FEDERAL LEARNING

cross-silo FL

- Third party ~~owns the~~ acts as the ~~cent~~ central server. ~~on~~
- incentive should not rely on private information.
- computational and communication resources provided by ~~or~~ gnizations are public goods.

~~Social welfare maximization problem~~

Convex problem

A real valued function f of x defined on an interval is called convex if the line segment between any two points on the graph of the function ~~has~~ lies above the graph.

For convex, it can have only one local minima. Finding minima is the problem, can be achieved using gradient descent.

Social Welfare Maximization

It is summary of static long-run general equilibrium conditions of a perfectly competitive economy.

~~Convex prof~~

Social ~~wef~~ welfare function

$W = U_1 + U_2 + U_3 + \ldots + U_n$

Problem: Two people with the following utility functions for goods X

$U_1 = 20 X_1^{1/2}$

$U_2 = 2X_2$

~~let~~ Let there are 140 units of goods X, so $140 = X_1 + X_2$

So, how to allocate x to maximize welfare.

$\mathcal{L} = 20X_1^{1/2} + 2X_2 + \lambda(140 - X_1 - X_2)$

$\frac{\partial \mathcal{L}}{\partial X_1} = 10 X_1^{-1/2} - \lambda = 0 \Rightarrow X_1 = 25$ → So $X_1 = 25$

$\frac{\partial \mathcal{L}}{\partial X_2} = 2 - \lambda = 0 \Rightarrow \lambda = 2$ ↑ $X_2 = 115$

$\frac{\partial \mathcal{L}}{\partial \lambda} = 140 - X_1 - X_2 = 0$

$\max W(u_1(x), \ldots, u_n(x))$

$$\text{s.t.} \begin{cases} \sum_{i=1}^{n} x_i^1 = X^1 \\ \vdots \\ \sum_{i=1}^{n} x_i^K = X^K \end{cases}$$ constraints, K constraints

can be linear or unstable, always exists, unstable.

~~Assumption~~:

Result: The strategic interaction among the organizations is modelled as a non-cooperative game with perfect information (given that organizations know the private information of each other.

Read again

Nash Equilibrium

It is a concept within game theory where the optimal outcome of a game is where there is no incentive to deviate from initial strategy. Overall, an individual can receive no incremental benefits from changing actions, assuming other players remain constant in their strategies. A game may have multiple Nash equilibrium or none at all.

Players have their strategy and other player also knows the strategy we can predict result or equilibrium.

To see if it even exists, reveal each player's strategy to other players. If no one changes their strategy, then the Nash equilibrium is proven.

Example:

SAM \ TOM	A	B
A	-1, 1	1, -1
B	-1, 1	0, 0

Both player choose to gain $1 by choosing strategy A.

Prisoner's dilemma: we might actually end up for a worse solution.

SAM \ TOM	A	B
A	9, 9	1, 10
B	10, 1	2, 2

→ Nash equilibrium not the best soln

If SAM choose A, Tom should choose B

	B	B
Tom	A, Sam	B
	B	B -

Both think they are doing better for themselves but are actually leading to a worse outcome.

SYSTEM Model

N organisations $N = \{0, 1, \ldots, N-1\}$

Let S_n denote the collected dataset of organisation.

$S_n \rightarrow$ no of data units in set S_n, $S_n = |S_n|$

$w \rightarrow$ weights of global model

$w^* \rightarrow$ optimal weights for global model

$$w^* = \arg\min \left\{ L(w) \triangleq \sum_{n \in N} \frac{S_n}{\sum_{n' \in N} S_{n'}} L(w, S_n) \right\}$$

(congruent)

$L(w, S_n) \rightarrow$ loss over dataset S_n

FL model

→ Same neural net on structure as global model

$w^r \rightarrow$ weights of global model after round r

$w_n^r \rightarrow$ local

1. w^{r-1} + K local updates corresponding to mini-batch stochastic descent.
2. upload w_n^r

$$\omega^r = \sum_{n \in N} S_n \omega_n^r \Big/ \sum_{n \in N} S_n'$$

K → no of updates on downloaded model

f_n → processing capacity used by organisation N.

