Distributed and Private Stochastic EM Methods via Quantized and Compressed MCMC

Anonymous Author(s)
Affiliation
Address
email

Abstract

To be completed

1 Introduction

We consider the distributed minimization of the following negated log incomplete data likelihood

$$\min_{\theta \in \Theta} \overline{L}(\theta) := L(\theta) + r(\theta) \quad \text{with } L(\theta) = \frac{1}{n} \sum_{i=1}^{n} L_i(\theta) := \frac{1}{n} \sum_{i=1}^{n} \left\{ -\log g(y_i;\theta) \right\} , \tag{1}$$

where n denotes the number of workers, $\{y_i\}_{i=1}^n$ are observations, $\theta \subset \mathbb{R}^d$ is the parameters set and $\mathrm{R} : \theta \to \mathbb{R}$ is a smooth regularizer.

The objective $L(\theta)$ is possibly nonconvex and is assumed to be lower bounded. In the latent data model, the likelihood $g(y_i;\theta)$, is the marginal distribution of the complete data likelihood, noted $f(z_i, y_i; \theta)$, such that

$$g(y_i;\theta) = \int_{\mathsf{Z}} f(z_i, y_i; \theta) \mu(\mathrm{d}z_i), \tag{2}$$

where $\{z_i\}_{i=1}^n$ are the vectors of latent variables associated to the observations $\{y_i\}_{i=1}^n$.

We also consider a special case of that problem since the complete likelihood pertains to the curved exponential family:

$$f(z_i, y_i; \theta) = h(z_i, y_i) \exp\left(\langle S(z_i, y_i) \,|\, \phi(\theta) \rangle - \psi(\theta) \right) , \tag{3}$$

where $\psi(\theta)$, $h(z_i, y_i)$ are scalar functions, $\phi(\theta) \in \mathbb{R}^k$ is a vector function, and $\{S(z_i, y_i) \in \mathbb{R}^k\}_{i=1}^n$ is the vector of sufficient statistics. We refer the readers to [Efron, 1975] for details on this subclass of problems which is of high interest given the broad range of problems that fall under this assumption. In the centralized settings, *i.e.,* when all data points are stored in a central server, a reference tool for learning such a model is called the EM algorithm [Dempster et al., 1977, Wu, 1983]. Comprised of two steps, the E-step computes an aggregated sum of expectations as follows:

$$\overline{s}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \overline{s}_i(\theta) \quad \text{where} \quad \overline{s}_i(\theta) := \int_{\mathsf{Z}} S(z_i, y_i) p(z_i | y_i; \theta) \mathrm{d}z_i , \tag{4}$$

and the M-step is given by

$$\overline{\theta}(\overline{s}(\theta)) := \arg\min_{\vartheta \in \theta} \left\{ r(\vartheta) + \psi(\vartheta) - \langle \overline{s}(\theta) \,|\, \phi(\vartheta) \rangle \right\} . \tag{5}$$

Submitted to 34th Conference on Neural Information Processing Systems (NeurIPS 2020). Do not distribute.

1.1 Our motivations

Expectations are not tractable: Sampling for those approximations are costly.

Need for distributed computing: MovieLens, Large n, compute time, decentralized infrastructure

Need for privacy and communication efficiency: Sensible data (hospital, user data...) that can not be moved. Low bandwidth devices (compute should be light).

1.2 Our contributions

2 Related Work

EM algorithms:

Distributed methods:

MCMC and Quantization:

Federated Learning methods:

3 On the Decentralization of the EM algorithm

3.1 Distributed SAEM

We first consider the plain distributed version of the sEM which does not tackle any privacy or communication bottlenecks. We precise that we perform periodic locals models averaging. It goes as follows:

Algorithm 1 Distributed SAEM with Periodic Locals Models Averaging

1: **Input**: Compression operator $\mathcal{C}(\cdot)$, number of rounds R, initial parameter θ_0.
2: **for** $r = 1$ to R **do**
3: **for** parallel for device $i \in D^r$ **do**
4: Set $\hat{\theta}_i^{(r)} = \hat{\theta}^{(r)}$. {Initialize each worker with current global model}
5: Draw M samples $z_{i,m}^{(r+1)}$ under model $\hat{\theta}_i^{(r)}$ via MCMC: {Local MCMC step}
6: Compute the local statistics $\tilde{S}_i^{(r+1)} = S(z_{i,m}^{(r+1)})$. {Local statistics}
7: Worker computes **local model**: {(Local) M-Step using local statistics}

$$\hat{\theta}_i^{(r+1)} = \overline{\theta}(\tilde{S}_i^{(r+1)})$$

8: Worker sends local model $\hat{\theta}_i^{(r+1)}$ to server.
9: **end for**
10: Server computes **global model** by periodic averaging {Local model averaging}

$$\hat{\theta}^{(r+1)} := \frac{1}{n} \sum_{i=1}^{n} \hat{\theta}_i^{(r+1)}$$

11: **end for**

3.2 Federated SAEM with Quantization and Compression

While Algorithm 2 is a distributed variant of the SAEM, it is neither (a) *p*rivate nor (b) *c*ommunication-efficient.

