Federated Expectation-Maximization with heterogeneity mitigation and variance reduction

Anonymous Author(s)
Affiliation
Address
email

Abstract

The Expectation Maximization (EM) algorithm is the default algorithm for inference in latent variable models. As in any other field of machine learning, applications of latent variable models to very large datasets make the use of advanced parallel and distributed architecture mandatory. This paper introduces `FedEM`, which is the first extension of the EM algorithm to the federated learning context. `FedEM` is a new communication efficient method, which handles partial participation of local devices, and is robust to heterogeneous distribution of the datasets. To alleviate the communication bottleneck, `FedEM` compresses appropriately defined complete data sufficient statistics. We also develop and analyze an extension of `FedEM` to further incorporate a variance reduction scheme. In all cases, we derive finite-time complexity bounds for smooth non-convex problems. Numerical results are presented to support our theoretical findings, as well as an application to federated missing values imputation for biodiversity monitoring.

1 Introduction

The Expectation-Maximization algorithm is the most popular approach for inference in latent variable models. The EM algorithm, a special instance of the Majorize-Minimization (MM) algorithm [17], was formalized by [5] and is without doubt one of the fundamental algorithms in machine learning. Applications include among many others finite mixture analysis, latent factor models inference, and missing data; see [27, 21, 18, 9] and the references therein. As in any other field of machine learning, training latent variable models on very large datasets make the use of advanced parallel and distributed architecture mandatory. Federated learning (FL) [15, 28], which exploits the computation power of a large number of edge devices to perform distributed machine learning, is a powerful framework to achieve this goal.

The conventional EM algorithm is not suitable for federated learning settings. We propose several new distributed versions of the EM algorithm supporting compressed communication. More precisely, our objective is to minimise a nonconvex finite-sum smooth objective function

$$\mathrm{Argmin}_{\theta\in\Theta} F(\theta), \qquad F(\theta) \stackrel{\mathrm{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_i(\theta) + \mathsf{R}(\theta) \quad \Theta \subset \mathbb{R}^d \,, \tag{1}$$

where n is the number of workers/devices which are connected to a central server, and worker $\#i$ only has access to the local loss function $\mathcal{L}_i$; finally R is a penalty term which may be introduced to promote sparsity, regularity, etc. In latent variable models, $\mathcal{L}_i(\theta) = -m^{-1}\sum_{j=1}^{m} \log p(y_{ij};\theta)$, where $\{y_{i,j}\}_{j=1}^{m}$ are the observations available for worker i, and $p(y;\theta)$ is the *incomplete* likelihood. In addition, $p(y;\theta)$ is defined by marginalizing the *complete-data* likelihood $p(y,z;\theta)$ defined as the joint probability density function of the observation y and a non-observed latent variable $z \in \mathsf{Z}$, i.e.

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

$p(y;\theta) = \int p(y,z;\theta)\mu(\mathrm{d}z)$ where Z is the *latent space* and μ is a measure on Z. We focus in this paper on the case where $p(y,z;\theta)$ belongs to a curved exponential family, given by

$$p(y,z;\theta) \stackrel{\text{def}}{=} \rho(y,z)\exp\left\{\langle s(y,z),\phi(\theta)\rangle - \psi(\theta)\right\} ; \tag{2}$$

where $s(y,z) \in \mathbb{R}^q$ is the *complete data sufficient statistics*, $\phi : \Theta \to \mathbb{R}^q$ and $\psi : \Theta \to \mathbb{R}$, $\rho : \mathsf{Y} \times \mathsf{Z} \to \mathbb{R}^+$ are vector/scalar functions.

In absence of communication constraints, the EM algorithm is a popular method to solve (1). It alternates between two steps: in the Expectation (E) step, using the current value of the iterate θ_{curr}, we compute a majorizing function $\theta \mapsto \mathsf{Q}(\theta,\theta_{\text{curr}})$ given up to an additive constant by

$$\mathsf{Q}(\theta,\theta_{\text{curr}}) \stackrel{\text{def}}{=} -\langle \bar{\mathsf{s}}(\theta_{\text{curr}}),\phi(\theta)\rangle + \psi(\theta) + \mathsf{R}(\theta) \quad \text{where} \quad \bar{\mathsf{s}}(\theta) \stackrel{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n}\bar{\mathsf{s}}_i(\theta) ; \tag{3}$$

and $\bar{\mathsf{s}}_i(\theta)$ is the ith device conditional expectation of the complete data sufficient statistics:

$$\bar{\mathsf{s}}_i(\theta) \stackrel{\text{def}}{=} \frac{1}{m}\sum_{j=1}^{m}\bar{\mathsf{s}}_{ij}(\theta) , \quad \bar{\mathsf{s}}_{ij}(\theta) \stackrel{\text{def}}{=} \int_{\mathsf{Z}} s(y_{ij},z)p(z|y_{ij};\theta)\mu(\mathrm{d}z) , \tag{4}$$

where $p(z|y_{ij};\theta) \stackrel{\text{def}}{=} p(y_{ij},z;\theta)/p(y_{ij};\theta)$. As for the Maximization (M) step, an updated value θ_{curr} is computed as a minimizer of $\theta \mapsto \mathsf{Q}(\theta,\theta_{\text{curr}})$. The majorizing function is then updated with the new θ_{curr}; this process is iterated until convergence. The EM algorithm is most useful when for any $\theta_{\text{curr}} \in \Theta$, the function $\theta \mapsto \mathsf{Q}(\theta,\theta_{\text{curr}})$ is a convex function of the parameter θ which is solvable in θ either explicitly or with little computational effort. A major advantage of the EM algorithm stems from its invariance under homeomorphisms, contrary to classical first-order methods: the EM updates are the same for any continuous invertible re-parametrization [16].

In the federated learning context, the vanilla EM algorithm is affected by three major problems: (1) the communication bottleneck, (2) data heterogeneity, and (3) partial participation of the workers.

When the number of workers is large, the cost of communication becomes overwhelming. A classical technique to alleviate this problem is to use *communication compression*. Most federated learning algorithms are first order methods: compression is typically applied to stochastic gradients. Yet, these methods are not appropriate to solve (1), since they do not preserve the desirable homeomorphic invariance property.

Since workers are often user personal devices, the issue of data heterogeneity naturally arises. Contrary to classical federated learning settings where heterogeneity is characterized by the non-cancellation of the local gradients at the optimum, in the EM algorithm, heterogeneity manifests itself by the non-equality of *local* conditional expectation of the complete data sufficient statistics $\bar{\mathsf{s}}_i$; conditions must therefore be derived to ensure convergence at the central server.

Finally, only a subset of devices are active in each learning round, since all users may not be available or willing to participate. Thus, taking into account partial participation of the workers, and its impact on the convergence rate of algorithms, is a major issue.

