Weekly Report KARIMI 2021-07-16

This week, I have mainly focussed my work towards developing a Federated EM algorithm. Two settings are possible:

- The expectations are tractable and we want to scale to large datasets with a random data index sampling while being distributed and private (this would be a sEM method, where sEM stands for Stochastic EM).
- The expectations are not tractable and thus we would use the SAEM under the FL settings (this is the setting of the my talk from last week).

1 SAEM for Federated Learning

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 1 FL-SAEM with statistics averaging

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

1.1 Challenges with Algorithm 1

While Algorithm 1 is a distributed variant of the SAEM, it is neither (a) $p$rivate nor (b) $c$ommunication-efficient. 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. 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.

I am incorporating respective solutions to those problems below.

1.2 Algorithmic solutions

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 1. Inspired by [1], we use an extension of the QSGD algorithm for our latent samples. Define the quantization operator as follows:

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

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 2 Langevin Dynamics with Quantization for worker i

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

$$g_i(k, m) = \mathsf{C}_j^{(\ell)}\left(\nabla_j f_{\theta_t}(z_i^{(k-1,m)}), \xi_j^{(k)}\right) \tag{2}$$

where $\xi_j^{(k)}$ is a realization of a uniform random variable.
5: Construct the Markov Chain:

$$z_i^{(k,m)} = z_i^{(k-1,m)} + \frac{\gamma_k}{2} g_i(k, m) + \sqrt{\gamma_k}\mathsf{B}_k\ , \tag{3}$$

where B_t denotes the Brownian motion.
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 the set of size $k < p$. Recall that after Line 6 we compute the local statistics $\tilde{S}_i^{(k+1)}$ using the output latent variables from Algorithm 2. We now use those statistics and compress them using Algorithm 3 as follows:

Algorithm 3 Sparsified Statistics with **Top-k**

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

$$\ddot{S}_i^{(k+1)} = \textbf{Top-}k\left(\tilde{S}_i^{(k+1)}\right) \tag{4}$$

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

References

[1] Dan Alistarh, Demjan Grubic, Jerry Li, Ryota Tomioka, and Milan Vojnovic. Qsgd: Communication-efficient sgd via gradient quantization and encoding. In *Advances in Neural Information Processing Systems*, pages 1709–1720, 2017.