The Perturbed Prox-Preconditioned SPIDER algorithm for EM-based large scale learning

Gersende Fort, Eric Moulines

THE PERTURBED PROX-PRECONDITIONED SPIDER ALGORITHM FOR EM-BASED LARGE SCALE LEARNING

G. Fort[1]*, E. Moulines*[2]

[1] IMT, Université de Toulouse & CNRS, F-31062 Toulouse, France.
[2] CMAP, Ecole Polytechnique, Route de Saclay, 91128 Palaiseau Cedex, France.

ABSTRACT

Incremental Expectation Maximization (EM) algorithms were introduced to design EM for the large scale learning framework by avoiding the full data set to be processed at each iteration. Nevertheless, these algorithms all assume that the conditional expectations of the sufficient statistics are explicit. In this paper, we propose a novel algorithm named `Perturbed Prox-Preconditioned SPIDER` (3P-SPIDER), which builds on the `Stochastic Path Integral Differential EstimatoR EM` (SPIDER-EM) algorithm. The `3P-SPIDER` algorithm addresses many intractabilities of the E-step of EM; it also deals with non-smooth regularization and convex constraint set. Numerical experiments show that `3P-SPIDER` outperforms other incremental EM methods and discuss the role of some design parameters.

Index Terms— Statistical Learning, Large Scale Learning, Expectation Maximization algorithm, Finite-sum Optimization, Accelerated Stochastic Approximation, Control Variates.

1. INTRODUCTION

EM [1, 2] is a very popular computational tool, designed to solve non convex minimization problems on $\mathbb{R}^d$ when the objective function is not explicit but defined by an integral $F(\theta) = -\log \int_{\mathsf{Z}} G(z;\theta)\mathrm{d}\mu(z)$. EM is a Majorize-Minimization algorithm which, based on the current value of the parameter θ_{curr}, defines a majorizing function $\theta \mapsto Q(\theta;\theta_{\mathrm{curr}})$ through a Kullback-Leibler argument; then, the new point is chosen as the/a minimum of $Q(\cdot;\theta_{\mathrm{curr}})$. The computation of Q is straightforward when there exist (known and explicit) functions $\mathsf{R}, \phi, \mathsf{s}$ such that $Q(\cdot;\theta_{\mathrm{curr}}) = \mathsf{R}(\cdot) - \langle \bar{\mathsf{s}}(\theta_{\mathrm{curr}}), \phi(\cdot)\rangle$ and $\bar{\mathsf{s}}(\tau) \propto \int_{\mathsf{Z}} \mathsf{s}(z)G(z;\tau)\mathrm{d}\mu(z)$ is the expectation of the function s with respect to (w.r.t.) the probability measure $G(\cdot;\tau)\exp(-F(\tau))\mathrm{d}\mu$. In these cases, the vector $\bar{\mathsf{s}}(\theta_{\mathrm{curr}})$ defines the function Q.

It may happen that the vector $\bar{\mathsf{s}}(\theta_{\mathrm{curr}})$ is not explicit (see e.g. [3, section 6]); a natural idea is to substitute $\bar{\mathsf{s}}$ for an approximation, possibly random. A first level of intractability occurs when the integral $\bar{\mathsf{s}}(\theta_{\mathrm{curr}})$ is not explicit. Many stochastic EM versions were proposed and studied to overcome this intractability: among them, let us cite Monte Carlo EM [4, 5] where $\bar{\mathsf{s}}$ is approximated by a Monte Carlo integration; and SA EM [6, 7] where $\bar{\mathsf{s}}$ is approximated by a Stochastic Approximation (SA) scheme [8]. With the Big Data era, a second level of intractability occurred: EM applied to statistical learning evolved into online versions and large scale versions in order to minimize a loss function associated to a set of observations (also called *examples*). In large scale versions, the number of training data n is too large to be processed at each iteration of EM: for example, when the majorizing function Q of EM is of the form $Q(\theta;\theta_{\mathrm{curr}}) = \mathsf{R}(\theta) - \langle \bar{\mathsf{s}}(\theta_{\mathrm{curr}}), \phi(\theta)\rangle$, the vector $\bar{\mathsf{s}}(\theta_{\mathrm{curr}})$ often has the form $n^{-1}\sum_{i=1}^{n} \bar{\mathsf{s}}_i(\theta_{\mathrm{curr}})$ and the sum over n terms can not be allowed at each iteration of EM. To overcome this intractability in this so-called *finite-sum* setting, incremental EM-based algorithms were proposed: let us cite `incremental EM` [9], `Online-EM` [10], `sEM-VR` [11], `FIEM` [12] (see also [13] for `opt-FIEM`) and `SPIDER-EM` [14, 15]. The three algorithms `sEM-vr`, `FIEM` and `SPIDER-EM` can be seen as a `Online-EM` algorithm combined with a variance reduction technique through the construction of a control variate; they all improve on `Online-EM` (see e.g. [15]). However, these EM-based algorithms designed for the *finite-sum* framework all consider that the functions $\theta \mapsto \bar{\mathsf{s}}_i(\theta)$ can be explicitly evaluated for any θ and $i = 1, \cdots, n$, while being defined as an expectation.

