Properties of the Stochastic Approximation EM Algorithm with Mini-batch Sampling

Estelle Kuhn, Catherine Matias, Tabea Rebafka

Properties of the Stochastic Approximation EM Algorithm with Mini-batch Sampling

Estelle Kuhn[1], Catherine Matias[2] and Tabea Rebafka[2]

[1] MaIAGE, INRA, Université Paris-Saclay, 78350 Jouy-en-Josas, France.
[2] Sorbonne Université, Université de Paris, CNRS, Laboratoire de Probabilités, Statistique et Modélisation (LPSM), Paris, France.

Abstract

To speed up convergence a mini-batch version of the Monte Carlo Markov Chain Stochastic Approximation Expectation-Maximization (MCMC-SAEM) algorithm for general latent variable models is proposed. For exponential models the algorithm is shown to be convergent under classical conditions as the number of iterations increases. Numerical experiments illustrate the performance of the mini-batch algorithm in various models. In particular, we highlight that an appropriate choice of the mini-batch size results in a tremendous speed-up of the convergence of the sequence of estimators generated by the algorithm. Moreover, insights on the effect of the mini-batch size on the limit distribution are presented.

1 Introduction

On very large datasets the computing time of the classical expectation-maximization (EM) algorithm (Dempster et al., 1977) as well as its variants such as MCEM, SAEM, MCMC-SAEM and others is long, since all data are used in every iteration. To circumvent this problem, a bunch of EM-type algorithms have been proposed, namely various mini-batch (Neal and Hinton, 1999; Liang and Klein, 2009; Karimi et al., 2018; Nguyen et al., 2019) and online (Titterington, 1984; Lange, 1995; Cappé and Moulines, 2009; Cappé, 2011) versions of the EM algorithm. They all consist in using only a part of the observations during one iteration in order to accelerate convergence. While online algorithms process a single observation per iteration handled in the order of arrival, mini-batch algorithms use larger, randomly chosen subsets of observations. The size of these subsets of data is called mini-batch size. Choosing large mini-batch sizes entails long computing times, while very small mini-batch sizes as well as online algorithms may result in a loss of accuracy of the algorithm. Hence, an optimal mini-batch size would achieve a compromise between accuracy and computing time. However this issue is generally overlooked.

In this article, we propose a mini-batch version of the MCMC-SAEM algorithm (Delyon et al., 1999; Kuhn and Lavielle, 2004). The original MCMC-SAEM algorithm is a powerful alternative to EM when the E-step is intractable. This is particularly interesting for nonlinear models or non-Gaussian models, where the unobserved data cannot be simulated exactly from the conditional distribution. Moreover, the MCMC-SAEM algorithm is also more computing efficient than the MCMC-EM algorithm, since only a single instance of the latent variable is sampled at every iteration of the algorithm. Nevertheless, when the dimension of the latent variable is huge, the sampling step can be time consuming. From this point of view the here proposed mini-batch version is computationally more efficient than the original MCMC-SAEM, since at each iteration only a small proportion of the latent variable is simulated and only the corresponding data are visited. For curved exponential models, we prove almost-sure convergence of the sequence of estimates generated by the mini-batch MCMC-SAEM algorithm under classical conditions as the number of iterations of the algorithm increases. Moreover, we assess

in simulation experiments the influence of the mini-batch size on the speed-up of the convergence in various models and highlight that an appropriate choice of the mini-batch size results in an important gain. We also present insights on the effect of the mini-batch size on the limit distribution of the estimates. Numerical results illustrate our findings.

2 Latent variable model and algorithm

This section first presents the general latent variable model and the required assumptions. Then the original MCMC-SAEM algorithm is described, before presenting the new mini-batch version of the MCMC-SAEM algorithm.

2.1 Model and assumptions

Consider the common latent variable model with observed (incomplete) data $\mathbf{y}$ and latent (unobserved) variable $\mathbf{z}$. Denote n the dimension of the latent variable $\mathbf{z} = (z_1 \dots, z_n) \in \mathbb{R}^n$. In many latent variable models, n corresponds to the number of observations, but it is not necessary that $\mathbf{z}$ and $\mathbf{y}$ have the same size or that each observation y_i depends only on a single latent component z_j.

Denote $\theta \in \Theta \subset \mathbb{R}^d$ the model parameter of the joint distribution of the complete data $(\mathbf{y}, \mathbf{z})$. In what follows, omitting all dependencies in the observations $\mathbf{y}$, which are considered as fixed realizations in the analysis, we assume that the complete-data likelihood function has the following form

$$f(\mathbf{z}; \theta) = \exp\left\{-\psi(\theta) + \langle S(\mathbf{z}), \phi(\theta)\rangle\right\} c(\mathbf{z}), \tag{1}$$

where $\langle \cdot, \cdot \rangle$ denotes the scalar product, $S(\mathbf{z})$ denotes a vector of sufficient statistics of the model with values in $\mathcal{S}$ and ψ and ϕ are functions on Θ. The posterior distribution of the latent variables $\mathbf{z}$ given the observations, is denoted by $\pi(\cdot; \theta)$.

2.2 Description of MCMC-SAEM algorithm

The original MCMC-SAEM algorithm proposed by Kuhn and Lavielle (2004) consists in replacing the E-step in the EM-algorithm by a simulation step combined with a stochastic approximation step. Here, we focus on a version where the MCMC part is a Metropolis-Hastings-within-Gibbs algorithm (Robert and Casella, 2004). More precisely, at iteration k, the following three steps are performed.

Simulation step A new realization $\mathbf{z}_k$ of the latent variable is sampled from an ergodic Markov transition kernel $\Pi(\mathbf{z}_{k-1}, \cdot|\theta_{k-1})$, whose stationary distribution is the posterior distribution $\pi(\cdot; \theta_{k-1})$. In practice, one iteration of a Metropolis-Hastings-within-Gibbs algorithm is used. We consider a collection $(\Pi_i)_{1\leq i\leq n}$ of symmetric random walk Metropolis kernels defined on $\mathbb{R}^n$ and acting only on the i-th coordinate, see Fort et al. (2003). More precisely, let$(\mathbf{e}_i)_{1\leq i\leq n}$ be the canonical basis of $\mathbb{R}^n$. Then, for each $i \in \{1, \dots, n\}$ starting from the n-vector $\mathbf{z} = (z_1, \dots, z_n)$, the proposal in the direction of $\mathbf{e}_i$ is given by $\mathbf{z} + x\mathbf{e}_i$, where $x \in \mathbb{R}$ is sampled from a symmetric increment density q_i. This proposal is then accepted with probability $\min\{1, \pi(\mathbf{z} + x\mathbf{e}_i; \theta_{k-1})/\pi(\mathbf{z}; \theta_{k-1})\}$.

