Exam of the Course: SOD314

26 March 2023

Notes: this exam lasts 2 hours, you can use the lecture notes that were distributed during the classes. There are 2 exercises. You can refer to Theorems that are present in the notes by indicating their number and page. In the interest of time, keep your answers concise and to the point.

Exercise 1. Overlapping images

In a modern medical imaging techniques, several sensors take partial and overlapping images of the same body element. Since the amount of data that each sensor receives is big, there is much research around how one could find the overall image by distributed optimization methods.

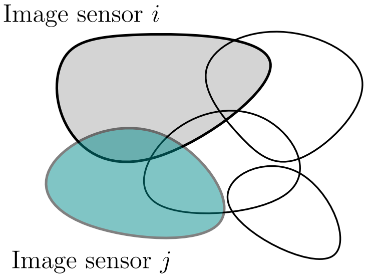

Figure 1: Partial images and their overlaps.

Consider Figure 1, where we have indicated the partial images of sensors i and j, and their overlap. Consider the measurements for sensor i as $y_i \in \mathbf{R}^{m_i}$ and the decision variables $x_i \in \mathbf{R}^{n_i}$. The problem can be modeled as solving the optimization problem involving N sensors,

$$\min_{x_i \in \mathbf{R}^{n_i}, i=1,\ldots,N} \quad \frac{1}{2}\sum_{i=1}^{N} \|y_i - C_i x_i\|_2^2, \qquad \text{subject to: } V^{ij}x_i = V^{ji}x_j, \quad \forall i \sim j,$$

where $C_i \in \mathbf{R}^{m_i \times n_i}$ is a matrix ($m_i \leq n_i$), and matrices $V^{ij} \in \mathbf{R}^{\ell_{ij} \times n_i}$, $V^{ji} \in \mathbf{R}^{\ell_{ij} \times n_j}$ are selection matrices that describe the overlap between decision x_i and x_j. Finally $i \sim j$ tells which sensor has overlaps with which other one. We assume that if two sensors have overlaps, then they can message each other.

1. Describe the problem: is it convex? Is the cost smooth? Strongly convex?
2. Start by removing the constraints and adding to the cost the penalty $\sum_{i\sim j} \frac{1}{2\alpha}\|V^{ij}x_i - V^{ji}x_j\|^2$ and derive a first-order decentralized gradient descent scheme.
3. Describe the decentralized method that you have obtained in the previous point. In particular, how many communication rounds per iteration you need? What do you exchange? Can you derive a convergence result, and if so, is it to the true optimizer or to an approximate one?
4. Consider now using the ADMM algorithm on the original problem. Derive a peer-to-peer ADMM algorithm that can solve the problem at optimality. Describe how many communication rounds per iteration you need and what you exchange.
5. Re-consider now using the ADMM algorithm, but now in a cloud-based setting. Let $w \in \mathbf{R}^n$ be the global decision variable vector and model the problem as solving,

   $$\min_{x_i \in \mathbf{R}^{n_i}, i=1,\ldots,N} \quad \frac{1}{2}\sum_{i=1}^{N} \|y_i - C_i x_i\|_2^2 + \nu\|w\|_1, \qquad \text{subject to: } x_i = U^i w, \quad \forall i,$$

   where $U^i \in \mathbf{R}^{n_i \times n}$ is a selection matrix that tells which part of w is x_i. Here we have added a regularization $\nu\|w\|_1$, with $\nu > 0$.

   Derive a cloud-based ADMM to solve the above problem, in which the sensors communicate all to a server. Describe how many communication rounds per iteration you need and what you exchange. What about convergence?

Exercise 2. Language models

Consider a federated learning setting, where a number of users use their phones to text messages. The phones record the messages to train a model to perform next-word prediction.

Phones in nearby geographical locations are connected to the same server and employ the same model architecture. Let $\ell_p(x;\theta_i)$ be the loss incurred by phone p, when a message θ_i is revealed, and a next-word prediction x is used. As in class, $x \in \mathbf{R}^n$ are the model weights, or loosely "the model". The aim is to find the best x sharing information with the server.

Let the loss $\ell_p(x;\theta_i) : \mathbf{R}^n \times \mathbf{R}^m \to \mathbf{R}$ be strongly convex and smooth. The messages $\theta_i \in \mathbf{R}^m$ can be thought of as random vectors drawn from a certain probability distribution Θ_p.

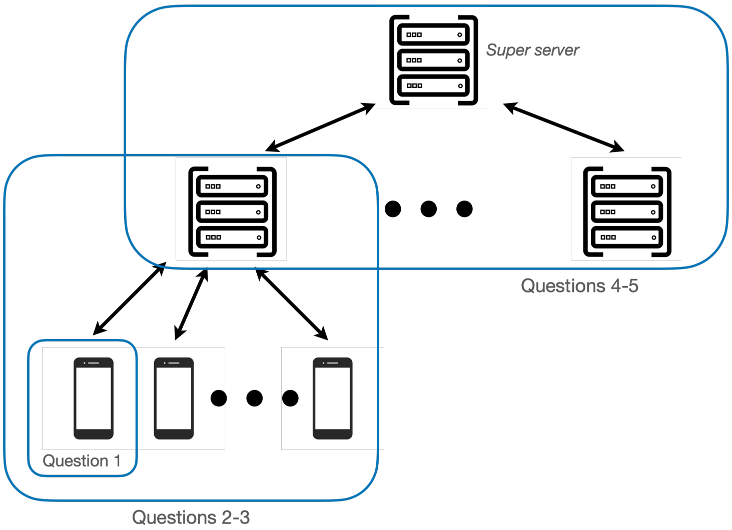

Figure 2: A depiction of this problem and the different questions.

1. Consider the p phone alone. Describe the best stochastic method we have seen to solve the training problem,
$$\min_{x\in\mathbf{R}^n} \mathbf{E}_{\theta_i\sim\Theta_p}[\ell(x;\theta_i)].$$
Detail the assumptions as well as the theoretical guarantees.

2. Consider now the phones collaborating with one server. Describe the FedAvg algorithm that each phone could use to train its model (i.e., to learn the best x) by collaborating with the server. Detail the assumptions as well as the theoretical guarantees. Is this a cross-device or cross-silo setting?

3. Describe a method to ensure privacy at the phone level when employing FedAvg. What is a (ϵ,δ)-differential privacy and what is a Gaussian mechanism?

4. The server is now collaborating to other servers to further improve the model. Other servers may be in other geographical locations and the probability distributions Θ_p may be very different. Describe a federated algorithm that can help the servers to collaborate to deliver the best model. In this setting, assume that you can have a *super*-server that coordinate all the servers.

5. Each server works on very special geographical locations and dialects. As such, a global model may not be well suited to its need. Describe a method to fine-tune (i.e., personalize) the global model for different servers.