This paper introduces a novel EM-based procedure, named `Perturbed Prox-Preconditioned SPIDER` which tackles the two difficulties: ; (i) the *finite-sum* setting; (ii) the intractability of the quantities $\bar{\mathsf{s}}_i(\theta_{\mathrm{curr}})$. It is proved in [14] that the complexity bounds of `SPIDER-EM`, expressed as the number of optimization steps and as the number of evaluations of the quantities $\bar{\mathsf{s}}_i(\theta_{\mathrm{curr}})$ required to reach an ϵ-approximate stationary point of F, improves over the state-of-the art. Therefore, our algorithm builds on `SPIDER-EM`. It is also designed to address a composite problem with a non-smooth term. `3P-SPIDER` is introduced in Section 2, with an emphasis on the case the quantities $\bar{\mathsf{s}}_i(\theta_{\mathrm{curr}})$ are approximated by a Monte Carlo sum. In Section 3, the algorithm is applied to the logistic regression problem; insights on the choice of some design parameters are also given. It is shown that this perturbed version of `SPIDER-EM` improves on the perturbed version of `Online-EM` thus illustrating that the variance reduction technique is still perceptible. This benefit is all the more visible that the error when approximating the $\bar{\mathsf{s}}_i(\theta_{\mathrm{curr}})$'s is small. Finally, since `3P-SPIDER` combines two approximations to address the intractability of the $\bar{\mathsf{s}}_i(\theta_{\mathrm{curr}})$'s and the finite-sum setting, it is advocated to regularly refresh the control-variate approximation with a full screening of the data set.

The complexity analysis of this algorithm is provided in [16]: under conditions on the approximations of the $\bar{\mathsf{s}}_i(\theta_{\mathrm{curr}})'s$, which are satisfied for example for a Monte Carlo approximation, it is shown that `3P-SPIDER` has the same complexity bounds as `SPIDER-EM`. In that sense, it remains optimal among the (perturbed) incremental EM algorithms.

Notations $\mathbb{R}_+^\star$ and $\mathbb{N}^\star$ denote respectively (resp.) the positive real line and the set of the positive integers. For $n \in \mathbb{N}^\star$, set

This work is partially supported by the *Fondation Simone et Cino Del Duca* through the project OpSiMorE and by the ANR-19-CHIA-0002-01. Part of this work was conducted under the auspices of the Lagrange Center in Mathematics and Computer Sciences

$[n]^\star \stackrel{\text{def}}{=} \{1, \cdots, n\}$ and $[n] \stackrel{\text{def}}{=} \{0, \cdots, n\}$. For $x \in \mathbb{R}$, $\lceil x \rceil$ is the nearest integer greater than or equal to x. Vectors are column-vectors; for a, b in $\mathbb{R}^\ell$, $\langle a, b \rangle$ denotes the Euclidean scalar product, and $\|a\|$ the associated norm. For a matrix A, A^T and A^{-1} are resp. its transpose and its inverse. I_d is the $d \times d$ identity matrix. The random variables are defined on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$; $\mathbb{E}$ denotes the associated expectation. For random variables U, V, $\mathbb{E}[U|V]$ is the conditional expectation of U given V. For a smooth function f, $\nabla_x f$ (or simply ∇f when clear enough) is the gradient of f with respect to the variable x; $\nabla^2 f$ is its hessian. For a proper lower semi-continuous convex function g and x in its (assumed) non-empty domain, $\partial g(x)$ is the subdifferential of g at x.

2. THE PERTURBED PROX-PRECONDITIONED SPIDER ALGORITHM

2.1. The optimization problem

We address the minimization of an objective function $F : \Theta \to \mathbb{R}$:

$$\theta \mapsto \frac{-1}{n} \sum_{i=1}^{n} \log \int_{\mathsf{Z}} h_i(z) \exp(\langle \mathsf{s}_i(z), \phi(\theta) \rangle) \mathrm{d}\mu(z) + \mathsf{R}(\theta) \quad (1)$$

where Θ is an open subset of $\mathbb{R}^d$, $(\mathsf{Z}, \mathcal{Z})$ is a measurable space, $\mathcal{Z}$ denoting a σ-algebra over Z; the functions $\phi : \Theta \to \mathbb{R}^q$, $\mathsf{R} : \Theta \to \mathbb{R}$ and for all $i \in [n]^\star$, $\mathsf{s}_i : \mathsf{Z} \to \mathbb{R}^q$ and $h_i : \mathsf{Z} \to \mathbb{R}_+^\star$ are measurable; and μ is a dominating measure on $(\mathsf{Z}, \mathcal{Z})$. The minimization of the negative log-likelihood in latent variable models provides examples of such a problem. As a first example, consider the maximum likelihood estimate of a mixture of densities from the curved exponential family (see e.g. [15, supp. material] for the Gaussian mixture model). As a second example, consider the following logistic regression model: given $\mathbb{R}^d$-valued covariate vectors $\{X_i, i \in [n]^\star\}$, for any $\theta \in \Theta \stackrel{\text{def}}{=} \mathbb{R}^d$, the binary observations $\{Y_i, i \in [n]^\star\}$ are independent with distribution

$$p_\theta(y_i) \propto \int_{\mathbb{R}^d} (1 + \exp(-y_i \langle X_i, z_i \rangle))^{-1} \times \exp\left(-(2\sigma^2)^{-1} \|z_i - \theta\|^2\right) \mathrm{d}z_i \ ,$$