Stochastic approximation step The approximation of the sufficient statistic is updated by

$$\mathbf{s}_k = (1 - \gamma_k)\mathbf{s}_{k-1} + \gamma_k S(\mathbf{z}_k), \tag{2}$$

where $(\gamma_k)_{k\geq 1}$ is a decreasing sequence of positive step-sizes such that $\sum_{k=1}^{\infty} \gamma_k = \infty$ and $\sum_{k=1}^{\infty} \gamma_k^2 < \infty$. That is, the current approximation $\mathbf{s}_k$ of the sufficient statistic $S(\mathbf{z})$ is a weighted mean of its previous value $\mathbf{s}_{k-1}$ and the value $S(\mathbf{z}_k)$ obtained by the current simulation step.

Algorithm 1 Mini-batch MCMC-SAEM

Input: data $\mathbf{y}$.
Initialization: Choose initial values θ_0, $\mathbf{s}_0$, $\mathbf{z}_0$.
Set $k = 1$.
while not converged **do**
  Sample $r \sim \text{Bin}(n, \alpha)$.
  Sample r indices from $\{1, \dots, n\}$, denoted by $\mathcal{I}_k$.
  Set $\mathbf{z}_k = \mathbf{z}_{k-1}$
  for $i \in \mathcal{I}_k$ **do**
    Sample $\mathbf{z} \sim \Pi_i(\mathbf{z}_k, \cdot|\theta_{k-1})$.
    Set $\mathbf{z}_k = \mathbf{z}$.
  end for
  $\mathbf{s}_k = (1 - \gamma_k)\mathbf{s}_{k-1} + \gamma_k S(\mathbf{z}_k)$.
  Update parameter $\theta_k = \hat{\theta}(\mathbf{s}_k)$.
  Increment k.
end while

Maximization step The model parameter θ is updated by

$$\theta_k = \hat{\theta}(\mathbf{s}_k),$$

with $\hat{\theta}(\mathbf{s}) = \arg\max_{\theta \in \Theta} L(\theta; \mathbf{s})$ where $L(\theta; \mathbf{s}) = -\psi(\theta) + \langle \mathbf{s}, \phi(\theta) \rangle$.

Depending on the model the maximization problem may not have a closed-form solution.

2.3 Mini-batch MCMC-SAEM algorithm

When n is large, the simulation step can be very time-consuming. Indeed, simulating *all* components z_i of the latent variable at every iteration is costly in time. Thus, according to the spirit of other mini-batch algorithms, updating only a part of the latent components may speed up the convergence of the algorithm. With this idea in mind, denote $0 < \alpha \leq 1$ the (mean) proportion of components of the latent variable $\mathbf{z}$ that are updated during one iteration.

Mini-batch simulation step In the mini-batch version of the MCMC-SAEM algorithm the simulation step consists of two parts. First, select the indices of the components z_i of the latent variable that will be updated. That is, we sample the number r of indices from a binomial distribution $\text{Bin}(n, \alpha)$ and then select randomly r indices among $\{1, \dots, n\}$ without replacement. Denote $\mathcal{I}_k$ this set of selected indices at iteration k. Next, at iteration k, instead of sampling all the components $z_{k,i}$, we may sample only the components $z_{k,i}$ with index $i \in \mathcal{I}_k$ from the Markov kernel $\Pi_i(\cdot, \cdot|\theta_{k-1})$ using one iteration of a Metropolis-Hastings-within-Gibbs algorithm.

Stochastic approximation and maximization step These two steps are identical to the original SAEM algorithm. However, in many models the evaluation of the sufficient statistic $S(\mathbf{z}_k)$ can be performed by an update of a small part of the components of its previous value $S(\mathbf{z}_{k-1})$. This may be computationally more efficient than computing $S(\mathbf{z}_k)$ naively.

Initialization Initial values θ_0, $\mathbf{s}_0$ and $\mathbf{z}_0$ for the model parameter, the sufficient statistic and the latent variable, respectively, have to be chosen by the user or at random.

See Algorithm 1 for a complete description of the algorithm.

3 Convergence of the algorithm

In this section we show that the mini-batch MCMC-SAEM algorithm converges as the number of iterations increase under classical conditions (which are the ones that ensure convergence of the original MCMC-SAEM algorithm). Indeed, compared to classical MCMC-SAEM algorithm, the mini-batch version involves an additional stochastic part that comes from the selection of indexes of the latent variable to be updated. This additional randomness is ruled by the value of α.

3.1 Equivalent descriptions

The above description of the simulation step is convenient for achieving maximal computing efficiency. We now focus on an equivalent framework that underlines the fact that our mini-batch algorithm is a special case of the MCMC-SAEM classical algorithm. For two kernels P_1, P_2, we denote their composition by

$$P_2 \circ P_1(z, z') = \int_{\tilde{\mathbf{z}}} P_2(\tilde{\mathbf{z}}, \mathbf{z}') P_1(\mathbf{z}, d\tilde{\mathbf{z}}).$$

With this notation at hand, the Metropolis-Hastings-within-Gibbs relies on the kernel $\Pi = \Pi_n \circ \cdots \circ \Pi_1$. Now, in the mini-batch simulation step, the update of only a part of the latent variable is equivalent to generating latent vectors $\mathbf{z} = (z_1, \ldots, z_n)$ according to the Markov kernel $\Pi_\alpha(\cdot, \cdot|\theta)$ defined as the following recursive composition

$$\begin{aligned}
\Pi_\alpha(\mathbf{z}, \mathbf{z}'|\theta) &= (\Pi_{\alpha,n} \circ \cdots \circ \Pi_{\alpha,1})(\mathbf{z}, \mathbf{z}'|\theta) \\
\text{where } \Pi_{\alpha,i}(\mathbf{z}, \mathbf{z}'|\theta) &= \alpha \Pi_i(\mathbf{z}, (z_1, \ldots, z_i', \ldots, z_n)|\theta) + (1-\alpha)\delta_{\mathbf{z}}(\mathbf{z}'), \\
\text{and thus } \Pi_\alpha(\mathbf{z}, \mathbf{z}'|\theta) &= \sum_{k=0}^{n} \alpha^k (1-\alpha)^{n-k} \sum_{1 \le i_1 < \cdots < i_k \le n} (\Pi_{i_k} \circ \cdots \circ \Pi_{i_1})(\mathbf{z}, \mathbf{z}'|\theta), \qquad (3)
\end{aligned}$$

where $\delta_a(\cdot)$ denotes the Dirac measure at a. That is, $\Pi_\alpha(\cdot, \cdot|\theta)$ is the composition of mixtures of the original kernels $\Pi_i(\cdot, \cdot|\theta)$ and a deterministic component with mixing weight α. In other words, the mini-batch MCMC-SAEM algorithm corresponds to the original MCMC-SAEM algorithm with a particular choice of the transition kernel. Note that it can also be seen as a mixture over different trajectories (the choice of indexes $i_1 < \cdots < i_k$ to be updated) of Metropolis-Hastings-within-Gibbs kernels acting on a subpart of the latent vector $\mathbf{z}$.