- `FedEM`**.** The main contribution of our paper is a new method called `FedEM`, supporting communication compression, partial participation and data heterogeneity. In this algorithm, the workers compute an estimate of the *local complete data sufficient statistics* $\bar{\mathsf{s}}_i$ using a minibatch of data, apply an unbiased compression operator to a noise compensated version (using a technique inspired by [11, 10]) and send the result to the central server, which performs aggregation and model update.
- `VR-FedEM`**.** We improve `FedEM` by adding a variance reduction method inspired by the `SPIDER` framework [6] which has recently been extended to the EM framework [7]. When compared to `FedEM`, `VR-FedEM` additionally performs variance reduction for each worker, progressively alleviating the variance brought by the stochastic approximation of the local complete data sufficient statistics.
- **Theoretical analysis.** We introduce a unified theoretical framework which covers the convergence of `FedEM` and `VR-FedEM` algorithms, and establish convergence guarantees for finding an ϵ-stationary point (see Theorems 1 and 4). In both cases, we provide the number of optimization

steps $K_{\rm opt}(\epsilon)$, the number of per-sample conditional expectations $K_{\rm CE}(\epsilon)$. These results show that VR-FedEM is an improvement of VR-DIANA [11], as (i) it allows step sizes independent of the number of observations per server m and (ii) the complexity grows at the rate $\sqrt{m}$ rather than $m^{2/3}$. In this respect VR-FedEM has, compared to VR-DIANA, the same virtues as SPIDER [6] compared to SVRG [13] and SAGA [3], or as SPIDER-EM [8] compared to incremental EM methods such as [14].

Notations For two vectors $a, b \in \mathbb{R}^q$, $\langle a, b\rangle$ is the scalar product, and $\|\cdot\|$ the associated norm. For a positive integer n, set $[n]^\star \stackrel{\rm def}{=} \{1, \cdots, n\}$ and $[n] \stackrel{\rm def}{=} \{0, \cdots, n\}$.

2 FedEM: Expectation Maximization algorithms for federated learning

Recall the definition of the negated penalized log-likelihood $F(\theta)$ from (1). Along the entire paper, we make the following assumptions (we omit them in most of our statements for the sake of brevity).

A1. *The parameter set $\Theta \subseteq \mathbb{R}^d$ is a convex open set. The functions $\mathsf{R} : \Theta \to \mathbb{R}$, $\phi : \Theta \to \mathbb{R}^q$, $\psi : \Theta \to \mathbb{R}$, and $\rho(y_{ij}, \cdot) : \mathsf{Z} \to \mathbb{R}_+$, $s(y_{ij}, \cdot) : \mathsf{Z} \to \mathbb{R}^q$ for $i \in [n]^\star$ and $j \in [m]^\star$ are measurable functions. For any $\theta \in \Theta$ and $i \in [n]^\star$, the log-likelihood is bounded as $-\infty < \mathcal{L}_i(\theta) < \infty$.*

A2. *For all $\theta \in \Theta$ and $i, j \in [n]^* \times [m]^*$ the conditional expectation $\bar{\mathsf{s}}_{ij}(\theta)$ is well-defined.*

A3. *For any $s \in \mathbb{R}^q$, the map $s \mapsto \mathrm{Argmin}_{\theta\in\Theta}\ \{\psi(\theta) + \mathsf{R}(\theta) - \langle s, \phi(\theta)\rangle\}$ exists and is unique; the singleton is denoted by $\{\mathsf{T}(s)\}$.*

The EM algorithm defines a sequence $\{\theta_k, k \geq 0\}$ that can be computed recursively as $\theta_{k+1} = \mathsf{T} \circ \bar{\mathsf{s}}(\theta_k)$, where the map T is defined in A3 and $\bar{\mathsf{s}}$ is defined in (3). On the other hand, the EM algorithm can be defined through a mapping in the complete data sufficient statistics, referred to as the *expectation space*. In this setting, the EM iteration defines a sequence in $\mathbb{R}^q$ $\{\widehat{S}_k, k \geq 0\}$ given by $\widehat{S}_{k+1} = \bar{\mathsf{s}} \circ \mathsf{T}(\widehat{S}_k)$. Thus, we observe that the EM algorithm admits two equivalent representations:

$$\text{(Parameter space)}\ \ \theta_{k+1} = \mathsf{T} \circ \bar{\mathsf{s}}(\theta_k); \quad \text{(Expectation space)}\ \ \widehat{S}_{k+1} = \bar{\mathsf{s}} \circ \mathsf{T}(\widehat{S}_k). \tag{5}$$

In this paper, we focus on the expectation space representation; see [16] for an interesting discussion on the connection of EM and mirror descent. It has been shown in [4] that if $s_\star$ is a fixed point to the EM algorithm in the expectation space, then $\theta_\star = \mathsf{T}(s_\star)$ is a fixed point of the EM algorithm in the parameter space, i.e., $\theta_\star = \mathsf{T} \circ \bar{\mathsf{s}}(\theta_\star)$; note that the converse is also true. Define the functions $\mathsf{h}_i, \mathsf{h} : \mathbb{R}^q \to \mathbb{R}^q$:

$$\mathsf{h}(s) \stackrel{\rm def}{=} \frac{1}{n}\sum_{i=1}^n \mathsf{h}_i(s)\ , \qquad \mathsf{h}_i(s) \stackrel{\rm def}{=} \bar{\mathsf{s}}_i \circ \mathsf{T}(s) - s\ . \tag{6}$$

The limit points of the EM algorithm in the expectation space are the roots of the *mean field* $\mathsf{h}(s)$.

A4. *For any $i \in [n]^\star$, there exists $L_i > 0$ such that for any $s, s' \in \mathbb{R}^q$,*

$$\|\mathsf{h}_i(s) - \mathsf{h}_i(s')\| = \|\bar{\mathsf{s}}_i \circ \mathsf{T}(s) - s - \bar{\mathsf{s}}_i \circ \mathsf{T}(s') + s'\| \leq L_i\|s - s'\|\ .$$

A5. *The function $\mathrm{W} \stackrel{\rm def}{=} F \circ \mathsf{T} : \mathbb{R}^q \to \mathbb{R}$ is continuously differentiable on $\mathbb{R}^q$and its gradient is globally Lipschitz with constant $L_{\dot{\mathrm{W}}}$. Furthermore, for any $s \in \mathbb{R}^q$,*

$$\nabla \mathrm{W}(s) = -B(s)\mathsf{h}(s)$$

where $B(s)$ is a $q \times q$ positive definite matrix. In addition, there exist $0 < v_{\min} \leq v_{\max}$ such that for any $s \in \mathbb{R}^q$, the spectrum of $B(s)$ is in $[v_{\min}, v_{\max}]$.

At each iteration of the FedEM algorithm (see algorithm 1), the central server releases an aggregated estimate $\widehat{S}_k$ of the complete data sufficient statistics and the associated current fit of the parameter $\mathsf{T}(\widehat{S}_k)$. The workers each sample a mini-batch of data independently and compute $\{\mathsf{S}_{k+1,i}\}_{i=1}^n$ which are, conditionally to the past (see Section 7.2 for a rigorous definition), independent; they are moreover assumed unbiased approximations of $\bar{\mathsf{s}}_i \circ \mathsf{T}(\widehat{S}_k)$. In addition, it is assumed (see A6) that the conditional variance is (uniformly) bounded. For each $k \in \mathbb{N}$, denote by $\mathcal{F}_k$ the σ-algebra generated by $\{\mathsf{S}_{\ell,i}, i \in [n]^\star, \ell \leq k\}$. The workers then transmit to the central server a compressed version of the new sufficient statistics. The algorithm encompasses partial participation of the workers: at iteration $k+1$, only a subset $\mathcal{A}_{k+1}$ of active workers participate to the training. The averaged fraction of participating workers is denoted p. In the sequel, we consider a large class of unbiased compression operators Quant satisfying a variance bound (see A7).

