Collaborative Machine Learning Markets

Olga Ohrimenko **Shruti Tople** **Sebastian Tschiatschek**

Microsoft Research

Abstract

We study the problem of collaborative machine learning markets where multiple parties can achieve improved performance on their machine learning tasks by combining their training data. We discuss desired properties for these machine learning markets in terms of fair revenue distribution and potential threats, including data replication. We then instantiate a collaborative market for cases where parties share a common machine learning task and where parties' tasks are different. Our marketplace incentivizes parties to submit high quality training and true validation data using a novel payment-division function that is robust-to-replication.

1 Introduction

One of the main obstacles for training well-performing machine learning models is the limited availability of sufficiently diverse labeled training data. However, the data needed to train good models often exists but is not easy to leverage as it is distributed and owned by multiple parties. For instance, in the medical domain, important data about patients that could be used for learning diagnostic support systems for cancer might be in possession of different hospitals, each of which holding different data, (e.g., from a specific geographical region with different demographics). Typically, by pooling the available data, the hospitals could train better machine learning models for their application than they could using only their own data. As all hospitals would benefit from a better machine learning model obtained through data sharing, there is a need for collaborative machine learning.

Naturally, this type of collaboration raises questions in terms of how to incentivize parties to participate in such a collaborative machine learning effort and how to respect privacy and integrity requirements of the parties regarding their data. In this work, we propose a cloud-based collaborative machine learning platform accessible to parties for submitting data and machine learning tasks that addresses the above concerns. Submitted training data represents the data that a party is willing to contribute/sell, while the validation data can be seen as a specification of the machine learning model a party is willing to buy. After a *trade* in this market, a participating party obtains a model trained on the data available to the market and customized for its task. Crucially, there is no sharing of data and parties are only provided this customized model (or query interface). As a result, only information relevant to the validation task is released through the model, limiting the possibility of copying and reusing the data for other tasks. Such markets allow multiple parties to jointly train machine learning models based on the training data provided by all of the parties and achieve improved performance on their own tasks. Parties pay *to* the market for the improvement on their validation tasks and get paid *by* the market for the contribution of their training data to the tasks of others. The market can support a single validation task scenario, for example, where hospitals bring together their data to train a single model for detecting cancer. Furthermore, it also supports scenarios where one's data can contribute to multiple tasks. An overview of our envisioned marketplace is shown in Figure 1.

Our market is enabled by three main components: 1. *Data valuation.* Each model in the market is trained on the data that is best suited for the specified validation task. Similar to recent and concurrent work [1, 2] we can use Shapley values [3], a solution concept from game theory discussed later, to match data to tasks. 2. *Customized model training.* We ensure that the model that a party receives is

Abstract for: Smooth Games Optimization and Machine Learning Workshop (NeurIPS 2019), Vancouver, Canada.

only suited for its own task but not the other tasks available on the market. As a result each party is incentivized to provide its true validation task. 3. *Payment division.* Each party receives a reward proportional to how useful its data is for training other models. We again use Shapley values, however, in this case to determine fair payoffs.

One of the key challenges in designing such data marketplaces comes from the very nature of data, i.e., free replication. This means that the reward of a party submitting copies of the same dataset should not be more than the party submitting it once. Indeed the authors of [4] also point out that, if used naively for machine learning, Shapley value is not robust to replication (albeit in a different market setup than ours). We design a market that is robust to replication. The key idea behind our approach is to allow a party to contribute data only if it also submits a validation task for which it requires a trained model. Submitting a task, in turn, requires a participation fee. Hence, for every replica, a party has to pay a participation fee that depends on its validation task. As a result the fee it pays for the improvement on its task balances the payoff it gets from the use of its training data.