A third description of the algorithm will be appropriate for the analysis of its theoretical properties. In order to stress the role of the mini-batch procedure, we denote by $(\mathbf{z}_k^\alpha)_k$ the sequence of latent variables obtained by the mini-batch algorithm with mini-batch size α. In the k-th mini-batch simulation step, for each $i \in \{1, \ldots, n\}$, we sample a Bernoulli random variable $U_{k,i}$ with parameter α. This is an indicator of whether the latent variable $\mathbf{z}_{k-1,i}^\alpha$ is updated at iteration k. Next, we sample a realization $\tilde{\mathbf{z}}_k$ from the transition kernel Π_i and set

$$\mathbf{z}_{k,i}^\alpha = U_{k,i}\tilde{\mathbf{z}}_{k,i} + (1 - U_{k,i})\mathbf{z}_{k-1,i}^\alpha. \qquad (4)$$

When $\alpha = 1$, the sequence $(z_k^1)_k$ generated by the batch algorithm is a Markov chain with transition kernel $\Pi = \Pi_n \circ \cdots \circ \Pi_1$.

Fort et al. (2003) establish results on the geometric ergodicity of hybrid samplers and in particular for the random-scan Gibbs sampler. The latter is defined as $n^{-1}\sum_{i=1}^n \Pi_i$, where each Π_i is a kernel on $\mathbb{R}^n$ acting only on the i-th component. More generally the random-scan Gibbs sampler may be defined as $\sum_{i=1}^n a_i \Pi_i$ where $(a_1, \ldots, a_n)$ is a probability distribution. This means that at each step of the algorithm, only one component i is drawn from the probability distribution $(a_1, \ldots, a_n)$ and then updated. These probabilities may be chosen uniformly ($a_i = 1/n$) or for e.g. can help focusing on a component that is more difficult to simulate. We

generalize their results to a setup where at each step k, we rely on a kernel Π_α iterated from a random-scan Gibbs sampler $\widetilde{\Pi}_\alpha$ as follows

$$\Pi_\alpha(\cdot,\cdot|\theta_k) = \widetilde{\Pi}_\alpha(\cdot,\cdot|\theta_k)^{\sum_i U_{i,k}} \tag{5}$$

$$\text{and} \quad \widetilde{\Pi}_\alpha(\cdot,\cdot|\theta_k) = \begin{cases} (\sum_{i=1}^n U_{k,i})^{-1} \sum_{i=1}^n U_{k,i}\Pi_i(\cdot,\cdot|\theta_k) & \text{if } \sum_{i=1}^n U_{k,i} \geq 1, \\ \mathrm{Id} & \text{else} \end{cases}$$

where Id denotes the identity kernel $\mathrm{Id}(\mathbf{z},\mathbf{z}') = \mathbf{1}\{\mathbf{z}=\mathbf{z}'\}$. Note that this is not exactly the kernel corresponding to the algorithm described above, as here this one could formally update the same component i more than once in one step of the algorithm. Nonetheless, we neglect this effect and establish our result for the algorithm corresponding to this kernel.

3.2 Assumptions and result

Assume that the random variables $\mathbf{s}_0, \mathbf{z}_1, \mathbf{z}_2, \ldots$ are defined on the same probability space $(\Omega, \mathcal{A}, P)$. We denote $\mathcal{F} = \{\mathcal{F}_k\}_{k\geq 0}$ the increasing family of σ-algebras generated by the random variables $\mathbf{s}_0, \mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k$. Assume that the following regularity conditions on the model hold.

(M1) The parameter space Θ is an open subset of $\mathbb{R}^d$. The complete data likelihood function is given by

$$f(\mathbf{z};\theta) = \exp\left(-\psi(\theta) + \langle S(\mathbf{z}), \phi(\theta)\rangle\right) c(\mathbf{z}),$$

where S is a continuous function on $\mathbb{R}^n$ taking its values in an open subset $\mathcal{S}$ of $\mathbb{R}^m$. Moreover, the convex hull of $S(\mathbb{R}^n)$ is included in $\mathcal{S}$ and for all $\theta \in \Theta$

$$\int S(\mathbf{z})\pi(\mathbf{z};\theta)d\mathbf{z} < \infty.$$

(M2) The functions ψ and ϕ are twice continuously differentiable on Θ.

(M3) The function $\bar{s} : \Theta \to \mathcal{S}$ defined as $\bar{\mathbf{s}}(\theta) \triangleq \int S(\mathbf{z})\pi(\mathbf{z};\theta)d\mathbf{z}$ is continuously differentiable on Θ.

(M4) The observed-data log-likelihood function $l : \Theta \to \mathbb{R}$ defined as $l(\theta) \triangleq \log \int f(\mathbf{z};\theta)d\mathbf{z}$ is continuously differentiable on Θ and $\partial_\theta \int f(\mathbf{z};\theta)d\mathbf{z} = \int \partial_\theta f(\mathbf{z};\theta)d\mathbf{z}$.

(M5) Define $L : \mathcal{S} \times \Theta \to \mathbb{R}$ as $L(\mathbf{s};\theta) \triangleq -\psi(\theta) + \langle \mathbf{s}, \phi(\theta)\rangle$. There exists a continuously differentiable function $\widehat{\theta} : \mathcal{S} \to \Theta$, such that

$$\forall \mathbf{s} \in \mathcal{S}, \quad \forall \theta \in \Theta, \quad L(\mathbf{s};\widehat{\theta}(\mathbf{s})) \geq L(\mathbf{s};\theta).$$

We now introduce the usually required conditions for proving convergence of the SAEM procedure.

(SAEM1) For all k in $\mathbb{N}$, $\gamma_k \in [0,1]$, $\sum_{k=1}^\infty \gamma_k = \infty$ and $\sum_{k=1}^\infty \gamma_k^2 < \infty$.

(SAEM2) The functions $l : \Theta \to \mathbb{R}$ and $\widehat{\theta} : \mathcal{S} \to \Theta$ are m times differentiable, where we recall that $\mathcal{S}$ is an open subset of $\mathbb{R}^m$.

For any $\mathbf{s} \in \mathcal{S}$, we define $H_\mathbf{s}(\mathbf{z}) = S(\mathbf{z}) - \mathbf{s}$ and its expectation with respect to the posterior distribution $\pi(\cdot;\hat{\theta}(\mathbf{s}))$ denoted by $h(\mathbf{s}) = \mathrm{E}_{\hat{\theta}(\mathbf{s})}[S(\mathbf{z})] - \mathbf{s}$. For any $\rho > 0$, denote $V_\rho(\mathbf{z}) = \sup_{\theta\in\Theta}[\pi(\mathbf{z};\theta)]^\rho$. We consider the following additional assumptions.

(H1) There exists a constant M_0 such that

$$\mathcal{L} = \left\{\mathbf{s} \in \mathcal{S}, \langle \nabla\ell(\hat{\theta}(\mathbf{s})), h(\mathbf{s})\rangle = 0\right\} \subset \{\mathbf{s} \in \mathcal{S}, -\ell(\hat{\theta}(\mathbf{s})) < M_0\}.$$

In addition, there exits $M_1 \in (M_0, \infty]$ such that $\{\mathbf{s} \in \mathcal{S}, -\ell(\hat{\theta}(\mathbf{s})) \leq M_1\}$ is a compact set.