A6. *For all $k \in \mathbb{N}$, conditional to $\mathcal{F}_k$, $\{\mathsf{S}_{k+1,i}\}_{i=1}^n$ are independent. Moreover, for any $i \in [n]^\star$, $\mathbb{E}\left[\mathsf{S}_{k+1,i}|\mathcal{F}_k\right] = \bar{\mathsf{s}}_i \circ \mathsf{T}(\widehat{S}_k)$ and there exists $\sigma_i^2 > 0$ such that for any $k \geq 0$, and $\mathbb{E}[\|\mathsf{S}_{k+1,i} - \bar{\mathsf{s}}_i \circ \mathsf{T}(\widehat{S}_k)\|^2] \leq \sigma_i^2$.*

We denote $\sigma^2 \stackrel{\text{def}}{=} n^{-1}\sum_{i=1}^n \sigma_i^2$. The workers then transmit to the central server a compressed version of the new sufficient statistics. In the sequel, we consider a large class of unbiased compression operators Quant satisfying a variance bound.

A7. *There exists $\omega \geq 0$ such that for any $s \in \mathbb{R}^q$: $\mathbb{E}\left[\mathrm{Quant}(s)\right] = s$, and $\mathbb{E}\left[\|\mathrm{Quant}(s)\|^2\right] \leq (1+\omega)\|s\|^2$.*

An example of quantization operator satisfying A7 is random dithering that can be described as the random operator $\mathrm{Quant} : \mathbb{R}^d \to \mathbb{R}^d$, $\mathrm{Quant}(x) = \mathrm{sign}(x) \cdot \|x\|_p \cdot (1/s_{\mathrm{quant}}) \cdot \lfloor s_{\mathrm{quant}}|x|/\|x\|_p + \xi \rfloor$ where is $p \geq 1$ is user-defined, ξ is a uniform random variable on $[0,1]^d$, $s_{\mathrm{quant}} \in \mathbb{N}^*$ is the number of levels of roundings; see [11, 2]. This operator satisfies A 7 with $\omega = s_{\mathrm{quant}}^{-1} O(d^{1/p} + d^{1/2})$; see [11, Example 1].

Algorithm 1: `FedEM` with partial participation

Data: $k_{\max} \in \mathbb{N}$; for $i \in [n]^\star$, $V_{0,i} \in \mathbb{R}^q$; $\widehat{S}_0 \in \mathbb{R}^q$; a positive sequence $\{\gamma_{k+1}, k \in [k_{\max}-1]\}$; $\alpha > 0$; a coefficient $p = \mathbb{E}_{\mathcal{A}\sim\mathbb{P}_{\mathrm{PP}}}[\mathrm{card}(\mathcal{A})]/n$.

Result: The `FedEM-PP` sequence: $\{\widehat{S}_k, k \in [k_{\max}]\}$

for $k = 0, \ldots, k_{\max}-1$ **do**
  Sample $\mathcal{A}_{k+1} \sim \mathbb{P}_{\mathrm{PP}}$;
  for $i \in \mathcal{A}_{k+1}$ **do**
    (worker #i);
    Sample $\mathsf{S}_{k+1,i}$, an approximation of $\bar{\mathsf{s}}_i \circ \mathsf{T}(\widehat{S}_k)$;
    Set $\Delta_{k+1,i} = \mathsf{S}_{k+1,i} - V_{k,i} - \widehat{S}_k$;
    Set $V_{k+1,i} = V_{k,i} + \alpha p^{-1}\,\mathrm{Quant}(\Delta_{k+1,i})$. ;
    Send $\mathrm{Quant}(\Delta_{k+1,i})$ to the controller ;
  for $i \notin \mathcal{A}_{k+1}$ **do**
    (worker #i);
    Set $V_{k+1,i} = V_{k,i}$ (no update) ;
  (the controller) ;
  Set $H_{k+1} = V_k + (np)^{-1}\sum_{i\in\mathcal{A}_{k+1}} \mathrm{Quant}(\Delta_{k+1,i})$;
  Set $\widehat{S}_{k+1} = \widehat{S}_k + \gamma_{k+1} H_{k+1}$;
  Set $V_{k+1} = V_k + \alpha(np)^{-1}\sum_{i\in\mathcal{A}_{k+1}} \mathrm{Quant}(\Delta_{k+1,i})$;
  Send $\widehat{S}_{k+1}$ and $\mathsf{T}(\widehat{S}_{k+1})$ to the n workers

We do not compress $\mathsf{S}_{k+1,i} - \hat{S}_k$ and transmit it to the central server as data heterogeneity between servers may prevent these local differences to vanish at the optimum, leading to large compression errors. Following [20], we instead introduce at each worker a memory $V_{k,i}$ (initialized to $\mathsf{h}_i(\widehat{S}_0)$ at $k=0$ for simplicity), and quantize the differences $\Delta_{k+1,i} = \mathsf{S}_{k+1,i} - \hat{S}_k - V_{k,i}$, $i \in [n]^\star$. The local workers update $V_{k+1,i} = V_{k,i} + \alpha\,\mathrm{Quant}(\Delta_{k+1,i})$ with $\alpha > 0$ (typically set to $1/(1+\omega)$ where ω in defined in A7). The central server updates $V_{k+1} = V_k + \alpha n^{-1}\sum_{i=1}^n \mathrm{Quant}(\Delta_{k+1,i})$. Then, it computes an estimate $\widehat{S}_{k+1}$ of the complete data sufficient statistics by averaging the quantized difference $n^{-1}\sum_{i=1}^n \mathrm{Quant}(\Delta_{k+1,i})$ and by adding V_k. The final step consists in solving the M step of the EM algorithm, i.e. to compute $\theta_{k+1} = \mathsf{T}(\widehat{S}_{k+1})$ (see A3).

Convergence analysis, full participation regime. In this paragraph, we focus on the full-participation regime, i.e., assume that for all $k \in \mathbb{N}$, $\mathcal{A}_k = [n]^*$, and consequently $p = 1$. We now present in Theorem 1 our key result, from which complexity expressions are derived. The proof is postponed to Section 7.

Theorem 1. *Under Assumptions A 1 to 7 and set $L^2 \stackrel{\text{def}}{=} n^{-1}\sum_{i=1}^n L_i^2$. Let $\{\widehat{S}_k, k \in [k_{\max}]\}$ be given by algorithm 1, with $\alpha = (1+\omega)^{-1}$ and $\gamma_k = \gamma \in (0, \gamma_{\max}]$, with $\gamma_{\max} = \min\left(\frac{L_{\dot{\mathrm{W}}}}{2v_{\min}}, \frac{L}{2\sqrt{2}(1+\omega)\sqrt{\omega/n}}\right)$. Denote by K the uniform random variable on $[k_{\max}-1]$. Then, taking $V_{0,i} = \mathsf{h}_i(\widehat{S}_0)$, for $i \in [n]^\star$, we get*

$$v_{\min}\left(1 - \gamma\frac{L_{\dot{\mathrm{W}}}}{v_{\min}}\right)\mathbb{E}\left[\|\mathsf{h}(\widehat{S}_K)\|^2\right] \leq \gamma^{-1}k_{\max}^{-1}\left(\mathrm{W}(\widehat{S}_0) - \min \mathrm{W}\right) + \gamma L_{\dot{\mathrm{W}}}\frac{5+\omega}{n}\left(\frac{1}{n}\sum_{i=1}^n \sigma_i^2\right) .$$

Optimizing the learning rate γ in Theorem 1, we derive the following corollary.

