A Variance Reduced Expectation Maximization algorithm for finite-sum optimization

Gersende Fort
(CNRS, Institut de Mathématiques de Toulouse, France)

Joint work with Eric Moulines (CMAP, Ecole Polytechnique, France) and Hoi-To Wai (Chinese Univ. of Hong Kong, Hong-Kong)

Séminaire Equipe MaIAGE
INRAE Jouy-en-Josas

In this talk

Motivated by the Large scale Learning setting,

- Design a novel algorithm for the optimization problem:

$$\text{find } s_\star \in \mathbb{R}^q \text{ s.t.} \qquad \mathsf{h}(s_\star) = 0$$

- Adapted to the finite sum setting (large number of examples n)

$$\text{when} \qquad \mathsf{h}(s) = \frac{1}{n}\sum_{i=1}^{n} \mathsf{h}_i(s)$$

- Stochastic optimization: it combines
  - the Stochastic Approximation method Robbins and Monro (1951); Benveniste et al. (1990)

$$\widehat{S}_{n+1} = \widehat{S}_n + \gamma_{n+1} H_{n+1} \qquad H_{n+1} \approx \mathsf{h}(\widehat{S}_n)$$

  - a variance reduction technique

I. Motivation: Expectation Maximization for inference in latent variable models (large scale learning)

Reminder 1: latent variable models

- The observations $Y = (Y_1, \cdots, Y_n)$
- A parametric statistical model indexed by $\theta \in \Theta$
- Some latent or hidden variables $Z = (Z_1, \cdots, Z_n)$
- A *complete data* vector: (Y, Z), make easier or more general the definition of the observations

Example 1: Mixture models

$$Y_i \overset{i.i.d}{\sim} \sum_{g=1}^{G} \omega_g \, f_g(y_i; \theta_g) \mathrm{d}\mu_i \qquad \theta = (\theta_{1:G}, \omega_{1:G})$$

Or equivalently

$$Z_i \sim \omega_{\bullet} \qquad Y_i | (Z_i = g) \sim f_g(y_i; \theta_g) \mathrm{d}\mu_i$$

Example 2: Mixed Effect models

Random effects $Z_{\bullet} \rightarrow$ non explicit expression of the likielihood of the observations

Reminder 2: The (standard) EM algorithm to optimize the likelihood

- The objective function:
  - The likelihood, non explicit

$$\theta \mapsto \log p(Y_{1:n};\theta) = \sum_{i=1}^{n} \log p(Y_i;\theta) = \sum_{i=1}^{n} \log \int \underbrace{\bar{p}(Y_i, z_i; \theta)}_{\text{complete date likel.}} \, \mathrm{d}\nu(z_i)$$

- EM solves the optimization problem by iterating
  - the Expectation-step:

$$Q(\theta, \theta^t) = \sum_{i=1}^{n} \mathbb{E}\left[\log \bar{p}(Y_i, \mathbf{Z_i}; \theta) | Y_{1:n}, \theta^t\right]$$

    the latent variables are *imputed* with their best approximation at time $\#t$
  - The Maximization step:

$$\theta^{t+1} \in \mathrm{argmax}_{\theta \in \theta^t} Q(\theta, \theta^t)$$

Reminder 3: The curved exponential family

- Pbm: computation of the *function*

$$\theta \mapsto Q(\theta, \theta^t) = \sum_{i=1}^{n} \mathbb{E}\left[\log \bar{p}(Y_i, \mathbf{Z_i}; \theta) | Y_{1:n}, \theta^t\right]$$

- Realistic when

$$\log \bar{p}(Y_i, \mathbf{Z_i}; \theta) = \psi(\theta) + \langle s_i(\mathbf{Z_i}), \phi(\theta) \rangle$$

In that case,

$$Q(\theta, \theta^t) = \psi(\theta) + \left\langle \sum_{i=1}^{n} \underbrace{\mathbb{E}\left[s_i(\mathbf{Z_i}) | Y_{1:n}, \theta^t\right]}_{\bar{\mathsf{s}}_i(\theta^t)}, \phi(\theta) \right\rangle$$

and the two steps of EM are

- E-step: compute $\sum_{i=1}^{n} \bar{\mathsf{s}}_i(\theta^t)$
- M-step: update

$$\theta^{t+1} \in \operatorname{argmax} \ \psi(\theta) + \left\langle \sum_{i=1}^{n} \bar{\mathsf{s}}_i(\theta^t), \phi(\theta) \right\rangle$$

Optimization problem: finite sum setting, for curved exponential families

In this talk

- Solve on $\Theta \subseteq \mathbb{R}^d$ the **minimization** problem

$$\mathrm{argmin}_{\theta \in \Theta} - \frac{1}{n} \sum_{i=1}^{n} \log \int_{\mathsf{Z}} p_i(z_i; \theta) \mathsf{d}\mu(z_i) + \tilde{\mathsf{R}}(\theta), \qquad p_i(z_i;\theta) > 0$$