(H2) The family $\{q_i\}_{1\leq i\leq n}$ of symmetric densities is such that there exist some constants $\eta_i > 0$ and $\delta_i < \infty$ (for $i = 1, \ldots, n$) such that $q_i(x) > \eta_i$ whenever $|x| < \delta_i$.

(H3) There are constants δ and Δ with $0 \leq \delta \leq \Delta \leq \infty$ such that

$$\xi \triangleq \inf_{i=1,\ldots,n} \int_{\delta}^{\Delta} q_i(x)dx > 0,$$

and, for any sequence $\{\mathbf{z}^j\}$ with $\lim_j |\mathbf{z}^j| = \infty$, we may extract a subsequence $\{\tilde{\mathbf{z}}^j\}$ with the property that, for some $i \in \{1, \ldots, n\}$, all $x \in [\delta, \Delta]$, and all $\theta \in \Theta$

$$\lim_j \frac{\pi(\tilde{\mathbf{z}}^j; \theta)}{\pi(\tilde{\mathbf{z}}^j - \mathrm{sign}(z_i^j)x\mathbf{e}_i; \theta)} = 0, \quad \text{and} \quad \lim_j \frac{\pi(\tilde{\mathbf{z}}^j + \mathrm{sign}(z_i^j)x\mathbf{e}_i; \theta)}{\pi(\tilde{\mathbf{z}}^j; \theta)} = 0.$$

(H4) There exist $C > 1$, $\rho \in (0, 1)$ and $\theta_0 \in \Theta$ such that for all $\mathbf{z} \in \mathbb{R}^n$,

$$|S(\mathbf{z})| \leq C\pi(\mathbf{z}; \theta_0)^{-\rho}.$$

To state our convergence result, we consider the version of the algorithm with truncation on random boundaries studied by Andrieu et al. (2005). This additional projection step ensures in particular the stability of the algorithm for the theoretical analysis and is only a technical tool for the proof without any practical consequences.

Theorem 1. *Assume* **(M1)**–**(M5)**, **(SAEM1)**–**(SAEM2)** *and* **(H1)**–**(H4)**. *Then, for all* $0 < \alpha \leq 1$, *the sequence* $(\theta_k)_{k\geq 1}$ *generated by the mini-batch MCMC-SAEM algorithm with corresponding Markov kernel* $\Pi_\alpha(\cdot, \cdot|\theta)$ *converges towards the set of critical points of the observed likelihood* $\ell(\theta)$ *as the number of iterations increases.*

Proof of Theorem 1. The proof consists of two steps. First, we prove the convergence of the sequence of sufficient statistics $(\mathbf{s}_k)_k$ towards the set of zeros of function h using Theorem 5.5 in Andrieu et al. (2005). Then, in a second step, following the usual reasoning for EM-type algorithms, described for instance in Delyon et al. (1999), we deduce that the sequence $(\theta_k)_k$ converges to the set of critical points of the observed data log-likelihood ℓ.

First step. In order to apply Theorem 5.5 in Andrieu et al. (2005), we need to establish that their conditions (A1) to (A4) are satisfied. In what follows, (A1) to (A4) always refer to the conditions stated in Andrieu et al. (2005). First, note that under our assumptions **(H1)**, **(M1)**–**(M5)** and **(SAEM2)**, condition (A1) is satisfied. Indeed, this is a consequence of Lemma 2 in Delyon et al. (1999). To establish (A2) and (A3), as suggested in Andrieu et al. (2005), we establish their drift conditions (DRI), see Proposition 6.1 in Andrieu et al. (2005). We first focus on establishing (DRI1) in Andrieu et al. (2005). To this aim, we rely on Fort et al. (2003) that establishes results for the random-scan Metropolis sampler. In their context, they consider a sampler $\Pi = n^{-1}\sum_{i=1}^n \Pi_i$. We generalize their results to our setup according to (5). Following the lines of the proof of Theorem 2 in Fort et al. (2003), we can show that Equations (6.1) and (6.3) appearing in the drift condition (DRI1) in Andrieu et al. (2005) are satisfied as soon as **(H2)-(H3)** hold. Indeed following the strategy developed in Allassonnière et al. (2010), we first establish Equations (6.1) and (6.3) using a drift function depending on θ namely $V_\theta(\mathbf{z}) = \pi(\mathbf{z}; \theta)^{-\rho}$ where ρ is given in **(H4)**. Then we define the common drift function V as follow. Let $\theta_0 \in \Theta$ and ρ given in **(H4)** and define $V(\mathbf{z}) = \pi(\mathbf{z}; \theta_0)^{-\rho}$. Then for any compact $\mathcal{K} \in \Theta$, there exist two positive constants $c_\mathcal{K}$ and $C_\mathcal{K}$ such that for all $\theta \in \mathcal{K}$ and for all $\mathbf{z}$, we get $c_\mathcal{K} V(\mathbf{z}) \leq \pi(\mathbf{z}; \theta)^{-\rho} \leq C_\mathcal{K} V(\mathbf{z})$. We then establish Equations (6.1) and (6.3) for this drift function V. Moreover, using Proposition 1 and Proposition 2 in Fort et al. (2003) we obtain that Equation (6.2) in (DRI1) from Andrieu et al. (2005) holds. Under assumption

(H4) we have the first part of (DRI2) in Andrieu et al. (2005). The second part of the same equation is true with $\beta = 1$ in our case. Finally, (DRI3) in Andrieu et al. (2005) is true in our context with $\beta = 1$ because $\mathbf{s} \mapsto \hat{\theta}(\mathbf{s})$ is twice continuously differentiable, thus Lipschitz on any compact set. To prove this, we decompose the space in the acceptance and rejection regions and consider the integral over four sets leading to four different expressions of the acceptance ratio (see for example proof of Lemma 4.7 in Fort et al., 2015). This concludes that (DRI) and therefore (A2)–(A3) in Andrieu et al. (2005) are satisfied. Notice that **(SAEM1)** ensures (A4). This concludes the first step of the proof.

Second step. As the function $\mathbf{s} \mapsto \hat{\theta}(\mathbf{s})$ is continuous, the second step is immediate by applying Lemma 2 in Delyon et al. (1999). □

4 Experiments

We carry out various simulation experiments in a frailty model, a nonlinear mixed effects model and a Bayesian deformable template model to illustrate the performance of the proposed mini-batch MCMC-SAEM algorithm and the potential gain in efficiency.

4.1 Frailty model in survival analysis

In survival analysis the frailty model is an extension of the well-known Cox model (Duchateau and Janssen, 2008). Indeed, the hazard rate function in the frailty model includes an additional random effect called frailty to account for unexplained heterogeneity.