Corollary 2. *Under the conditions of Theorem 1, for a large enough $k_{\max}$, for $\gamma = \left(\frac{(\mathrm{W}(\widehat{S}_0)-\min \mathrm{W})n}{k_{\max} L_{\dot{\mathrm{W}}}(5+\omega)\sigma^2}\right)^{1/2}$, we get*

$$\mathbb{E}\left[\|\mathsf{h}(\widehat{S}_K)\|^2\right] \leq \frac{2}{v_{\min}} \sqrt{\frac{\left(\mathrm{W}(\widehat{S}_0) - \min \mathrm{W}\right) L_{\dot{\mathrm{W}}}(5+\omega)\sigma^2}{n k_{\max}}} \vee \frac{\left(\mathrm{W}(\widehat{S}_0) - \min \mathrm{W}\right)}{\gamma_{\max} k_{\max}} .$$

Note that, Theorem 1 and Corollary 2 do not require any assumption regarding the distributional heterogeneity of workers. They remain thus valid when workers have access to data resulting from different distributions — a widespread situation in FL frameworks. Crucially, a naive implementation of compressed distributed EM without assumptions on the heterogeneity would not converge.

We express the complexity of `FedEM` in terms of the number $\mathcal{K}_{\mathrm{Opt}}(\epsilon)$ of M-steps updates. in order to find an ϵ-approximate stationary point of $F \circ \mathsf{T}(s))$, i.e. a model such that $\mathbb{E}\left[\|\mathsf{h}(\widehat{S}_K)\|^2\right] \leq \epsilon$. From Corollary 2, we have that:

$$\mathcal{K}_{\mathrm{opt}}(\epsilon) = \Omega\left(\frac{(1+\omega)\sigma^2}{n\epsilon^2} + \frac{1}{\gamma_{\max}\epsilon}\right)$$

Maximal learning rate. Without compression, the maximal learning rate γ to ensure convergence is $\frac{L_{\dot{\mathrm{W}}}}{2v_{\min}}$. Thus, compression also impacts $\gamma_{\max}$ by a factor proportional to $\sqrt{n}/\omega^{3/2}$ as ω increases (similar constraints were observed in the risk optimization literature, e.g. in [11, 23]). This highlights two different regimes depending on the ratio $\sqrt{n}/\omega^{3/2}$. If the number of workers n scales at least as ω^3, the maximal learning rate is not impacted by compression. On the other hand, for smaller numbers of workers $n \ll \omega^3$, compression can degrade the maximal learning rate.

These observations show the benefit of using compressed operators in particular in low noise regimes, and with a large number of workers. We highlight this conclusion with a small example. Recall that in the case of scalar quantization, $\omega \sim s_{\mathrm{quant}}^{-1} d^{1/2}$, for $d = 100$ and $s = 4$ (obtaining a compression rate of a factor 16), the maximal learning rate is almost unchanged if $n \geq 10$.

Dependency on ϵ. The complexity $\mathcal{K}_{\mathrm{opt}}$ is decomposed into two terms scaling respectively as $\sigma^2\epsilon^{-2}$ (asymptotically dominant), and as $\gamma_{\max}\epsilon^{-1}$. This first observation highlights two different regimes, i.e. a *high noise regime* corresponding to $\sigma^2\epsilon^{-1} \geq 1$ where the complexity is of order $\sigma^2\epsilon^{-2}$, and a *low noise regime* where $\sigma^2\epsilon^{-1} \leq 1$ and the complexity is of order $\gamma_{\max}\epsilon^{-1}$; a most critical example of the low noise regime being the case where full batches are used at each iteration, resulting in $\sigma^2 = 0$. The benefit of compression is thus most significant in the *low noise* regime, when the maximal learning rate is not reduced. We then improve the communication cost of each iteration without increasing the optimization complexity. On the other hand, in the *high noise* regime while we preserve a complexity $O(\epsilon^{-2})$, it increases by a factor $(1+\omega)$. We finally observe that, in the case $\omega = 0$ (i.e., without compression) we recover the classical rate for online EM [14].

`FedEM` **with partial participation and compression** In this paragraph, we extend the previous result to the *partial participation* regime, in which only a fraction of the workers participate to the training at each step of the learning process. This is a key feature in the FL framework, as individuals may not always be available or willing to participate [19]. To analyze the convergence in this situation, we make the following assumption.

A8. *For all $k \in \mathbb{N}$, $\mathcal{A}_{k+1} = \{i \in [n] \text{ s.t.} B_{k+1,i} = 1\}$ where $(B_{k+1,i})_{i\in[n]^*,k\geq 0} \overset{i.i.d.}{\sim} \mathcal{B}(p)$, are i.i.d. Bernoulli random variables of parameter p.*

This assumption is standard in the FL literature [24, 26, 22], and can easily be extended to worker dependent probabilities of participation [12]. Algorithm 1 is naturally defined in the partial participation regime. A noteworthy point is that, upon receiving $(\mathrm{Quant}(\Delta_{k+1;i}))_{i\in\mathcal{A}_{k+1}}$, the central node performs the update $H_{k+1} = V_k + (np)^{-1}\sum_{i\in\mathcal{A}_{k+1}} \mathrm{Quant}(\Delta_{k+1;i})$ and *not* $H_{k+1} = (np)^{-1}\sum_{i\in\mathcal{A}_{k+1}}(\mathrm{Quant}(\Delta_{k+1;i}) + V_{k,i})$. Though the later solution may appear more natural, it would actually not only require to store all values $(V_{k,i})_{i\in[n]^*}$ on the central server, but also impair convergence in the heterogeneous setting. In other words, even in the *uncompressed*

regime, in which $\mathrm{Quant}(\Delta_{k;i}) = \Delta_{k;i}$, our algorithm differs from a naive implementation of distributed EM, that would not be robust to data heterogeneity. The following proposition extends Theorem 1 to the partial participation regime.

Proposition 3. *Assumptions A 1 to 8 and set* $L^2 \stackrel{\mathrm{def}}{=} n^{-1}\sum_{i=1}^n L_i^2$. *Let* $\{\widehat{S}_k, k \in [k_{\max}]\}$ *be given by algorithm 1, run with* $\alpha = (1+\omega)^{-1}$ *and* $\gamma_k = \gamma \in (0, \gamma_{\max}]$, *with* $\gamma_{\max} = \min\left(\frac{L_{\dot{\mathrm{W}}}}{2v_{\min}}, \frac{pL}{2\sqrt{2}(1+\omega)\sqrt{\omega/(pn)}}\right)$ *Denote by* K *the uniform random variable on* $[k_{\max}-1]$. *Then, taking* $V_{0,i} = \mathsf{h}_i(\widehat{S}_0)$, *for* $i \in [n]^\star$, *we get* $v_{\min}\mathbb{E}\left[\|\mathsf{h}(\widehat{S}_K)\|^2\right] \le 2\gamma^{-1}k_{\max}^{-1}\left(\mathrm{W}(\widehat{S}_0) - \min \mathrm{W}\right) + 2\gamma L_{\dot{\mathrm{W}}}\frac{5+\omega}{pn}\left(\frac{1}{n}\sum_{i=1}^n \sigma_i^2\right)$.

3 VR-FedEM: Federated EM algorithm with variance reduction

A novel algorithm, called VR-FedEM and described by algorithm 2, is derived to additionally incorporate a variance reduction scheme in FedEM.