for any $i \in [n]^\star$, $y_i \in \{-1, 1\}$. In words, each individual $\#i$ in the training set has an individual predictor Z_i. Given Z_i, the success probability $\mathbb{P}(Y_i = 1 \mid Z_i)$ is $(1 + \exp(-\langle X_i, Z_i \rangle))^{-1}$. The individual predictors $Z_1, \cdots, Z_n$ are assumed to have a Gaussian distribution with expectation θ, assumed to be unknown, and (known) diagonal covariance matrix $\sigma^2 \mathrm{I}_d$. The ridge-regularized negative log-likelihood, given by $-n^{-1} \sum_{i=1}^n \log p_\theta(Y_i) + \tau \|\theta\|^2$ may be written as (1) with $\mathsf{Z} \stackrel{\text{def}}{=} \mathbb{R}$, $\phi(\theta) \stackrel{\text{def}}{=} \theta$, $\mathrm{d}\mu(z) \stackrel{\text{def}}{=} \exp(-z^2/(2\sigma^2))\mathrm{d}z$,

$$h_i(z) \stackrel{\text{def}}{=} \left(1 + \exp\left(-Y_i \|X_i\| z\right)\right)^{-1} \ , \quad \mathsf{s}_i(z) \stackrel{\text{def}}{=} z \, \frac{X_i}{\sigma^2 \|X_i\|} \ ,$$

$$\mathsf{R}(\theta) \stackrel{\text{def}}{=} \frac{1}{2} \theta^T \left(\frac{1}{\sigma^2 n} \sum_{i=1}^{n} \frac{X_i X_i^T}{\|X_i\|^2} + 2\tau \mathrm{I}_d \right) \theta \ .$$

2.2. EM in the expectation space

For solving this optimization problem, EM defines a sequence $\{\theta_k, k \geq 0\}$ taking values in Θ, by repeating *(i)* E-step: compute

$$Q(\theta; \theta_k) \stackrel{\text{def}}{=} -\frac{1}{n} \sum_{i=1}^{n} \int_{\mathsf{Z}} \langle \mathsf{s}(z), \phi(\theta) \rangle \ p_i(z; \theta_k) \mathrm{d}\mu(z) + \mathsf{R}(\theta)$$

where for any $z \in \mathsf{Z}$, $\theta \in \Theta$, $i \in [n]^\star$,

$$p_i(z; \theta) \propto h_i(z) \exp(\langle \mathsf{s}_i(z), \phi(\theta) \rangle) \quad (2)$$

is a probability density; *(ii)* M-step: compute the minimum

$$\theta_{k+1} \stackrel{\text{def}}{=} \operatorname{argmin}_{\theta \in \Theta} Q(\theta; \theta_k), \quad Q(\theta; \theta_k) = \mathsf{R}(\theta) - \langle \bar{\mathsf{s}}(\theta_k), \phi(\theta) \rangle \ ,$$

with

$$\bar{\mathsf{s}}(\theta) \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \bar{\mathsf{s}}_i(\theta), \qquad \bar{\mathsf{s}}_i(\theta) \stackrel{\text{def}}{=} \int_{\mathsf{Z}} \mathsf{s}_i(z) \, p_i(z; \theta) \mathrm{d}\mu(z) \ .$$

Hereafter, we assume that for any $\theta_{\mathrm{curr}} \in \Theta$, $\theta \mapsto Q(\theta; \theta_{\mathrm{curr}})$ possesses an unique minimum and we define for any s in a closed convex set $\mathcal{S} \supseteq \bar{\mathsf{s}}(\Theta)$,

$$\mathsf{T}(s) \stackrel{\text{def}}{=} \operatorname{argmin}_{\theta \in \Theta} \left(\mathsf{R}(\theta) - \langle s, \phi(\theta) \rangle \right) \ .$$

With these notations, it holds: $\theta_{k+1} = \mathsf{T} \circ \bar{\mathsf{s}}(\theta_k)$.

In the logistic regression example, $p_i(z; \theta)\mathrm{d}\mu(z)$ is the a posteriori distribution of the hidden variable Z_i given the observation Y_i; $q = d$; $\theta \mapsto \mathsf{R}(\theta) - \langle s, \phi(\theta) \rangle$ possesses an unique minimum; for any $s \in \mathcal{S} \stackrel{\text{def}}{=} \mathbb{R}^d$, $\mathsf{T}(s) = \Omega s$ where

$$\Omega \stackrel{\text{def}}{=} \left(\frac{1}{\sigma^2 n} \sum_{i=1}^{n} \frac{X_i X_i^T}{\|X_i\|^2} + 2\tau \mathrm{I}_d \right)^{-1} \ . \quad (3)$$

When such a map T exists, it is well known that EM can be equivalently defined in the expectation step: the computation of the Θ-valued sequence $\{\theta_k, k \geq 0\}$ through $\theta_{k+1} = \mathsf{T} \circ \bar{\mathsf{s}}(\theta_k)$ is equivalent to the computation of the $\bar{\mathsf{s}}(\Theta)$-valued sequence $\{s_k, k \geq 0\}$ through $s_{k+1} = \bar{\mathsf{s}} \circ \mathsf{T}(s_k)$. The limiting points of these sequences are resp. the roots of $\theta \mapsto \mathsf{T} \circ \bar{\mathsf{s}}(\theta) - \theta$ and $s \mapsto \bar{\mathsf{s}} \circ \mathsf{T}(s) - s$ (see e.g. [7]). Hereafter, we will see EM as an algorithm in the expectation space: EM is an iterative procedure designed to find the roots of *the mean field* $\mathsf{h} \colon \mathcal{S} \to \mathbb{R}^q$

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

In the large scale learning setting, $\bar{\mathsf{s}}$ has a prohibitive computational cost since it involves a sum over the full data set of size n: EM can not be applied exactly. A popular alternative in the literature is to replace EM iterations with SA iterations, where the SA algorithm is designed to find the roots of h [7]. `3P-SPIDER` is in the same vein.

