Data Exchange Markets via Utility Balancing

Aditya Bhaskara
University of Utah
Salt Lake City, UT, USA
bhaskaraaditya@gmail.com

Sreenivas Gollapudi
Google Research
Mountain View, CA, USA
sgollapu@google.com

Sungjin Im
University of California
Merced, CA, USA
sim3@ucmerced.edu

Kostas Kollias
Google Research
Mountain View, CA, USA
kostaskollias@google.com

Kamesh Munagala
Duke University
Durham, NC, USA
kamesh@cs.duke.edu

Govind S. Sankar
Duke University
Durham, NC, USA
govind.subash.sankar@duke.edu

ABSTRACT

This paper explores the design of a balanced data-sharing marketplace for entities with heterogeneous datasets and machine learning models that they seek to refine using data from other agents. The goal of the marketplace is to encourage participation for data sharing in the presence of such heterogeneity. Our market design approach for data sharing focuses on interim utility balance, where participants contribute and receive equitable utility from refinement of their models. We present such a market model for which we study computational complexity, solution existence, and approximation algorithms for welfare maximization and core stability. We finally support our theoretical insights with simulations on a mean estimation task inspired by road traffic delay estimation.

CCS CONCEPTS

• **Theory of computation → Algorithmic game theory**; **Rounding techniques**.

KEYWORDS

Data markets, Utility balancing, Approximation Algorithms, Core Stability

ACM Reference Format:
Aditya Bhaskara, Sreenivas Gollapudi, Sungjin Im, Kostas Kollias, Kamesh Munagala, and Govind S. Sankar. 2024. Data Exchange Markets via Utility Balancing. In *Proceedings of the ACM Web Conference 2024 (WWW '24), May 13–17, 2024, Singapore, Singapore.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3589334.3645364

1 INTRODUCTION

The power of big data comes from the improved decision making it enables via training and refining machine learning models. To unlock this power to the fullest, it is critical to enable and facilitate data sharing among different units in an organization and between different organizations. The market for big data "*accounted for USD 163.5 Billion in 2021 and is projected to occupy a market size of USD 473.6 Billion by 2030 growing at a CAGR of 12.7%*" [27]. Motivated by the emergence of online marketplaces for data such as SnowFlake [11], in this paper we consider the timely question:

> How can we design a principled marketplace for sharing data between entities (organizations or applications) with heterogeneous datasets they own and machine learning models they seek to refine, so that each entity is encouraged to voluntarily participate?

Towards this end, we assume agents have diverse ML models for decision making that they seek to refine with data. At the same time, each agent possesses data that may be relevant to the tasks of other agents. As an example, a retailer may have sales data for certain products in certain geographic locations, but may want data for related products in other markets to make a better prediction of sales trends. This data could be in the hands of competing retailers. Similarly, a hospital system seeking to build its in-house model for a disease condition based on potentially idiosyncratic variables may want patient data from other hospital systems to refine this model.

In our paper, we assume the participants in the market have no value for money. We further assume that the agents seeking data are the same as those seeking to refine models. Therefore we consider an exchange economy without money as opposed to a two-sided market with buyers and sellers. This is a reasonable assumption for non-profits such as hospital systems or universities, where student or patient data can be "exchanged" but not sold for profit. Though we seek a market design without money, the agents in the market still need to be incentivized to voluntarily participate in the market and exchange data, and this is the main focus of our design.

In the settings we consider, data is often sensitive and private [2, 15]. As in [2], we address this issue by having a trusted central entity (or clearinghouse) with who all agents share their ML tasks and datasets. This entity can refine or retrain the model for one agent using samples of the data from other agents. For instance, if each agent specifies the gradient of their loss function and their in-house model parameters, the central entity can run stochastic gradient descent to update the parameters using the other data. This way, the central entity can efficiently compute the loss of the refined model and hence the utility of a collection of datasets to a model. By using a utility sharing method such as Shapley value that has been well-studied in machine learning [16, 17], the entity can use the same process to attribute this utility gain fairly to the agents that contributed data to the refinement. The entity then

WWW '24, May 13–17, 2024, Singapore, Singapore

ACM ISBN 979-8-4007-0171-9/24/05…$15.00
https://doi.org/10.1145/3589334.3645364

sends the refined models back to the respective agents, preserving data privacy in the process.

1.1 Model and Results

Our approach to market design for data exchange without money is to view it as *utility balancing* – to encourage voluntary participation, an agent should contribute as much utility to other agents as they receive from them. In market design terminology, this corresponds to having a common endogenous price per unit utility bought or sold, so that each agent is revenue-neutral. This can be viewed as a form of fairness in the exchange. The goal of the central entity is to find the right amount of data any set of agents should exchange, so that the overall solution is utility balanced. The solution is randomized, where for each agent, we compute a distribution over sets of other agents. When this agent chooses a set from this distribution to obtain data from, then utility balance holds in expectation (or *interim*). We motivate interim balance in settings where the same agents trade over many epochs so that the total utility across these epochs approaches its expectation. The objective of the central entity could be to either social welfare or fairness in the utilities of agents.

We call this overall problem the Data Exchange Problem. We study computational complexity and existence results for the Data Exchange Problem under natural utility functions and how that utility is shared among contributors. Our main results are the following, most of which are in the full paper [8].

(1) We present a formal model for the Data Exchange Problem in Section 2 based on interim utility balancing, codifying the objectives of welfare maximization and stability.
(2) We show NP-Hardness and polynomial time approximation algorithms for welfare maximization (Section 3). We present a logarithmic approximation in Theorem 4 for submodular utilities and a general class of sharing rules that includes the well-known Shapley value. We present a PTAS for concave utilities with proportional sharing.
(3) We show that the same solution framework also handles the case where the balance condition can be relaxed by compensating or extracting payment from agents using a convex function on the extent of imbalance.
(4) We show the existence of core stable and strategyproof solutions and the trade-offs achievable between these notions and welfare. We also show that a specific type of stable and strategyproof solution can be efficiently computed via greedy matchings.
(5) In Section 4 finally perform simulations on a road network where agents are paths that are interested in minimizing sample variance. We show that our approximation algorithms significantly outperform a pairwise trade benchmark, showing the efficacy of our model and algorithms.

We present the statements of these results more formally in Section 2 after we present the formal mathematical model.

1.2 Related Work

The emerging field of data markets already has unearthed several novel challenges in data privacy, market design, strategyproofness, and so on. Please see recent work [2, 15, 22] for a comprehensive enumeration of research challenges. Our paper proposes a market design via a central clearinghouse and utility balancing, with computational and stability analysis.

Exchange Economies. Our paper falls in the framework of *market design*. Though market design for exchange economies – where agents voluntarily participate in trade given their utility functions and the market constraints – is a classic problem, much of this work concerns markets for goods that cannot be freely replicated. The key challenge in our setting is that data can be freely replicated, which makes the market design problem very different.