The algorithm is decomposed into k_{out} outer loops (indexed by t), each of them having k_{in} inner loops (indexed by k). At iteration $\#(k+1)$ of the inner loop, each worker $\#i$ updates a local statistic $\widehat{S}_{t,k+1,i}$ based on a mini-batch $\mathcal{B}_{t,k+1,i}$ of its own examples $\{\bar{\mathsf{s}}_{ij}, j \in \mathcal{B}_{t,k+1,i}\}$ (see Line 8): starting from $\widehat{S}_{t,0,i} \stackrel{\mathrm{def}}{=} m^{-1}\sum_{j=1}^m \bar{\mathsf{s}}_{ij} \circ \mathsf{T}(\widehat{S}_{t,-1})$, $\widehat{S}_{t,k+1,i}$ is defined in such a way that it approximates $m^{-1}\sum_{j=1}^m \bar{\mathsf{s}}_{ij} \circ \mathsf{T}(\widehat{S}_{t,k})$ (see 15). Then, worker $\#i$ sends to the controller a quantization of $\Delta_{t,k+1,i}$ (see Line 12) which can be seen as an approximation of $\alpha^{-1}\{\mathsf{h}_i(\widehat{S}_{t,k}) - \mathsf{h}_i(\widehat{S}_{t,k-1})\}$ upon noting that the variable $V_{t,k+1,i}$ defined by Line 10 approximates $\mathsf{h}_i(\widehat{S}_{t,k})$ (see Proposition 18). The controller learns the mean value $V_{t,k+1} = n^{-1}\sum_{i=1}^n V_{t,k+1,i}$ (see Line 15 and Lemma 17) and, by adding the quantized quantities, defines a field $H_{t,k+1}$ which approximates $n^{-1}\sum_{i=1}^n \mathsf{h}_i(\widehat{S}_{t,k})$ (see Proposition 20). Line 14 can be seen as a Stochastic Approximation update, with learning rate $\gamma_{t,k+1}$ and mean field $n^{-1}\sum_{i=1}^n \mathsf{h}_i(\widehat{S}_{t,k})$.

Algorithm 2: VR-FedEM

Data: $k_{\mathrm{out}}, k_{\mathrm{in}}, \mathsf{b} \in \mathbb{N}_\star$; for $i \in [n]^\star$, $V_{1,0,i} \in \mathbb{R}^q$; $\widehat{S}_{\mathrm{init}} \in \mathbb{R}^q$; a positive sequence $\{\gamma_{t,k+1}, t \in [k_{\mathrm{out}}]^\star, k \in [k_{\mathrm{in}}-1]\}$; $\alpha > 0$

Result: sequence: $\{\widehat{S}_{t,k}, t \in [k_{\mathrm{out}}]^\star, k \in [k_{\mathrm{in}}]\}$

1 $\widehat{S}_{1,0} = \widehat{S}_{1,-1} = \widehat{S}_{\mathrm{init}}$, $V_{1,0} = n^{-1}\sum_{i=1}^n V_{1,0,i}$;
2 **for** $i = 1, \dots, n$ **do**
3 $\quad \mathsf{S}_{1,0,i} = \frac{1}{m}\sum_{j=1}^m \bar{\mathsf{s}}_{ij} \circ \mathsf{T}(\widehat{S}_{\mathrm{init}})$
4 **for** $t = 1, \dots, k_{\mathrm{out}}$ **do**
5 $\quad$ **for** $k = 0, \dots, k_{\mathrm{in}} - 1$ **do**
6 $\quad\quad$ **for** $i = 1, \dots, n$ (worker $\#i$, locally) **do**
7 $\quad\quad\quad$ Sample at random a batch $\mathcal{B}_{t,k+1,i}$ of size b in $[m]^\star$;
8 $\quad\quad\quad$ Set $\mathsf{S}_{t,k+1,i} = \mathsf{S}_{t,k,i} + \mathsf{b}^{-1}\sum_{j\in\mathcal{B}_{t,k+1,i}}\left(\bar{\mathsf{s}}_{ij} \circ \mathsf{T}(\widehat{S}_{t,k}) - \bar{\mathsf{s}}_{ij} \circ \mathsf{T}(\widehat{S}_{t,k-1})\right)$;
9 $\quad\quad\quad$ Set $\Delta_{t,k+1,i} = \mathsf{S}_{t,k+1,i} - \widehat{S}_{t,k} - V_{t,k,i}$;
10 $\quad\quad\quad$ Set $V_{t,k+1,i} = V_{t,k,i} + \alpha\,\mathrm{Quant}(\Delta_{t,k+1,i})$. ;
11 $\quad\quad\quad$ Send $\mathrm{Quant}(\Delta_{t,k+1,i})$ to the controller ;
12 $\quad\quad$ *(the controller)* ;
13 $\quad\quad$ Set $H_{t,k+1} = V_{t,k} + n^{-1}\sum_{i=1}^n \mathrm{Quant}(\Delta_{t,k+1,i})$;
14 $\quad\quad$ Set $\widehat{S}_{t,k+1} = \widehat{S}_{t,k} + \gamma_{t,k+1}H_{t,k+1}$;
15 $\quad\quad$ Set $V_{t,k+1} = V_{t,k} + \alpha n^{-1}\sum_{i=1}^n \mathrm{Quant}(\Delta_{t,k+1,i})$;
16 $\quad\quad$ Send $\widehat{S}_{t,k+1}$ and $\mathsf{T}(\widehat{S}_{t,k+1})$ to the n workers ;
17 $\quad$ $\widehat{S}_{t+1,0} = \widehat{S}_{t+1,-1} = \widehat{S}_{t,k_{\mathrm{in}}}$;
18 $\quad$ $V_{t+1,0} = V_{t,k_{\mathrm{in}}}$;
19 $\quad$ **for** $i = 1, \dots, n$ **do**
20 $\quad\quad$ $\mathsf{S}_{t+1,0,i} = \frac{1}{m}\sum_{j=1}^m \bar{\mathsf{s}}_{ij} \circ \mathsf{T}(\widehat{S}_{t+1,0})$;
21 $\quad\quad$ $V_{t+1,0,i} = V_{t,k_{\mathrm{in}},i}$

At the end of each outer loop, the local approximations $\mathsf{S}_{t+1,0,i}$ are initialized to the full sum $m^{-1}\sum_{j=1}^m \bar{\mathsf{s}}_{ij} \circ \mathsf{T}(\widehat{S}_{t,k_{\mathrm{in}}})$ (see Line 20) thus canceling the bias of $\mathsf{S}_{\cdot,i}$ (see Proposition 14).

The convergence in expectation of VR-FedEM to the roots of the function h defined in (6) is derived under the following assumption.

A9. *For any* $i \in [n]^\star$ *and* $j \in [m]^\star$, *there exists* L_{ij} *such that for any* $s, s' \in \mathbb{R}^q$,

$$\|\bar{\mathsf{s}}_{ij} \circ \mathsf{T}(s) - s - \bar{\mathsf{s}}_{ij} \circ \mathsf{T}(s') + s'\| \le L_{ij}\|s - s'\| .$$

Theorem 4. *Assume A 1 to 5, 7 and 9. Set* $L^2 \stackrel{\text{def}}{=} n^{-1}m^{-1}\sum_{i=1}^n\sum_{j=1}^m L_{ij}^2$. *Let* $\{\widehat{S}_{t,k}, t \in [k_{\text{out}}]^\star, k \in [k_{\text{in}}-1]\}$ *be given by algorithm 2 run with* $\alpha = 1/(2(1+\omega))$, $V_{1,0,i} = \mathsf{h}_i(\widehat{S}_{1,0})$ *for any* $i \in [n]^\star$, *and* $b = \frac{k_{\text{in}}}{(1+\omega)^2}$:

$$\gamma_{t,k} = \gamma \stackrel{\text{def}}{=} \frac{v_{\min}}{L_{\dot{\mathrm{W}}}}\left(1 + \frac{2L}{L_{\dot{\mathrm{W}}}}\frac{v_{\max}}{\sqrt{n}}\left(32(1+\omega)^2\omega + 1\right)^{1/2}\right)^{-1} .$$

