Fairness in Federated Learning via Core-Stability

Bhaskar Ray Chaudhury   Linyi Li   Mintong Kang   Bo Li   Ruta Mehta

University of Illinois at Urbana Champaign

Introduction

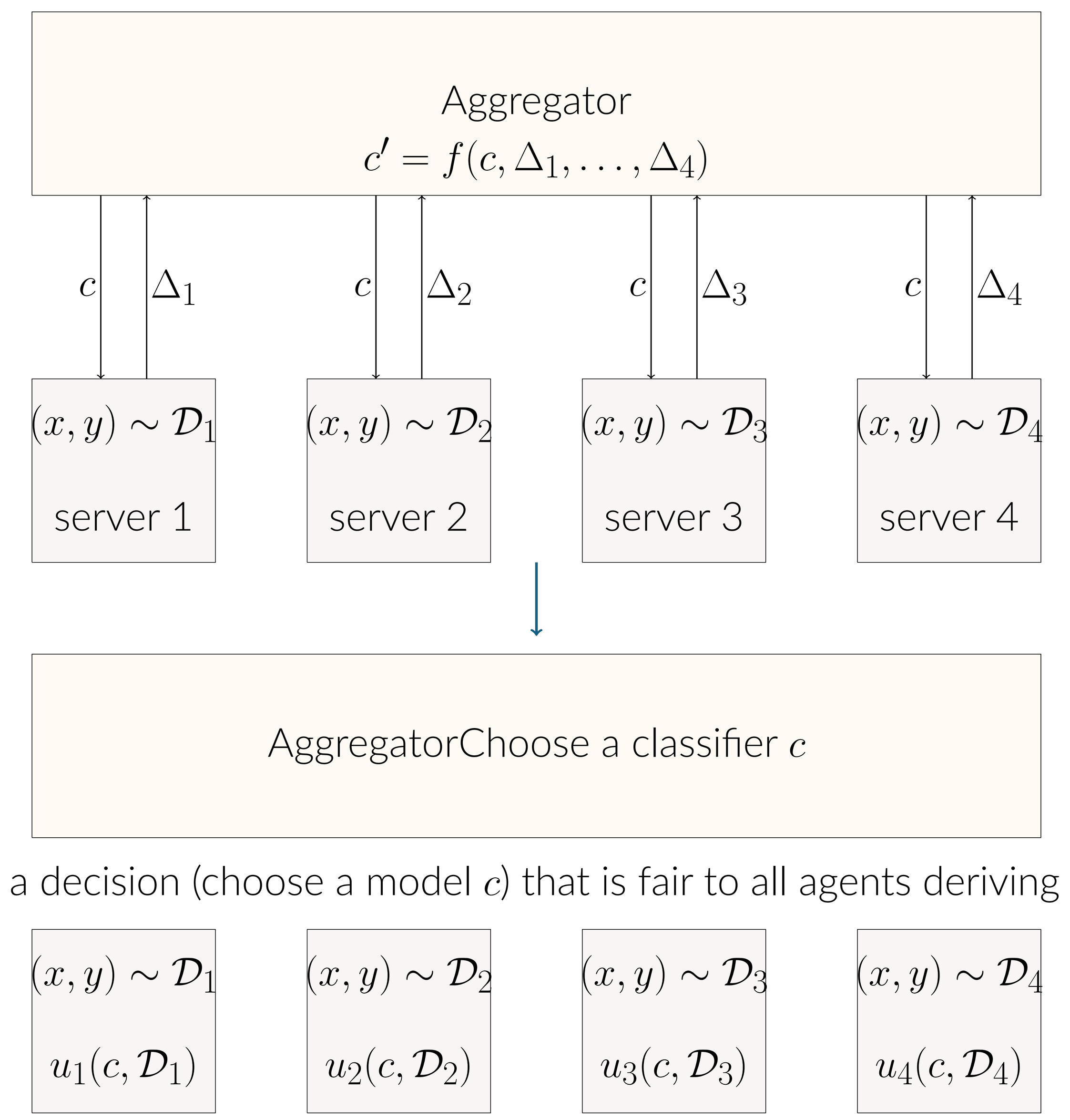
- We formally extend core-stability from co-operative game theory to define fairness in federated learning.
- We show core-stability exists under some conditions proved with a fixed point formulation.
  - **Linear / logistic regression**: Hold
  - **Smooth Neural Nets (DNN)**: approximate core-stable within a local neighborhood
- We design an effective FL protocol CoreFed to realize core-stable training when possible.
- On three datasets, CoreFed achieves core-stable fairness, while maintaining similar utility with the standard FedAvg protocol.

