FL-Market: Trading Private Models in Federated Learning

Shuyuan ZHENG, Kyoto University
Yang Cao, Hokkaido University
Masatoshi Yoshikawa, Kyoto University
Huizhong Li, WeBank
Qiang Yan, Singapore Management University
1. Background
Dilemma of ML

1. Huge amounts of data required
   - Facebook’s object detection system has been reported to be trained on 3.5 billion images from Instagram.

2. Privacy concerns
   - Millions of Facebook users' personal data was acquired without the individuals' consent by Cambridge Analytica, predominantly to be used for political advertising.

3. Expensive datasets
   - People are becoming increasingly aware of the economic value of their data.
Model Trading

- Selling trained ML models
  - Cheaper than datasets
- Buyers do not contact training data.
  - Relieve privacy concerns
- Problem: Models still contain private information.
Existing Model Marketplaces

- No privacy protection supported [1, 2]
- Privacy protection against buyers [3, 4, 5]
  - A trusted broker injects noise into models
  - Uniform privacy protection levels

[1] Chen et al., “Towards model-based pricing for machine learning in a data marketplace,” SIGMOD, 2019.
[2] Jia et al., “Efficient task-specific data valuation for nearest neighbor algorithms,” PVLDB, 2019.
[3] Agarwal et al., “A marketplace for data: An algorithmic solution,” in ACM-EC, 2019.
[4] Liu et al., “Dealer: An end-to-end model marketplace with differential privacy,” PVLDB, 2021.
[5] Jiang et al., “Pricing GAN-based data generators under R’enyi differential privacy,” Information Sciences, 2022.
Problems

1. Unrealistic assumption: trusted broker.
   • Many giant companies were involved in privacy scandals and data breaches
   • Data owners need local privacy.
     • Privacy against both model buyers and the broker

2. Uniform privacy protection levels
   • Data owners have different privacy preferences
   • Data owners need personalized privacy protection.

Our goal: to design a model marketplace that supports local and personalized privacy.
Local and Personalized Privacy by FL + LDP

- Federated learning (FL) [6]
  - Data owners collaboratively train a model by submitting local gradients.
  - The local gradients are aggregated into a global gradient for model updating.
  - **Local privacy**: Training data maintained on the local sides

- Local differential privacy (LDP) [7]
  - Ensure the indistinguishability of any two local gradients.
  - **Local privacy**: Data owners perturb local gradients on the local sides.
  - **Personalized privacy**: Data owners can set different privacy losses $\epsilon_i$.

[6] McMahan et al., “Communication-efficient learning of deep networks from decentralized data,” AISTATS, 2017.
[7] Evfimievski et al., “Limiting privacy breaches in privacy preserving data mining,” PODS, 2003.
FL-Market: A Model Marketplace with Local and Personalized Privacy
Challenges

1. **Gradients aggregation** under personalized privacy losses
   - The conventional aggregation method only considers data size.
   - Different privacy losses result in **different accuracy levels**

2. **Gradients procurement** given a budget
   - Some gradients expensive, some cheap.
   - Purchase in a way that **maximizes the model utility**.
2. Trading Framework
Federated Learning

\[ w^{r+1} = w^r - \eta \cdot \frac{\sum_i g_i}{n} \]

1. **Model broadcasting**: The server broadcasts the global model.

2. **Local training**: Each data owner trains its model on its local data to derive a local gradient.

3. **Gradient aggregation**: The servers aggregate all the local gradients to derive a global gradient.

4. **Model updating**: The server updates the global model by the global gradient.
• **Auction mech.** for gradients procurement
• **Aggregation mech.** for gradients aggregation
FL-Market

1. financial budget $B$
2. valuation functions $v'_1, ..., v'_n$
3. privacy loss $\epsilon_1, ..., \epsilon_n$
4. payments $p_1, ..., p_n$
5. perturbed gradient $\hat{g}_\lambda = \sum_{i=1}^{n} \lambda_i \cdot \bar{g}_i$

**Step 1: Auction announcement**
FL-Market

Step 2: Bidding

1. financial budget $B$
2. valuation functions $v'_1, ..., v'_n$
3. privacy loss $\epsilon_1, ..., \epsilon_n$
4. perturbed local gradients $\bar{g}_1, ..., \bar{g}_n$
5. perturbed gradient $\bar{g}_\lambda = \sum_{i=1}^{n} \lambda_i \cdot \bar{g}_i$

Data owners

FL broker

Model buyer
Step 3: Privacy loss and payment decision

1. financial budget $B$
2. valuation functions $v'_1, ..., v'_n$
3. privacy loss $\epsilon_1, ..., \epsilon_n$
4. perturbed local gradients $\tilde{g}_1, ..., \tilde{g}_n$
5. perturbed gradient $\tilde{g}' = \sum_{i=1}^{n} \lambda_i \cdot \tilde{g}_i$

Note: $\forall i, \epsilon_i \leq \tilde{\epsilon}_i$ and $p_i \geq v_i(\epsilon_i)$. 
Step 4: Local gradient computing

Note: each $\tilde{g}_i$ satisfies $\epsilon_i$-LDP.
Note: $\lambda_i \in [0, 1]$, $\sum_i \lambda_i = 1$. 

