD hard drive. As with the normal experiments, all codes can be found at `https://github.com/hiendn/StoEMMIX`.

We again, set the learning rate sequence $\{\gamma_r\}_{r=1}^{\infty}$ to $\gamma_r = (1-10^{-10}) \times r^{6/10}$, and set the allowed data access budget for each of our assessed algorithms to 10 epochs, or epoch equivalents. We assess the standard EM algorithm (implemented via bespoke `C` code), as well as mini-batch EM algorithm variants with batch sizes of sizes $N = n/10$ and $N = n/5$, and using Polyak averaging. We do not consider truncation here, as we found it unnecessary. As before, for each estimation problem, we use the randomization method of McLachlan & Peel (2000, Sec. 3.9.3) to generate a single initial parameter vector $\boldsymbol{\theta}^{(0)}$ that is used for all of the assessed algorithms.

To assess the exponential distribution mixture EM algorithm and mini-batch variants, we simulate data from $g = 3$ component exponential mixture scenario. Specifically, with data generated from a process characterized by a PDF of form (8), with parameters $\pi_1 = 0.2$, $\pi_2 = 0.1$, $\pi_3 = 0.7$, $\lambda_1 = 1$, $\lambda_2 = 9$, and $\lambda_3 = 15$, following a suggestion of Jewell (1982). We use the names Exp1 and Exp2 to indicate test experimental instances generated by simulating $n = 10^6$ and $n = 10^7$

observations from the process described above, respectively.

To assess the Poisson distribution mixture EM algorithm and mini-batch variants, we simulate data from $g = 3$ component exponential mixture scenario. Specifically, with data generated from a process characterized by a PMF of form (10), with parameters $\pi_1 = 0.8$, $\pi_2 = 0.1$, $\pi_3 = 0.1$, $\lambda_1 = 1$, $\lambda_2 = 5$, and $\lambda_3 = 12$, following a suggestion of Jewell (1982); Karlis & Xekalaki (1999). We use the names Poi1 and Poi2 to indicate test experimental instances generated by simulating $n = 10^6$ and $n = 10^7$ observations from the process described above, respectively.

As with the Flea and ELKI scenarios, we measure the performances of the assessed methods, we again measure the computation times, log-likelihoods, SE and ARI. The experiments are each repeated $\text{Rep} = 100$ times.