2.3. The Perturbed Prox-Preconditioned SPIDER algorithm

Given a sequence of positive step sizes $\{\gamma_k, k \geq 0\}$, SA defines a sequence $\{\widehat{S}_k, k \geq 0\}$ such that

$$\widehat{S}_{k+1} = \widehat{S}_k + \gamma_{k+1} H_{k+1} \quad (4)$$

where H_{k+1} is an approximation of $\mathsf{h}(\widehat{S}_k)$. Observing that $\bar{\mathsf{s}}(\theta) = \mathbb{E}\left[\bar{\mathsf{s}}_I(\theta)\right]$ for some $[n]^\star$-valued uniform random variable I, a natural idea to mimic the asymptotic behavior of EM is the definition

$$H_{k+1} \stackrel{\text{def}}{=} \frac{1}{\mathsf{b}} \sum_{i \in \mathcal{B}_{k+1}} \bar{\mathsf{s}}_i(\widehat{S}_k) - \widehat{S}_k$$

where $\mathcal{B}_{k+1}$ is a batch of size b sampled uniformly from $[n]^\star$ (with or without replacement) and independently of $\widehat{S}_k$. Such a strategy corresponds to the `Online-EM` algorithm. The incremental EM-based algorithms with variance reduction techniques use the property $\mathsf{h}(\widehat{S}_k) = \mathbb{E}\left[H_{k+1} + V | \widehat{S}_k\right]$ for any (conditionally) centered

random variable V. This implies that, thanks to an adequate construction of the *control variate* V, the variance of the approximation of $\mathsf{h}(\widehat{S}_k)$ can be reduced (see e.g. [17] for an introduction to variance reduction methods in Monte Carlo sampling). This is the essence of `sEM-vr`, `FIEM` and `SPIDER-EM` which essentially differ in the definition of V.

`3P-SPIDER` is described in Algorithm 1. As in `SPIDER-EM`, the control variate is refreshed regularly, let us say at the beginning of each *outer* loop $\#t$ (see lines 2 and 10). In `SPIDER-EM`, it is defined as $\bar{\mathsf{s}} \circ \mathsf{T}(\widehat{S}_{t,-1}) = n^{-1} \sum_{i=1}^n \bar{\mathsf{s}}_i(\widehat{S}_{t,-1})$. Here, two perturbations are allowed: the approximation of $\bar{\mathsf{s}}_i(\widehat{S}_{t,-1})$ with a quantity denoted by $\hat{\mathsf{s}}_i^{t,-1}$, and an error $\mathcal{E}_t$ which may include for example the situation when a sub-sample of the n examples is used when computing the sum instead of the full data set. At each *inner* loop $\#(k+1)$, the control variate is modified in order to track the ideal quantity $\bar{\mathsf{s}} \circ \mathsf{T}(\widehat{S}_{t,k})$: note indeed that $\mathsf{S}_{t,0} \approx \bar{\mathsf{s}} \circ \mathsf{T}(\widehat{S}_{t,-1})$ and, from line 6, $\mathsf{S}_{t,k+1} - \mathsf{S}_{t,k} \approx \bar{\mathsf{s}} \circ \mathsf{T}(\widehat{S}_{t,k}) - \bar{\mathsf{s}} \circ \mathsf{T}(\widehat{S}_{t,k-1})$.

The sequence of interest $\{\widehat{S}_{t,k}, t \in [k_{\rm out}]^\star, k \in [k_{\rm in}]\}$ is updated first by a SA step (see Line 7) followed with a proximal step (see Line 8). In the SA step, the mean field $\mathsf{h}(\widehat{S}_{t,k}) = \bar{\mathsf{s}} \circ \mathsf{T}(\widehat{S}_{t,k}) - \widehat{S}_{t,k}$ is approximated with (see Lines 6 and 7)

$$H_{k+1} \stackrel{\rm def}{=} \frac{1}{\mathsf{b}} \sum_{i \in \mathcal{B}_{t,k+1}} \hat{\mathsf{s}}_i^{t,k} + V_{k+1} - \widehat{S}_{t,k}$$

where $V_{k+1} \stackrel{\rm def}{=} \mathsf{S}_{t,k} - \mathsf{b}^{-1} \sum_{i \in \mathcal{B}_{t,k+1}} \hat{\mathsf{s}}_i^{t,k-1}$ is a control variate. Here again, $\mathcal{B}_{t,k+1}$ is a batch of size b sampled from $[n]^\star$, with or without replacement and independently of the past of the algorithm.

The proximal step in lines 8 and 12 is a novelty (with respect to `SPIDER-EM`) introduced to force the path of the algorithm $\{\widehat{S}_{t,k}, t \in [k_{\rm out}]^\star, k \in [k_{\rm in}]\}$ to remain in the set $\mathcal{S}$ and possibly to inherit other properties from an adequate definition of g (see section 3 for an example). The proof of the convergence in expectation of the algorithm (see [16]) relies on the observation that the algorithm (4) is a perturbed *preconditioned*-gradient method: by setting $\mathrm{W}(s) \stackrel{\rm def}{=} F \circ \mathsf{T}(s)$, we have under regularity assumptions on the functions $\phi, \mathsf{s}, \mathsf{R}$ that $\nabla \mathrm{W}(s) = -B(s)\mathsf{h}(s)$ for any $s \in \mathcal{S}$, where (see e.g. [13, Proposition 1])