Model We observe survival times $\mathbf{t} = (t_{ij})_{1 \leq i \leq n, 1 \leq j \leq m}$ measured over n groups with m measurements per group, and covariates $\mathbf{X}_{ij} \in \mathbb{R}^p$. The latent random variables $\mathbf{z} = (z_1, \ldots, z_n)$ correspond to the frailty terms, each component being the frailty of one group. The z_i's are supposed to be i.i.d. with centered Gaussian distribution with variance σ^2. We denote by λ_0 the baseline hazard function. Here we choose for λ_0 the Weibull function given by

$$\lambda_0(t) = \lambda_0 \rho t^{\rho - 1}, \quad t > 0,$$

with $\lambda_0 > 0$ and $\rho > 1$. The conditional hazard rate $\lambda_{ij}(\cdot|z_i)$ of observation t_{ij} given the frailty z_i is given by

$$\lambda_{ij}(\cdot|z_i) = \lambda_0(\cdot) \exp(\mathbf{X}_{ij}{}^\intercal \boldsymbol{\beta} + z_i),$$

where $\boldsymbol{\beta} \in \mathbb{R}^p$. Thus the unknown model parameter is $\theta = (\boldsymbol{\beta}, \sigma^2, \lambda_0, \rho)$. In practical applications the main interest lies in the estimation of the regression parameter $\boldsymbol{\beta}$.

Likelihood In the frailty model the conditional survival function is given by

$$\begin{aligned} G_{ij}(t|z_i) &= \mathbb{P}(t_{ij} > t|z_i) \\ &= \exp[-\lambda_0 t^\rho \exp(\mathbf{X}_{ij}{}^\intercal \boldsymbol{\beta} + z_i)]. \end{aligned}$$

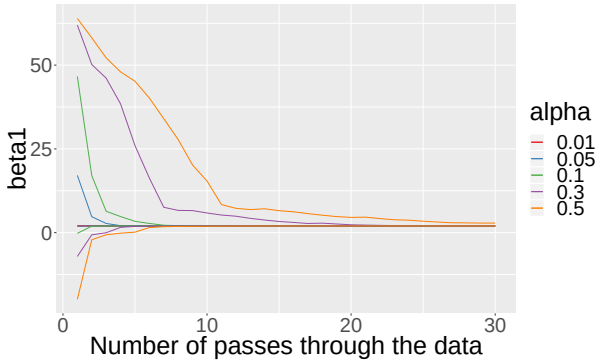
In other words, the conditional distribution of the survival time t_{ij} given z_i is the Weibull distribution with scale $\lambda_0 \exp(\mathbf{X}_{ij}{}^\intercal \boldsymbol{\beta} + z_i)$ and shape ρ. For the conditional and the complete likelihood function we obtain

$$\mathbb{P}_\theta(\mathbf{t}|\mathbf{z}) = \prod_{i=1}^{n} \prod_{j=1}^{m} G_{ij}(t_{ij}|z_i) \lambda_{ij}(t_{ij}|z_i),$$

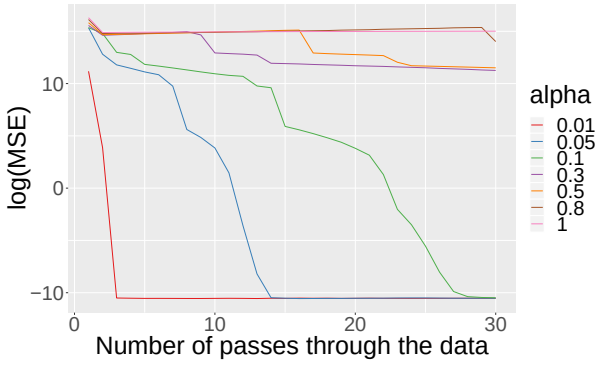
$$\mathbb{P}_\theta(\mathbf{t}, \mathbf{z}) = \mathbb{P}_\theta(\mathbf{t}|\mathbf{z}) \prod_{i=1}^{n} \varphi\left(\frac{z_i}{\sigma}\right),$$

where φ denotes the density of the standard normal distribution.

The complete likelihood may be written in the form (1), with sufficient statistics $S(\mathbf{z}) = (S_0(\mathbf{z}), S_1(\mathbf{z}), \ldots, S_n(\mathbf{z}))$ where $S_0(\mathbf{z}) = \sum_{i=1}^{n} z_i^2/n$ and $S_i(\mathbf{z}) = \exp(z_i)$ for $i = 1, \ldots, n$.



a)



b)

Figure 1: Evolution of the precision of the mean of the estimates of β_1 with respect to the number of passes through the data for proportions $\alpha \in \{0.01, 0.05, 0.1, 0.3, 0.5, 0.8, 1\}$. a) 80%-confidence bands for β_1. b) Logarithm of the mean squared error of β_1.

Simulation step In the simulation step we use the following sampling procedure. Let $\mathcal{I}_k$ be the subset of indices of latent variable components z_i that have to be updated at iteration k. For each $i \in \mathcal{I}_k$, we use a Metropolis-Hastings procedure: first, draw a candidate $\tilde{z}_{k,i}$ from the normal distribution $\mathcal{N}(z_{k-1,i}, 0.2)$, then compute the logarithm of the acceptance ratio given by

$$\log(\tau_{k,i}) = m(\tilde{z}_{k,i} - z_{k-1,i}) - \frac{1}{2\sigma_{k-1}^2}(\tilde{z}_{k,i}^2 - z_{k-1,i}^2)$$
$$- \lambda_0 \left(\mathrm{e}^{\tilde{z}_{k,i}} - \mathrm{e}^{z_{k-1,i}}\right) \left(\sum_{j=1}^{m} t_{ij}^{\rho} \exp[\mathbf{X}_{ij}{}^{\intercal} \boldsymbol{\beta}_{k-1}]\right).$$

Then, for a realization $\omega_{k,i}$ of the uniform distribution $U[0,1]$, we set $z_{k,i} = \tilde{z}_{k,i}$ if $\omega_{k,i} < \tau_{k,i}$, and $z_{k,i} = z_{k-1,i}$ otherwise.

Stochastic approximation step Compute the stochastic approximation of the sufficient statistics $(s_{k,i})_{0 \le i \le n}$ according to Eq. (2), that is, for $l = 0, \ldots, n$, compute

$$s_{k,l} = (1 - \gamma_k) s_{k-1,l} + \gamma_k S_l(\mathbf{z}_k),$$

where the sequence $(\gamma_k)_{k \ge 1}$ is chosen as $\gamma_k = 0.6$ for $1 \le k \le 20$ and $\gamma_k = (k - 100)^{-0.6}$ otherwise.

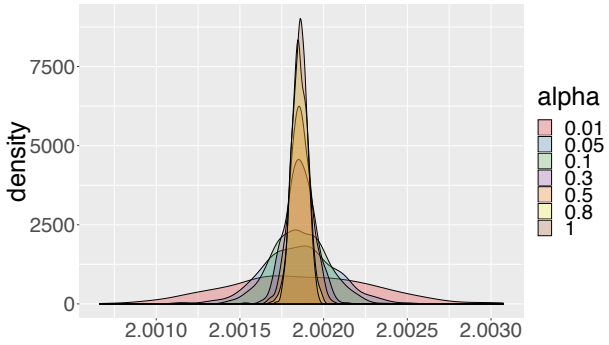


Figure 2: Estimation of the limit distribution of the estimate of β_1 after $K_0 = 8000$ iterations for the mini-batch algorithm with proportions $\alpha \in \{0.01, 0.05, 0.1, 0.3, 0.5, 0.8, 1\}$.