- In the curved exponential family:

$$-\frac{1}{n} \sum_{i=1}^{n} \log \int_{\mathsf{Z}} h_i(z_i) \exp\left(\langle s_i(z_i), \phi(\theta)\rangle - \psi(\theta)\right) \mathsf{d}\mu(z_i) + \tilde{\mathsf{R}}(\theta)$$

- Via EM-based methods

Intractable EM Dempster, Laird, Rubin (1977)

Objective function:

$$-\sum_{i=1}^{n} \log \int_{\mathsf{Z}} p_i(z_i;\theta) \mathrm{d}\mu(z_i) + \tilde{\mathsf{R}}(\theta), \qquad p_i(z_i;\theta) = h_i(z_i) \exp\left(\langle s_i(z_i), \phi(\theta)\rangle - \psi(\theta)\right)$$

- **EM algorithm:** Repeat for $t = 0, \ldots$

E-step $\quad \bar{\mathsf{s}}(\theta_t) = \frac{1}{n} \sum_{i=1}^{n} \bar{\mathsf{s}}_i(\theta_t) \qquad$ where $\bar{\mathsf{s}}_i(\theta) \stackrel{\mathrm{def}}{=} \int_{\mathsf{Z}} \mathsf{s}_i(z_i) \frac{p_i(z_i;\theta)}{\int p_i(u;\theta) \mathrm{d}\mu(u)} \mathrm{d}\mu(z_i)$

M-step $\quad \theta_{t+1} = \mathsf{T}\left(\bar{\mathsf{s}}(\theta_t)\right)$

where

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

E-step

$\rightarrow$ sum over n expectations $\rightarrow$ Large computational cost of each EM iteration, when n is large

$\rightarrow$ in some cases, the expectations $\bar{\mathsf{s}}_i$'s are intractable

We consider the case when the M-step (computation of T) is explicit

EM in the expectation space

- EM: an algorithm in the *expectation space*

$$\theta_{t+1} = \mathsf{T} \circ \bar{\mathsf{s}}(\theta_t) = \mathsf{T} \circ \underbrace{\bar{\mathsf{s}} \circ \mathsf{T}} \circ \bar{\mathsf{s}} \dots \underbrace{\bar{\mathsf{s}} \circ \mathsf{T}} \circ \bar{\mathsf{s}}(\theta_0)$$

$$S_{t+1} = \bar{\mathsf{s}} \circ \mathsf{T}(S_t) = \frac{1}{n} \sum_{i=1}^{n} \bar{\mathsf{s}}_i \circ \mathsf{T}(S_t)$$

- EM designed to find the roots of

$$\begin{aligned} \mathsf{h}(s) &\stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \underbrace{\bar{\mathsf{s}}_i \circ \mathsf{T}(s) - s}_{\mathsf{h}_i(s)} \\ &= \mathbb{E}\left[\mathsf{h}_I(s)\right] \\ &= \mathbb{E}\left[\mathsf{h}_I(s) + V\right] \qquad \mathbb{E}[V] = 0 \end{aligned}$$

where $I \sim \mathcal{U}(\{1, \dots, n\})$ and V is a *control variate* i.e. r.v. correlated with h_I and centered.

A Lyapunov function

- EM designed to solve on $\Theta \subseteq \mathbb{R}^d$

$$\mathrm{argmin}_{\theta \in \Theta} F(\theta), \qquad F(\theta) \stackrel{\mathrm{def}}{=} \mathsf{R}(\theta) - \frac{1}{n} \sum_{i=1}^{n} \log \int_{\mathsf{Z}} p_i(z;\theta) \mathsf{d}\mu(z)$$

- For exact EM: F is a Lyapunov function

$$F(\theta_{t+1}) \leq F(\theta_t)$$

- EM in the expectation space:

$$W \stackrel{\mathrm{def}}{=} F \circ \mathsf{T}$$

it holds (under regularity conditions)

$$\nabla W(s) = -B(s)\, \mathsf{h}(s) \qquad \mathsf{h}(s) \stackrel{\mathrm{def}}{=} \frac{1}{n} \sum_{i=1}^{n} (\bar{\mathsf{s}}_i \circ \mathsf{T}(s) - s)$$

$\hookrightarrow$ An algorithm designed to find the roots of h is among the stochastic preconditioned gradient algorithms, with preconditioning matrix $B^{-1}(s)$.

II. Algorithm and Convergence analysis

Variance reduced incremental algorithms (in the EM context $\mathsf{h}_i = \bar{\mathsf{s}}_i \circ \mathsf{T}(s) - s$)

solve on $\mathbb{R}^q$: $\mathsf{h}(s) = 0$ with $\mathsf{h}(s) = n^{-1} \sum_{i=1}^n \mathsf{h}_i(s) = \mathbb{E}[\mathsf{h}_I(s)]$

$$\widehat{S}_{t+1} = \widehat{S}_t + \gamma_{t+1} \left(\frac{1}{\mathsf{b}} \sum_{i \in \mathcal{B}_{t+1}} \mathsf{h}_i(\widehat{S}_t) + V_{t+1} \right)$$

where $\mathcal{B}_{t+1}$ is a mini-batch of examples of size $\mathsf{b} << n$.

