

Notation and Conventions

General

- If $f = g + h$ and g is differentiable but h is non-differentiable, overload the ∇ operator by defining $\nabla f := \nabla g$.
- $\|\cdot\|$: Euclidean norm
- $\angle(v, w)$: the angle between vectors v and w

Non-Federated Sparse Linear Regression

- (X, Y) : training dataset
- λ : L1 regularization parameter
- β : model
- $f(\beta) := f(X, Y, \lambda, \beta)$: objective function of sparse linear regression
- $\beta^* := \arg \min_{\beta} f(\beta)$: optimal model
- β^s : model at s -th iteration
- η : step-size
- $S_{\lambda\eta}(\cdot)$: soft-thresholding operator

Federated Sparse Linear Regression

- K : number of clients
- G : communication graph
- $\text{ne}(i)$: neighborhood of i -th client
- $K_i := 1 + |\text{ne}(i)|$: the number of neighbors of the i -th client including itself
- Let the index i range over $\{1, \dots, K\}$, and the index j range over $\{i\} \cup \text{ne}(i)$.
- (X_i, Y_i) : i -th private training dataset
- $(X, Y) := \bigcup_i (X_i, Y_i)$: joint training dataset
- $f_i(\beta) := f(X_i, Y_i, \lambda, \beta)$: i -th objective function
- β_i : i -th local model

Problem Statement

Non-Federated Sparse Linear Regression

The most basic setting is the non-federated setting, in which a single client possesses a training dataset (X, Y) and wants to find an optimal model $\beta^* := \arg \min_{\beta} f(X, Y, \lambda, \beta)$. For conciseness we can suppress X and Y and simply write the objective function as $f(\beta)$. The objective function $f(\beta)$ is the objective function for sparse linear regression.

Federated Sparse Linear Regression

More generally, in the federated setting there are K clients, each possessing a private training dataset (X_i, Y_i) . Define the joint training dataset $(X, Y) = \bigcup_i (X_i, Y_i)$. The clients want to find an optimal model $\beta^* = \arg \min_{\beta} f(X, Y, \lambda, \beta) = \arg \min_{\beta} f(\beta)$. To collaborate, the clients broadcast messages to their neighbors on a graph G . These messages must not include private training datasets, but can include models and gradients.

Federated Sparse Linear Regression with Poisoning Attacks

Even more generally, assume that all clients are either benign or adversarial. Benign clients want to recover the optimal model as before, but adversaries will intentionally broadcast incorrect models and/or gradients to their neighbors in order to prevent convergence.

Algorithms

Proximal Gradient Descent

All algorithms to follow will generalize proximal gradient descent, which solves non-federated sparse linear regression and has the following steps:

Gradient (G): $\beta_i \leftarrow \beta_i - \eta \nabla f_i(\beta_i)$.

Threshold (T): $\beta_i \leftarrow S_{\lambda\eta}(\beta_i)$.

Federated Proximal Gradient Descent

The first generalization of proximal gradient descent is to the federated setting. This requires two new kinds of steps: a broadcasting step:

Broadcast (B): $\{\beta_i, \nabla f_i(\beta_i)\} \mapsto j$

and at least one aggregation step:

Consensus (C): $\beta_i \leftarrow \beta_i + \frac{\eta}{K_i} \sum (\beta_j - \beta_i)$

Aggregate (A): $\beta_i \leftarrow \beta_i - \frac{\eta}{K_i} \sum \nabla f_j(\beta_j)$

Aggregate+ (A+): $\beta_i \leftarrow \beta_i - \frac{\eta}{K_i} \sum \nabla f_j(\beta_i)$.

The aggregation steps can be executed sequentially or simultaneously. For example, there are the steps

(AC): $\beta_i \leftarrow \beta_i - \frac{\eta}{2K_i} \sum (\nabla f_j(\beta_j) - (\beta_j - \beta_i))$

(GC): $\beta_i \leftarrow \beta_i - \frac{\eta}{2K_i} (K_i \nabla f_i(\beta_i) - \sum (\beta_j - \beta_i))$.

Stipulating some assumptions, we can constrain the set of possible algorithms solving federated sparse linear regression.

Assumption: Output must be sparse.

1. T occurs exactly once and occurs last.

Assumption: No more gradients than necessary.

2. A+ is banned.
3. Exactly one of A and G occurs.

Assumption: No more broadcast rounds than necessary.

4. If A or C occurs, then B occurs exactly once and occurs directly before the earliest A or C.
5. C occurs at most once.

Assumption: Don't use information that is out of date.

6. AC and CA are banned.

Thus, the complete list of methods: G, A, (AC), CG, GC, (GC). (Note: in the written qualifying exam, the algorithm A+ was investigated, which we are excluding from this list because it is expensive in both computation and communication).

Name	Rule	Cost	"Informativeness"
G	$\beta_i \leftarrow \beta_i - \eta \nabla f_i(\beta_i)$	Low	Low
C	$\beta_i \leftarrow \beta_i + \frac{\eta}{K_i} \sum (\beta_j - \beta_i)$	Medium	Medium
A	$\beta_i \leftarrow \beta_i - \frac{\eta}{K_i} \sum \nabla f_j(\beta_j)$	Medium	Medium
A+	$\beta_i \leftarrow \beta_i - \frac{\eta}{K_i} \sum \nabla f_j(\beta_i)$	High	High

Federated Proximal Gradient Descent Robust to Poisoning Attacks

To generalize further to the setting with poisoning attacks, we can take the aggregation scheme (i.e., A, (AC), etc.) and turn it into a weighted sum rather than a regular sum. The weight given to the term from client j should depend on the amount that client i “trusts” client j . How trust is computed depends on the information available to client i .

For example, suppose that client i can see β_i^s , β_i^{s-1} , β_j^s , and β_j^{s-1} , that is, its own local model during the current and previous iteration, as well as the local model of client j during the current and previous iteration. It would not be suspicious if the models of client i and client j approach each other over time. Therefore, we can check

$$\|\beta_i^s - \beta_j^s\| < \|\beta_i^{s-1} - \beta_j^{s-1}\|.$$

This is usable by (AC), CG, GC, and (GC) at no additional communication cost.

WRITE ABOUT GRADIENT-BASED TRUST

Argument	Metric	Rule	Use Case	Add'l Cost
Models	L2 norm	$\ \beta_i^s - \beta_j^s\ < \ \beta_i^{s-1} - \beta_j^{s-1}\ $	(AC), CG, GC, (GC)	Low
Gradients	Cosine	$\cos(\angle(\nabla f_i(\beta_j), \nabla f_j(\beta_j)))$	A, (AC)	High