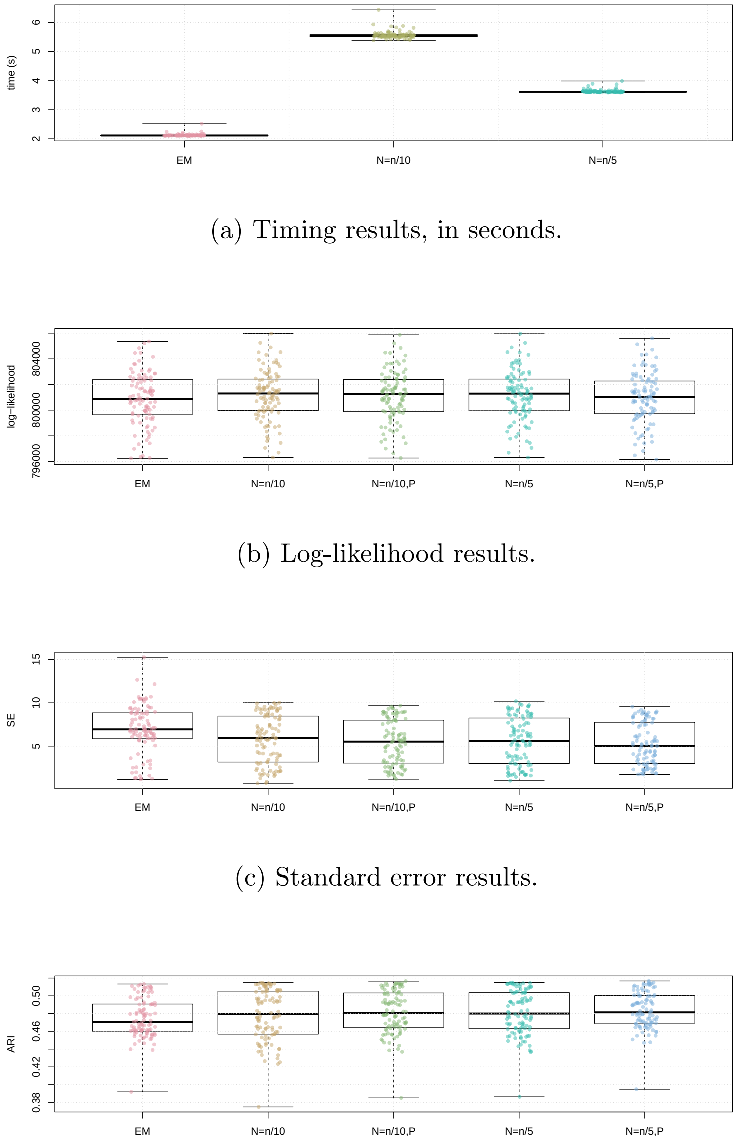
5.4 Results

Figures 7 and 8 contain box plots that summarize the results of Exp1 and Exp2, and Figures 9 and 10 contain box plots that summarize the results of Poi1 and Poi2, respectively.

As with the normal mixture algorithms, we note that Polyak averaging requires no additional amount of computational effort and thus are not timed, seperately. From the timing data, we observe that, as before, the standard EM algorithm is uniformly faster than the mini-batch algorithms. This is as expected from the previous results.

We observe from the Exp1 and Exp2 results that the mini-batch variants tended to yield a small improvement in average performance across the metrics of log-likelihood, SE, and ARI. This is most pronounced when observing the SE results.

We note that batch size $N = n/10$ tended to yield better log-likelhood results than $N = n/5$, however, it is the opposite in the case of SE and ARI. Furthermore, unlike the normal mixture



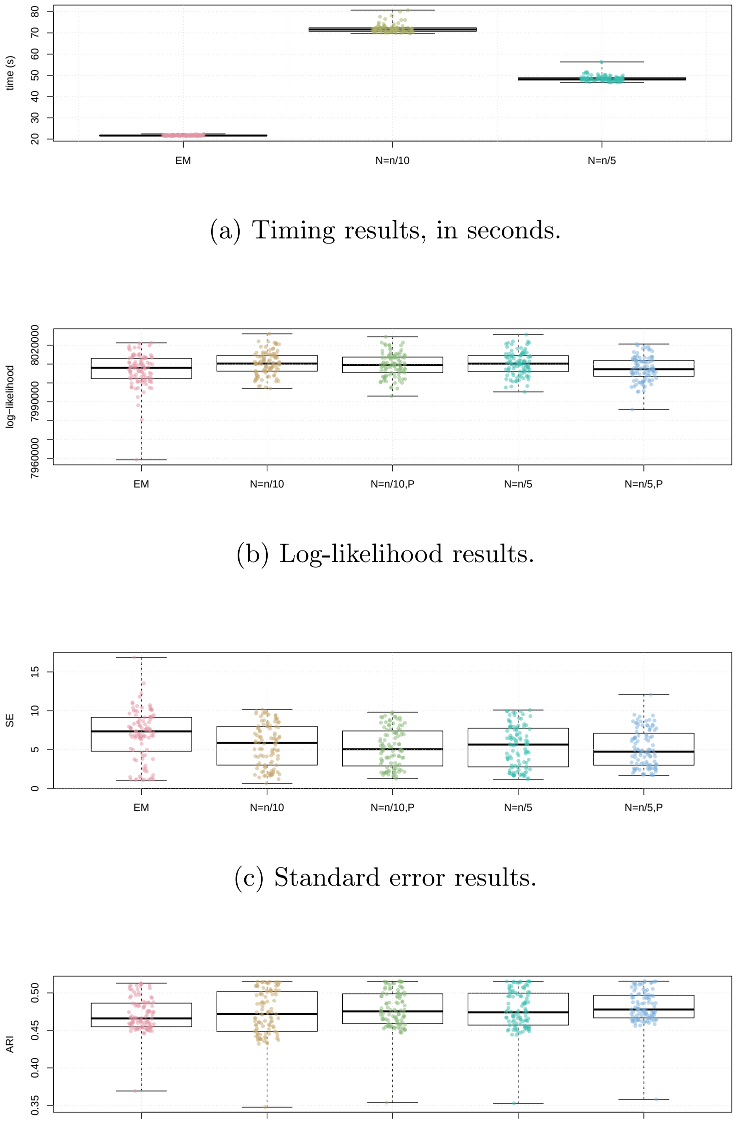
(a) Timing results, in seconds.