- Online-EM (Neal and Hinton, 1998; Cappé and Moulines, 2009). NO variance reduction ($V_{t+1} = 0$).
- sEM-vr: Stochastic EM with Variance Reduction Chen et al, 2018
- FIEM: Fast Incremental EM Karimi et al, 2019; Fort et al, 2021
- **SPIDER-EM** Fort, Moulines, Wai - NeurIPS 2020: Stochastic Path Integrated Differential EstimatoR EM

$$V_{t+1} = \sum_{\ell=0}^{t} \left\{ \frac{1}{\mathsf{b}} \sum_{i \in \mathcal{B}_\ell} \mathsf{h}_i(\widehat{S}_{\ell-1}) - \frac{1}{\mathsf{b}} \sum_{i \in \mathcal{B}_{\ell+1}} \mathsf{h}_i(\widehat{S}_{\ell-1}) \right\}$$

Nguyen et al. (2017), Fang et al. (2018), Wang et al. (2019)

SPIDER-EM (Stochastic Path Integrated Differential EstimatoR Expectation Maximization)

1: $\widehat{S}_{1,0} = \widehat{S}_{1,-1} = \widehat{S}_{\text{init}} \qquad V_{1,0} = 0 \qquad \mathcal{B}_{1,0} = \{1, \cdots, n\}$
2: **for** $t = 1, \cdots, k_{\text{out}}$ **do**
3: **for** $k = 0, \ldots, \xi_t - 1$ **do**
4: Sample a mini batch $\mathcal{B}_{t,k+1}$ of size b from $\{1, \cdots, n\}$
5: $V_{t,k+1} = V_{t,k} + \left(\mathsf{b}^{-1} \sum_{i \in \mathcal{B}_{t,k}} \mathsf{h}_i(\widehat{S}_{t,k-1}) - \mathsf{b}^{-1} \sum_{i \in \mathcal{B}_{t,k+1}} \mathsf{h}_i(\widehat{S}_{t,k-1})\right)$
6: $\widehat{S}_{t,k+1} = \widehat{S}_{t,k} + \gamma_{t,k+1} \left(\mathsf{b}^{-1} \sum_{i \in \mathcal{B}_{t,k+1}} \mathsf{h}_i(\widehat{S}_{t,k}) + V_{t,k+1}\right)$
7: **end for**
8: $\widehat{S}_{t+1,-1} = \widehat{S}_{t,\xi_t}$
9: $V_{t+1,0} = 0 \qquad \mathcal{B}_{t+1,0} = \{1, \cdots, n\}$
10: $\widehat{S}_{t+1,0} = \widehat{S}_{t+1,-1} + \gamma_{t+1,0} \left(n^{-1} \sum_{i=1}^{n} \mathsf{h}_i(\widehat{S}_{t+1,-1}) + V_{t+1,0}\right)$
11: **end for**

- k_{out} outer loops, the outer $\#t$ is of length ξ_t
- The control variate is refreshed at each *outer loop* $\#t$ (see Line 9)
- A full scan of the examples at each *outer loop* (see Line 9).

Extensions

- The length of the outer loop is a Geometric random variable with expectation ξ_t. Fort, Moulines, Wai - ICASSP 2021
- Avoid the full scan of the examples when starting each outer loop $\to$ reduction of the computational cost. Fort, Moulines, Wai - ICASSP 2021
- An approximation of h_i Fort, Moulines - SSP 2021

$$\widehat{\mathsf{h}_i(\widehat{S}_{t,k})} = \mathsf{h}_i(\widehat{S}_{t,k}) + \eta_{t,k+1}$$

for example: in EM, $\mathsf{h}_i(s) = \bar{\mathsf{s}}_i \circ \mathsf{T}(s) - s$ and $\bar{\mathsf{s}}_i$ is an expectation w.r.t. the a posteriori distribution of the latent variables $\to$ Monte Carlo approximation.

- A Proximal operator for constrained optimization Fort, Moulines - SSP 2021

$$\widehat{S}_{t,k+1} = \mathrm{Prox}_{B(\widehat{S}_k),\gamma_{t,k+1}\, g}\left(\widehat{S}_{t,k} + \gamma_{t,k+1}\left(\mathsf{b}^{-1}\sum_{i\in\mathcal{B}_{t,k+1}} \widehat{\mathsf{h}_i(\widehat{S}_{t,k})} + V_{t,k+1}\right)\right)$$

for example: find the roots of h in a compact set.