D_n → no. of cycles required by n to process one data unit

T_n^{UL} → time for uploading

T_n^{DL} → time for downloading

duration of round $T(f) = \max_{n \in N} \left\{ \frac{S_n D_n K}{f_n} + T_n^{UL} + T_n^{DL} \right\}$

↓ max time among models

total time ↓

no. of rounds $r(f) = T / T(f)$

m_n (in dollars) → money transfer to organisation

Let vector $m = (m_n, n \in N)$

1) Utility: precision of trained global model.

$\epsilon(r(f)) \geq L(\omega^{r(f)}) - L(\omega^*)$

actual, expected

Assume: $L(\omega)$ is convex (non increasing and convex)

$\epsilon(r(f)) = \epsilon_0 / (\epsilon_1 + K r(f))$

ϵ_0, ϵ_1 are constants.

↳ i.e. $\epsilon(r(f))$ is non increasing and convex in $r(f)$ as well as bounded. $\epsilon(0)$

Utility, $U_n(r(f)) = u_n(\epsilon(0) - \epsilon(r(f))), n \in N$

u_n → unit revenue that organization n can earn from it market using global model.

2) Cost.

$$C_n(f_n, r(f)) = (C_n^{VL} + C_n^{DL})\, r(f) + \underbrace{C_n^{inv} f_n}_{\text{investment cost}} + \underbrace{C_n^{comp} (f_n)^2 S_n D_n K r(f)}_{\text{operating costs}}, \quad n \in N$$

$C_n(f_n, r(f))$ may be unknown to central server and other organisations

$(f_n)^2$ because of quadratic energy consumption

$r(f)$ → no. of cycles

3) Pay off

$$V_n(f_n, r(f), u_n) = U_n(r(f)) - C_n(f_n, r(f)) + m_n$$

1) Social welfare Maximization

$$\max_{f} \sum_{n \in N} (U_n(r(f)) - C_n(f_n, r(f))) \quad \text{--- (8)}$$

subject to $f_n \geq 0, n \in N$

$$\sum m = 0$$

2) Incentive Mechanism Design Problem

By organisation:

message profile (γ_n, π_n), γ_n → no. of rounds that organisation n expects

π_n → unit monetary transfer per training round that organisation n expects to pay or receive.

By central server:

$f(\gamma) = (f_n(\gamma), n \in N)$ process capacity vector

$m(\gamma, \pi) = (m_n(\gamma, \pi), n \in N)$, monetary transfer vector

(γ^{NE}, π^{NE}) NE of game, i.e. no one cannot increase payoff by deviating from it.

$f(\gamma)$ and $m(\gamma, \pi)$ optimised

Properties

P1 Social efficiency: (8)

P2 Individual rationality

$$\forall n \left(f_n(\gamma^{NE}),\ r(f(\gamma^{NE})),\ m_n(\gamma^{NE}, \pi^{NE}) \right) \geq 0$$

P3 Budget Balance: $\sum_{n \in N} m_n(\gamma^{NE}, \pi^{NE}) = 0$

INCENTIVE MECHANISM DESIGN

P1-P3 + social welfare maximization (8) is non-convex.

~~+ each organization is both~~

max no. of possible rounds $\leftarrow \bar{r} \triangleq \max_{f_n \geq 0} \sum_{n \in N} r(f)$

$$\bar{r} = \min_{n \in N} \left\{ T / T_n^{UL} + T_n^{DL} \right\}$$

Mechanism 1

$\gamma_n \in [0, \bar{r}]$, $\pi_n \in R$

$\tilde{r}(\gamma)$, no. of rounds each organisation needs to perform

$\tilde{r}(\gamma) = \left(\sum_{n \in N} \gamma_n \right) / N$

$f_n(\gamma)$ → processing capacity that each org should use

$$f_n(\gamma) = f_n^{\infty}(\tilde{r}(\gamma)) \triangleq \frac{S_n D_n K}{\frac{T}{\tilde{r}(\gamma)} - T_n^{UL} - T_n^{DL}}, \quad n \in N$$