(b) Log-likelihood results.

(c) Standard error results.

(d) Adjusted-Rand index results.

Figure 7: Results from Rep = 100 replications of the Exp1 simulation experiment. The 'EM' box plot summarizes the performance of the standard EM algorithm. The other plots are labelled by which variant of the mini-batch EM algorithm is summarized. The value of the batch size N is indicated (either $N = n/10$ or $N = n/5$), and a 'P' indicates that Polyak averaging was used, respectively.



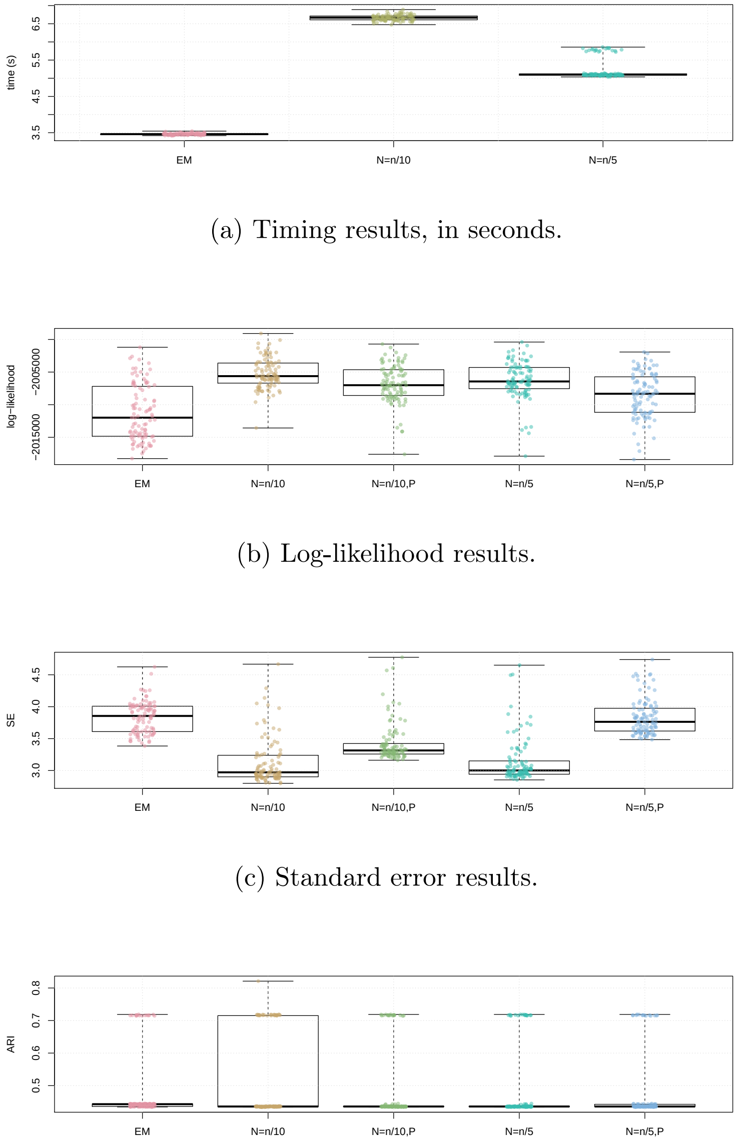
(a) Timing results, in seconds.

(b) Log-likelihood results.

(c) Standard error results.

(d) Adjusted-Rand index results.