Assumptions

1. There exists a continuously differentiable function $W : \mathbb{R}^q \to \mathbb{R}$ such that

$$\nabla W(s) \overset{\text{def}}{=} -B(s)\,\mathsf{h}(s) \qquad \mathsf{h}(s) \overset{\text{def}}{=} \frac{1}{n}\sum_{i=1}^{n} \mathsf{h}_i(s)$$

   where $B(s)$ is a $q \times q$ positive definite matrix.
   In addition, ∇W is globally Lipschitz with constant $L_{\dot{W}}$,
   and there exist $0 < v_{\min} \leq v_{\max}$ such that the spectrum of $B(s)$ is in $[v_{\min}, v_{\max}]$.

2. For any $i \in \{1, \cdots, n\}$, the function h_i is globally Lipschitz with constant L_i.

Convergence in expectation, explicit h_i's

Under the previous assumptions:

(Fort, Moulines, Wai, NeurIPS 2020)

Set $L^2 \stackrel{\text{def}}{=} n^{-1} \sum_{i=1}^n L_i^2$. *Fix* $k_{\text{out}}, k_{\text{in}}, \mathsf{b} \in \mathbb{N}_\star$. *Choose* $\alpha \in (0, v_{\min}/\mu_\star(k_{\text{in}}, \mathsf{b}))$ *with*

$$\mu_\star(k_{\text{in}}, \mathsf{b}) \stackrel{\text{def}}{=} v_{\max} \frac{\sqrt{k_{\text{in}}}}{\sqrt{\mathsf{b}}} + \frac{L_{\dot{W}}}{2L}.$$

Run the algorithm with $\xi_t = k_{\text{in}}$ *and* $\gamma_{t,k} \stackrel{\text{def}}{=} \alpha / L$. *Then*

$$\mathbb{E}\left[\| \mathsf{h}\left(\widehat{S}_{\tau, \xi - 1}\right) \|^2 \right] \leq \left(\frac{1}{k_{\text{in}}} + \frac{\alpha^2}{\mathsf{b}} \right) \frac{1}{k_{\text{out}}} \frac{2L}{\alpha \{ v_{\min} - \alpha \mu_\star(k_{\text{in}}, \mathsf{b}) \}} \left(\mathbb{E}\left[W(\widehat{S}_{\text{init}}) \right] - \min W \right)$$

where (τ, ξ) *is a uniform r.v. on* $\{1, \cdots, k_{\text{out}}\} \times \{0, \cdots, k_{\text{in}} - 1\}$ *indep of* $\{\widehat{S}_{t,k}\}$.

Complexity for ϵ-approximate stationarity

From this **explicit** expression of an upper bound for

$$\mathbb{E}\left[\|\mathsf{h}(\widehat{S}_{\tau,\xi-1})\|^2\right]$$

- in the `non convex` setting
- with a `random stopping rule`
- as a function of $k_{\text{out}}, k_{\text{in}}, \mathsf{b}, n$ and the learning rate γ $(= \gamma_{t,k}$ for any $t, k > 0)$

To reach ϵ-stationarity, the complexity of SPIDER-EM

With: $k_{\text{in}} = \mathsf{b} = O(\sqrt{n}), \quad k_{\text{out}} = O(1/(\epsilon k_{\text{in}}))$

Nbr of optimization steps: $O(1/\epsilon)$
Nbr of $\bar{\mathsf{s}}_i$'s evaluations: $\mathcal{K} = O(\sqrt{n}\,\epsilon^{-1}) \rightarrow$ *state of the art !*

Algorithm	Complexity $\mathcal{K}$
Online-EM	ϵ^{-2}
iEM	$n\,\epsilon^{-1}$
sEM-vr	$n^{2/3}\,\epsilon^{-1}$
FIEM	$n^{2/3}\,\epsilon^{-1} \wedge \sqrt{n}\,\epsilon^{-3/2}$

Sketch of proof

Inside an outer loop $\#t$, then sum along the inner loops $k = 0$ to $k = k_{\text{in}} - 1$; then sum along the outer loops $t = 1$ to $t = k_{\text{out}}$.

- W is Gradient-Lipschitz, and its gradient is a linear function of h

$$W(\widehat{S}_{t,k+1}) - W(\widehat{S}_{t,k}) \leq \left\langle \nabla W(\widehat{S}_{t,k}), \widehat{S}_{t,k+1} - \widehat{S}_{t,k} \right\rangle + \frac{L_{\dot{W}}}{2} \|\widehat{S}_{t,k+1} - \widehat{S}_{t,k}\|^2$$
$$\leq -\gamma_{t,k+1} v_{\min} \|H_{t,k+1}\|^2 + \gamma_{t,k+1} \left(\beta^2 v_{\max} + \gamma_{t,k+1} \frac{L_{\dot{W}}}{2} \right) \|H_{t,k+1}\|^2$$
$$+ \frac{\gamma_{t,k+1}}{\beta^2} v_{\max} \|H_{t,k+1} - \mathsf{h}(\widehat{S}_{t,k})\|^2 \qquad \forall \beta > 0; \text{choice: } \beta^2 \propto \gamma_{t,k+1}$$