$$B(s) \stackrel{\rm def}{=} \left(\nabla \mathsf{T}(s)\right)^T \, \nabla_\theta^2 \left(\mathsf{R}(\theta) - \langle s, \phi(\theta) \rangle\right)|_{\theta = \mathsf{T}(s)} \, \left(\nabla \mathsf{T}(s)\right) \ ,$$

is a positive-definite matrix. Therefore, given a lower semi-continuous proper convex function $g : \mathcal{S} \to \mathbb{R} \cup \{+\infty\}$, we use a *weighted* proximal operator defined by

$$\mathrm{Prox}_{B,\gamma g}(s') \stackrel{\rm def}{=} \mathrm{argmin}_{s \in \mathcal{S}} \left(\gamma g(s) + \frac{1}{2}(s - s')^T B (s - s') \right)$$

for any $\gamma > 0$ and any $q \times q$ positive-definite matrix B.

`3P-SPIDER` extends `SPIDER-EM` in the following directions. First, in the definition of the control variates $\mathsf{S}_{t,k}$, it allows to substitute the intractable $\bar{\mathsf{s}}_i \circ \mathsf{T}(\widehat{S}_{t,k})$ with an approximation $\hat{\mathsf{s}}_i^{t,k}$. Second, it adds a proximal step in order to force the sequence $\{\widehat{S}_{t,k}, t \in [k_{\rm out}]^\star, k \in [k_{\rm in}]\}$ to have some properties (see [18, 19] for a similar idea applied to the `SPIDER` algorithm, with $B(s) = \mathrm{I}_d$). Finally, it allows a perturbation $\mathcal{E}_t$ when initializing the control variate $\mathsf{S}_{t,0}$.

In [20] (see also [16]), the convergence in expectation of the sequence $\{\widehat{S}_{t,k}, t \in [k_{\rm out}]^\star, k \in [k_{\rm in}]\}$ towards the set

$$\begin{aligned} \mathcal{L} &\stackrel{\rm def}{=} \{s : \mathrm{Prox}_{B(s),\gamma g}(s + \gamma \mathsf{h}(s)) = s\} \qquad \forall \gamma > 0 \ , \\ &= \{s : 0 \in \partial g(s) - B(s)\mathsf{h}(s)\} = \{s : 0 \in \partial g(s) + \nabla \mathrm{W}(s)\} \end{aligned}$$

Data: $k_{\rm out}, k_{\rm in} \in \mathbb{N}^\star$; $\widehat{S}_{\rm init} \in \mathcal{S}$; $\gamma_{t,0} \geq 0$, $\gamma_{t,k} > 0$ for $t \in [k_{\rm out}]^\star$, $k \in [k_{\rm in}]^\star$, a lower semi-continuous proper convex function g

Result: The 3P-SPIDER sequence $\{\widehat{S}_{t,k}, t \in [k_{\rm in}]^\star, k \in [k_{\rm in}]\}$

1 $\widehat{S}_{1,0} = \widehat{S}_{1,-1} = \widehat{S}_{\rm init}$;
2 $\mathsf{S}_{1,0} = n^{-1} \sum_{i=1}^n \hat{\mathsf{s}}_i^{1,-1} + \mathcal{E}_1$;
3 **for** $t = 1, \cdots, k_{\rm out}$ **do**
4 **for** $k = 0, \ldots, k_{\rm in} - 1$ **do**
5 Sample a mini batch $\mathcal{B}_{t,k+1}$ of size b in $[n]^\star$;
6 $\mathsf{S}_{t,k+1} = \mathsf{S}_{t,k} + \mathsf{b}^{-1} \sum_{i \in \mathcal{B}_{t,k+1}} \left(\hat{\mathsf{s}}_i^{t,k} - \hat{\mathsf{s}}_i^{t,k-1} \right)$;
7 $\widehat{S}_{t,k+1/2} = \widehat{S}_{t,k} + \gamma_{t,k+1} \left(\mathsf{S}_{t,k+1} - \widehat{S}_{t,k} \right)$;
8 $\widehat{S}_{t,k+1} = \mathrm{Prox}_{B(\widehat{S}_{t,k}), \gamma_{t,k+1} g} \left(\widehat{S}_{t,k+1/2} \right)$;
9 $\widehat{S}_{t+1,-1} = \widehat{S}_{t,k_{\rm in}}$;
10 $\mathsf{S}_{t+1,0} = n^{-1} \sum_{i=1}^n \hat{\mathsf{s}}_i^{t+1,-1} + \mathcal{E}_{t+1}$;
11 $\widehat{S}_{t+1,-1/2} = \widehat{S}_{t+1,-1} + \gamma_{t+1,0} \left(\mathsf{S}_{t+1,0} - \widehat{S}_{t+1,-1} \right)$;
12 $\widehat{S}_{t+1,0} = \mathrm{Prox}_{B(\widehat{S}_{t+1,-1}), \gamma_{t+1,0} g}(\widehat{S}_{t+1,-1/2})$

Algorithm 1: The Perturbed Prox-Preconditioned SPIDER (3P-SPIDER) algorithm.

is proved. In the case g is the indicator function of a closed convex set $\mathcal{K}$ and $B(s)$ is invertible for any $s \in \mathcal{K} \cap \mathcal{S}$, the limiting points are the roots of $\nabla \mathrm{W}$ which are in $\mathcal{K}$, that is the roots of $\mathsf{h}(s)$ in $\mathcal{K}$: `3P-SPIDER` has the same asymptotic behavior as EM.