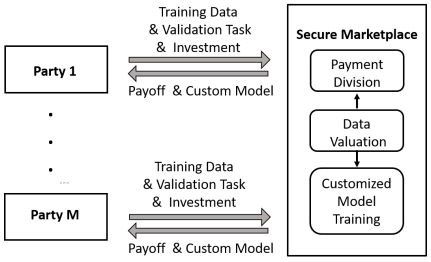

Figure 1: Our marketplace setup

Finally, we ensure that our cloud-based market is secure. That is, the market itself cannot resell, copy or release the data or the trained models. Such a market can be instantiated using secure hardware, such as Intel SGX [5], which allows parties to submit their data encrypted and ensure that it is only decrypted and processed in secure enclaves (e.g., see [6]).

We summarize our contribution as follows:

1. *Marketplace Definition*: We introduce a collaborative marketplace to sell data and buy machine learning models for learning single and multiple validation tasks.
2. *Payment Division*: We instantiate a collaborative market for a single validation task scenario with a novel and robust-to-replication payment division function.

In the full version of the paper [7] we extend these results to the multi-task scenario, propose customized model training and selection of training data for party-specific tasks, and empirically validate properties and assumptions of our marketplace.

Related Work Shapley values for valuating data have been used in [8, 2, 1, 9], while machine learning marketplaces have been studied in [10, 4, 11] The closest to our paper is work by Agarwal et al. [4]. They propose a general marketplace setup which makes use of Shapley values for computing payoffs. It addresses the problem of data replication heuristically by downweighing Shapley values according to data similarity. Critically, their approach reduces payoffs for honest (non-replicating) parties. In contrast, we achieve robustness to replication naturally through the multi-party setting in combination with a novel characteristic function.

2 Background and Notation

Performance (gain) function $\mathcal{G}$. Given training data $\mathcal{X}$, the goal of supervised machine learning is to identify function $\mathcal{M}$ that performs well on unseen data, i.e., test data, and, for instance, achieves good classification accuracy. Let $\mathcal{G}$ be the function that measures the performance of $\mathcal{M}$. We use validation data $\mathcal{V}$ as a proxy for estimating the performance. (E.g., the average classification performance is $\mathcal{G}(\mathcal{V}, \mathcal{M}(\mathcal{X})) = \frac{1}{|\mathcal{V}|} \sum_{(\omega,z)\in\mathcal{V}} \mathbf{1}_{\mathcal{M}(\omega)=z}$, where $\mathbf{1}$ is the indicator function.) Clearly, the performance measure is application dependent. For simplicity, we assume an idealized gain function $\mathcal{G}$ and dependencies of the model on the training data: for training datasets $\mathcal{X}, \mathcal{X}'$: (i) *replicated data does not change performance*: $\forall \mathcal{X}, \mathcal{X}' \colon \mathcal{M}(\mathcal{X}) = \mathcal{M}(\mathcal{X} \oplus \mathcal{X})$; (ii) *monotonicty*: $\mathcal{G}(\mathcal{V}, \mathcal{M}(\mathcal{X}))) \leq \mathcal{G}(\mathcal{V}, \mathcal{M}(\mathcal{X} \oplus \mathcal{X}')))$; (iii) *supermodularity*: $\mathcal{G}(\mathcal{V}, \mathcal{M}(\mathcal{X} \cup \{x\})) - \mathcal{G}(\mathcal{V}, \mathcal{M}(\mathcal{X})) \leq \mathcal{G}(\mathcal{V}, \mathcal{M}(\mathcal{X}' \cup \{x\})) - \mathcal{G}(\mathcal{V}, \mathcal{M}(\mathcal{X}'))$ for $\mathcal{X} \subseteq \mathcal{X}'$ and $x \notin \mathcal{X}'$; (iv) *boundedness*: $\mathcal{G}(\mathcal{V}, \mathcal{M}(\mathcal{X})) \leq 1$. This is for instance true for 1-NN classifiers.