- Biased field; full scan when refreshing $\rightarrow$ cancel the bias

$$\mathbb{E}\left[H_{t,k+1} | \mathcal{F}_{t,k}\right] = \mathsf{h}(\widehat{S}_{t,k}) + H_{t,k} - \mathsf{h}(\widehat{S}_{t,k-1}) \qquad \mathbb{E}\left[H_{t,k+1} | \mathcal{F}_{t,0}\right] = 0.$$

- L^2-error of the field

$$\mathbb{E}\left[\|H_{t,k+1} - \mathsf{h}(\widehat{S}_{t,k})\|^2 | \mathcal{F}_{t,0}\right] = \mathbb{E}\left[\|H_{t,k+1} - \mathbb{E}\left[H_{t,k+1} | \mathcal{F}_{t,k}\right]\|^2 | \mathcal{F}_{t,0}\right] + \mathbb{E}\left[\| \underbrace{\mathbb{E}\left[H_{t,k+1} | \mathcal{F}_{t,k}\right] - \mathsf{h}(\widehat{S}_{t,k})}_{H_{t,k} - \mathsf{h}(\widehat{S}_{t,k-1})} \|^2 | \mathcal{F}_{t,0}\right]$$

- Variance: specific form of $H_{t,k+1}$ $\rightarrow$ difference of h_i's

$$H_{t,k+1} - \mathbb{E}\left[H_{t,k+1} | \mathcal{F}_{t,k}\right] = \frac{1}{\mathsf{b}} \sum_{i \in \mathcal{B}_{t,k+1}} \{\mathsf{h}_i(\widehat{S}_{t,k}) - \mathsf{h}_i(\widehat{S}_{t,k-1})\} - \frac{1}{n} \sum_{i=1}^{n} \{\mathsf{h}_i(\widehat{S}_{t,k}) - \mathsf{h}_i(\widehat{S}_{t,k-1})\}$$

use: $\|\mathsf{h}_i(\widehat{S}_{t,k}) - \mathsf{h}_i(\widehat{S}_{t,k-1})\|^2 \leq L_i^2 \|\widehat{S}_{t,k} - \widehat{S}_{t,k-1}\|^2 = L_i^2 \gamma_{t,k}^2 \|H_{t,k}\|^2$

Assumptions (case: Monte Carlo approximation of h_i's)

In the case

$$\mathsf{h}_i(\widehat{S}_{t,k}) = \int \mathcal{H}(z) p_i(z; \widehat{S}_{t,k}) \mathsf{d}\mu(z) \approx \frac{1}{m_{t,k+1}} \sum_{r=1}^{m_{t,k+1}} \mathcal{H}(Z_r^{i,t,k})$$

error

$$\eta_{t,k+1} \stackrel{\text{def}}{=} \frac{1}{\mathsf{b}} \sum_{i \in \mathcal{B}_\bullet} \left(\frac{1}{m_{t,k+1}} \sum_{r=1}^{m_{t,k+1}} \mathcal{H}(Z_r^{i,t,k}) - \mathsf{h}_i(\widehat{S}_{t,k}) \right)$$

3. (bias) there exists $C_b \geq 0$ s.t. for any t, k, with probability one

$$\|\mathbb{E}\left[\eta_{t,k+1} | \mathcal{F}_{t,k}\right]\| \leq \frac{C_b}{m_{t,k+1}}$$

4. (variance) there exists C_v s.t. for any t, k with probability one

$$\mathbb{E}\left[\|\eta_{t,k+1} - \mathbb{E}\left[\eta_{t,k+1} | \mathcal{F}_{t,k}\right]\|^2 | \mathcal{F}_{t,k}\right] \leq \frac{C_v}{M_{t,k+1}}$$

Examples. i.i.d. case: $C_b = 0$; i.i.d. and MCMC cases: $M_{t,k+1} = \mathsf{b}\, m_{t,k+1}$

Convergence in expectation (i.i.d. case)

Fort, Moulines – SSP 2021; i.i.d. case and MCMC case

Choose $\xi_t = k_{\text{in}}$ *and* $\gamma_{t,k} = \gamma$ *where*

$$\gamma \stackrel{\text{def}}{=} \frac{v_{\min}}{L_{\dot{W}} + 2L v_{\max}\sqrt{k_{\text{in}}}/\sqrt{\mathsf{b}}}$$

Then

$$\gamma v_{\min} \mathbb{E}\left[\frac{\|\widehat{S}_{\tau,\xi} - \widehat{S}_{\tau,\xi-1}\|^2}{\gamma^2}\right] \leq \frac{1}{k_{\text{out}}(1+k_{\text{in}})}\left(W(\widehat{S}_{\text{init}}) - \min W\right) + C_1 \frac{v_{\max}}{L}\frac{1}{\sqrt{k_{\text{in}}\mathsf{b}}}\mathbb{E}\left[\frac{k_{\text{in}} - \xi}{m_{\tau,\xi+1}}\right]$$

where (τ, ξ) *is a uniform r.v. on* $\{1, \cdots, k_{\text{out}}\} \times \{0, \cdots, k_{\text{in}}\}$ *indep of* $\{\widehat{S}_{t,k}\}$.