2.4. Case of a Monte Carlo approximation

The intractable quantity $\bar{\mathsf{s}}_i \circ \mathsf{T}(s)$ is defined by

$$\bar{\mathsf{s}}_i \circ \mathsf{T}(s) \stackrel{\rm def}{=} \int_{\mathsf{Z}} \mathsf{s}_i(z) p_i(z; \mathsf{T}(s)) \mathrm{d}\mu(z) \ ,$$

where $p_i(z; \mathsf{T}(s)) \, \mathrm{d}\mu(z)$ is the distribution defined by (2). When this integral is not explicit, a natural idea is to approximate it by a Monte Carlo (MC) sum. For example,

$$\hat{\mathsf{s}}_i^{t,k} \stackrel{\rm def}{=} \frac{1}{m_{t,k+1}} \sum_{r=1}^{m_{t,k+1}} \mathsf{s}_i \left(Z_r^{i,t,k} \right) \ ,$$

where for $t \in [k_{\rm out}]^\star, k \in [k_{\rm in}]$ and $i \in [n]^\star$, $\{Z_r^{i,t,k}, r \geq 1\}$ is a Markov chain designed to be ergodic with unique invariant distribution $p_i(z; \mathsf{T}(\widehat{S}_{t,k})) \mathrm{d}\mu(z)$. Such a chain can be obtained by running a Markov chain Monte Carlo sampler (see e.g. [21, 22]); note that the independent and identically distributed (i.i.d) setting is a special case of the Markovian setting. When the random variables $\{Z_r^{i,t,k}, r \geq 1\}$ are i.i.d., we have $\mathbb{E}\left[\hat{\mathsf{s}}_i^{t,k} | \widehat{S}_{t,k}\right] = \bar{\mathsf{s}}_i \circ \mathsf{T}(\widehat{S}_{t,k})$; when the random variables $\{Z_r^{i,t,k}, r \geq 1\}$ are a Markov chain, the approximation is biased: $\mathbb{E}\left[\hat{\mathsf{s}}_i^{t,k} | \widehat{S}_{t,k}\right] \neq \bar{\mathsf{s}}_i \circ \mathsf{T}(\widehat{S}_{t,k})$. In this biased case, the algorithm still converges to $\mathcal{L}$ but its theoretical analysis is more technical (see [20]).

3. APPLICATION: INFERENCE IN THE LOGISTIC REGRESSION MODEL

Let us consider the logistic regression model described in Section 2.1. Since $\mathbb{P}(Y_i = y_i) \leq 1$, it can be proved that the minima of F are in the set $\{\theta \in \mathbb{R}^d : \tau\|\theta\|^2 \leq \ln 4\}$. Therefore, in the expectation space, the minima of $\mathsf{W} = F \circ \mathsf{T}$ are in the set $\{s \in \mathbb{R}^d : \tau s^T \Omega^2 s \leq \ln 4\}$ which is included in $\mathcal{K} \stackrel{\text{def}}{=} \{s \in \mathbb{R}^d : \tau s^T \Omega s \leq \ln 4/\lambda_{\min}\}$ where $\lambda_{\min}$ is the positive minimal eigenvalue of Ω (see (3)). `3P-SPIDER` is applied with $g \stackrel{\text{def}}{=} \chi_{\mathcal{K}}$, the characteristic function of the compact convex set $\mathcal{K}$. With this definition of $\mathcal{K}$ and since $B(s) = \Omega$ for any $s \in \mathcal{S} \stackrel{\text{def}}{=} \mathbb{R}^d$, the computation of the operator $\text{Prox}_{B(s),\gamma g}$ is explicit. $\bar{\mathsf{s}}_i(\Omega s)$ is equal to

$$\frac{X_i}{\sigma^2\|X_i\|}\frac{1}{Z(s)}\int_{\mathbb{R}} z\,\frac{\exp(z\,\langle X_i, \Omega s\rangle/(\sigma^2\|X_i\|))}{1+\exp(-Y_i\|X_i\|z)}\,\exp(\frac{-z^2}{2\sigma^2})\mathrm{d}z$$

where the normalizing constant $Z(s)$ is given by

$$Z(s) \stackrel{\text{def}}{=} \int_{\mathbb{R}} \frac{\exp(z\,\langle X_i, \Omega s\rangle/(\sigma^2\|X_i\|))}{1+\exp(-Y_i\|X_i\|z)}\,\exp(-z^2/(2\sigma^2))\mathrm{d}z\ .$$

These integrals are not explicit; we consider the approximation $\hat{\mathsf{s}}_i^{t,k}$ of $\bar{\mathsf{s}}_i(\Omega\widehat{S}_{t,k})$ given by a MC sum as described in Section 2.4; the samples $Z_r^{i,t,k}$ are obtained by the Gibbs sampler given in [23].

The numerical illustrations use the MNIST data set: the class "1" contains the 12 873 images in the training set labeled 1 and 3 and the class "−1" contains the 12 116 images in the training set labeled 7 and 8; hence $n = 24\,989$. The 787 pixels are compressed in 50 features through PCA (see [13, section 5] for the details). An intercept is included in the covariates so $d = 51$. `3P-SPIDER` is run with $\sigma^2 = 0.1$, $\tau = 1$, $k_{\text{out}} = 20$, $k_{\text{in}} = \lceil\sqrt{n}/10\rceil = 16$ and $\mathsf{b} = \lceil 10\sqrt{n}\rceil = 1581$. Note that $k_{\text{in}} \times \mathsf{b} = n$ so that each outer loop requires n examples; it corresponds to an *epoch*.