Privacy: Indeed, we remark that broadcasting the vector of statistics are a potential breach to the data observations as their expression is related y and the latent data z. With a simple knowledge of the model used, the data could be retrieved if one extracts those statistics.

Communication bottlenecks: Also regarding (b), the broadcast of n vector of statistics $S(y_i, z_i)$ can be cumbersome when the size of the latent space and the parameter space of the model are huge.

For computational purposes and privacy enhanced matter, I have chosen to study and develop the second algorithms that I proposed in my last week's report. In that algorithm, one does not compute a periodic averaging of the local models (this would requires performing as many M-steps as there are workers). Rather, workers compute local statistics and send them to the central server for a periodic averaging of those vectors and the latter computes one M-step to update the global model.

Algorithm 2 FL-SAEM with Periodic Statistics Averaging

1: **Input**: TO COMPLETE
2: Init: $\theta_0 \in \Theta \subseteq \mathbb{R}^d$, as the global model and $\bar{\theta}_0 = \frac{1}{n}\sum_{i=1}^n \theta_0$.
3: **for** $r = 1$ to R **do**
4: **for** parallel for device $i \in D^r$ **do**
5: Set $\hat{\theta}_i^{(0,r)} = \hat{\theta}^{(r)}$.
6: Draw M samples $z_{i,m}^{(r)}$ under model $\hat{\theta}_i^{(r)}$
7: Compute the surrogate sufficient statistics $\tilde{S}_i^{(r+1)}$
8: Workers send local statistics $\tilde{S}_i^{(k+1)}$ to server.
9: **end for**
10: Server computes **global model using the aggregated statistics**:

$$\hat{\theta}^{(r+1)} = \overline{\theta}(\tilde{S}^{(r+1)})$$

where $\tilde{S}^{(r+1)} = (\tilde{S}_i^{(r+1)}, i \in D_r)$ and send global model back to the devices.
11: **end for**

3.3 Embedded Engines to comply with Federated settings

Line 6 – Quantization: The first step is to quantize the gradient in the Stochastic Langevin Dynamics step used in our sampling scheme Line 6 of Algorithm 2. Inspired by [Alistarh et al., 2017], we use an extension of the QSGD algorithm for our latent samples. Define the quantization operator as follows:

$$\mathsf{C}_j^{(\ell)}(g, \xi_j) = \|v\| \cdot \operatorname{sign}(g_j) \cdot \left(\lfloor \ell\,|g_j| / \|v\| \rfloor + \mathbf{1}\left\{\xi_j \leq \ell\,|g_j| / \|v\| - \lfloor \ell\,|g_j| / \|v\| \rfloor\right\}\right) / \ell \tag{6}$$

where ℓ is the level of quantization and $j \in [d]$ denotes the dimension of the gradient.

Hence, for the sampling step, Line 6, we use the modified SGLD below, to be compliant with the privacy of our method.

Algorithm 3 Langevin Dynamics with Quantization for worker i

1: **Input**: Current local model $\hat{\theta}_i^{(r)}$ for worker $i \in [\![1, n]\!]$.
2: Draw M samples $\{z_i^{(r,m)}\}_{m=1}^M$ from the posterior distribution $p(z_i|y_i; \hat{\theta}_i^{(k)})$ via Langevin diffusion with a quantized gradient:
3: **for** $k = 1$ to K **do**
4: Compute the quantized gradient of $\nabla \log p(z_i|y_i; \hat{\theta}_i^{(k)})$:

$$g_i(k, m) = \mathsf{C}_j^{(\ell)}\left(\nabla_j f_{\theta_t}(z_i^{(k-1,m)}), \xi_j^{(k)}\right) \tag{7}$$

where $\xi_j^{(k)}$ is a realization of a uniform random variable.
5: Sample the latent data using the following chain:

$$z_i^{(k,m)} = z_i^{(k-1,m)} + \frac{\gamma_k}{2} g_i(k, m) + \sqrt{\gamma_k}\mathsf{B}_k\ , \tag{8}$$

where B_t denotes the Brownian motion and $m \in [M]$ denotes the MC sample.
6: **end for**
7: Assign $\{z_i^{(r,m)}\}_{m=1}^M \leftarrow \{z_i^{(K,m)}\}_{m=1}^M$.
8: **Output:** latent data $z_{i,m}^{(k)}$ under model $\hat{\theta}_i^{(t,k)}$