There are two classic exchange economies that relate to our work – the trading of indivisible goods [30] and market clearing [4]. The first classic problem is also termed the *house allocation problem*. Here, every agent owns a house and has a preference ordering over other houses. The goal is to allocate a house to each agent in a fashion that lies in the core: No subset of agents can trade houses and improve their outcome. Shapley and Scarf [30] showed that the elegant *top trading cycles* algorithm finds such a core-stable allocation. A practical application of this framework is to *kidney exchanges* [5, 28], which is widely studied and implemented. Our problem falls in the same framework as house allocation, albeit with data instead of houses. Data is a replicable resource, and leads to complex utilities for agents; these aspects make the algorithm design problem very different, as we compare in the full paper [8].

In the same vein, the second classic problem of market clearing for non-replicable goods dates back to Arrow and Debreau [4], and has elegant solutions via equilibrium pricing of the goods. However, equilibrium prices are harder to come by for *replicable* digital goods such as music or video [20]. We bypass this issue by having a common price per unit of utility traded, which translates, via eliminating the price, to our flow formulation on utilities.

Federated Learning. Our work is closely related to recent work by Donahue and Kleinberg [13, 14] on forming coalitions for data exchange in federated learning. However, in their settings, all agents have the same learning objective (either regression or mean estimation), but have data with different bias, leading to local models with different bias. The goal is to form coalitions where the error of the model for individual agents, measured against their own data distribution, is minimized. The authors present optimal coalitional structures for maximizing welfare, as well as achieving core stability. Similarly, the work of Rasouli and Jordan [26] considers data sharing where all agents have similar preferences over other agents. In contrast, we consider agents with *heterogeneous* tasks and data requirements, which makes even welfare maximization NP-Hard.

Pricing and Shapley Value. In the settings we study, agents are both producers and consumers of data, motivating an exchange economy like the works cited above. When sellers of data are distinct from buyers, various works [2, 7, 10, 12, 18] have studied pricing and incentives for selling aspects such as privacy and accuracy. See [24] for a survey.

One important aspect of our work is allocating utility shares to the agents contributing data. For most of our paper, we adopt the Shapley value [31]. Though this method has its roots in cost sharing in Economics, it has seen a resurgence in interest as a method to measure the utility of individual datasets for a machine learning

task [16, 17]. This method has many nice properties; see [2] for a discussion of these properties in a data sharing context. We note that our work presents a general framework and as we show in the paper, it can be adapted to other utility sharing rules.

2 THE DATA EXCHANGE PROBLEM AND OUR RESULTS

Without further ado, we formally present the Data Exchange Problem and a summary of our results. We are given a set of agents X. Each agent $i \in X$ has a dataset $\mathcal{D}_i$ and a machine learning task t_i. (Our results easily extend to the setting where each agent has multiple datasets and tasks.) The accuracy of the task t_i can be improved if agent i obtains the datasets of other users.

2.1 Utility Functions

Suppose agent i obtains the datasets $\cup_{j\in S}\mathcal{D}_j$ of a subset S of agents, then the improvement in accuracy is captured by a *utility function* $u_i(S)$. We assume this function can be computed efficiently for a given set S of agents. Further, this set function is assumed to satisfy the following:

- **Non-negativity and Boundedness:** $u_i(S) \in [0, 1]$ for all $S \subseteq X \setminus \{i\}$. Furthermore, $u_i(\emptyset) = 0$. By scaling, we can also assume that $\max_i u_i(X) = 1$.
- **Monotonicity:** $u_i(S) \geq u_i(T)$ for all $T \subset S$.
- **Submodularity:** This captures diminishing returns from obtaining more data. For all $T \subset S$ and $q \notin S$, we have $u_i(S \cup \{q\}) - u_i(S) \leq u_i(T \cup \{q\}) - u_i(T)$.

A special case of submodular utilities is the **symmetric weighted** setting: Here, there is a concave non-decreasing function f_i for each agent i. Suppose agent j's dataset that she contributes to i has size s_{ij}, then we have $u_i(S) = f_i\left(\sum_{j\in S} s_{ij}\right)$. In other words, the utility only depends on the total size of the datasets contributed by the agents in S.

Example 1. *Suppose each agent i is interested in estimating the population mean of data in its geographical vicinity, and its utility function is the improvement in variance of this estimate. In this case, agent j can contribute s_{ij} amount of data to agent i, and we let $D_i(S) = \sum_{j\in S} s_{ij}$. Assuming $s_{ii} = 1$ and that these data are drawn i.i.d. from a population with variance σ_i^2, we have $u_i(S) = \sigma_i^2\left(1 - \frac{1}{1+D_i(S)}\right)$ and falls in the symmetric weighted setting.*

Continuous Utilities. Though the bulk of the paper focuses on utilities modeled as set functions, in the full paper [8], we also consider the setting where agents can exchange fractions of data. Suppose agent j transfers y_{ij} fraction of her data to agent i, then agent i's utility is modeled as a continuous, monotonically non-decreasing function $u_i(\vec{y_i}) \in [0, 1]$, where $\vec{y_i} = \langle y_{i1}, y_{i2}, \ldots\rangle$. As we show later, such utilities lead to more tractable algorithmic formulations.

2.2 Utility Sharing

The utility $u_i(S)$ that i gains from the set S of agents is attributed to the agents in S according to a fixed rule. We let $h_{ij}(S)$ denote the contribution of agent $j \in S$ to the utility $u_i(S)$, so that $\sum_{j\in S} h_{ij}(S) = u_i(S)$. In this paper, we consider two classes of sharing rules that have been studied in cooperative game theory, and more recently in machine learning:

- **Shapley Value:** This is a classic "gold-standard" rule from cooperative game theory [16, 17, 31], and works as follows: Take a random permutation of the agents in S. Start with W as the empty set and consider adding the agents in S one at a time to W. At the point where j is added, let $\Delta_j = u_i(W \cup \{j\}) - u_i(W)$ be the increase in utility due to the datasets in W. The Shapley value $h_{ij}(S)$ is the expectation of Δ_j over all random permutations of S.
- **Proportional Value:** In this class of rules [9, 21], there is a fixed set of weights $\{w_{ij}\}$, and we define $h_{ij}(S) = \frac{w_{ij}}{\sum_{k\in S} w_{ik}} \cdot u_i(S)$. The natural special case is the setting $w_{ij} = u_i(\{j\})$, so that the utility is shared proportionally to how much j's dataset would have individually contributed to i.

For submodular utilities, the Shapley value satisfies a property called *cross-monotonicity* [23]: if $T \subset S$ and $j \in T$, then $h_{ij}(T) \geq h_{ij}(S)$. Note that there is an entire class of rules that satisfy cross-monotonicity for submodular utilities; please see [16, 17] for a detailed discussion of the Shapley value and related cross-monotonic rules in the context of machine learning. In contrast, the proportional value does not satisfy this property. We contrast these rules in the following example.