Figure 8: Results from Rep = 100 replications of the Exp2 simulation experiment. The 'EM' box plot summarizes the performance of the standard EM algorithm. The other plots are labelled by which variant of the mini-batch EM algorithm is summarized. The value of the batch size N is indicated (either $N = n/10$ or $N = n/5$), and a 'P' indicates that Polyak averaging was used, respectively.



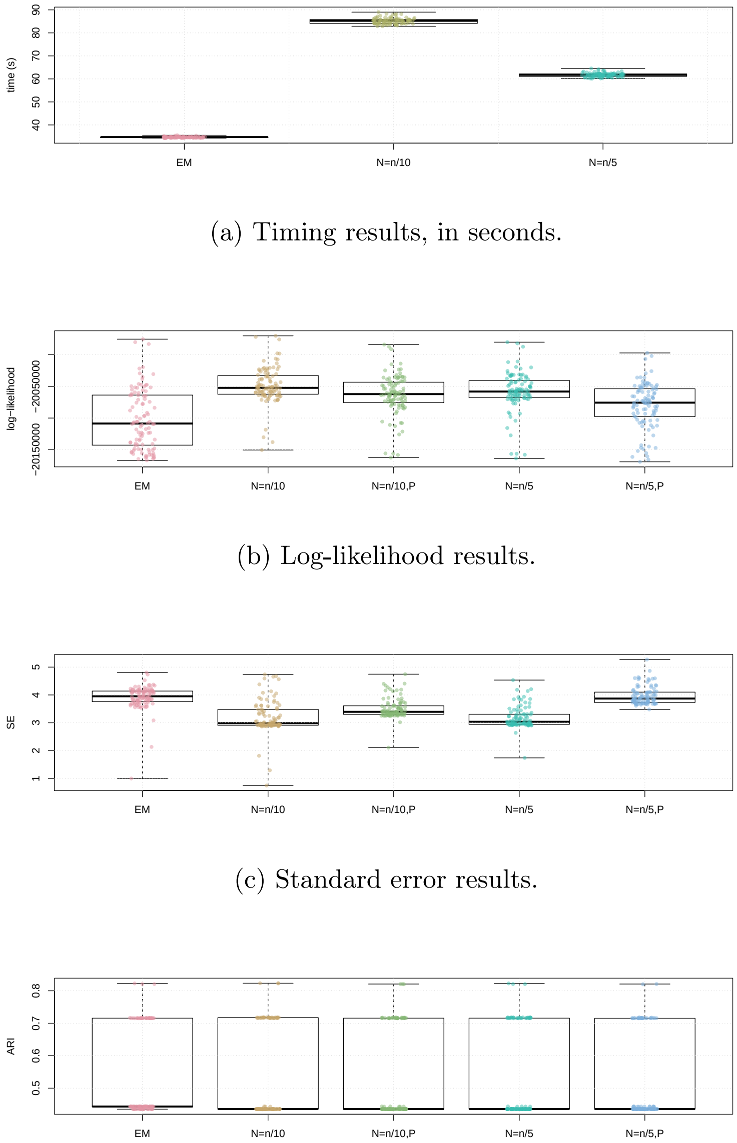
(a) Timing results, in seconds.

(b) Log-likelihood results.

(c) Standard error results.

(d) Adjusted-Rand index results.

Figure 9: Results from Rep = 100 replications of the Poi1 simulation experiment. The 'EM' box plot summarizes the performance of the standard EM algorithm. The other plots are labelled by which variant of the mini-batch EM algorithm is summarized. The value of the batch size N is indicated (either $N = n/10$ or $N = n/5$), and a 'P' indicates that Polyak averaging was used, respectively.



(a) Timing results, in seconds.

(b) Log-likelihood results.

(c) Standard error results.

(d) Adjusted-Rand index results.