Shapley Values for Fair Payoffs Consider a (machine learning) task which M parties $\mathbf{M} = \{1, \ldots, M\}$ aim to solve together. To quantify the value of the contribution of each party towards solving the task, we consider a characteristic function $v \colon 2^{\mathbf{M}} \to \mathbb{R}$. For every set $S \subseteq \mathbf{M}$ of parties,

$v(S)$ quantifies how well the parties in S can solve the task, e.g., $v(S)$ could be the prediction accuracy of the best model the parties in S can train by combining their data, i.e., $v(S) = \mathcal{G}(\mathcal{V}, \cup_{i\in S}\mathcal{X}_i)$, where $\mathcal{X}_i$ is the ith party's training data. In order to compensate parties for helping to solve a machine learning task, we consider Shapley values [3] studied in game theory for collaborative games:

$$\psi(v,i) = \sum_{S\subseteq \mathbf{M}\setminus\{i\}} \frac{|S|!(M-|S|-1)!}{M!}\big(v(S\cup\{i\}) - v(S)\big), \tag{1}$$

i.e., $\psi(v,i)$ quantifies the average marginal contribution of party i wrt all possible subsets of parties. Shapley values $\psi(v,i)$ are the unique payoffs satisfying properties of *efficiency, symmetry, linearity and null player*, ensuring that parties are paid equally for equal contributions and all gains are distributed among the parties. However, Shapley values are not robust to replication.

3 Collaborative Marketplace

In our marketplace, we consider M parties $P_1, \ldots, P_M$ which aim to collaborate towards training machine learning models for their tasks. Each party simultaneously takes the role of a seller and a buyer. The ith party has training data $\mathcal{X}_i$ and validation data $\mathcal{V}_i$. The performance of a machine learning model $\mathcal{M}(\mathcal{X})$ trained on some training data $\mathcal{X}$ is evaluated using a performance/gain function $\mathcal{G}(\mathcal{V}_i, \mathcal{M}(\mathcal{X})) \in [0,1]$.

The goal of the market is to provide party i with a customized model trained on the subset of (or potentially all) datasets of other parties' that best fits its task (based on $\mathcal{V}_i$). At the same time the market uses $\mathcal{X}_i$ to train models of parties where this dataset fits the corresponding task (based on validation data of other parties).

Definition 1 (Marketplace)**.** *A* marketplace *is a tuple $(\mathcal{P}, PD)$, where $\mathcal{P} = (P_1, \ldots, P_M)$ is the list of parties engaging with the market place and PD is the payment division function.*

We consider the following interaction with the data marketplace:

1. Parties $P_1, \ldots, P_M$ arrive.
2. The marketplace collects all training data sets $\mathcal{X}_1, \ldots, \mathcal{X}_M$ and validation tasks $\mathcal{V}_1, \ldots, \mathcal{V}_M$.
3. Every party pays the market a participation fee A_i that is determined proportional to unit increase in personal performance gain $A_i = 1 - \mathcal{G}(\mathcal{V}_i, \mathcal{M}(\mathcal{X}_i))$, i.e., valuation for increasing performance to 100% on their validation data. The market can also set a fixed value for a unit increase $c > 0$ in performance, such that $A_i = c \cdot (1 - \mathcal{G}(\mathcal{V}_i, \mathcal{M}(\mathcal{X}_i)))$.
4. The marketplace trains a machine learning model $\mathcal{M}^i$ for every party i where $\mathcal{M}^i = \mathcal{M}(\mathcal{V}_i, \oplus_{j\in S_i}\mathcal{X}_j)$ and $S_i \subseteq \mathbf{M}$.
5. The model $\mathcal{M}^i$ is shared with party i. Let $a_i = \mathcal{G}(\mathcal{V}_i, \mathcal{M}^i) - \mathcal{G}(\mathcal{V}_i, \mathcal{M}(\mathcal{X}_i))$. Party i receives payoff t_i which depends on the increase in performance on its validation data $\mathcal{V}_i$ (i.e., they receive $A_i - a_i$) and how much its data helps in improving performance of models for other validation tasks, b_i. Hence, in total party i gains (or loses) $(A_i - a_i) + b_i - A_i = b_i - a_i$.

Single validation task This is a special case of the marketplace where all parties agree on the same validation set $\mathcal{V}$ and there is only one model $\mathcal{M}$ that is trained and returned to all parties.