Example 2. *There are n agents each contributing data to agent 0. The first $n-1$ agents have identical data, so that $u_0(S) = 0.5$ for any non-empty $S \subseteq [n-1]$. Agent n has a unique dataset so that $u_0(\{n\}) = 0.5$, and $u_0(S \cup \{n\}) = 1$ for any non-empty $S \subseteq [n-1]$. Then, for $S \subseteq [n-1]$, the Shapley value is $h_{0n}(S \cup \{n\}) = 0.5$ and $h_{0j}(S \cup \{n\}) = \frac{1}{2|S|}$ for $j \in S$. However, the proportional share with $w_{ij} = u_i(\{j\})$ is $h_{0j}(S \cup \{n\}) = \frac{1}{|S|+1}$ for all $j \in S \cup \{n\}$.*

In the above example, the Shapley value is more reflective of the actual contributions of the individual agents compared to proportional value; however, the latter rule sometimes leads to better algorithmic results. In particular, for continuous concave utilities and the symmetric weighted setting, the proportional sharing rule is more tractable, while for general submodular utilities, the Shapley value is more tractable.

2.3 Constraints for Data Exchange: Utility Flow

We now present the constraints of the Data Exchange Problem. We assume there is a central entity that computes this exchange. The key constraint is that each agent receives as much utility from the exchange as it contributes. In this exchange, each agent i is associated with a distribution $\{x_{iS}\}$ over sets $S \subseteq X \setminus \{i\}$ of agents whose datasets she could receive. In other words, with mutually exclusive probability x_{iS}, agent i receives the datasets from S and receives utility u_{iS} as a result.

The first constraint encodes that $\{x_{iS}\}$ define a probability distribution over possible sets S.

$$\forall\, i, \sum_S x_{iS} \leq 1 \tag{1}$$

where the remaining probability is assigned to $S = \emptyset$.

The BALANCE condition captures that the expected utility contributed by an agent to other agents is equal to the expected utility she receives.

$$\forall\, i, \sum_S \sum_{j \in S} h_{ij}(S) x_{iS} = \sum_j \sum_{S | i \in S} h_{ji}(S) x_{jS} \tag{2}$$

Note that the balance condition is *interim*, meaning it holds for the expected utility. Any solution that satisfies the BALANCE condition subject to Eq. (1) is said to be a *feasible* solution to the DATA EXCHANGE PROBLEM.

Remarks. First note that for deterministic exchange where $x_{iS} \in \{0, 1\}$, the balance constraints may cause very low utility or lack of a feasible solution. This motivates our use of randomization and interim balance. A randomized solution is justified when agents interact over many epochs with different datasets and models. Though any specific interaction is ex-post imbalanced, these even out over time by the law of large numbers. Such interim balance also makes our algorithmic problem more tractable.

Next, though we don't discuss it in the paper, it is easy to generalize the model to the setting where each agent i has a collection of datasets and a collection of tasks, and each needs different datasets. Further, in the full paper [8], we discuss the changes that need to be made to the constraints to handle continuous, concave utilities.

Eq. (2) is a strict constraint, and therefore trades off with objectives such as social welfare. In the full paper [8], we also consider the case where the balance can be violated.

Finally, as mentioned before, we assume the clearinghouse has accurate access to all datasets and tasks, and can hence compute utilities, their shares, and the feasible DATA EXCHANGE solution. We ignore strategic misreporting on the part of the agents for most of the paper, but we will discuss this aspect and its trade-off with other objectives in the full paper [8].

2.4 Social Welfare Objective

Our goal is to find the optimal DATA EXCHANGE subject to feasibility. Towards this end, we mainly consider the *social welfare* objective where the goal is to find the distributions $\{x_{iS}\}$ that maximizes:

$$\text{Social Welfare} = \sum_{i \in X} \sum_{S \subseteq X \setminus \{i\}} u_i(S) x_{iS} = \sum_{i \in X} \sum_{S \subseteq X \setminus \{i\}} \sum_{j \in S} x_{iS} h_{ij}(S). \tag{3}$$

We will study the computational complexity of this problem.

Remark about running times. Throughout, we assume that there is an efficient subroutine MLSUB that given an agent i and set $S \subseteq X \setminus \{i\}$ returns the utility $u_i(S)$ and the shares $h_{ij}(S)$ for all $j \in S$. We remark that by "polynomial" running time, we mean polynomially many calls to MLSUB, combined with polynomially many ancillary computations. Such an approach decouples the exact running time of MLSUB from our results. For ML tasks, estimating $u_i(S)$ will require retraining the model using data from S; this can typically be done efficiently. Further, estimate $h_{ij}(S)$ can be done to a good approximation via sampling permutations; see [2, 17].

Computational complexity of welfare maximization. The welfare maximization problem is a linear program with $2n$ constraints, so that the optimum solution has at most $2n$ non-zero variables. Nevertheless, we show NP-HARDNESS by a reduction from EXACT 3-COVER. We note that the hardness result holds even when for any S, both $u_i(S)$ and $h_{ij}(S)$ are computable in near-linear time.

THEOREM 3 (PROVED IN THE FULL PAPER [8]). *The welfare maximization objective in* DATA EXCHANGE *is* NP-HARD *for submodular utilities and Shapley value sharing.*

In Section 3, we develop polynomial time algorithms that multiplicatively approximate social welfare.[1] Our algorithms achieve approximate feasibility, where we relax the BALANCE constraint to ϵ-BALANCE (where $\epsilon \in (0, 1)$):

$$\left| \sum_S \sum_{j \in S} h_{ij}(S) x_{iS} - \sum_j \sum_{S | i \in S} h_{ji}(S) x_{jS} \right| \le \epsilon \qquad \forall i. \tag{4}$$

The running times we achieve are now polynomial in $\frac{1}{\epsilon}$, with the assumption that there are analogously many calls to MLSUB. We show the following theorem in Section 3; the precise running time and approximation factors are presented there.

THEOREM 4 (PROVED IN SECTION 3). *We can achieve the following approximation factors to the social welfare objective for* DATA EXCHANGE *via an algorithm that runs in time polynomial in the input size and $\frac{1}{\epsilon}$ and finds a feasible solution that satisfies ϵ-*BALANCE*:*

- *A $O(\log n)$ approximation for arbitrary submodular utilities[2] and any cross-monotonic utility sharing rule (including the Shapley value rule).*
- *A $1 + \epsilon$ approximation for symmetric weighted setting and proportional value with $w_{ij} = s_{ij}$.*

Our results follow by writing the social welfare optimization problem as a Linear Program (LP) with exponentially many variables of the form $\{x_{iS}\}$. Since the number of feasibility constraints is $2n$, we use the multiplicative weight method to approximately solve it. This requires developing a dual oracle for the constraints, which for each agent i, is a constrained maximization problem over a weighted sum of $\{h_{ij}(S)\}$, and we need to find the set $S \subseteq X \setminus \{i\}$ that maximizes this weighted sum. We show approximation algorithms for this problem, leading to the proof of the above theorem.

