Federated Uncertainty Quantification: a survey

Eric Moulines

March 2023

Many machine learning applications require training a centralized model on decentralized, heterogeneous, and potentially private data sets. Federated learning (FL, McMahan et al., 2017; Kairouz et al., 2021; Wang et al., 2021) has emerged as a privacy-friendly training paradigm that does not require clients' private data to leave their local devices. FL brings new challenges in addition to "traditional" distributed learning: expensive communication, statistical heterogeneity, partial participation, and privacy (Li et al., 2020a).

The "classical" formulation of FL treats it as a distributed optimization problem where the model parameters θ are trained on K private data sets $\mathcal{D} = \bigcup_{k\in[K]} \mathcal{D}_k$,

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} L(\boldsymbol{\theta}), \quad \text{where} \quad L(\boldsymbol{\theta}) = \sum_{k\in[K]} -\log p\left(\mathcal{D}_k \mid \boldsymbol{\theta}\right).$$

Standard distributed optimization algorithms (e.g., data-parallel SGD) are too communication-intensive to be practical at FL. Federated Averaging (FedAvg, McMahan et al., 2017) reduces communication costs by allowing clients to perform multiple local SGD steps/epochs before parameter updates are sent back to the central server and aggregated.

An alternative approach is to consider a Bayesian formulation of the FL problem. In this setting, the goal is to estimate the posterior of parameters $p(\boldsymbol{\theta} \mid \mathcal{D})$ when a prior $p(\boldsymbol{\theta})$ (e.g. an improper uniform or a Gaussian prior) and a collection of client likelihoods $p\left(\mathcal{D}_k \mid \boldsymbol{\theta}\right)$ are given, which are independent of the model parameters,

$$p(\boldsymbol{\theta} \mid \mathcal{D}) \propto p(\boldsymbol{\theta}) \prod_{k\in[K]} p\left(\mathcal{D}_k \mid \boldsymbol{\theta}\right)$$

In this case, the posterior naturally factors over partitioned client data, with the global posterior corresponding to a multiplicative aggregate of local factors (and the prior). However exact posterior inference is untractable even for models and data sets of modest size. Approximate inference methods should therefore be considered.

In this talk, we will two approximate inference approaches:

- Markov Chain Monte Carlo. Among the many methods which have been proposed, we will concentrate on the Federated Averaging Langevin Dynamics (FALD), studied in Plassier et al. (2023) . FALD, proposed in Deng et al. (2021), is an extension to the Bayesian setting of FEDAvG (McMahan et al., 2017). The updates performed on the i th client define a sequence of local parameters which are transmitted according to some preset schedule to a central server. The central server averages the local parameters to update the global parameter. This global parameter is finally transmitted back to each client, and is used as a starting point of a new round of local interactions. To mitigate the impact of local stochastic gradients, we adapt variance-reduction techniques (Wang et al., 2013; Kovalev et al., 2020) and bias-reduction techniques (Horváth et al., 2022; Gorbunov et al., 2021). The local update rule is based on a reference point common to all clients. This mechanism eliminates the "infamous non-stationarity of the local methods" (paraphrasing Gorbunov et al. (2021)) and therefore avoids extra bias.

- Variational inference methods. In this setting, the solution of federated learning is obtained as a mode of a variational (posterior) distribution $q \in \mathcal{Q}$ with a divergence function $D(\cdot\|\cdot)$ (e.g., KL -divergence),

$$\boldsymbol{\theta} = \arg\max_{\boldsymbol{\theta}} q(\boldsymbol{\theta}), \qquad \text{where } q(\boldsymbol{\theta}) = \arg\min_{q \in \mathcal{Q}} D(p(\boldsymbol{\theta} \mid \mathcal{D}) \| q(\boldsymbol{\theta})).$$

  In this approach, clients use local computations to perform posterior inference (instead of parameter/gradient estimates) in parallel. In turn, fewer lockstep synchronization and communication steps may be required between clients and servers.