Federated Learning (FL)

- A distributive Machine Learning framework – set of federating agents train a joint classifier without sharing data.
- Widely applied in many applications, e.g., self-driving cars and medical imaging.
- Different clients in FL may have heterogeneous data. **How to train a centralized model that is fair to all agents?**

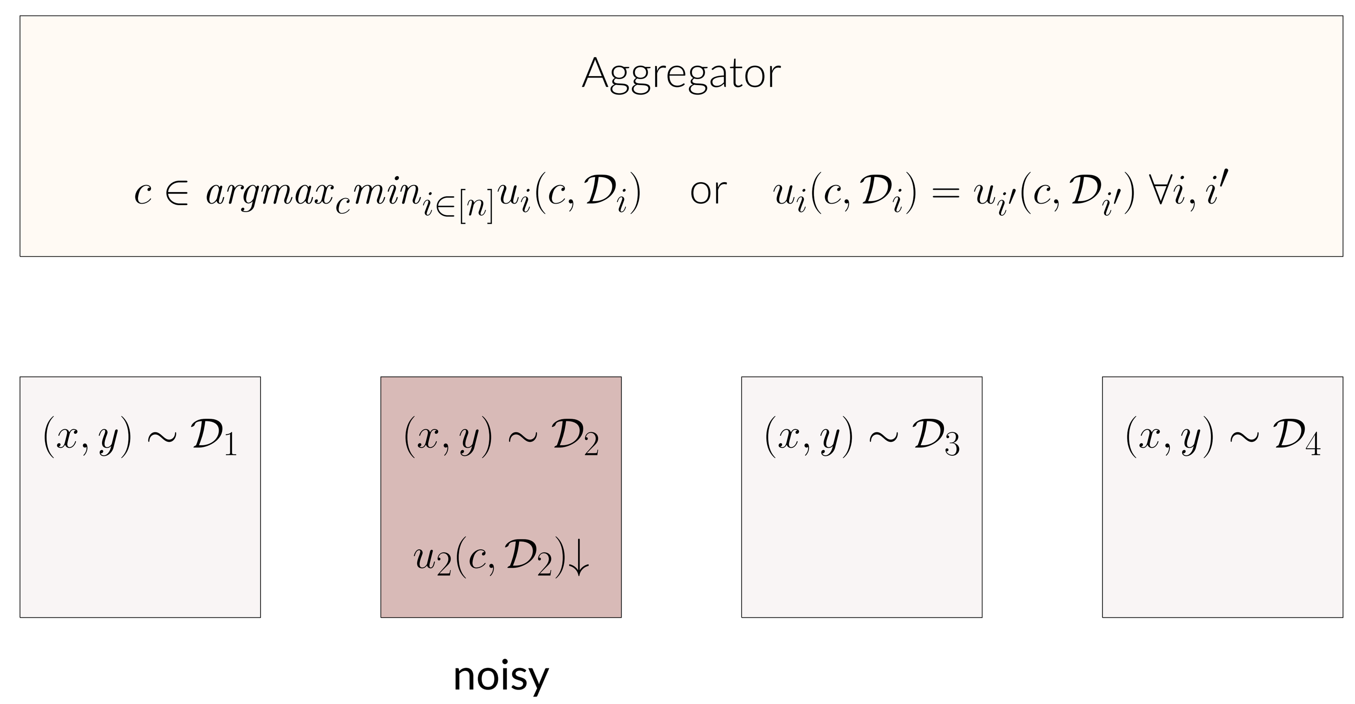
FL as Public Decision Making

Find a model that performs well on all types of data distributions (**Representational Parity**).



Some Existing Fairness Notions

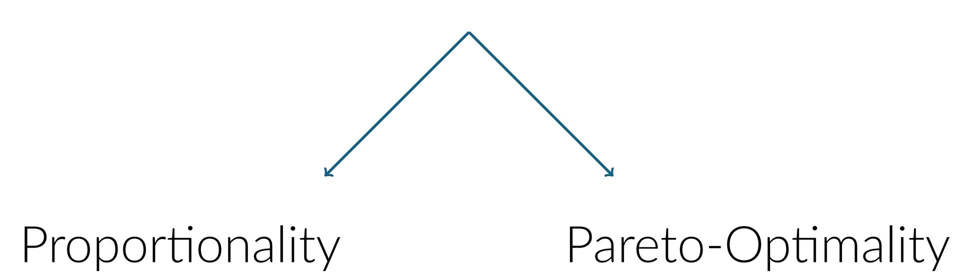
- Egalitarian Fairness[Donahue, Kleinberg'21]: Find c such that $max_c min_{i\in[n]} u_i(c, \mathcal{D}_i)$.
- Equity Based Fairness [Donahue, Kleinberg'21]: Find c such that $\frac{u_i(c,\mathcal{D}_i)}{n_i} = \frac{u_{i'}(c,\mathcal{D}_{i'})}{n_{i'}}$ $\forall i, i'$.
- **Problem:** Final outcome will be tailored towards the agent who is hard to satisfy, i.e., is susceptible to noisy data from particular agents.