Further, in the full paper [8], we show the following theorem:

THEOREM 5 (PROVED IN THE FULL PAPER [8]). *For* DATA EXCHANGE *with continuous concave utility functions and proportional sharing, for any $\epsilon \in (0, 1)$, there is an algorithm running in time polynomial in the input size and $\frac{1}{\epsilon}$ and that finds a $(1+\epsilon)$ approximation to social welfare, while violating* BALANCE *by an additive ϵ.*

Finally, in the full paper [8], we show that the same solution ideas extend to the case where the balance condition Eq. (2) can be violated by compensating/extracting payments from agents using a convex function of the extent of imbalance in the utilities. Such an approach can lead to much larger social welfare.

[1] By α-approximation for $\alpha \ge 1$, we mean our algorithm achieves at least $\frac{1}{\alpha}$ fraction of the optimal social welfare.

[2] The results hold for arbitrary monotone utilities and only require cross-monotonic sharing; however, cross-monotonicity typically does not hold unless utilities are submodular.

2.5 Core Stability and Strategyproofness

Stability is a widely studied notion in cooperative game theory, and seeks solutions that are robust to coalitional deviations. In our context, we have the following definition.

Definition 6. *A feasible solution $\mathcal{F}$ to* Data Exchange *is* core stable *if there is no $U \subseteq X$ of users and another feasible solution $\mathcal{F}'$ just on the users in U such that for all $i \in U$, $u_i(\mathcal{F}') > u_i(\mathcal{F})$. A solution $\mathcal{F}$ is c-stable if there is no such U with $|U| \leq c$.*

In other words, suppose a coalition $U \subseteq X$ of agents deviates and trades just among themselves via a feasible solution $\mathcal{F}'$ so that all their utilities improve, then this coalition is *blocking*. A core solution has no blocking coalitions.

In the full paper [8], we first show that regardless of the utility function and choice of sharing rule, there is always a feasible Data Exchange solution that is core-stable to an arbitrarily good approximation. This is a consequence of Scarf's lemma [29] from cooperative game theory. Though it is unclear how to efficiently compute such a solution in general, we show an algorithm to find a 2-stable solution via Greedy maximal weight matching.

In the full paper [8], we next study the trade-off between core and welfare. On the negative side, we show an instance in the symmetric weighted setting with proportional sharing, where any core solution has social welfare that is $\Omega(\sqrt{n})$ times smaller than the optimal social welfare, showing the two concepts of core and welfare maximization can be far from each other. Nevertheless, we show how to achieve approximate core-stability and social welfare simultaneously via randomizing between them.

We finally consider strategic behavior by agents, where they hide either their tasks or data. We define feasible misreports, and again show that for the symmetric weighted setting, strategyproofness and approximate welfare maximization are simultaneously incompatible. On the positive side, we show that a Greedy cycle canceling algorithm that generalizes greedy matching is strategyproof.

3 ALGORITHMS FOR WELFARE MAXIMIZATION: PROOF OF THEOREM 4

In this section, we present approximation algorithms for welfare maximization. We present the overall framework in Section 3.1, which reduces the problem to solving an oracle problem, one for each agent (Eq. (10)), so that an approximation algorithm to the oracle translates to the same approximation to welfare maximization, while achieving ϵ-balance (Eq. (4)). We present the approximations to the oracle for submodular utilities with Shapley value in Section 3.3, and for symmetric weighted concave utilities with proportional sharing in the full paper [8]. We also present an extension to continuous concave utilities with proportional sharing in the full paper [8].

As mentioned before, the welfare maximization problem can be written as an exponential-sized LP, where the non-negative variables are $\{x_{iS}\}$; the objective is to maximize Eq. (3) subject to the constraints Eqs. (1) and (2).

3.1 Multiplicative Weight Algorithm

We solve this using the multiplicative weights framework of Plotkin, Shmoys, and Tardos (PST) [25]. Since our final solution loses an additive ϵ in the balance constraints (Eq. (2)), we assume at the outset that these constraints are violated by an additive ϵ, that is, Eq. (4). The problem with relaxed constraints can only have a larger objective value (social welfare). The relaxation helps us achieve polynomial running time.

Lemma 7. *Let OPT denote the optimal solution value to the instance with relaxed balance constraints. Then $OPT \geq \epsilon$.*

Proof. To see this, recall that we assumed $\max_i u_i(X) = 1$. For the maximizer i, set $x_{iX} = \epsilon$ and set all other variables to zero. This gives us a guarantee that $OPT \geq \epsilon$. □

Now, we try all objective values in powers of $(1+\epsilon)$ using binary search. Consider some guess B for this value; we want to check if this value is feasible. By Lemma 7 we assume that $B \geq \epsilon$. We therefore want to check the feasibility of the following LP, where the objective Eq. (3) is encoded in Eq. (5); the balance constraints Eq. (4) is enconded in Eqs. (6) and (7); and the probability constraint Eq. (1) is encoded in Eq. (8). Call this LP1(B, ϵ). Our final solution will correspond to the largest B for which LP1(B, ϵ) is feasible.

$$\text{(LP1)} \qquad \begin{aligned} \sum_{i,j,S} h_{ij}(S) x_{iS} &\geq B && (5)\\ \forall\, i, \sum_{j,S} h_{ij}(S) x_{iS} - \sum_{j,S|i\in S} h_{ji}(S) x_{jS} &\geq -\epsilon && (6)\\ \forall\, i, -\sum_{j,S} h_{ij}(S) x_{iS} + \sum_{j,S|i\in S} h_{ji}(S) x_{jS} &\geq -\epsilon && (7)\\ \forall\, i, \sum_{S} x_{iS} &\leq 1 && (8)\\ \forall\, i, S, \sum_{S} x_{iS} &\geq 0 && (9) \end{aligned}$$

We will use the PST framework to solve the feasibility of the above LP. Let Eqs. (5) to (7) be represented by the coefficient matrices A, b and let P be the polytope of vectors satisfying Eqs. (8) and (9). We are testing whether $\exists ? x \in P, Ax \geq b$. The PST framework requires an oracle to solve $\max_{x\in P} p^\top Ax$ for arbitrary vectors $p \geq 0$. In our setting, this becomes

$$\text{Oracle} = \max_{x\in P} \sum_{i,j,S} Q_{ij} h_{ij}(S) x_{iS}$$

for possibly negative weights Q_{ij}. Since the constraints across i are now independent, the maximum solution will select the optimum solution S to Eq. (10) and sets $x_{iS} = 1$, for each i.

$$\text{Oracle for agent } i = \max_{S} \sum_{j\in S} Q_{ij} h_{ij}(S) \qquad (10)$$

Using a similar proof as Theorem 3, it can be shown the Oracle problem is NP-Hard. We will therefore develop approximation algorithms, and show two such algorithms in the full paper [8]. As we show below, this will translate to an approximation for the social welfare. The overall algorithm is presented in Algorithm 1.

3.2 Analysis

Suppose the multiplicative approximation ratio of the oracle Eq. (10) is $\alpha \geq 1$; this means the oracle subroutine finds a solution whose value is at least OPT/α when OPT is the optimal solution to the oracle. Define ρ be the maximum value that any of the constraints

Algorithm 1 Multiplicative Weights Update to solve LP1.