We study the quantity

$$\mathcal{D}_{t,k} \stackrel{\text{def}}{=} \mathbb{E}\left[\frac{\|\widehat{S}_{t,k+1} - \widehat{S}_{t,k}\|^2}{\gamma_{t,k+1}^2}\right], \qquad t \in [k_{\text{out}}]^\star, k \in [k_{\text{in}} - 1],$$

which quantifies how far `3P-SPIDER` is from its limiting set (see the definition of $\mathcal{L}$); this expectation is estimated by a MC sum over 25 independent runs. All the runs start from the same value $\widehat{S}_{\text{init}}$. For the computation of the quantity $\mathsf{S}_{t,0}$ at each outer loop $\#t$, all the examples are used and the expectations $\bar{\mathsf{s}}_i \circ \mathsf{T}(\widehat{S}_{t,-1})$ are approximated by a MC sum with $m' = 10\lceil\sqrt{n}\rceil = 1590$ points. Hence, $\mathcal{E}_t = 0$ except when specified (see the second analysis below); the computation of $\mathsf{S}_{t,0}$ requires n examples: it corresponds to an epoch.

First, `3P-SPIDER` is run with $m_{t,k} = 2\lceil\sqrt{n}\rceil$ and $\gamma_{t,k} = \gamma_{t,0} = 0.1$; `Online-EM` is run with a step size equal to $\gamma_t = 0.1$, a batch size $\mathsf{b} = \lceil 10\sqrt{n}\rceil$ (case "sqr") and $\mathsf{b} = n$ (case "full"), and a MC approximation for $\bar{\mathsf{s}}_i$ computed with $2\lceil\sqrt{n}\rceil$ points. Figure 1(a) displays $\mathcal{D}_{t,0}$ for `3P-SPIDER` and $\|\widehat{S}_{t+1} - \widehat{S}_t\|^2/\gamma_t^2$ for `Online-EM`. The x-axis scales as the number of *epoch*, that is the use of n examples. The plot shows that, even when the expectations $\bar{\mathsf{s}}_i$ have to be replaced with approximations, `3P-SPIDER` is far more efficient than `Online-EM` (in which the exact expectations are also replaced with MC approximations).

Second, we analyze the role of $\mathcal{E}_t$ when initializing the control variate $\mathsf{S}_{t,0}$. We run `3P-SPIDER` with $\gamma_{t,k} = \gamma_{t,0} = 0.1$ and a number of MC points $m_{t,k} = 2\lceil\sqrt{n}\rceil$; the quantity $\mathcal{D}_{t,k}$ is displayed on Figure 1(b) vs the cumulated number of inner loops; the squares, circles and diamonds indicate $\mathcal{D}_{t,k_{\text{in}}}$ for every outer loop. The case "full" corresponds to $\mathcal{E}_t = 0$, the case "half" (resp. "quarter") corresponds to $\mathsf{S}_{t,0}$ computed with a batch of size $\lceil n/2\rceil$ examples (resp. $\lceil n/4\rceil$). The control variate is too poor in the case "half" and "quarter" and, after the transient phase when the possibly bad initialization is forgotten, it weakens the benefit of its use: we definitely advice $\mathcal{E}_t = 0$.

Third, we analyze how the variability of the MC approximation and the choice of the step sizes affect the rate of convergence of `3P-SPIDER`. In "Case 1", the values are the same as in Figure 1(a). In "Case 2", $\gamma_{t,k} = \gamma_{t,0} = 0.1$ during the first three outer loops and then $\gamma_{t,k} = \gamma_{t,0} = 10^{-3}$; $m_{t,k}$ is as in "Case 1" until the outer loop $\#10$ and then $m_{t,k}$ is multiplied by 5. In "Case 3", the step sizes and the number of MC points are as in "Case 2", except that the step size decreases later, at outer loop $\#6$. On Figure 1(c), we display $\mathcal{D}_{t,k}$ vs the cumulated number of inner loops, starting from the number $\#32$ (that is, at the end of the second outer loop); the diamonds, circles and squares indicate $\mathcal{D}_{t,k_{\text{in}}}$. First, `3P-SPIDER` is improved when the number of MC points increases; when the fluctuations of the algorithm is of the same order as the fluctuations of the MC errors, `3P-SPIDER` can not go forward anymore in order to reach a more precise estimation of the parameter (compare "Case 1" and "Case 3"). Small step sizes penalize the algorithm (compare "Case 1" and "Case 2").

Finally, we discuss the strategies $\gamma_{t,0} = 0$ and $\gamma_{t,0} \neq 0$. `3P-SPIDER` run as in Figure 1(a) corresponds to "Case 1". In "Case 2" and "Case 3", the number of MC points is multiplied by 5 from the outer loop $\#11$, and $\gamma_{t,k} = 0.1$ for any $k > 0$. In "Case 1" and "Case 2", $\gamma_{t,0} = 0.1$ and in "Case 3", $\gamma_{t,0} = 0$. On Figure 1(d), we display $\mathcal{D}_{t,k}$ vs the cumulated number of inner loops, starting from the loop $\#32$; the diamonds, circles and squares indicate $\mathcal{D}_{t,k_{\text{in}}}$. Here again, we observe the benefit of reducing the MC variability by increasing the number of MC points (compare "Case 1" to the other cases); "Case 2" and "Case 3" are almost similar, maybe with a slightly better behavior for "Case 2".