Let (τ, K) *be a uniform random variable on* $[k_{\text{out}}]^\star \times [k_{\text{in}}-1]$, *independent of* $\{\widehat{S}_{t,k}, t \in [k_{\text{out}}]^\star, k \in [k_{\text{in}}-1]\}$. *Then, it holds*

$$\frac{v_{\min}}{2}\gamma\mathbb{E}\left[\|H_{\tau,K+1}\|^2\right] \le \frac{1}{k_{\text{in}}k_{\text{out}}}\left(\mathbb{E}\left[\mathrm{W}(\widehat{S}_{1,0})\right] - \min \mathrm{W}\right) , \tag{7}$$

$$\mathbb{E}\left[\|\mathsf{h}(\widehat{S}_{\tau,K})\|^2\right] \le 2\left(1 + 4\gamma^2\frac{L^2}{(1+\omega)^2 n}\right)\mathbb{E}\left[\|H_{\tau,K+1}\|^2\right] . \tag{8}$$

The proof is postponed to (9). (7) provides the convergence of $\mathbb{E}\left[\|H_{\tau,K+1}\|^2\right]$, and Equation (8) ensures that the quantity of interest $\mathbb{E}\left[\|\mathsf{h}(\widehat{S}_{\tau,K})\|^2\right]$ is controlled by $\mathbb{E}\left[\|H_{\tau,K+1}\|^2\right]$. We observe that $\left(1 + 4\gamma^2\frac{L^2}{(1+\omega)^2 n}\right) \le 2$. As in Theorem 1, the maximal learning rate is reduced by a factor $\omega^{3/2}/\sqrt{n}$, w.r.t. the uncompressed case. which does not impact the maximal learning rate for n large enough.

Contrary to Theorem 4 in [11] (that provides convergence for a variance reduced algorithm in the risk optimization framework), our $\gamma_{\max}$ does not depend on $k_{\text{in}}, k_{\text{out}}$ and does not decrease with m. Moreover, to reach an ϵ-stationary point, (i) the number of optimization steps required is $\mathcal{K}_{\text{opt}}(\epsilon) = k_{\text{out}}k_{\text{in}} = \Omega(\frac{1+\omega^{3/2}/\sqrt{n}}{\epsilon})$; (ii) The number of calls to one of the function $\bar{\mathsf{s}}_{ij}$ is $\mathcal{K}_{\text{CE}} = nm + nmk_{\text{out}} + n2\mathsf{b}k_{\text{in}}k_{\text{out}} = \Omega\left(nm + \frac{n}{\epsilon}\left(1 + \frac{\omega^{3/2}}{\sqrt{n}}\right)\left(\frac{m}{k_{\text{in}}} + \frac{k_{\text{in}}}{(1+\omega)^2}\right)\right)$. With $k_{\text{in}} = \sqrt{m}(1+\omega)$, we obtain an overall call-complexity of $\Omega\left(nm + \frac{n}{\epsilon}\left(1 + \frac{\omega^{3/2}}{\sqrt{n}}\right)\frac{\sqrt{m}}{1+\omega}\right)$. We stress the dependency as $\sqrt{m}$: for $n = 1$, we get a complexity scaling as $\sqrt{m}\epsilon^{-1}$ that corresponds to the optimal $\mathcal{K}_{\text{CE}}$ of `SPIDER-EM`.

Up to our knowledge, this is the first result on Federated EM, that leverages advanced variance reduction techniques, while being robust to distribution heterogeneity and reducing the communication cost between workers.

4 Numerical illustration

In this section, we illustrate the performance of `FedEM` and `VR-FedEM` on Gaussian Mixture Model (GMM) inference, with simulated data and the analysis of the MNIST data set. We also present an application to Federated missing data imputation, in the context of citizen science data analysis for biodiversity monitoring with the analysis of a subsample of the eBird data set [25, 1].

Synthetic data We illustrate the performance of `FedEM` and `VR-FedEM` on the inference of a GMM. We generate a two-dimensional, two-components GMM where, for all $i \in [N]^\star$, and $k \in \{0, 1\}$, $\mathbb{P}(Z_i = k) = \pi_k$ and $Y_i|Z_i = k \sim \mathcal{N}(\mu_k, \Sigma)$. The covariance matrix Σ is assumed known, and we fit the weights (π_0, π_1) and the means (μ_0, μ_1) to the observed data. We set the number of samples to $N = 10^4$ and the number of agents to $n = 100$. We fit a two-dimensional GMM, using the same values of the parameters for both algorithms: $\gamma = 10^{-2}$, and $\alpha = 10^{-2}$. For `FedEM`, we set the size of the minibatch to 20; for `VR-FedEM`, we set the size of the minibatch to 5, and the number of iterations in the inner loop to $k_{in} = 20$. We run the two algorithms for 500 epochs (500 passes on the whole data set), and record the squared norm of the mean field $\|H^k\|^2$.

In Figure 1, we display the 0.25 and 0.75 quantiles of the empirical distribution of $\|H^k\|^2$ for `FedEM` and `VR-FedEM`. The results illustrate the effect of variance reduction, and gives insight on the variability of the trajectories resulting from the two algorithms.

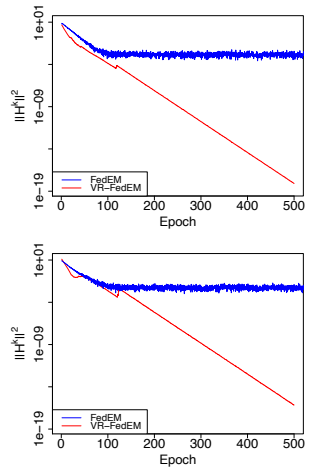


Figure 1: Quantile 0.25 (top) and 0.75 (bottom) of $\|H^k\|^2$ vs the number of epochs; estimated from 10 independent samples.

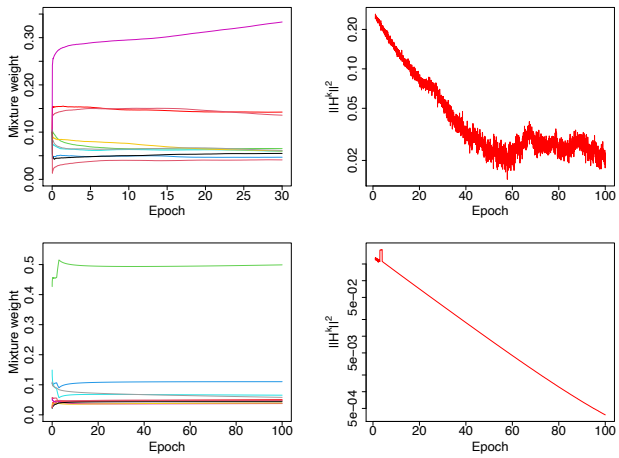


Figure 2: [Left] Evolution of the estimates of the weights π_ℓ for $\ell = 1, \dots, 10$ by `FedEM` (top) and `VR-FedEM` (bottom) vs the number of epochs. [Right] Evolution of the squared norm of the mean field $\|H^k\|^2$ vs the number of epochs.