Choose parameters $\epsilon, \delta \le 1$ and $\eta = \frac{\epsilon}{4n\alpha}$.
Try values for B via in powers of $(1+\delta)$.
Let $A \in \mathbb{R}^{(2n+1)\times n}, b \in \mathbb{R}^{2n+1}$ denote the coefficients of LP1(B, ϵ).
Let $\mathbf{w}^{(1)} = \mathbf{1}^{2n+1}$.
for $t = 1, \ldots, T = \frac{32n^2\alpha^2 \log n}{\epsilon^2}$ **do**
  Let $\mathbf{p}^{(t)} := \frac{\mathbf{w}^{(t)}}{\sum_i w_i^{(t)}}$.
  Let $\mathbf{x}^{(t)}$ be the output of the α-approximate oracle with input $\mathbf{p}^{(t)\top} A\mathbf{x}$.
  if $\mathbf{p}^{(t)\top} A\mathbf{x}^{(t)} < \mathbf{p}^{(t)^\top} \frac{b}{\alpha}$ **then**
    Return infeasible and decrease the guess for B.
  else
    $\mathbf{m}^{(t)} := \frac{1}{\rho}(A\mathbf{x}^{(t)} - \frac{\mathbf{b}}{\alpha})$.
    $\forall\, i, w_i{}^{(t+1)} := w_i^{(t)}(1 - \eta m_t^{(t)})$.
  end if
end for
Return $\bar{\mathbf{x}} = \frac{\sum_i \mathbf{x}^{(t)}}{T}$.

in $Ax \ge b, x \in P$ can be additively violated. Since we assume $u_i(X) \le 1$ for all i, it is clear that $\rho = \sum_i u_i(X) \le n$.

Our main theorem is the following.

Theorem 8. *Suppose the oracle problem Eq.* (10) *can be solved to a multiplicative approximation factor of α. Then, with $O(\frac{n^2\alpha^2 \log n}{\epsilon^2})$ calls to the oracle subproblem and $O(n)$ time overhead per call to the oracle, Algorithm 1 returns a solution $\mathbf{x}$ that satisfies Eqs.* (6) *to* (9) *and that satisfies:*

$$\sum_{i,j,S} h_{ij}(S) x_{iS} \ge \frac{OPT}{2\alpha(1+3\delta)}.$$

To prove this theorem, we require a result from [3].

Lemma 9 (Theorem 2.1 in [3]). *After T rounds in Algorithm 1, for every i,*

$$\sum_{t=1}^{T} \mathbf{m}^{(t)} \cdot \mathbf{p}^{(t)} \le \sum_{t=1}^{T} m_i^{(t)} + \eta \sum_{t=1}^{T} \left| m_i^{(t)} \right| + \frac{2\log n}{\eta}. \tag{11}$$

Proof of Theorem 8. Suppose the algorithm did T iterations without declaring infeasibility. Since the algorithm did not declare it infeasible, then we have that

$$\mathbf{p}^{(t)\top} A\mathbf{x}^{(t)} \ge \mathbf{p}^{(t)^\top} \frac{b}{\alpha}$$

for every time step t. Thus, the left hand side of Eq. (11) is non-negative.

$$\begin{aligned} 0 &\le \sum_{t=1}^{T} m_i^{(t)} + \eta \sum_{t=1}^{T} \left| m_i^{(t)} \right| + \frac{2\log n}{\eta} \\ &= \frac{1}{n} \sum_{t=1}^{T} (A_i \mathbf{x}^{(t)} - \frac{b_i}{\alpha}) + \eta T + \frac{2\log n}{\eta} \end{aligned}$$

Dividing by T, and choosing $\eta = \frac{\epsilon}{4n\alpha}$ and $T = \frac{32n^2\alpha^2 \log n}{\epsilon^2}$, we get

$$A_i \bar{\mathbf{x}} \ge \frac{b_i}{\alpha} - \eta n - \frac{2n \log n}{\eta T} \implies A_i \bar{\mathbf{x}} \ge \frac{b_i}{\alpha} - \frac{\epsilon}{2\alpha}.$$

The theorem statement then follows by choosing $\delta \le \frac{1}{3}$ and with the observation that for some guess B for the optimal value, we have $B \ge \frac{OPT}{1+\delta} \ge \frac{\epsilon}{1+\delta}$. □

3.3 Oracle for Cross-monotonic Sharing

We now consider the case where $h_{ij}(S)$ is cross-monotonic in S, and $u_i(S)$ is a non-decreasing submodular set function. Note that cross-monotonicity captures the Shapley value. We will present a $O(\log n)$ approximation to the oracle (Eq. (10)) for this setting, which when combined with Theorem 8, completes the proof of the first part of Theorem 4. The key hurdle with devising an approximation algorithm is that the quantities Q_{ij} in Eq. (10) can be negative; we show this is not an issue for cross-monotonic sharing.

Simplifying the data exchange *problem.* Before considering the oracle problem (Eq. (10)), we consider the overall data exchange problem (Eqs. (5) to (9)) and show some bounds for it. Let $u_{ij} := h_{ij}(\{j\}) = u_i(\{j\})$. Note that by cross-monotonicity, we have $h_{ij}(S) \le u_{ij}$ for all $j \in S$.

Lemma 10. *By losing a multiplicative factor of $(1-\epsilon)$ in social welfare, for every i, we can set $x_{iS} = 0$ for any S that contain some j such that $u_{ij} := h_{ij}(\{j\}) \le \frac{\epsilon^2}{n^2}$.*

Proof. Fix some i. Let $S_{\text{small}} = \left\{ j \mid h_{ij}(\{j\}) \le \frac{\epsilon^2}{n^2} \right\}$. Consider any solution $\mathbf{x}$. We claim that modifying $\mathbf{x}$ such that we add the value of x_{iS} to $x_{iS \setminus S_{\text{small}}}$, and set $x_{iS} = 0$ only loses $(1-\epsilon)$ factor in the objective. Since the utility sharing rule is cross-monotone, for any set S we have $h_{ij}(S \setminus S_{\text{small}}) \ge h_{ij}(S)$ for all $j \in S \setminus S_{\text{small}}$. Further, we have $h_{ij}(S_{\text{small}}) \le h_{ij}(\{j\})$ for all $j \in S_{\text{small}}$. Therefore, we have

$$\begin{aligned} u_i(S) = \sum_{j \in S} h_{ij}(S) &= \sum_{j \in S_{\text{small}}} h_{ij}(S) + \sum_{j \in S \setminus S_{\text{small}}} h_{ij}(S) \\ &\le \sum_{j \in S_{\text{small}}} h_{ij}(\{j\}) + \sum_{j \in S \setminus S_{\text{small}}} h_{ij}(S \setminus S_{\text{small}}) \\ &\le \frac{\epsilon^2}{n} + u_i(S \setminus S_{\text{small}}). \end{aligned}$$

Adding up the losses, we lose a $\frac{\epsilon^2}{n}$ for each user i, leading to a loss of ϵ^2 overall. By Lemma 7, the initial optimum was at least ϵ. We therefore lose a factor of at most $(1-\epsilon)$ in social welfare. □

We therefore assume $x_{iS} = 0$ for all S s.t. $j \in S$ and $u_{ij} < \frac{\epsilon^2}{n^2}$.