$m_n(\gamma, \pi)$ → monetary transfer

$$m_n(\gamma, \pi) = \tilde{r}(\gamma)(\pi_{\mu}(n+1) - \pi_{\mu}(n+2))$$

i.e. no of rounds X difference between the unit monetary transfer submitted by organisation with indices $\mu(n+1)$ and $\mu(n+2)$

$(n+1)\%N$ $(n+2)\%N$

1) Game of Organisation

Game 1 (Message Profile Submission)

- Player: all organisations, $n \in N$
- strategy: message profile (γ_n, π_n) with $\gamma_n \in [0, \bar{r}]$ and $\pi_n \in R$ for each organisation $n \in N$.
- Pay off fn: V_n.

(γ_{-n}, π_{-n}) = all message profiles except $n \in N$

$\gamma_{-n} = (\gamma_{n'}, n' \in N \setminus \{n\})$ $\pi_{-n} = (\pi_{n'}, n' \in N \setminus \{n\})$

Definition 1.

$$V_n(f_n(\gamma^{NE}), \tilde{r}(\gamma^{NE}), m_n(\gamma^{NE}, \pi^{NE})) \geq V_n(f_n(\gamma_n, \gamma_{-n}^{NE}), \tilde{r}(\gamma_n, \gamma_{-n}^{NE}), m_n(\gamma_n, \pi_n, \gamma_{-n}^{NE}, \pi_{-n}^{NE}))$$

$\gamma_n \in [0, \bar{r}], \pi_n \in R, n \in N$

Assumption or Result:

Read again

$m_n(\gamma_n, \pi_n, \gamma_{-n}^{NE}, \pi_{-n}^{NE}) = m_n(\gamma_n, \pi_n^{NE}, \gamma_{-n}^{NE}, \pi_{-n}^{NE})$ for $\pi_n \in R$ under any $N \geq 3$.

For $N = 2$, make a virtual org with zero utility and cost.

2) Nash Equilibrium and Properties:

Lemma 1. (Nash equilibrium).

A message profile is NE of game 1 iff.

$$r_n^{NE} = \underset{r \in [0,\bar{r}]}{N\arg\max}\ V_n(f_n^*(r), r, m_n(r1, \Pi^{NE})) - \sum_{n' \in N \setminus \{n\}} r_{n'}^{NE},\ n \in N$$

where 1 is an all-one vector with length N.

$$m_n(r1, \Pi^{NE}) = r(\Pi_{\mu(n+1)}^{NE} - \Pi_{\mu(n+2)}^{NE})$$

DISTRIBUTED ALGORITHM DESIGN

1. ~~Problem Reformulation~~

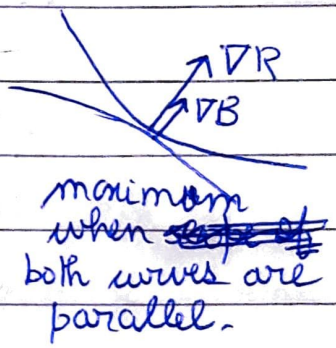
Lagrangian Multipliers

↳ converge to saddle point of Lagrangian hence NE of the game 1.

$f(x,y) = x^2 e^y y$

↓ Maximise

$g(x,y) = x^2 + y^2 = 4$ (constraint)



maximum when ~~scope of~~ both curves are parallel.

To maximise f subject to $g(x,y)=0$ means to find the level ~~curve~~ curve of f with greatest k-value that intersects the constraint curve. It will be the place where two curves are tangent.

$$\nabla f = c \nabla g$$

$$\nabla f(x,y) = \lambda \nabla g(x,y)$$

x, y are critical points and λ is lagrainge multiplier.

$$L(x,y,\lambda) = R(x,y) - \lambda(B(x,y) - b)$$

$$\nabla L = 0$$

$$\begin{bmatrix} \partial L/\partial x \\ \partial L/\partial y \\ \partial L/\partial \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Problem Reformulation

$r = (r_n, n \in N)$, r_n is no. of rounds organisation n performs.

$$\underset{r}{\text{maximise}} \sum_{n \in N} (U_n(r_n) - C_n(f_n^*(r_n), r_n))$$

subject to $r_{\mu(n-2)} = r_{\mu(n-1)}$ $n \in N$

$r_n \in [0, \bar{r}]$, $n \in N$

Let r^* be optimal soln.