From

$$\widehat{S}_{t,k+1} - \widehat{S}_{t,k} = \gamma_{t,k+1} H_{t,k+1} \neq \gamma_{t,k+1}\,\mathsf{h}(\widehat{S}_{t,k}),$$

a control is then obtained on $\mathbb{E}\left[\|\mathsf{h}(\widehat{S}_{\tau,\xi})\|^2\right]$

Complexity for ϵ-approximate stationarity

From this **explicit** expression of an upper bound for

$$\mathbb{E}\left[\|\mathsf{h}(\widehat{S}_{\tau,\xi-1})\|^2\right]$$

- in the `non convex` setting
- with a `random stopping rule`
- as a function of $k_{\text{out}}, k_{\text{in}}, \mathsf{b}, n$ and the learning rate γ
- with a Monte Carlo approximation of the h_i's

To reach ϵ-stationarity, the complexity of Perturbed-SPIDER-EM

With: $k_{\text{in}} = \mathsf{b} = O(\sqrt{n}), \quad k_{\text{out}} = O(1/(\epsilon k_{\text{in}})), \quad m_{t,k} = \epsilon^{-1}$

Nbr of optimization steps: $O(1/\epsilon)$
Nbr of $\bar{\mathsf{s}}_i$'s evaluations: $\mathcal{K} = O(\sqrt{n}\,\epsilon^{-1}) \rightarrow$ *same as SPIDER-EM*
Nbr of Monte Carlo draws: $O(\sqrt{n}/\epsilon^2)$

III. Numerical illustrations

SPIDER-EM: state-of-the-art among the incremental EM algorithms

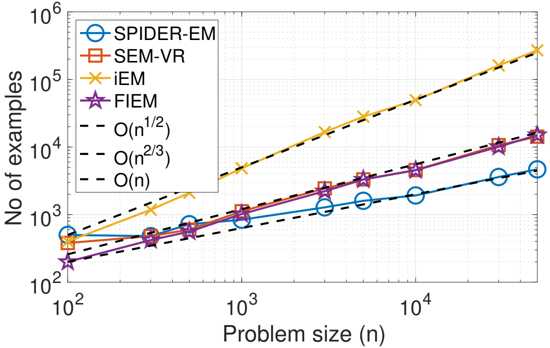

Figure: Nbr of processed examples required to reach convergence, as a function of the problem size n

Estimation of the parameters (1/2)

Case: inference in a mixture of Gaussian distributions (from the MNIST data set). Gaussian mixture models in $\mathbb{R}^{20}$; $G = 12$ components with the same cov matrix; $n = 6\,10^4$ examples

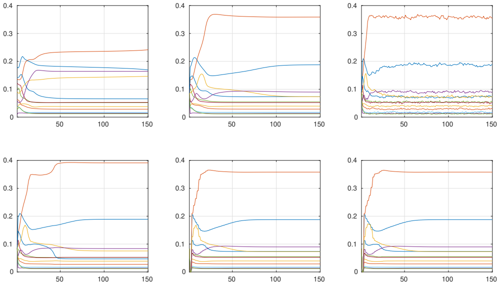

Figure: Evolution of the $L = 12$ iterates $\alpha_k = (\alpha_{k,1}, \ldots, \alpha_{k,L})$ as a function of the number of epochs, for `EM`, `iEM` and `Online EM` on the top from left to right; `FIEM`, `sEM-vr` and `SPIDER-EM` on the bottom from left to right.

Estimation of the parameters (2/2)

Case: inference in a mixture of Gaussian distributions (from the MNIST data set). Gaussian mixture models in $\mathbb{R}^{20}$; $G = 12$ components with the same cov matrix; $n = 6\,10^4$ examples

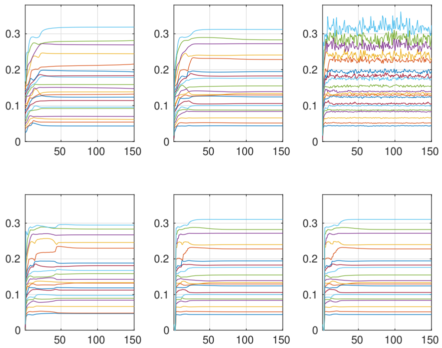

Figure: Evolution of the $p = 20$ eigenvalues of the iterates Σ_k as a function of the number of epochs, for `EM`, `iEM` and `Online EM` on the top from left to right; `FIEM`, `sEM-vr` and `SPIDER-EM` on the bottom from left to right.