Approximating the Oracle. For agent i, let

$$S^* = \arg\max \sum_{j \in S} Q_{ij} h_{ij}(S) \qquad OPT = \sum_{j \in S^*} Q_{ij} h_{ij}(S^*).$$

For given $\epsilon > 0$, the algorithm works as follows:

(1) Guess OPT in powers of $(1+\epsilon)$ by binary search.
(2) For constant $\delta = e - 1$, divide the agents into buckets based on the Q_{ij} value. The k^{th} bucket B_k is defined as

$$B_k = \{ j \mid Q_{ij} \in \left(u_0(1+\delta)^k, u_0(1+\delta)^{k+1} \right] \}$$

where $u_0 = \frac{\epsilon \cdot OPT}{n}$ and $k \in \{0, 1, \ldots, 3\lceil \log_{1+\delta}(\frac{n}{\epsilon})\rceil - 1\}$.
(3) For each bucket B_k, let $V_k = \sum_{j \in B_k} Q_{ij} h_{ij}(B_k)$.
(4) For this guess of OPT, the final solution is S_z where $z = \text{argmax}_k V_k$.
(5) The solution is valid for this value of OPT if $V_z \geq OPT/\hat{\alpha}$, where $\hat{\alpha} = 3e(1+3\epsilon)\ln n$. We use the largest OPT for which the solution returned is valid, and return this solution.

In the analysis below, we assume OPT can be precisely guessed.

Theorem 11. *For $\epsilon > 0$, when utilities $u_i(S)$ are monotone non-decreasing in S and the utility sharing rule is cross-monotone, the* Oracle *problem can be approximated to factor $\alpha \leq 3e(1+2\epsilon)\ln n$ in $O(\frac{n \log n}{\log(1+\epsilon)})$ time and correspondingly many calls to* MLSub.

Proof. Let $S_0 = \{j \in S^* | Q_{ij} < 0\}$. We have:

$$\sum_{j \in S^*} Q_{ij} h_{ij}(S^*) = \sum_{j \in S_0} Q_{ij} h_{ij}(S^*) + \sum_{j \in S^* \setminus S_0} Q_{ij} h_{ij}(S^*)$$
$$\leq \sum_{j \in S^* \setminus S_0} Q_{ij} h_{ij}(S^*) \leq \sum_{j \in S^* \setminus S_0} Q_{ij} h_{ij}(S^* \setminus S_0).$$

where the final inequality follows by cross-monotonicity. Since S^* is optimal, this means $S_0 = \emptyset$. Therefore, we assume $Q_{ij} > 0$.

Next note that $OPT \geq \sum_{j \in S} Q_{ij} h_{ij}(S)$ for $S = \{j\}$, which means $Q_{ij} u_{ij} \leq OPT$ for all j. Given constant $\epsilon \in (0, 1]$, let $S_{\text{small}} = \left\{ j \mid Q_{ij} u_{ij} < \epsilon \cdot \frac{OPT}{n} \right\}$. By the same argument as in the proof of Lemma 10, we can restrict to agents in $X \setminus S_{\text{small}}$ by losing a $(1-\epsilon)$ factor in OPT. Let $\hat{X} = X \setminus S_{\text{small}}$, so that these are now the only agents of interest. The above implies $Q_{ij} u_{ij} \in OPT \cdot \left[\frac{\epsilon}{n}, 1\right]$ for $j \in \hat{X}$. Since $u_{ij} \in \left[\frac{\epsilon^2}{n^2}, 1\right]$ by Lemma 10, this implies $Q_{ij} \in OPT \cdot \left[\frac{\epsilon}{n}, \frac{n^2}{\epsilon^2}\right]$. Therefore, the buckets constructed by the algorithm only use agents from $\hat{X}$.

Let $\hat{S} = S^* \cap \hat{X}$. By the Pigeonhole principle, the elements of some bucket must contribute at least $\frac{\log(1+\delta)}{3 \log \frac{n}{\epsilon}}$ fraction of the objective, OPT. Suppose this is the k^{th} bucket B_k. We therefore have:

$$\frac{\log(1+\delta)}{\log \frac{n}{\epsilon}} \cdot OPT \leq \sum_{j \in \hat{S} \cap B_k} Q_{ij} h_{ij}(\hat{S}) \leq (1+\delta)^{k+1} \sum_{j \in \hat{S} \cap B_k} h_{ij}(\hat{S}).$$

Suppose we choose B_k as the solution instead. We have

$$\sum_{j \in B_k} Q_{ij} h_{ij}(B_k) \geq (1+\delta)^k \sum_{j \in B_k} h_{ij}(B_k) = (1+\delta)^k u_i(B_k)$$
$$\geq (1+\delta)^k u_i(B_k \cap \hat{S}) = (1+\delta)^k \sum_{j \in B_k \cap \hat{S}} h_{ij}(B_k \cap \hat{S})$$
$$\geq (1+\delta)^k \sum_{j \in B_k \cap \hat{S}} h_{ij}(\hat{S}) \geq \frac{(1-\epsilon)\log(1+\delta)}{3(1+\delta)\log \frac{n}{\epsilon}} OPT.$$

Here, the second inequality holds because u_i is monotonically non-decreasing, and the next inequality holds since h_{ij} is cross-monotone, so that $h_{ij}(B_k \cap \hat{S}) \geq h_{ij}(\hat{S})$. Thus, the largest of the solutions V_k is a $\frac{3(1+\delta)\log \frac{n}{\epsilon}}{(1-\epsilon)\log(1+\delta)}$-approximation to the optimal solution. This is minimized at $\delta = e - 1$, giving us an approximation ratio of $\frac{3e \log \frac{n}{\epsilon}}{(1-\epsilon)} \leq 3e(1+2\epsilon)\log n$ for $\epsilon < \frac{1}{2}$ and for large enough n.

We can execute this algorithm in almost linear time in the following way: guess the right value of OPT by a binary search, which takes $O(\frac{\log n}{\log(1+\epsilon)})$ time to find OPT up to multiplicative error of $(1+\epsilon)$. Throw out all elements that have $Q_{ij} u_{ij} < \frac{\epsilon \cdot OPT}{n}$ and $u_{ij} < \frac{\epsilon^2}{n^2}$, and find the bucket with the largest utility. This takes time $O(n)$, leading to an overall time of $O(\frac{n \log n}{\log(1+\epsilon)})$. □

4 EXPERIMENTS

We will now empirically compare the performance of our approximation algorithm in Section 3 with a no-sharing baseline, and with a pair-wise trade benchmark, showing we outperform both. In our experiments, each agent corresponds to a path in a road network. The delay of each edge in the road network is a random variable and each agent has a set of samples for each edge on its path that it can trade with other agents. The goal of each agent is to trade her samples in order minimize the sample variance in the estimate of the delay on her path.