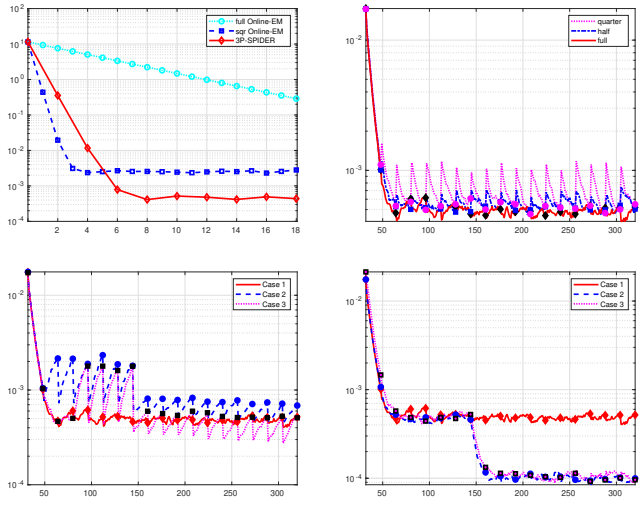

Fig. 1. [(a) top left] Comparison of algorithms; [(b) top right] Role of the size of the batch when computing $\mathsf{S}_{t,0}$; [(c) bottom left] Role of the step sizes $\gamma_{t,k}$ and the number of Monte Carlo points when computing $\hat{\mathsf{s}}_{t,k}^i$; [(d) bottom right] Role of $\gamma_{t,0}$

4. REFERENCES

[1] A.P. Dempster, N.M. Laird, and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Roy. Stat. Soc. B Met.*, vol. 39, no. 1, pp. 1–38, 1977.

[2] C.F.J. Wu, "On the Convergence Properties of the EM Algorithm," *Ann. Statist.*, vol. 11, no. 1, pp. 95–103, 1983.

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

[4] G.C.G. Wei and M.A. Tanner, "A Monte Carlo Implementation of the EM Algorithm and the Poor Man's Data Augmentation Algorithms," *J. Am. Stat. Assoc.*, vol. 85, no. 411, pp. 699–704, 1990.

[5] G. Fort and E. Moulines, "Convergence of the Monte Carlo expectation maximization for curved exponential families," *Ann. Statist.*, vol. 31, no. 4, pp. 1220–1259, 2003.

[6] G. Celeux and J. Diebolt, "The SEM algorithm: a probabilistic teacher algorithm derived from the EM algorithm for the mixture problem," *Computational Statistics Quarterly*, vol. 2, pp. 73–82, 1985.

[7] B. Delyon, M. Lavielle, and E. Moulines, "Convergence of a Stochastic Approximation version of the EM algorithm," *Ann. Statist.*, vol. 27, no. 1, pp. 94–128, 1999.

[8] A. Benveniste, M. Métivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations*, Springer Verlag, 1990.

[9] R. M. Neal and G. E. Hinton, *A View of the EM Algorithm that Justifies Incremental, Sparse, and other Variants*, pp. 355–368, Springer Netherlands, Dordrecht, 1998.

[10] O. Cappé and E. Moulines, "On-line Expectation Maximization algorithm for latent data models," *J. Roy. Stat. Soc. B Met.*, vol. 71, no. 3, pp. 593–613, 2009.

[11] J. Chen, J. Zhu, Y.W. Teh, and T. Zhang, "Stochastic Expectation Maximization with Variance Reduction," in *Advances in Neural Information Processing Systems 31*, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., pp. 7967–7977. Curran Associates, Inc., 2018.

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

[13] G. Fort, P. Gach, and E. Moulines, "Fast Incremental Expectation Maximization for finite-sum optimization: nonasymptotic convergence," *Statistics and Computing*, 2021, Accepted for publication.

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

[15] G. Fort, E. Moulines, and H.-T. Wai, "Faster Variance Reduced Stochastic Expectation Maximization for Nonconvex Finite-Sum Optimization," in *proceedings of the 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, Supplementary material available on HAL.

[16] G. Fort and E. Moulines, "The Perturbed Prox-Preconditioned SPIDER algorithm: non-asymptotic convergence bounds," in *2021 IEEE Statistical Signal Processing Workshop (SSP)*, 2021, Accepted.

[17] P. Glasserman, *Monte Carlo methods in financial engineering*, Springer, New York, 2004.

[18] Z. Li and J. Li, "A simple proximal stochastic gradient method for nonsmooth nonconvex optimization," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2018, NIPS'18, p. 5569–5579, Curran Associates Inc.

[19] Z. Wang, K. Ji, Y. Zhou, Y. Liang, and V. Tarokh, "Spiderboost and momentum: Faster variance reduction algorithms," in *Advances in Neural Information Processing Systems*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, Eds. 2019, vol. 32, pp. 2406–2416, Curran Associates, Inc.

[20] G. Fort and E. Moulines, "Perturbed Prox-Preconditioned SPIDER for finite-sum optimization," Tech. Rep., 2021, work in progress.

[21] O. Cappé and C. Robert, "Markov Chain Monte Carlo: 10 years and still running!," *Journal of the American Statistical Association*, vol. 95, no. 452, pp. 1282–1286, 2000.

[22] S. Brooks, A. Gelman, G. Jones, and X.-L. Meng, *Handbook of Markov Chain Monte Carlo*, CRC press, 2011.

[23] N. G. Polson, J.G. Scott, and J. Windle, "Bayesian Inference for Logistic Models Using Pólya–Gamma Latent Variables," *Journal of the American Statistical Association*, vol. 108, no. 504, pp. 1339–1349, 2013.