Evolution of the objective function

Case: inference in a mixture of Gaussian distributions (from the MNIST data set). Gaussian mixture models in $\mathbb{R}^{20}$; $G = 12$ components with the same cov matrix; $n = 6\,10^4$ examples

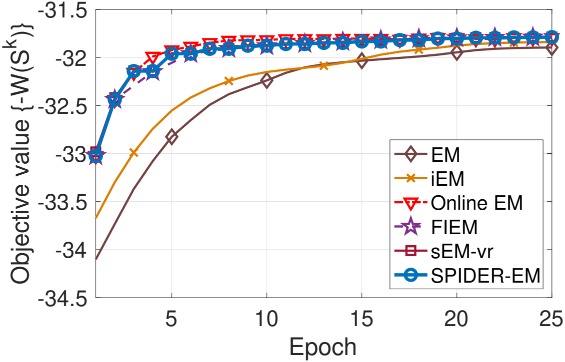

Figure: Evolution of the objective function $-W(\widehat{S}_k)$ vs the number of epochs.

Deterministic or geometric length of the outer loops? Full scan when refreshing ? (1/2)

Case: inference in a mixture of Gaussian distributions (from the MNIST data set). Gaussian mixture models in $\mathbb{R}^{20}$; $G = 12$ components with the same cov matrix; $n = 6\,10^4$ examples

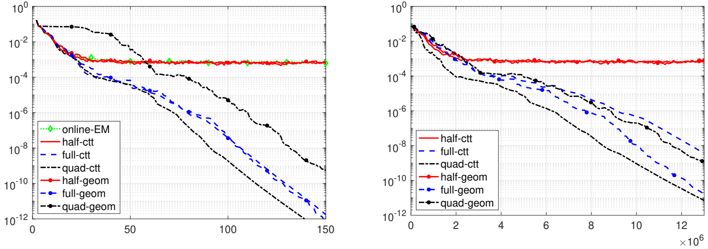

Figure: Quantile of order 0.5 of $\|\mathsf{h}(\widehat{S}_{t,\xi_t})\|^2$ vs the number of epochs (left) and vs the number of $\bar{\mathsf{s}}_i$'s evaluations (right)

Length of each outer loop: either constant (`ctt`) $\xi_t = k_{\text{in}}$, or a geometric r.v. (`geom`) with expectation k_{in}

When refreshing the control variate: use the full data set (`full`), or the half data set (`half`) or a quadratically increasing nbr of examples (`quad`).

Deterministic or geometric length of the inner loops? Full scan when refreshing ? (2/2)

Case: inference in a mixture of Gaussian distributions (from the MNIST data set). Gaussian mixture models in $\mathbb{R}^{20}$; $G = 12$ components with the same cov matrix; $n = 6\,10^4$ examples

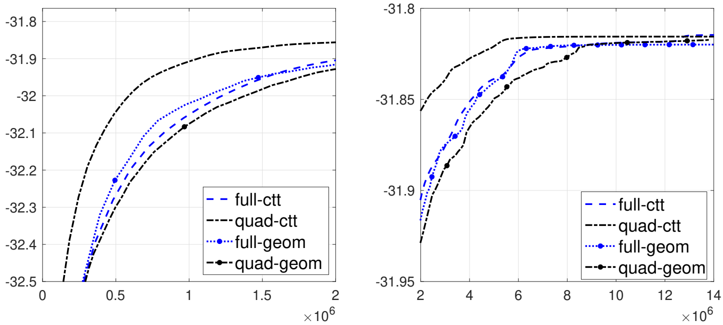

Figure: Evolution of the normalized log-likelihood vs the number of $\bar{s}_i$'s evaluations until $2e6$ (left) and after (right).

Monte Carlo approximations: benefit of variance reduction

Case: Ridge-penalized inference in a logistic regression model (from the MNIST data set). An individual regression vector $Z_i \in \mathbb{R}^{1+50}$ assumed i.i.d. $\mathcal{N}_{51}(\theta, 0.1\, I)$. $n = 24\,989$, 2 classes.

$$\Delta_{t,k+1} \stackrel{\text{def}}{=} \|\widehat{S}_{t,k+1} - \widehat{S}_{t,k}\|^2 / \gamma_{t,k+1}^2$$

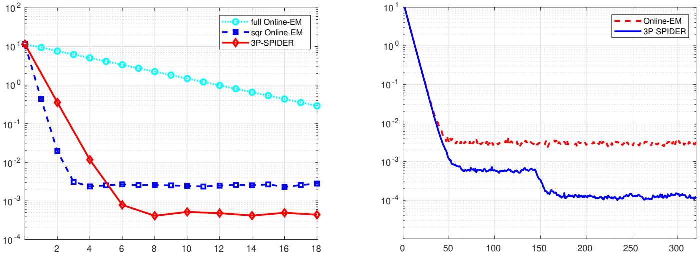

Figure: [left] Monte Carlo estimation of $\mathbb{E}\left[\Delta_{t,k+1}\right]$ vs the number of epochs. Comparison of (Perturbed-Proximal-Preconditioned) `3P-SPIDER-EM` and `Online-EM` when b $= n$ (case `full`) and b $= 10\sqrt{n}$ (case `sqr`). Monte Carlo approximations with $m_{t,k} = 2\sqrt{n}$. [right] Quantiles 0.75 of $\Delta_{t,k}$ vs the number of epochs, for `Online-EM` and `3P-SPIDER-EM`. For `3P-SPIDER-EM` $m_{t,k} = 2\sqrt{n}$ for $t \leq 9$ and $m_{t,k} = 10\sqrt{n}$ for $t \geq 10$.