As motivation, consider trucking or cab companies sharing data to improve each other's routing and demand forecasting models. Vehicles of these entities traverse different sets of routes and collect data on traffic conditions on road segments that they can share to improve the overall routing of other entities that also use these segments. We note that the experiments are intended to be a proof of concept that for a realistic dataset with sufficient complexity, the method shows improvement over simpler baselines. Nevertheless, our dataset has sufficient nuance, for instance, overlap between participants and correlation structure, that the results should carry over to other datasets with this structure.

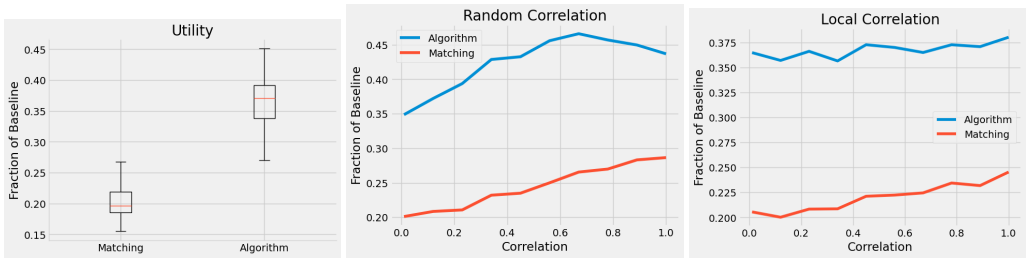
Setup. We sample a random neighborhood of radius 8 from the Manhattan road network in [1]. This will serve as the graph of interest for the rest of the experiment. We have $n = 20$ agents. Each agent i is assigned a path in the graph in the following way: Sample a random node u in the graph. Sample a length t uniformly at random between 5 and the depth of the BFS tree from u. Sample a node v uniformly at random at layer t of the BFS tree. The shortest path from u to v in the graph is the path P_i corresponding to agent i, and she is interested in minimizing the variance of the sample mean of the delay of this path.

The delay of each edge e is a random variable whose variance σ_e^2 is drawn uniformly from $[0, 1]$, independently of other edges. Agent i starts with $z^{(i)}$ data points for the delay of her path P_i, where $z^{(i)}$ is chosen uniformly at random between 2 and 9. Therefore, she starts with $z_e^{(i)} = z^{(i)}$ data points for each edge e in her path.

The agent's objective is to minimize the sum of the sample variances of the delays of the edges in her path P_i. Her initial sample variance is $\frac{\sigma_e^2}{z_e^{(i)}}$ and therefore, her initial total sample variance is

$$\text{Baseline for } i = v_0(i) := \sum_{e \in P_i} \frac{\sigma_e^2}{z_e^{(i)}}$$

Suppose she receives data from a set of other agents S, who collectively give her $z_e^{(S)}$ additional samples for edge e. Then, her utility



(a) (b) (c)

Figure 1: (a) Box plots of the total utility of the algorithm and benchmark (matching) solutions, measured as a fraction of the baseline. (b,c) Total utility of the algorithm and matching benchmark with varying levels of correlation, again measured as a fraction of the baseline. Figure (b) is Random correlation, and (c) is Local correlation.

is defined as the reduction in total sample variance. That is,

$$\text{Utility of } i = u_i(S) = v_0(i) - \sum_{e \in P_i} \frac{\sigma_e^2}{z_e^{(i)} + z_e^{(S)}}.$$

This is a monotonically increasing submodular function. We perform the cost-sharing via the Shapley value. We simulate the Shapley value by taking $m = 10$ random permutations, and use use $\epsilon = 0.01$ as the violation allowed in the BALANCE constraints.

Results. Since the optimal solution to DATA EXCHANGE is NP-Hard, we compare the total utility of our approximation algorithm (Section 3.3) to the baseline sample variance $\sum_i u_0(i)$, where no agents in the solution share their data. As a benchmark, we also find the best solution with trades only between pairs of agents, much like algorithms for kidney exchange. For this, we construct a graph on the agents where the weight for pair (i, j) is the maximum utility of DATA EXCHANGE with ϵ-BALANCE on just these two agents. We then find a maximum weight matching on this weighted graph. (See the full paper [8] for more details.)

In Fig. 1a, we present the total utility of our algorithm and the matching benchmark, measured as a fraction of the baseline sample variance, across several random samples of the road network. Our algorithm outperforms the benchmark, by a factor of 1.8 on average. Note that we can easily construct instances with a single long path with m edges and many paths sharing one edge with this path, where our algorithm outperforms matching by a factor of $\Omega(m)$. The goal of our experiment is to show that our algorithm has a significant advantage even in more realistic settings.

We now introduce *correlation* between the random variables of the edges. In this setting, we assume that correlated edges have their delays sampled from the same distribution. We introduce this correlation in two ways. In *random* correlation (Fig. 1b), we sample pairs of edges uniformly at random and correlate the pair. We measure the correlation (x-axis) as a ratio of the number of pairs sampled to the total number of edges in the graph. In *local* correlation (Fig. 1c), we sample vertices uniformly at random, and correlate all the edges incident to this edge. We measure the correlation (x-axis) as a ratio of the number of vertices sampled to the total number of vertices in the graph.

We measure how the total utility of our algorithm and the matching benchmark changes as a function of the correlation in Figs. 1b and 1c, again measured as a fraction of the baseline sample variance. Our algorithm outperforms the benchmark in both modes of correlation, and at both high and low levels of correlation.

5 CONCLUSION

There are several open questions that arise from our work. First, the approximation ratio for Shapley value sharing is $O(\log n)$ and we have not ruled out the existence of a constant approximation. Secondly, our algorithmic results require utilities to be submodular. Though this is a natural restriction, there are cases where it does not hold. For instance, if each dataset is a collection of features, the effect of combining features could be super-additive [16]. Devising efficient algorithms for special types of non-submodular functions that arise in learning is an interesting open question.

Next, for Shapley value sharing (as opposed to proportional sharing), our negative result for core-stability only shows the absence of a $(2 - \epsilon)$-approximation to welfare. Either strengthening this impossibility result or showing a constant approximation that lies in the exact core would be an interesting question. Further, it would be interesting to study strategyproofness for thick or random markets, analogous to results for stable matchings [6, 19].

Finally, our model can be viewed as budget balance with a single global price per unit utility transferred. Though there are hurdles to defining an Arrow-Debreau type market with endogenous prices for each data type, it would be interesting to define a richer and tractable class of markets along this direction.

Acknowledgment. We thank Jian Pei for several helpful suggestions. Aditya Bhaskara is supported by NSF awards CCF-2008688 and CCF-2047288. Sungjin Im is supported in part by NSF grants CCF-1844939 and CCF-2121745. Kamesh Munagala and Govind S. Sankar are supported by NSF grant CCF-2113798.

REFERENCES

[1] [n. d.]. Street Network of New York in GraphML. https://www.kaggle.com/datasets/crailtap/street-network-of-new-york-in-graphml. Accessed: 2023-09-20.