Line 7 – Compression MCMC output: We use the notorious **Top-k** operator that we define as $\mathcal{C}(x)_i = x_i$, if $i \in \mathcal{S}$; $\mathcal{C}(x)_i = 0$ otherwise and where $\mathcal{S}$ is defined as the size-k set of $i \in [p]$. Recall that after Line 6 we compute the local statistics $\tilde{S}_i^{(k+1)}$ using the output latent variables from Algorithm 3. We now use those statistics and compress them using Algorithm 4 as follows:

Algorithm 4 Sparsified Statistics with **Top-k**

Input: Current local statistics $\tilde{S}_i^{(k+1)}$ for worker $i \in [\![1, n]\!]$. Sparsification level k.
Apply **Top-k**:

$$\ddot{S}_i^{(k+1)} = \mathcal{C}\left(\tilde{S}_i^{(k+1)}\right) \tag{9}$$

Output: Compressed local statistics for worker i denoted $\ddot{S}_i^{(k+1)}$.

We present our final method in Algorithm 5, that performs SAEM under the federated settings.

Algorithm 5 Quantized and Compressed FL-SAEM with Periodic Statistics Averaging

Input: Compression operator $\mathcal{C}(\cdot)$, number of rounds R, initial parameter θ_0.
for $r = 1$ to R **do**
  for parallel for device $i \in D^r$ **do**
    Set $\hat{\theta}_i^{(0,r)} = \hat{\theta}^{(r)}$. {Initialize each worker with current global model}
    Draw M samples $z_{i,m}^{(r)}$ under model $\hat{\theta}_i^{(r)}$ via Quantized LD: {Local Quantized MCMC step}
    for $k = 1$ to K **do**
      Compute the quantized gradient of $\nabla \log p(z_i | y_i; \hat{\theta}_i^{(k)})$:

$$g_i(k, m) = \mathsf{C}_j^{(\ell)}\left(\nabla_j f_{\theta_t}(z_i^{(k-1,m)}), \xi_j^{(k)}\right) \quad \text{where} \quad \xi_j^{(k)} \sim \mathcal{U}_{[a,b]}$$

      Sample the latent data using the following chain:

$$z_i^{(k,m)} = z_i^{(k-1,m)} + \frac{\gamma_k}{2} g_i(k, m) + \sqrt{\gamma_k}\mathsf{B}_k,$$

      where B_t denotes the Brownian motion and $m \in [M]$ denotes the MC sample.
    end for
    Assign $\{z_i^{(r,m)}\}_{m=1}^M \leftarrow \{z_i^{(K,m)}\}_{m=1}^M$.
    Compute $\tilde{S}_i^{(r+1)}$ and its **Top-k** variant $\ddot{S}_i^{(r+1)} = \mathcal{C}\left(\tilde{S}_i^{(r+1)}\right)$. {Compressed local statistics}
    Worker send local statistics $\tilde{S}_i^{(r+1)}$ to server. {Single round of communication}
  end for
  Server computes **global model**: {(Global) M-Step using aggregated statistics}

$$\hat{\theta}^{(r+1)} = \overline{\theta}(\ddot{S}^{(r+1)})$$

  where $\ddot{S}^{(r+1)} = (\ddot{S}_i^{(r+1)}, i \in D_r)$ and send global model back to the devices.
end for

4 Theoretical Analysis

5 Numerical Experiments

5.1 Nonlinear Mixed Models under Distributed Settings

Compare SAEM, MCEM, dist-SAEM and maybe one distributed Gradient Descent as baseline

Same for Private settings with Sketched SGD or another good baseline

Fitting a linear mixed model on Oxford boys dataset [Pinheiro and Bates, 2006]

Fitting a nonlinear mixed model on Warfarin dataset [Consortium, 2009]

5.2 Probabilistic Latent Dirichlet Allocation

5.3 Bi-factor models under the Federated Learning settings

6 Conclusion

References

D. Alistarh, D. Grubic, J. Li, R. Tomioka, and M. Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, pages 1709–1720, 2017.

I. W. P. Consortium. Estimation of the warfarin dose with clinical and pharmacogenetic data. *New England Journal of Medicine*, 360(8):753–764, 2009.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the royal statistical society. Series B (methodological)*, pages 1–38, 1977.

B. Efron. Defining the curvature of a statistical problem (with applications to second order efficiency). *The Annals of Statistics*, 3(6):1189–1242, 1975.

J. Pinheiro and D. Bates. *Mixed-effects models in S and S-PLUS*. Springer Science & Business Media, 2006.

C. J. Wu. On the convergence properties of the EM algorithm. *The Annals of statistics*, pages 95–103, 1983.