$r^* = r_0^* = \_\_ = r_{N-1}^*$

$f^*(r^*) = (f_n^*(r^*), n \in N)$ optimizes 8

Then $L : [0, \bar{r}]^N \times R^N \to R$

$$L(r, \lambda) = \sum_{n \in N} (U_n(r_n) - C_n(f_n^*(r_n), r_n)) - \sum_{n \in N} \lambda_n (r_{\mu(n-2)} - r_{\mu(n-1)}),$$

$\lambda = (\lambda_n, n \in N)$ is the vector of lagrange multiplier.

$$L(r, n) = \sum_{n \in N} L_n(r_n, \lambda)$$

$$L_n(r_n, \lambda) = U_n(r_n) - C_n(f^*(r_n), r_n) - [\lambda_{\mu(n+2)} - \lambda_{\mu(n+1)}] r_n \quad n \in N$$

Lagrangian multipliers

↓

corresponding to constraints

$r_n = r_{\mu(n+1)}$ $r_{\mu(n-1)} = r_n$ resp

So $L_n(r_n, \lambda) = V_n(f_n^*(r_n), r_n, m_n(r_{n1}, \lambda))$ for $n \in \mathcal{N}$.

~~Dual~~ Duality holds according to Slater's condition. Hence saddle point $\mathcal{L}(r, \lambda)$, (r^*, λ^*) exists.

$\mathcal{L}(r, \lambda^*) \leq \mathcal{L}(r^*, \lambda^*) \leq \mathcal{L}(r^*, \lambda)$ for any $r \in [0, \bar{r}]$

(r^*, λ^*) is the saddle point of $\mathcal{L}(r, \lambda)$ then it is an NE of game 1.

Lemma 2 (Saddle point and NE). For any saddle point of $\mathcal{L}(r, \lambda)$ denoted (r^*, λ^*). the message profile $(\gamma^{NE} = r^*, \pi^{NE} = \lambda^*)$ is NE of game 1.

Algorithm 1: Distributed Algorithm for Cross-silo FL

Organisation $n \in \mathcal{N}$ randomly initializes $\gamma_n(0), \pi_n(0)$

$t \leftarrow 0$, Convg_Indicator $\leftarrow 0$

while Convg_Indicator do

Organization $n \in \mathcal{N}$ submits $(\gamma_n(t), \pi_n(t))$;

Central server sends organization $\gamma(t)$ and $\pi(t)$;

for organization $n \in \mathcal{N}$ in parallel do

$\hat{\gamma}_n \leftarrow \text{argmax}_{\gamma_n \in [0, \bar{r}]} V_n^p(\gamma_n, \gamma_{-n}(t), \pi(t))$;

$\gamma_n(t+1) \leftarrow \gamma_n(t) + \eta(\hat{\gamma}_n(t) - \gamma_n(t))$

$\pi_n(t+1) \leftarrow \pi_n(t) + p\eta(\gamma_{\mu(n-2)}(t) - \gamma_{\mu(n-1)}(t))$;

end

$t \leftarrow t+1$;

if $|\gamma_n(t+1) - \gamma_n(t)| \leq \phi$ $n \in \mathcal{N}$ then

convg_Indicator $\leftarrow 1$;

end

end

find saddle point of

$$L(\gamma,\Pi)=\sum_{n\in N} L_n(\gamma_n,\Pi)=\sum_{n\in N} V_n(f_n^*(\gamma_n),\gamma_n,m_n(\gamma_{n1},\Pi))$$

$$V_n^p(\gamma_n,\gamma_{-n},\Pi)=V_n(f_n^*(\gamma_n),\gamma_n,m_n(\gamma_{n1},\Pi))$$
$$-p\sum_{n\in N}(\gamma_{\mu(n-2)}-\gamma_{\mu(n-1)})^2$$

penalty coefficient