[2] Anish Agarwal, Munther Dahleh, and Tuhin Sarkar. 2019. A Marketplace for Data: An Algorithmic Solution. In *Proceedings of the 2019 ACM Conference on Economics and Computation* (Phoenix, AZ, USA) *(EC '19)*. Association for Computing Machinery, New York, NY, USA, 701–726. https://doi.org/10.1145/3328526.3329589

[3] Sanjeev Arora, Elad Hazan, and Satyen Kale. 2012. The Multiplicative Weights Update Method: a Meta-Algorithm and Applications. *Theory of Computing* 8, 6 (2012), 121–164. https://doi.org/10.4086/toc.2012.v008a006

[4] Kenneth J. Arrow and Gerard Debreu. 1954. Existence of an Equilibrium for a Competitive Economy. *Econometrica* 22, 3 (1954), 265–290.

[5] Itai Ashlagi, Felix Fischer, Ian A. Kash, and Ariel D. Procaccia. 2015. Mix and match: A strategyproof mechanism for multi-hospital kidney exchange. *Games and Economic Behavior* 91 (2015), 284–296. https://doi.org/10.1016/j.geb.2013.05.008

[6] Itai Ashlagi, Yash Kanoria, and Jacob D. Leshno. 2017. Unbalanced Random Matching Markets: The Stark Effect of Competition. *Journal of Political Economy* 125, 1 (2017), 69–98. https://doi.org/10.1086/689869

[7] Moshe Babaioff, Robert Kleinberg, and Renato Paes Leme. 2012. Optimal Mechanisms for Selling Information. In *Proceedings of the 13th ACM Conference on Electronic Commerce* (Valencia, Spain) *(EC '12)*. Association for Computing Machinery, New York, NY, USA, 92–109. https://doi.org/10.1145/2229012.2229024

[8] Aditya Bhaskara, Sreenivas Gollapudi, Sungjin Im, Kostas Kollias, Kamesh Munagala, and Govind S. Sankar. 2024. Data Exchange Markets via Utility Balancing. arXiv:2401.13053 [cs.GT]

[9] Simina Brânzei, Nikhil Devanur, and Yuval Rabani. 2021. Proportional Dynamics in Exchange Economies. In *Proceedings of the 22nd ACM Conference on Economics and Computation* (Budapest, Hungary) *(EC '21)*. Association for Computing Machinery, New York, NY, USA, 180–201.

[10] Shuchi Chawla, Shaleen Deep, Paraschos Koutrisw, and Yifeng Teng. 2019. Revenue Maximization for Query Pricing. *Proc. VLDB Endow.* 13, 1 (sep 2019), 1–14. https://doi.org/10.14778/3357377.3357378

[11] Snowflake Data Cloud. 2021. https://www.snowflake.com/en/.

[12] Rachel Cummings, Katrina Ligett, Aaron Roth, Zhiwei Steven Wu, and Juba Ziani. 2015. Accuracy for Sale: Aggregating Data with a Variance Constraint. In *Proceedings of the 2015 Conference on Innovations in Theoretical Computer Science* (Rehovot, Israel) *(ITCS '15)*. Association for Computing Machinery, New York, NY, USA, 317–324. https://doi.org/10.1145/2688073.2688106

[13] Kate Donahue and Jon Kleinberg. 2020. Model-sharing Games: Analyzing Federated Learning Under Voluntary Participation. arXiv:2010.00753 [cs.GT]

[14] Kate Donahue and Jon Kleinberg. 2021. Optimality and Stability in Federated Learning: A Game-theoretic Approach. arXiv:2106.09580 [cs.GT]

[15] Raul Castro Fernandez, Pranav Subramaniam, and Michael J. Franklin. 2020. Data Market Platforms: Trading Data Assets to Solve Data Problems. *Proc. VLDB Endow.* 13, 12 (jul 2020), 1933–1947. https://doi.org/10.14778/3407790.3407800

[16] Daniel Fryer, Inga Strümke, and Hien Nguyen. 2021. Shapley values for feature selection: The good, the bad, and the axioms. arXiv:2102.10936 [cs.LG]

[17] Amirata Ghorbani and James Zou. 2019. Data Shapley: Equitable Valuation of Data for Machine Learning. arXiv:1904.02868 [stat.ML]

[18] Arpita Ghosh and Aaron Roth. 2011. Selling Privacy at Auction. In *Proceedings of the 12th ACM Conference on Electronic Commerce* (San Jose, California, USA) *(EC '11)*. Association for Computing Machinery, New York, NY, USA, 199–208. https://doi.org/10.1145/1993574.1993605

[19] Nicole Immorlica and Mohammad Mahdian. 2005. Marriage, Honesty, and Stability. In *Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms* (Vancouver, British Columbia) *(SODA '05)*. Society for Industrial and Applied Mathematics, USA, 53–62.

[20] Kamal Jain and Vijay Vazirani. 2012. Equilibrium Pricing of Semantically Substitutable Digital Goods. arXiv:1007.4586 [cs.GT]

[21] Ramesh Johari and John N. Tsitsiklis. 2004. Efficiency Loss in a Network Resource Allocation Game. *Mathematics of Operations Research* 29, 3 (2004), 407–435.

[22] Aida Manzano Kharman, Christian Jursitzky, Quan Zhou, Pietro Ferraro, Jakub Marecek, Pierre Pinson, and Robert Shorten. 2023. An adversarially robust data-market for spatial, crowd-sourced data. arXiv:2206.06299 [cs.DS]

[23] Hervi Moulin. 1988. *Axioms of Cooperative Decision Making*. Cambridge University Press. https://doi.org/10.1017/CCOL0521360552

[24] Jian Pei. 2022. A Survey on Data Pricing: From Economics to Data Science. *IEEE Transactions on Knowledge and Data Engineering* 34, 10 (oct 2022), 4586–4608.

[25] Serge A. Plotkin, David B. Shmoys, and Éva Tardos. 1995. Fast Approximation Algorithms for Fractional Packing and Covering Problems. *Mathematics of Operations Research* 20, 2 (1995), 257–301.

[26] Mohammad Rasouli and Michael I. Jordan. 2021. Data Sharing Markets. arXiv:2107.08630 [econ.TH]

[27] Acumen Research. 2021. Big Data Market Size: Global Industry, Share, Analysis, Trends and Forecast 2022 - 2030. https://www.acumenresearchandconsulting.com/big-data-market.

[28] Alvin E. Roth, Tayfun Sönmez, and M. Utku Ünver. 2004. Kidney Exchange. *The Quarterly Journal of Economics* 119, 2 (2004), 457–488. http://www.jstor.org/stable/25098691

[29] Herbert E Scarf. 1967. The core of an N person game. *Econometrica: Journal of the Econometric Society* (1967), 50–69.

[30] Lloyd Shapley and Herbert Scarf. 1974. On cores and indivisibility. *Journal of Mathematical Economics* 1, 1 (1974), 23–37. https://doi.org/10.1016/0304-4068(74)90033-0

[31] Lloyd S. Shapley. 1951. *Notes on the N-Person Game II: The Value of an N-Person Game*. RAND Corporation, Santa Monica, CA. https://doi.org/10.7249/RM0670