Maximization step The maximization of $\theta \mapsto \ell(\theta; \mathbf{s}_k)$ is explicit in σ^2 and λ, with solutions given by

$$\sigma_k^2 = s_{k,0}$$
$$\lambda_{0,k} = nm \left(\sum_{i=1}^{n} \sum_{j=1}^{m} t_{ij}^{\rho} \exp[\mathbf{X}_{ij}{}^{\intercal} \boldsymbol{\beta}_k] s_{k,i} \right)^{-1}.$$

The update of $\boldsymbol{\beta}$ and ρ are done by Newton's method, as these maximizations are not explicit.

Numerical results In a simulation study we consider the frailty model with parameters fixed to $\boldsymbol{\beta} = (\beta_1, \beta_2) = (2, 3), \lambda_0 = 3$, $\sigma^2 = 2$ and $\rho = 3.6$. We set $n = 5000$ and $m = 100$. The covariates $\mathbf{X}_{ij}$ are drawn independently from the uniform distribution for every dataset.

In the first setting, 500 datasets are generated to which the mini-batch MCMC-SAEM with random initial values and different mini-batch sizes is applied, that is, with varying values of the proportion $\alpha \in \{0.01, 0.05, 0.1, 0.3, 0.5, 0.8, 1\}$. Figure 1 shows the evolution of the precision of the mean of the estimates $\bar{\beta}_{1,k} = \sum_{l=1}^{k} \beta_{1,l}/k$ of parameter component β_1 as a function of the number of passes through the data. That means that the value 10 on the x-axis, for example, corresponds to 10 iterations in the batch MCMC-SAEM ($\alpha = 1$) and to 100 iterations in the mini-batch MCMC-SAEM with $\alpha = 0.1$. That is, parameter estimators are compared when the different algorithms have visited (approximately) the same amount of data, or, to put it differently, when the algorithms have generated the same number of latent components z_i.

Figure 1 a) shows 80%-confidence bands for parameter component β_1, and graph b) shows the corresponding evolution of the logarithm of the mean squared error. Obviously, for all algorithms estimation improves when the number of passes through the data increases. Moreover, it is clear from both graphs that the rate of convergence depends extremely on the mini-batch size. Indeed, at (almost) any number of passes through the data, the mean squared error and the width of the confidence intervals is increasing in the proportion α. In this sense, the best choice that achieves the fastest convergence is a mini-batch size corresponding to the smallest value of α, here 0.01. From graph b) we see that convergence seems to be attained after only three passes through the data when $\alpha = 0.01$, while almost 30 are required when $\alpha = 0.1$. For larger mini-batch sizes convergence is even much slower.

In the second setting, we aim at studying the asymptotic behavior of the estimates, when the algorithms are supposed to have converged. Here, estimates of the different algorithms are compared at a fixed large number K_0 of iterations. That is, after K_0 iterations the batch algorithm has visited the whole dataset K_0 times, while the mini-batch version with e.g. proportion $\alpha = 0.5$ has only visited half as much information. To compute the variance of the estimates

Table 1: Standard deviation after $K_0 = 8000$ iterations.

Proportion α	β_1
0.01	4.19×10^{-4}
0.05	2.12×10^{-4}
0.1	1.58×10^{-4}
0.3	0.86×10^{-4}
0.5	0.65×10^{-4}
0.8	0.48×10^{-4}
1	0.42×10^{-4}

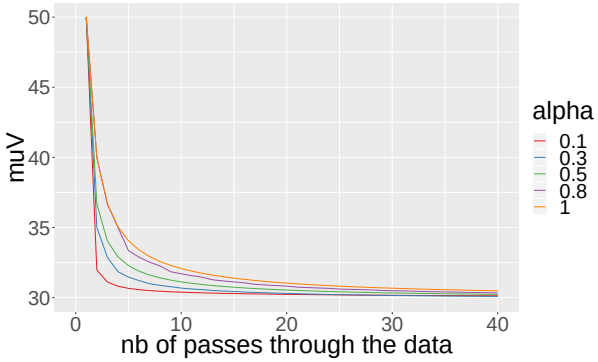


Figure 3: Estimates of the parameter μ_V using mini-batch MCMC-SAEM with $\alpha \in \{0.1, 0.3, 0.5, 0.8, 1\}$ as a function of the number of passes through the dataset.

that is only due to the stochasticity of the algorithm, we fix a dataset and run the mini-batch MCMC-SAEM algorithm 500 times using different initial values. We choose $K_0 = 8000$ iterations and consider proportions α varying in $\{0.01, 0.05, 0.1, 0.3, 0.5, 0.8, 1\}$. Figure 2 illustrates the limit distributions of the estimates of β_1 for different mini-batch sizes, which seem to be Gaussians centered at the same value but with varying variances. Table 1 gives the corresponding standard deviation of the estimates of β_1, which is clearly decreasing in α. This is expected as more data are visited for larger α. These results give some insight into the asymptotic behavior of the algorithms and in particular into the impact of the mini-batch size on the limit distribution, which is generally overlooked in the literature.

4.2 Nonlinear mixed model for pharmacokinetic study

In this section we consider a classical one-compartment model used in clinical pharmacokinetic studies. The model presented in Davidian and Giltinan (1995) serves to analyze the kinetic of the drug theophylline used in therapy for respiratory diseases. For $i = 1, \ldots, n$ and $j = 1, \ldots, J$, we define

$$y_{ij} = \frac{d_i \mathrm{ka}_i}{V_i \mathrm{ka}_i - \mathrm{Cl}_i} \left[e^{-\mathrm{Cl}_i t_{ij}/V_i} - e^{-\mathrm{ka}_i t_{ij}} \right] + \epsilon_{ij} \ ,$$

where the observation y_{ij} is the measure of drug concentration on individual i at time t_{ij}. The drug dose administered to individual i is denoted d_i. The parameters for individual i are the volume V_i of the central compartment, the constant ka_i of the drug absorption rate, and the drug's clearance Cl_i. The random measurement error is denoted by ϵ_{ij} and supposed to have a centered normal distribution with variance σ^2. For the individual parameters V_i, Cl_i and ka_i

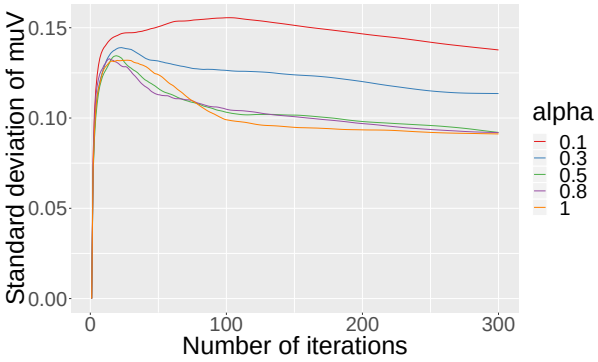


Figure 4: Estimates of the empirical standard deviation for the parameter μ_V using mini-batch MCMC-SAEM with $\alpha \in \{0.1, 0.3, 0.5, 0.8, 1\}$ as a function of the iteration number.