3.1 Desired Properties

In the following we enumerate desired properties for a machine learning marketplace such that participating parties receive *fair* payoffs for their engagement and benefit from participation.

Revenue division and payment Our list of properties is inspired by the "standard axioms of fairness" since they are the de facto method to assess the marginal value of goods (i.e., features in our setting) in a cooperative game (i.e., prediction task in our setting). They include: *Balance:* $\sum b_i = \sum a_i$. *Symmetry:* if two parties enter the market with same training and validation sets then $t_i = t_j$. *Zero element buyer:* if there is a party whose performance does not increase, it should at least get its participation fee back, $t_i = A_i$.

Incentives Party i decides on whether to enter the collaborative market or not and what $\mathcal{V}_i$ and $\mathcal{X}_i$ to contribute. As a result, the marketplace should incentivize the parties to join the market with good training data and honest validation data:

Joining the market: The market should incentivize new parties to join. Our setting incentivizes this as follows. A new party brings a new task to the market, hence, existing parties' data may be useful for this task, increasing their payoff b_i. At the same time, the new party is also bringing new training data which can increase performance of tasks submitted in the existing market, resulting in an increased payoff. Hence a party joining the market can benefit from increased utility.

Validation data: A dishonest party can try to manipulate training and validation data (including their relationship) in order to gain more than it would with its true training and validation datasets. For example, $\mathcal{V}_i$ can be seen as an implicit bid that party i places on the model $\mathcal{M}^i$ it will obtain. There can be a case that $\mathcal{G}(\mathcal{V}', \mathcal{M}^i) - \mathcal{G}(\mathcal{V}', \mathcal{M}(\mathcal{X}_i)) > \mathcal{G}(\mathcal{V}_i, \mathcal{M}^i) - \mathcal{G}(\mathcal{V}_i, \mathcal{M}(\mathcal{X}_i))$. If it is the case, $\mathcal{M}^i$ has higher utility than what is determined by $\mathcal{V}_i$. To this end, the marketplace needs to ensure that the model $\mathcal{M}^i$ that is returned to the party does not allow for existence of $\mathcal{V}'$ in the current marketplace, incentivizing the party to provide the best $\mathcal{V}_i$ to get the best utility model.

Training data: The market needs to ensure that the payment b_i that party i receives for the use of its data to train other models incentivizes it to provide its best $\mathcal{X}_i$. We note that our market does not have an explicit way for parties to bid or price training data.

Robustness to replication Parties may not behave honestly and may replicate their data and create new replica parties to join the market on their behalf. The market should be robust to replication. That is, a party that replicates its training data should not earn more than it would in the original market. This is a crucial property for any data market since data, as compared to physical goods, is easily replicable. This problem was also highlighted for a different marketplace setup in [4].

4 Market Instantiation

We consider a special case of the marketplace where all parties agree on the same validation set $\mathcal{V}$ and there is only one model $\mathcal{M}$ that is trained and returned to all M parties. (The scenario with multiple validation tasks is described in the full version [7].) We slightly abuse the notation by letting $\mathcal{M}_S = \mathcal{M}(\oplus_{k \in S} \mathcal{X}_k)$ be the model trained on data from parties in S for task $\mathcal{V}$. The parties agree on the same marginal payment per increase of the model performance they gain (for example, per percentage increase): if $\mathcal{G}(\mathcal{V}, \mathcal{M}_{\mathbf{M}}) = 1$ then party i pays the amount proportional to $A_i = 1 - \mathcal{G}(\mathcal{V}, \mathcal{M}_i)$. Hence, the largest amount that can be distributed among market participants is $\sum_{i \in \mathbf{M}} A_i$. Recall that, when entering the market, the party submits $\mathcal{X}_i$ and fee A_i. After the market completes, i obtains $\mathcal{M}_{\mathbf{M}}$ and payout $b_i \geq 0$.