Final c will be tailored towards to $\mathcal{D}_2$

Core-Stability

Choose c that maximizes $\prod_{i\in[n]} u_i(c, \mathcal{D}_i)$ (can be implemented via SGD)



- Proportionality: $u_i(c, \mathcal{D}_i) \geq \frac{u_i(c', \mathcal{D}_i)}{n}\ \forall c'$
- Pareto-Optimality: $\exists$ no c' s.t. $u_i(c, \mathcal{D}_i) \geq u_i(c', \mathcal{D}_i)$ with at least one strict inequality.

- **Core-Stability:** No set of agents have "significant incentive" to break and train a classifier with their own data ,i.e., $\exists$ no $S \subseteq [n]$, and no c' such that $\frac{|S|}{n} \cdot u_i(c', \mathcal{D}_i) \geq u_i(c, \mathcal{D}_i)\ \forall i \in S$ with at least one strict inequality.

Core-Stability generalizes both **Proportionality** ($S = \{i\}$ for each i) and **Pareto-Optimality** ($S = [n]$).

Distributed Algorithm

- **Input:** Number of clients K, number of rounds T, epochs E, learning rate η.
- **Output:** Model weights θ^T
- **For** $t = 0, 1, \cdots, T-1$,
  - Server selects a subset of K devices S_t
  - Server sends weights θ^t to all selected devices
  - Each selected device $s \in S_t$ updates θ^t for E epochs of SGD with learning rate η to obtain new weights $\bar{\theta}^t_s$
  - Each selected device $s \in S_t$ computes

$$\Delta\theta^t_s = \bar{\theta}^t_s - \theta^t,$$
$$\mathcal{L}^t_s = \frac{1}{|\mathcal{D}_s|}\sum_{i=1}^{|\mathcal{D}_s|} \ell(f_{\theta^t}(x^{(i)}_s), y^{(i)}_s)$$

  where $\mathcal{D}_s = \{(x^{(i)}_s, y^{(i)}_s) : 1 \leq i \leq |\mathcal{D}_s|\}$ is the training dataset on device s
  - Each selected device $s \in S_t$ sends $\Delta\theta_s$ and $\mathcal{L}_s$ back to the server
  - Server updates θ^{t+1} following

$$\theta^{t+1} \leftarrow \theta^t + \frac{1}{|S_t|}\sum_{s\in S_t} \frac{\Delta\theta^t_s}{M_s - \mathcal{L}^t_s \text{(weighted update)}}.$$

Experimental Evaluation

1. Main baseline: FedAvg   CoreFed achieves core-stable fairness compared with FedAvg while maintaining similar utility.
2. "U(Average)": average utility, "U(Multi)": multiplicative utility of the trained global model   CoreFed achieves higher overall utilities, especially for the multiplicative case since FedAvg favors the average case in general.

Table 1. Comparison of utility for each agent trained with CORE-FED and FedAvg. We see that $\sum_{i\in[n]} \frac{u_i(\theta',\mathcal{D}_i)}{u_i(\theta^*,\mathcal{D}_i)} < n$ holds, where θ' denotes the weights of shared model trained by FedAvg and θ^* by CORE-FED.

Dataset	Method	Agent 0	Agent 1	Agent 2	U(Average)	U(Multi)	$\sum_{i\in[n]} \frac{u_i(\theta',\mathcal{D}_i)}{u_i(\theta^*,\mathcal{D}_i)}$
Adult	FedAvg	2.59	0.77	1.46	1.61	2.91	2.80 (<3)
	CoreFed	2.62	0.90	1.53	1.68	3.61	
MNIST	FedAvg	0.34	0.29	0.92	0.52	0.091	2.66 (<3)
	CoreFed	0.36	0.41	0.91	0.56	0.13	
CIFAR-10	FedAvg	0.63	1.40	0.51	0.84	0.45	2.62 (<3)
	CoreFed	0.73	1.35	0.71	0.93	0.70	

Table 2. Comparison of utility for each agent trained with CORE-FED and FedAvg on CIFAR-10 with network VGG-11.

Method	Agent 0	Agent 1	Agent 2	U(Average)	U(Multi)	$\sum_{i\in[n]} \frac{u_i(\theta',\mathcal{D}_i)}{u_i(\theta^*,\mathcal{D}_i)}$
FedAvg	0.25	3.25	3.46	2.35	2.89	2.25 (<3)
CoreFed	1.63	3.17	3.32	2.71	17.15	

Table 3. Comparison of utility for each agent trained with CORE-FED and FedAvg on CIFAR-10 in the scenario that some agents have data of low quality (i.e., with added Gaussian noise). The variance of added Gaussian noise is 0.0,0.5,1.0 for agent 0,1,2, respectively.

Method	Agent 0	Agent 1	Agent 2	U(Average)	U(Multi)	$\sum_{i\in[n]} \frac{u_i(\theta',\mathcal{D}_i)}{u_i(\theta^*,\mathcal{D}_i)}$
FedAvg	3.28	3.30	1.42	2.67	15.37	2.74 (<3)
CoreFed	3.26	3.27	1.95	2.83	20.79	