log-normal distributions are considered given by

$$\begin{aligned} \log V_i &= \log(\mu_V) + z_{i,1}, \\ \log \mathrm{ka}_i &= \log(\mu_{\mathrm{ka}}) + z_{i,2}, \\ \log \mathrm{Cl}_i &= \log(\mu_{\mathrm{Cl}}) + z_{i,3}, \end{aligned}$$

where $(z_{i,1}, z_{i,2}, z_{i,3})$ are independent following a centered normal distribution with variance $\Omega = \mathrm{diag}(\omega_V^2, \omega_{\mathrm{ka}}^2, \omega_{\mathrm{Cl}}^2)$. Then the model parameters are $\theta = (\mu_k, \mu_V, \mu_{\mathrm{Cl}}, \omega_V^2, \omega_{\mathrm{ka}}^2, \omega_{\mathrm{Cl}}^2, \sigma^2)$.

In a simulation study we estimate the parameters by both the original MCMC-SAEM algorithm (see Kuhn and Lavielle, 2005, for implementation details) and our mini-batch version. More precisely, one dataset is generated with the following values: $n = 1000, J = 10, \mu_k = 1.8, \mu_V = 30, \mu_{\mathrm{Cl}} = 3.5, \omega_V = 0.02, \omega_{\mathrm{ka}} = 0.04, \omega_{\mathrm{Cl}} = 0.06, \sigma^2 = 2$. The dose d_i is constant, equal to 320. The times t_{ij} are such that $t_{ij} = j$ for all i. Then we perform 100 repetitions of the estimation task by using the mini-batch MCMC-SAEM algorithm with $\alpha \in \{0.1, 0.3, 0.5, 0.8, 1\}$. We set $\gamma_k = 1$ for $1 \leq k \leq 50$ and $\gamma_k = (k - 50)^{-0.6}$ otherwise.

The results for parameter μ_V are shown in Figures 3 and 4. The other results are similar and therefore omitted. Figure 3 presents the mean parameter estimate sequence $\overline{\mu_V}_k = \sum_{l=1}^{k} \mu_{V,l}/k$ with respect to the number of passes through the dataset as in Figure 1 for different values of the proportion α. We observe that the smaller the proportion α, the faster the sequence of estimates converges, in particular during the first passes through the data. Figure 4 presents for different values of the proportion α the estimates of the empirical standard deviation with respect to the number of iterations. We observe that as the number of iterations increases, the standard deviations are lower than for higher values of α. This illustrates in particular that including more data in the inference task leads to more accurate estimation results. This is indeed very intuitive. Therefore choosing an optimal value for α remains to achieve a trade-off between speeding up the convergence and involving enough data in the process to get accurate estimates.

4.3 Deformable template model for image analysis

We consider the dense deformation template model introduced in Allassonnière et al. (2007). Such models allow to represent observed images as deformations of a given common reference image called template. We deal with the formulation proposed in Allassonnière et al. (2010). We observe n gray level images denoted by $(y_i)_{1 \leq i \leq n}$. Each image y_i is defined on a grid of pixels $\Lambda \subset \mathbb{R}^2$ where for each $s \in \Lambda$, x_s is the location of pixel s in a domain $D \subset \mathbb{R}^2$. We assume that each image derives from the deformation of a common unknown template I, which is a

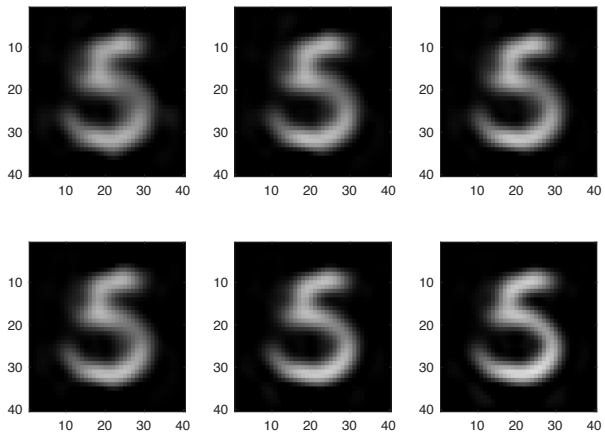


Figure 5: Estimation of the template: first row : using batch MCMC-SAEM ; second row : using mini-batch MCMC-SAEM with $\alpha = 0.1$; columns correspond respectively to 1, 2 and 3 passes through the dataset.

function from D to $\mathbb{R}$. Furthermore we assume for each image the existence of an unobserved deformation field $\Phi : \mathbb{R}^2 \to \mathbb{R}^2$ such that

$$y(s) = I(x_s - \Phi(x_s)) + \sigma\epsilon(s),$$

where $\epsilon(s)$ are distributed with respect to a normalized Gaussian distribution and σ^2 denotes the variance. To formulate this complex problem in a simpler parametric way, the template I and the deformation Φ are supposed to have parametric forms as follows. Given a set of landmarks denoted by $(p_k)_{1\leq k\leq k_p}$ which covers the domain D, the template function I is parametrized by coefficients $\xi \in \mathbb{R}^{k_p}$ through

$$I_\xi = \mathbf{K_p}\xi, \quad \text{where} \quad (\mathbf{K_p}\xi)(x) = \sum_{k=1}^{k_p} K_p(x, p_k)\xi(k),$$

and K_p is a fixed known kernel. Likewise, for another fixed set of landmarks $(g_k)_{1\leq k\leq k_g} \in D$, the deformation field is given by

$$\Phi_z(x) = (\mathbf{K_g}z)(x) = \sum_{k=1}^{k_g} K_g(x, g_k)(z^{(1)}(k), z^{(2)}(k)),$$

where $z = (z^{(1)}, z^{(2)}) \in \mathbb{R}^{k_g} \times \mathbb{R}^{k_g}$ and again, K_g is a fixed known kernel. The variables z are the latent variables of this model and are assumed to follow a centered Gaussian distribution with covariance matrix Γ.

We refer to Allassonnière et al. (2010) for further details on the model and for the implemention of the MCMC-SAEM algorithm. We estimate all model parameters, namely (ξ, Γ, σ^2), with both algorithms - the batch and the mini-batch with $\alpha = 0.1$. For the first experiment we use 25 images from the United States Postal Service database. We present the results obtained on digit 5 as an illustrating example in Figure 5. We observe that during the first iterations of the algorithms the convergence is sped up by using the mini-batch version leading to a more contrasted and accurate template estimate after three passes through the dataset.