Monte Carlo approximations: number of points in the Monte Carlo sum

Case: Ridge-penalized inference in a logistic regression model (from the MNIST data set). An individual predictor vector $Z_i \in \mathbb{R}^{1+50}$ assumed i.i.d. $\mathcal{N}_d(\theta, 0.1\, I)$. $n = 24\,989$, 2 classes.

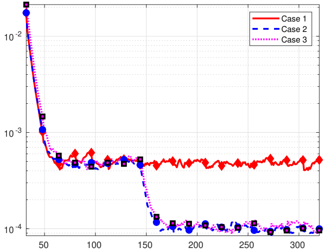

Figure: Monte Carlo estimation of $\mathbb{E}\left[\Delta_{t,k+1}\right]$ vs the number of epochs. `(Perturbed-Proximal-Preconditioned) SPIDER-EM` applied with $\gamma_{t,k} = 0.1$ and $m_{t,k} = 2\sqrt{n}$ in `Case 1`; and with $\gamma_{t,k} = 0.1$ and $m_{t,k} = 2\sqrt{n}$ for $t \leq 10$ and $m_{t,k} = 10\sqrt{n}$ for $t \geq 11$ on `Case 2` and `Case 3`. `Case 2` and `Case 3` differ in the choice of $\gamma_{t,0}$

IV. Bibliography

Results of this talk

- **G. Fort, E. Moulines, H.-T. Wai.** A Stochastic Path Integrated Differential Estimator Expectation Maximization Algorithm. *In Conference Proceedings NeurIPS, 2020.*
- **G. Fort, E. Moulines, H.-T. Wai**. Geom-SPIDER-EM: Faster Variance Reduced Stochastic Expectation Maximization for Nonconvex Finite-Sum Optimization, *ICASSP 2021 – 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP):3135–3139.*
- **G. Fort and E. Moulines.** The Perturbed Prox-Preconditioned SPIDER algorithm: non-asymptotic convergence bounds. *Accepted to IEEE Statistical Signal Processing Workshop (SSP 2021).*
- **G. Fort and E. Moulines.** The Perturbed Prox-Preconditioned SPIDER algorithm for EM-based large scale learning. *Accepted to IEEE Statistical Signal Processing Workshop (SSP 2021)*

Other references

- Benveniste, A. and Métivier, M. and Priouret P. Adaptive Algorithms and Stochastic Approxima-tions. Springer Verlag, 1990.
- Cappé, O. and Moulines, E. On-line expectation-maximization algorithm for latent data models. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 71(3):593–613, 2009.
- Chen, J. Zhu, Y. Teh, and T. Zhang. Stochastic Expectation Maximization with variance reduction. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Gar-nett, editors, *Advances in Neural Information Processing Systems 31*, pages 7967–7977. 2018.
- Dempster, A.P. and Laird, N.M. and Rubin, D.B. Maximum Likelihood from Incomplete Data Via the EM Algorithm *Journal of the Royal Statistical Society: Series B (Methodological)*, 1977.
- Fang, C. and Li, C. and Lin, Z. and Zhang, T. SPIDER: Near-Optimal Non-Convex Optimization viaStochastic 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.
- Fort, G. and Gach, P. and Moulines, E. The Fast Incremental Expectation Maximization for finite-sum optimization: asymptotic convergence, *Statistics and Computing*, 2021.
- Karimi, B. and Wai, H.-T., and Moulines, E. and Lavielle, M. On the Global Convergence of (Fast) In-cremental Expectation Maximization Methods. In H. Wallach, H. Larochelle, A. Beygelzimer,F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information ProcessingSystems 32*, pages 2837–2847. Curran Associates, Inc., 2019.
- Neal, R.M. and Hinton, G.E. A View of the EM Algorithm thatJustifies Incremental, Sparse,and other Variants. In M. I. Jordan, editor, *Learning in Graphical Models*, pages 355–368. Springer Netherlands, Dordrecht, 1998.
- Nguyen, L.M. and Liu, K. and Scheinberg,K. and Takác M. SARAH: A novel method for machine learning problems using stochastic recursive gradient. *In Proceedings of the 34th International Conference on Machine Learning* - Volume 70, ICML'17, page 2613–2621. 2017
- Robbins, H. and Monro, S.. A Stochastic Approximation Method. *The Annals of Mathematical Statistics*. 22 (3): 400, 1951.
- Wang, Z. and Ji, K. and Zhou, Y. and Liang, Y. and and Tarokh, V. SpiderBoost and Momentum: Faster Stochastic Variance Reduction Algorithms. 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 2406–2416. 2019.