Characteristic Function Our characteristic function captures the value of data of parties in a set S for the task $\mathcal{V}$ as the value of the model trained on the data as well as value the model brings to every party. As a result the characteristic function for this market is defined as

$$v(S) = \underbrace{\mathcal{G}(\mathcal{V}; \mathcal{M}_S)}_{\text{value of the model}} + \sum_{j \in S} \big[\underbrace{\mathcal{G}(\mathcal{V}; \mathcal{M}_S) - \mathcal{G}(\mathcal{V}; \mathcal{M}_j)}_{\text{model value for party } j} \big]$$

This function could be seen as the value of the model trained on the datasets of all parties in S plus marginal gains for each party. Note that for a single party the value of the data is expressed as the value of the model trained on its own training dataset.

Payment Division The total amount, $\mathbf{a}$, that is distributed among the participants depends on the individual gains obtained from the final model. Let $a_i = \mathcal{G}(\mathcal{V}, \mathcal{M}_{\mathbf{M}}) - \mathcal{G}(\mathcal{V}, \mathcal{M}_i)$. Then, $\mathbf{a} = \sum_{i \in \mathbf{M}} a_i$. We use Shapley values for characteristic function v to determine the distribution of $\mathbf{a}$ for each party i: $b_i = \mathbf{a} \times \psi(v, i)$. In total, party i obtains $t_i = (A_i - a_i) + b_i$ where $(A_i - a_i)$ is the return of the original investment if the final model performance is not 1. Hence, party i gains/loses the following amount by participating in the market: $(A_i - a_i) + b_i - A_i = \psi(v, i)\mathbf{a} - a_i$. Note that i gets $\psi(v, i)$ portion from each a_j and it pays $1 - \psi(v, i)$ of a_i to the market. In the full version of the paper we show that this market has the following properties: balance, symmetry, zero element buyer, and robustness to data replication.

References

[1] Ruoxi Jia, David Dao, Boxin Wang, Frances Ann Hubis, Nick Hynes, Nezihe Merve Gurel, Bo Li, Ce Zhang, Dawn Song, and Costas Spanos. Towards efficient data valuation based on the Shapley value. *arXiv preprint arXiv:1902.10275*, 2019.

[2] Amirata Ghorbani and James Zou. Data Shapley: Equitable valuation of data for machine learning. *arXiv preprint arXiv:1904.02868*, 2019.

[3] Lloyd S Shapley. A value for n-person games. *Contributions to the Theory of Games*, 2(28):307–317, 1953.

[4] Anish Agarwal, Munther Dahleh, and Tuhin Sarkar. A marketplace for data: An algorithmic solution. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, pages 701–726. ACM, 2019.

[5] Matthew Hoekstra, Reshma Lal, Pradeep Pappachan, Carlos Rozas, Vinay Phegade, and Juan del Cuvillo. Using innovative instructions to create trustworthy software solutions. In *Workshop on Hardware and Architectural Support for Security and Privacy (HASP)*, 2013.

[6] Olga Ohrimenko, Felix Schuster, Cédric Fournet, Aastha Mehta, Sebastian Nowozin, Kapil Vaswani, and Manuel Costa. Oblivious multi-party machine learning on trusted processors. In *USENIX Security Symposium*, 2016.

[7] Olga Ohrimenko, Shruti Tople, and Sebastian Tschiatschek. Collaborative machine learning markets with data-replication-robust payments. Technical Report MSR-TR-2019-27, Microsoft, November 2019.

[8] Anupam Datta, Shayak Sen, and Yair Zick. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 IEEE symposium on security and privacy (SP)*, pages 598–617. IEEE, 2016.

[9] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning (ICML)*, pages 1885–1894, 2017.

[10] Lingjiao Chen, Paraschos Koutris, and Arun Kumar. Model-based pricing for machine learning in a data marketplace. *arXiv preprint arXiv:1805.11450*, 2018.

[11] Jacob D. Abernethy and Rafael M. Frongillo. A collaborative mechanism for crowdsourcing prediction problems. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.*, pages 2600–2608, 2011.