In the second experiment we assess the effect of the mini-batch version in the asymptotic behavior of the algorithm. Therefore we run the batch version on 20 images of the dataset and the mini-batch version with $\alpha = 0.2$ on 100 images of the United States Postal Service

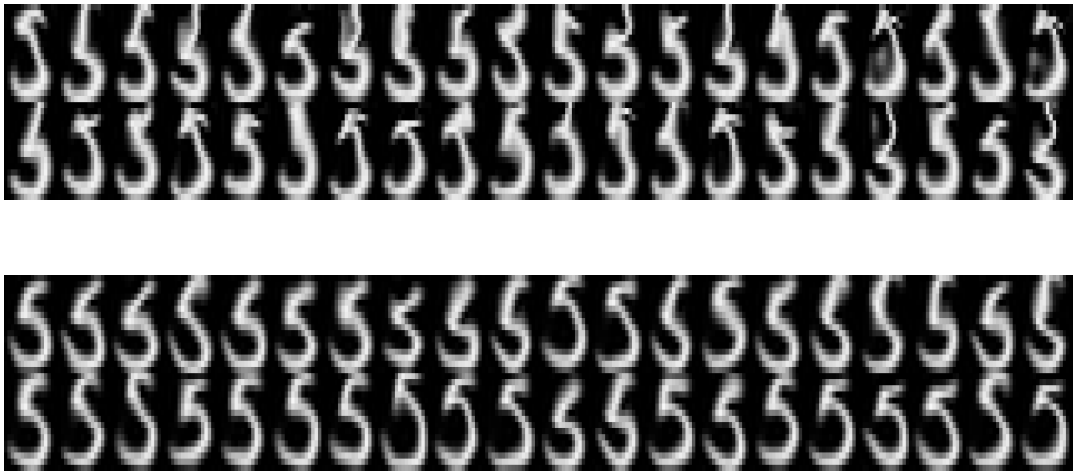


Figure 6: Generated synthetic samples from the model for digit 5 using the parameter estimates obtained with the batch version on 20 images (first row) and with the mini-batch version with $\alpha = 0.2$ on 100 images (second row).

database, both during 1000 iterations to reach asymptotic behavior. Note that choosing a small mini-batch size allows us to increase the number of images in the input, while the computing time of both algorithms is of the same order. To assess simultaneously the accuracy of the obtained estimates for the template and the deformation, we generate new samples from the model using both estimates of ξ and Γ. These results are presented in Figure 6. We observe that the samples of the second row look more like usual handwritten digits 5 as those of the database than the ones of the first row, highlighting that both template and deformation are better estimated by the mini-batch version performed on 100 images of the dataset. This shows that given a constraint on the computing time, more accuracy can be obtained by using the mini-batch MCMC-SAEM instead of the original algorithm.

5 Conclusion

The proposed mini-batch version of the MCMC-SAEM algorithm has a good theoretical foundation, as it is shown to be almost surely convergent whenever the original algorithm is. Moreover, in the simulations carried out in different models, the new algorithm turns out to achieve an important speed-up of convergence during the first passes through the data compared to the original algorithm. This opens the way to possibly drastic reductions of the computing time, or to increase accuracy by processing larger datasets and keeping the computing time fixed. Furthermore, our investigations on the limit distribution of the sequence of estimates generated by the algorithm yields: the larger the mini-batch size, the more concentrated is the limit distribution.

These results encourage the development of other mini-batch EM-type algorithms, as for example for the variational EM algorithm, in order to achieve similar speed-ups in further models and applications.

Finally, the question about an optimal choice of the mini-batch size arises naturally. According to our findings, an optimal value of the proportion α must simultaneously achieve two objectives: speeding up the convergence to drastically reduce the computing time, while estimating accurately model parameters. Therefore, it is of great interest to develop an empirical criterion leading to an optimal choice of the mini-batch size by operating a trade-off between accuracy and computing time.

All programs are available on request from the authors.

Acknowledgement

Work partly supported by the grant ANR-18-CE02-0010 of the French National Research Agency ANR (project EcoNet).

References

Allassonnière, S., Y. Amit, and A. Trouvé (2007). Toward a coherent statistical framework for dense deformable template estimation. *J. Roy. Statist. Soc. Ser. B 69*, 3–29.

Allassonnière, S., E. Kuhn, and A. Trouvé (2010). Construction of Bayesian deformable models via a stochastic approximation algorithm: A convergence study. *Bernoulli 16*(3), 641–678.

Andrieu, C., E. Moulines, and P. Priouret (2005). Stability of stochastic approximation under verifiable conditions. *SIAM J. Control Optim. 44*(1), 283–312.

Cappé, O. (2011). Online EM algorithm for hidden Markov models. *Journal Computational and Graphical Statistics 20*(3), 728–749.

Cappé, O. and E. Moulines (2009). On-line expectation-maximization algorithm for latent data models. *J. Roy. Statist. Soc. Ser. B 71*(3), 593–613.

Davidian, M. and D. M. Giltinan (1995). *Nonlinear Models for Repeated Measurement Data.* Chapman & Hall/CRC Press.

Delyon, B., M. Lavielle, and E. Moulines (1999). Convergence of a stochastic approximation version of the EM algorithm. *Ann. Statist. 27*(1), 94–128.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. Roy. Statist. Soc. Ser. B 39*(1), 1–38.

Duchateau, L. and P. Janssen (2008). *The frailty model.* New York, NY: Springer.

Fort, G., B. Jourdain, E. Kuhn, T. Lelièvre, and G. Stoltz (2015). Convergence of the wang-landau algorithm. *Mathematics of Computation 84*(295), 2297–2327.

Fort, G., E. Moulines, G. O. Roberts, and J. S. Rosenthal (2003). On the geometric ergodicity of hybrid samplers. *Journal of Applied Probability 40*, 123–146.

Karimi, B., M. Lavielle, and E. Moulines (2018). On the convergence properties of the mini-batch EM and MCEM algorithms. Unpublished.

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

Kuhn, E. and M. Lavielle (2005). Maximum likelihood estimation in nonlinear mixed effects models. *Comput Stat Data An. 49*(4), 1020–1038.

Lange, K. (1995, 01). A gradient algorithm locally equivalent to the EM algorithm. *J. Roy. Statist. Soc. Ser. B 2*(57), 425–437.

Liang, P. and D. Klein (2009). Online EM for unsupervised models. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pp. 611–619. Association for Computational Linguistics.

Neal, R. M. and G. E. Hinton (1999). A view of the EM algorithm that justifies incremental, sparse, and other variants. In M. I. Jordan (Ed.), *Learning in Graphical Models*, pp. 355–368. Cambridge, MA, USA: MIT Press.

Nguyen, H., F. Forbes, and G. McLachlan (2019). Mini-batch learning of exponential family finite mixture models. Technical report, arXiv:1902.03335.

Robert, C. P. and G. Casella (2004). *Monte Carlo statistical methods* (Second ed.). Springer Texts in Statistics. Springer-Verlag, New York.

Titterington, D. M. (1984). Recursive parameter estimation using incomplete data. *J. Roy. Statist. Soc. Ser. B 2*(46), 257–267.