Figure 10: Results from Rep = 100 replications of the Poi2 simulation experiment. The 'EM' box plot summarizes the performance of the standard EM algorithm. The other plots are labelled by which variant of the mini-batch EM algorithm is summarized. The value of the batch size N is indicated (either $N = n/10$ or $N = n/5$), and a 'P' indicates that Polyak averaging was used, respectively.

experiments, Polyak averaging appears to improve performance in SE and ARI, but not in log-likelihood. In fact, when $N = n/5$ and Polyak averaging was used, the log-likelihood performance was at parity with the standard EM algorithm.

In the Poi1 and Poi2 results, we observe that all mini-batch variants tended to improve the average performance across the log-likelihood and SE, although ARI appeared to be at parity with the standard EM algorithm. We observe that batch sizes $N = n/10$ performed a small amount better than $N = n/5$, and that Polyak averaging tended to decrease the performance in both log-likelhood and SE. Most notably, the SE result with $N = n/5$ and Polyak averaging appeared to be at the same performance level as the standard EM algorithm

References

Anderson, T. W. & Olkin, I. (1985). Maximum-likelihood estimation of the parameters of a multivariate normal distribution. *Linear Algebra and Its Applications*, 70, 147–171.

Atienza, N., Garcia-Heras, J., Munoz-Pichardo, J. M., & Villa, R. (2007). On the consistency of MLE in finite mixture models of exponential families. *Journal of Statistical Planning and Inference*, 137, 496–505.

Bengio, Y. (2012). Practical recommendations for gradient-based training of deep architechtures. In G. Montavon, G. B. Orr, & K.-R. Muller (Eds.), *Neural Networks: Tricks of the Trade.* Berlin: Springer.

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

Cappé, O. (2011). Online EM algorithm for hidden Markov models. *Journal of Computation and Graphical Statistics*, 20, 728–749.

Cappé, O. & Moulines, E. (2009). On-line expectation-maximization algorithm for latent data models. *Journal of the Royal Statistical Society B*, 71, 593–613.

Celeux, G., Chauveau, D., & Diebolt, J. (1996). Stochastic version of the EM algorithm: an expertimental study in the mixture case. *Journal of Statistical Computation and Simulation*, 55, 287–314.

Celeux, G. & Diebolt, J. (1992). A stochastic approximation type EM algorithm for the mixture problem. *Stochastics and Stochastic Reports*, 41, 119–134.

Celeux, G. & Govaert, G. (1992). A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis*, 14, 315–332.

Celeux, G. & Govaert, G. (1993). Comparison of the mixture and the classfication maximum likelihood in cluster analysis. *Journal of Statistical Computation and Simulation*, 47, 127–146.

Delyon, B., Lavielle, M., & Moulines, E. (1999). Counvergence of a stochastic approximation version of the EM algorithm. *Annals of Statistics*, 27, 94–128.

Dupuy, C. & Bach, F. (2017). Online but accurate inference for latent variable models with local Gibbs sampling. *Journal of Machine Learning Reseach*, 18, 1–45.

Eddelbuettel, D. (2013). *Seamless R and C++ Integration with Rcpp*. New York: Springer.

Feldmann, A. & Whitt, W. (1998). Fitting mixtures of exponentials to long-tail distributions to analyze network performance models. *Performance Evaluation*, 31(245-279).

Ganesalingam, S. & McLachlan, G. J. (1980). A comparison of the mixture and classification approaches to cluster analysis. *Communications in Statistics - Theory and Methods*, 9, 923–933.

Hasselblad, V. (1969). Estimation of finite mixtures of distributions from the exponential family. *Journal of the American Statistical Association*, 64, 1459–1471.

Horn, R. A. & Johnson, C. R. (2013). *Matrix Analysis*. Cambridge: Cambridge University Press.

Jaheen, Z. F. (2005). On record statistics from a mixture of two exponential distributions. *Journal of Statistical Computation and Simulation*, 75, 1–11.

Jewell, N. (1982). Mixtures of exponential distributions. *Annals of Statistics*, 10, 479–484.

Jordan, M. I. & Jacobs, R. A. (1994). Hierachical mixtures of experts and the EM algorithm. *Neural Computation*, 6, 181–214.