MNIST Data set We perform a similar experiment on the MNIST dataset to illustrate the behaviour of `FedEM` and `VR-FedEM` on a GMM inference problem with real data. The dataset consists of $n = 7 \times 10^4$ images of handwritten digits, each with 784 pixels. We pre-process the dataset by removing 67 uninformative pixels (which are always zero across all images) to obtain $d = 717$ pixels per image. Second, we apply principal component analysis (PCA) to reduce the data dimension. We keep the $d_{\mathrm{PC}} = 20$ principal components (PCs) of each observation. We estimate a multivariate GMM model with $g = 10$ components. Details on the multivariate Gaussian mixture model are given in the supplementary material (see section 10). The mini-batches $\{\mathcal{B}_\tau\}_\tau$ are independent, and sampled at random in $\{1, \dots, n\}$ with replacement. The minibatch size is set to be $\mathtt{b} = 20$ and the stepsize $\gamma = 10^{-3}$. The same initial value $\widehat{S}_{\mathrm{init}}$ is used for all experiments. We initialize the parameters as follows: $\pi_l^0 = 0.1$ for all $l \in \{1, \dots, L\}$, and for all $l \in \{1, \dots, L\}$ μ_l^0 is sampled uniformly at random among the available samples. We fix the covariance matrix of the GMM to be equal to the empirical covariance matrix $\hat{\Sigma}$. Then, we set $\widehat{S}_{\mathrm{init}} = \bar{s}(\pi^0, \mu^0, \hat{\Sigma})$.

Federated missing values imputation for citizen science We develop a special instance of `FedEM` to missing values imputation in the federated setting called `FedMissEM`, and apply it to the analysis of a sample of the eBird data base [25, 1], a citizen science smartphone application for biodiversity monitoring. In eBird, citizens record wildlife observations, specifying the ecological site they visited, the date, the species and the number of observed specimens. Two major challenges occur: (i) ecological sites are visited irregularly, which leads to missing values and (ii) non-professional observers have heterogeneous wildlife counting schemes. Thus, the development of missing values prediction methods which are applicable in such federated settings is an important problem.

`FedMissEM` **model and algorithm** Consider an unknown matrix $\theta^\star \in \mathbb{R}^{I \times J}$; for instance, $i \in \{1, \dots, I\}$ indexes ecological sites, $j \in \{1, \dots, J\}$ indexes time stamps, and the entry $\theta^\star_{i,j}$ is a scalar parameter characterizing the distribution of species individuals present at site i and time stamp j. For instance, $\theta^\star_{i,j}$ is the log-intensity of a Poisson distribution if the observations are count data or the log-odd of a binomial model if the observations are presence-absence data. We assume that n observers participate in the programme: For $c \in [n]^\star$, the c-th observer records a set of observations $Y^c = \{Y^c_{i,j}, (i,j) \in \Omega^c\}$, where $\Omega^c \subset [I]^\star \times [J]^\star$ is the set of field observations, and $Y^c_{i,j}$ is the noisy observation at site i and time stamp j. We set $X^c = (Y^c, Z^c)$, where $Z^c = \{Z^c_{i,j}, (i,j) \notin \Omega^c\}$ is the set of unobserved variables for observer c. We further assume that $(X^c_{i,j})_{1 \le c \le n, 1 \le i \le I, 1 \le j \le J}$ are independent with distribution from an exponential family with respect to some reference measure $\nu_{i,j}$. For all $(i,j) \in [I]^\star \times [J]^\star$ and for all $c \in [n]^\star$:

$$p^c_{i,j}(y, z; \theta^c_{ij}) = \rho^c_{i,j}(y, z) \exp\{s^c_{i,j}(y, z)\theta^c_{i,j} - \psi^c_{i,j}(\theta^c_{i,j})\},$$

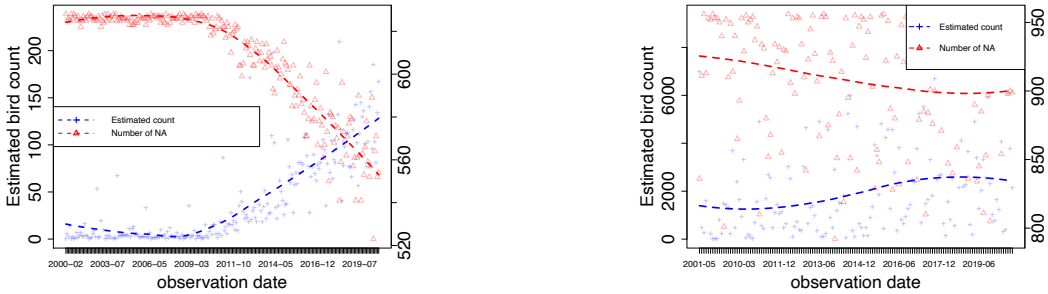
where $s^c_{i,j}(y,z) = \mathbb{1}_{(i,j)\in\Omega^c} y + (1-\mathbb{1}_{(i,j)\in\Omega^c})z$, and θ^c_{ij} is the scalar natural parameter. Heterogeneity between agents (observers) is translated by the fact that, for fixed $(i,j) \in [I]^\star \times [J]^\star$, we may have $\theta^c_{i,j} \neq \theta^{c'}_{i,j}$; i.e. observers c and c' may on average count different number of specimens at the same location and time stamp, for instance because they do not have access to the same material, or do not have the same level of expertise. The function $\psi_{i,j}$ is for instance defined by $\psi_{i,j}(\theta) = -\theta^2/2$ for a Gaussian model, $\psi_{i,j}(\theta) = \log(1+\mathrm{e}^\theta)$ for a Bernoulli model, and $\psi_{i,j}(\theta) = \mathrm{e}^\theta$ for a Poisson model.

We assume that θ is low-rank and use the parameterization $\theta = UV^\top$, where $U \in \mathbb{R}^{I\times r}$ and $V \in \mathbb{R}^{J\times r}$ with $\mathrm{rank}(\theta) = r$ and $r < \min(I,J)$. We minimize the negated penalized log-likelihood:

$$\min_{U\in\mathbb{R}^{I\times r}, V\in\mathbb{R}^{J\times r}} F(U,V) = \frac{1}{n}\sum_{c=1}^{n}\left(\sum_{i=1}^{I}\sum_{j=1}^{J}\mathcal{L}^c_{i,j}(UV^\top)\right) + \frac{\lambda}{2}\left(\|U\|_F^2 + \|V\|_F^2\right),$$

where for $\theta \in \mathbb{R}^{I\times J}$, $\mathcal{L}^c_{i,j}(\theta) = -\log\left\{\int_{\mathbb{R}^q} p^c_{i,j}(y,z;\theta_{i,j})\nu_{i,j}(dz)\right\}$. Algorithm 6 in Appendix 10.2 provides the pseudo-code for `FedMissEM`.

Application to eBird data analysis We apply `FedMissEM` to the analysis of a sample of the eBird data base [25, 1] of field observations reported in France by 2,465 observers, across 9,721 sites and at 525 monthly time points. We extract two subsamples corresponding to observations of two relatively common species: the Common Buzzard and the Mallard. These subsamples correspond to respectively to 5,980 and 12,185 field observations. We run the `FedMissEM` algorithm for 150 epochs, with parameters $\gamma = 1e-4$, $\alpha = 1e-3$, minibatch size $\mathtt{b} = 100$, fixed rank $r = 2$. We recover aggregated temporal trends at the national French level for these two bird species by summing the estimated counts across ecological sites, for each time stamp; the trends are displayed in Figure 3.



(a) Common Buzzard (b) Mallard

Figure 3: Estimated temporal trends for Common Buzzard (Left) and Mallard (right). Blue crosses: estimated monthly counts; Red triangles: number of missing values. Dotted lines: LOESS regressions for the estimated counts (blue) and the number of missing values (red).