Step 5: Gradient aggregation and delivery

\[
\hat{g}_\lambda = \sum_{i=1}^{n} \lambda_i \cdot \bar{g}_i
\]
Mechanism Design Problems

- **Aggregation mech.**
  - $\text{Aggr}(\epsilon_1, \ldots, \epsilon_n, d_1, \ldots, d_n) \rightarrow \lambda = [\lambda_1, \ldots, \lambda_n]$
  - Objective: To maximize the global gradient’s utility with respect to $\lambda$

- **Auction mech.**
  - $\text{Auc}(b'_1, \ldots, b'_n, B) \rightarrow \epsilon_1, \ldots, \epsilon_n, p_1, \ldots, p_n$
  - Objective: To maximize the global gradient’s utility with respect to $\epsilon_1, \ldots, \epsilon_n$
  - Constraints: truthfulness, individual rationality, budget feasibility…
3. Solution & Evaluation
Aggregation Mechanism: OptAggr

- Equivalent to a **convex** quadratic programming problem.
  - Can be well solved by existing solvers in polynomial time.
  - Only have **nonanalytical** solutions

- OptAggr decides optimal aggregation weights by employing an existing solver.
Auction Mechanism

• Challenge:
  • OptAggr does not provide an analytical solution
  • The auction objective is thus also nonanalytical.
  • Traditional auction theory only deals with analytical objectives.

• Solution: Automated mechanism design
  • To optimize the auction objective by machine learning.
RegretNet [8]

- SOTA automated mechanism design framework
  - Allocation network: for allocating privacy losses
  - Payment network: for setting payments

- Problems that makes optimization hard:
  - Only for single-unit auctions
  - Randomized auction results
    - When all $\epsilon_i = 0$, the expected error is unbounded.

[Duetting et al., “Optimal auctions through deep learning,” ICML, 2019.]
Auction Mechanism: DM-RegretNet

- Support **multi-unit** auctions
  - More possible values of privacy loss
- **Deterministic** auction results
  - Given the same bids and budget, the privacy losses are deterministic
Error Bound

• How do DM-RegretNet and OptAggr perform in terms of minimizing the error bound of the global gradient?
Model Accuracy

- How do DM-RegretNet and OptAggr perform in terms of optimizing model accuracy?
Thank you for listening!
Appendix
Local Differential Privacy

\[ \text{Gradient } g \quad \text{Randomized Algorithm } \mathcal{A} \quad \text{with probability } p_1 \]

\[ \text{Output } \tilde{g} \]

\[ \text{Gradient } g' \quad \text{Randomized Algorithm } \mathcal{A} \quad \text{with probability } p_2 \]

\[ \varepsilon\text{-LDP: for any possible } g, g', \text{ for any possible } \tilde{g}, \frac{p_1}{p_2} \leq e^\varepsilon \]
Mechanism Design Problems

- Aggregation mech:
  - Aggr\((\epsilon_1, ..., \epsilon_n, d_1, ..., d_n) \rightarrow \lambda = [\lambda_1, ..., \lambda_n]\)

- Auction mech:
  - Auc\((b'_1, ..., b'_n, B) \rightarrow \epsilon_1, ..., \epsilon_n, p_1, ..., p_n\)
  - Truthfulness: Obtain the highest profit by bidding the real preference.
  - Individual rationality (IR): Non-negative profit
  - Budget feasibility (BF)

Problem 1 (Error Bound-Minimizing Aggregation),
\[
\min_{\lambda = \text{Aggr}(\epsilon, d)} \quad E(RR(\tilde{g}_\lambda; \epsilon, d) = \sup_{g_1, ..., g_n} \text{err}(\tilde{g}_\lambda; \epsilon, d)
\]
\[\text{S.t.: } \forall i, \lambda_i \in [0, 1], \text{ and } \sum_{i=1}^{n} \lambda_i = 1\]

Problem 2 (Budget-Limited Multi-Unit Multi-Item Procurement Auction),
\[
\min_{\epsilon, p = \text{Auc}(b', B)} \quad \mathbb{E}[\nu, B] [E(RR(\tilde{g}_\lambda; \lambda = \text{Aggr}(\epsilon, d))]
\]
\[\text{S.t.: } \forall i, \epsilon_i \in [0, \epsilon'_i], \text{ truthfulness, IR, and BF.}\]
Training DM-RegretNet

1. Inference
2. Aggregation
3. Solve QP
4. Update DM-RegretNet

Auction Mech.: DM-RegretNet
Aggregation Mech.: OptAggr

Training bids → local gradients → global gradient
Joint Optimization

- Aggregation is affected by and feeds back into auction