Karlis, D. & Xekalaki, E. (1999). On testing for the number of components in a mixed Poisson model. *Annals of the Institute of Mathematical Statistics*, 51, 149–162.

Karlis, D. & Xekalaki, E. (2001). Robust inference for finite poisson mixtures. *Journal of Statistical Planning and Inference*, 93, 93–115.

Kuhn, E. & Lavielle, M. (2004). Coupling a stochastic approximation version of EM with an MCMC procedure. *ESAIM: Probability and Statistics*, 8, 115–131.

Kullback, S. & Leibler, R. A. (1951). On information and sufficiency. *Annals of Mathematical Statistics*, 22, 79–86.

Le Corff, S. & Fort, G. (2013a). Convergence of a particle-based approximation of the block online expectation maximization algorithm. *ACM Transactions on Modeling and Computer Simulation*, 23, 1–22.

Le Corff, S. & Fort, G. (2013b). Online expectation maximization based algorithms for interence in hidden Markov models. *Electronic Journal of Statistics*, 7, 763–792.

Li, D., Xu, L., & Goodman, E. (2014). On-line EM variants for multivariate normal mixture model in background learning and moving foreground detection. *Journal of Mathematical Imaging and Vision*, 48, 114–133.

Li, D., Xu, L., & Goodman, E. D. (2013a). Illumination-robust foreground detection in a video surveillance system. *IEEE Transactions on Circuits and Systems for Video Technology*, 23, 1637–1650.

Li, D., Xu, L., Goodman, E. D., Xu, Y., & Wu, Y. (2013b). Integrating a statistical background-foreground extraction algorithm and SVM classifier for pedestrian detection and tracking. *Integrated Computer-Aided Engineering*, 20, 201–216.

Lindsay, B. G. (1983). The geometry of mixture likelihoods: a general theory. *Annals of Statistics*, 11, 86–94.

McLachlan, G. J. & Peel, D. (2000). *Finite Mixture Models.* New York: Wiley.

Melnykov, V., Chen, W.-C., & Maitra, R. (2012). MixSim: an R package for simulating data to study performance of clustering algorithms. *Journal of Statistical Software*, 51, 1–25.

Neal, R. M. & Hinton, G. E. (1999). A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan (Ed.), *Learning in Graphical Models.* Cambridge: MIT Press.

Nielsen, F. (2012). K-MLE: a fast algorithm for learning statistical mixture models. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Pollard, D. (2002). *A User's Guide to Measure Theoretic Probability*. Cambridge: Cambridge University Press.

Redner, R. A. & Walker, H. F. (1984). Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26, 195–239.

Saint-Jean, C. & Nielsen, F. (2015). Online k-MLE for mixture modeling with exponential families. In F. Nielsen & F. Barbaresco (Eds.), *Geometric Sicence of Information*. Cham: Springer.

Same, A., Ambroise, C., & Govaert, G. (2007). An online classification EM algorithm based on the mixture model. *Statistics and Computing*, 17, 209–218.

Schubert, E., Koos, A., Emrich, T., Zufle, A., Schmid, K. A., & Zimek, A. (2015). A framework for clustering uncertain data. *Proceedings of the VLDB Endowment*, 8, 1976–1979.

Scrucca, L., Fop, M., Murphy, T. B., & Raftery, A. E. (2016). mclust: clustering, classification and density estimation using Gaussian finite mixture models. *R Journal*, 8, 289–317.

Seber, G. A. F. (2008). *A Matrix Handbook For Statisticians*. New York: Wiley.

Titterington, D. M. (1984). Recursive parameter estimation using incomplete data. *Journal of the Royal Statistical Society B*, 46, 257–267.

Wang, S. & Zhao, Y. (2006). Almost sure convergence of Titterington's recursive estimator for mixture models. *Statistics and Probability Letters*, 76, 2001–2006.

Wickham, H., Cook, D., Hofmann, H., & Buja, A. (2011). tourr: an R package for exploring multivariate data with projections. *Journal of Statistical Software*, 40, 1–18.

Yu, D. & Deng, L. (2015). *Automatic Speech Recognition: A Deep Learning Approach*. London: Springer.