5 Conclusions

We introduced the `FedEM` algorithm which is, to the best of our knowledge, the first method to implement the EM algorithm in a federated learning setting, and handles compression of exchanged information, data heterogeneity and partial participation. We further extended it to additionally incorporate a variance reduction scheme, and introduced the `VR-FedEM` algorithm. We derived complexity bounds which highlight the efficiency of the two algorithms, and illustrated our claims with numerical simulations, as well as an application to biodiversity monitoring data.

Broader Impact This work is mostly theoretical, and we believe it does not currently present any direct societal consequence. However, the methods described in this paper can be used to train machine learning models which could themselves have societal consequences. For instance, the deployment of machine learning models can suffer from gender and racial bias, or amplify existing inequalities.

References

[1] ebird. 2017. ebird: An online database of bird distribution and abundance [web application]. ebird, cornell lab of ornithology, ithaca, new york. available: http://www.ebird.org. (accessed: 21 march 2020).

[2] D. Alistarh, T. Hoefler, M. Johansson, S. Khirirat, N. Konstantinov, and C. Renggli. The convergence of sparsified gradient methods. *arXiv preprint arXiv:1809.10505*, 2018.

[3] A. Defazio, F. Bach, and S. Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1646–1654. Curran Associates, Inc., 2014.

[4] B. Delyon, M. Lavielle, and E. Moulines. Convergence of a stochastic approximation version of the EM algorithm. *Ann. Statist.*, 27(1):94–128, 1999. ISSN 0090-5364. doi: 10.1214/aos/1018031103. URL https://doi.org/10.1214/aos/1018031103.

[5] A. Dempster, N. Laird, and D. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. Roy. Stat. Soc. B Met.*, 39(1):1–38, 1977.

[6] C. Fang, C. Li, Z. Lin, and T. Zhang. SPIDER: Near-Optimal Non-Convex Optimization via Stochastic Path-Integrated Differential Estimator. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 689–699. Curran Associates, Inc., 2018.

[7] G. Fort, E. Moulines, and H.-T. Wai. A stochastic path integral differential estimator expectation maximization algorithm. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 16972–16982. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/c589c3a8f99401b24b9380e86d939842-Paper.pdf.

[8] G. Fort, E. Moulines, and H.-T. Wai. A stochastic path integral differential estimator expectation maximization algorithm. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 16972–16982. Curran Associates, Inc., 2020.

[9] S. Frühwirth-Schnatter, G. Celeux, and C. P. Robert, editors. *Handbook of mixture analysis*. Chapman & Hall/CRC Handbooks of Modern Statistical Methods. CRC Press, Boca Raton, FL, 2019.

[10] E. Gorbunov, F. Hanzely, and P. Richtárik. A unified theory of sgd: Variance reduction, sampling, quantization and coordinate descent. In *International Conference on Artificial Intelligence and Statistics*, pages 680–690. PMLR, 2020.

[11] S. Horváth, D. Kovalev, K. Mishchenko, S. Stich, and P. Richtárik. Stochastic distributed learning with gradient quantization and variance reduction. *arXiv preprint arXiv:1904.05115*, 2019.

[12] S. Horváth and P. Richtárik. A Better Alternative to Error Feedback for Communication-Efficient Distributed Learning. *arXiv:2006.11077 [cs, stat]*, June 2020. arXiv: 2006.11077.

[13] R. Johnson and T. Zhang. Accelerating Stochastic Gradient Descent using Predictive Variance Reduction. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 315–323. Curran Associates, Inc., 2013.

[14] B. Karimi, H.-T. Wai, E. Moulines, and M. Lavielle. On the Global Convergence of (Fast) Incremental Expectation Maximization Methods. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 2837–2847. Curran Associates, Inc., 2019.

[15] J. Konečnỳ, H. B. McMahan, F. X. Yu, P. Richtárik, A. T. Suresh, and D. Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016.

[16] F. Kunstner, R. Kumar, and M. Schmidt. Homeomorphic-invariance of em: Non-asymptotic convergence in kl divergence for exponential families via mirror descent. In *International Conference on Artificial Intelligence and Statistics*, pages 3295–3303. PMLR, 2021.

[17] K. Lange. *MM optimization algorithms*. SIAM, 2016.

[18] G. McLachlan and T. Krishnan. *The EM algorithm and extensions*. Wiley series in probability and statistics. Wiley, 2008.

[19] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y. Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Artificial Intelligence and Statistics*, pages 1273–1282. PMLR, Apr. 2017. ISSN: 2640-3498.

[20] K. Mishchenko, E. Gorbunov, M. Takáč, and P. Richtárik. Distributed learning with compressed gradient differences. *arXiv preprint arXiv:1901.09269*, 2019.

[21] K. P. Murphy. Dynamic bayesian networks: representation, inference and learning. 2002.

[22] C. Philippenko and A. Dieuleveut. Artemis: tight convergence guarantees for bidirectional compression in federated learning. *arXiv preprint arXiv:2006.14591*, 2020.

[23] C. Philippenko and A. Dieuleveut. Preserved central model for faster bidirectional compression in distributed settings. *arXiv preprint arXiv:2102.12528*, 2021.

[24] F. Sattler, S. Wiedemann, K.-R. Müller, and W. Samek. Robust and Communication-Efficient Federated Learning From Non-i.i.d. Data. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–14, 2019. ISSN 2162-2388. doi: 10.1109/TNNLS.2019.2944481. Conference Name: IEEE Transactions on Neural Networks and Learning Systems.

[25] B. L. Sullivan, C. L. Wood, M. J. Iliff, R. E. Bonney, D. Fink, and S. Kelling. ebird: A citizen-based bird observation network in the biological sciences. *Biological Conservation*, 142(10): 2282–2292, 2009. ISSN 0006-3207. doi: https://doi.org/10.1016/j.biocon.2009.05.006. URL https://www.sciencedirect.com/science/article/pii/S000632070900216X.

[26] H. Tang, C. Yu, X. Lian, T. Zhang, and J. Liu. DoubleSqueeze: Parallel Stochastic Gradient Descent with Double-pass Error-Compensated Compression. In *International Conference on Machine Learning*, pages 6155–6165. PMLR, May 2019. ISSN: 2640-3498.

[27] L. Xu and M. I. Jordan. On convergence properties of the em algorithm for gaussian mixtures. *Neural computation*, 8(1):129–151, 1996.

[28] Q. Yang, Y. Liu, T. Chen, and Y. Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19, 2019.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See **??** and **??** for methodological and theoretical results and Section 4 for associated experiments.
   (b) Did you describe the limitations of your work? [Yes] See broader impact and Appendix.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See broader impact and Appendix.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...

(a) Did you state the full set of assumptions of all theoretical results? [Yes] See **??** and **??**

(b) Did you include complete proofs of all theoretical results? [Yes] Also see Appendix in Supplemental Material.

3. If you ran experiments...

(a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code available in Supplementary Material. The data is available for synthetic and MNIST data set. For the eBird data set, we are not allowed to release it publically (though it is available to academic researchers on demand).

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See Section 4.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We reported 0.25 and 0.75 quantiles for our synthetic experiments (see Section 4).

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See Section 4 for further references.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...

(a) If your work uses existing assets, did you cite the creators? [Yes] We cited the appropriate references associated to the eBird dataset (see Section 4).

(b) Did you mention the license of the assets? [Yes] Only open source and/or Academic assets are used.

(c) Did you include any new assets either in the supplemental material or as a URL? [Yes] We included our code in Supplemental Material.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A] We used either synthetic or public data, or data available on demand to academic researchers.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]