Knowledge and Information Systems (2022) 64:1417–1455
https://doi.org/10.1007/s10115-022-01679-4

SURVEY PAPER

Data pricing in machine learning pipelines

Zicun Cong[1] · Xuan Luo[1] · Jian Pei[1] · Feida Zhu[2] · Yong Zhang[3]

Received: 14 July 2021 / Revised: 30 March 2022 / Accepted: 1 April 2022 /
Published online: 13 May 2022

Abstract

Machine learning is disruptive. At the same time, machine learning can only succeed by collaboration among many parties in multiple steps naturally as pipelines in an eco-system, such as collecting data for possible machine learning applications, collaboratively training models by multiple parties and delivering machine learning services to end users. Data are critical and penetrating in the whole machine learning pipelines. As machine learning pipelines involve many parties and, in order to be successful, have to form a constructive and dynamic eco-system, marketplaces and data pricing are fundamental in connecting and facilitating those many parties. In this article, we survey the principles and the latest research development of data pricing in machine learning pipelines. We start with a brief review of data marketplaces and pricing desiderata. Then, we focus on pricing in three important steps in machine learning pipelines. To understand pricing in the step of training data collection, we review pricing raw data sets and data labels. We also investigate pricing in the step of collaborative training of machine learning models and overview pricing machine learning models for end users in the step of machine learning deployment. We also discuss a series of possible future directions.

Zicun Cong, Xuan Luo, and Jian Pei's research is supported in part by the NSERC Discovery Grant program. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

✉ Yong Zhang
yong.zhang3@huawei.com

Zicun Cong
zicun_cong@cs.sfu.ca

Xuan Luo
xuan_luo@cs.sfu.ca

Jian Pei
jpei@cs.sfu.ca

Feida Zhu
fdzhu@smu.edu.sg

[1] Simon Fraser University, Burnaby, Canada

[2] Singapore Management University, Singapore, Singapore

[3] Huawei Technologies Canada, Burnaby, Canada

Keywords Data assets · Data pricing · Data products · Machine learning · AI

1 Introduction

The disruptive success of machine learning in many applications has led to an explosion in demand [1, 109]. Recent research predicts that the global machine learning market is expected to reach 20.83 billion dollars in 2024 [70]. To succeed in building a machine learning application, one party is far from enough. Many parties have to collaborate in one way or another. For example, one party may have to acquire raw data and data labeling services from some other parties to construct training data, multiple parties may need to collaborate in building a machine learning model, and one party may want to use some other parties' models to solve its business problems. Machine learning applications are indeed pipelines connecting many parties.

Data are critical for machine learning. Machine learning models, especially deep models, rely on large amounts of data for training and testing. Deploying machine learning services also needs data—machine learning models consume users' data as input, and return insights and recommend possible actions. Maintaining and updating machine learning models still need data. The importance of data for machine learning cannot be over emphasized. Data penetrate the whole machine learning pipelines.

Obtaining data for machine learning is far from easy [67]. For a party that wants to build a machine learning model, the challenges come from multiple aspects. First, within the party, in order to develop a training data set, more often than not it is costly to collect data, create proper labels and ensure data quality. Second, the party may realize that it does not have the necessary data to train the target model. Thus, the party may have to explore external sources for the data needed. This involves acquiring external data. Last, to build or strengthen business edges, the party may want to provide machine learning services to other parties. Then, the party has to exchange data with other parties, such as accessing data from end users and providing end users model output.

Connecting many parties in an eco-system in scale requires a general and principled mechanism. As data and models are essential in machine learning pipelines and data and model exchanges are the most fundamental interactions among different parties, data and model marketplaces become a natural choice for machine learning pipelines and eco-systems, and pricing becomes the core mechanism in machine learning pipelines.

In response to the massive and diversified demands for various data, data products become valuable assets for purchase and sale. Here, *data products* refer to data sets as products and information services derived from data sets [86]. Data commoditization motivates data owners to share their data products in exchange of rewards and thus helps data buyers to access data products of high quality and large quantities.

To enable tradings between data owners and data buyers, data must be priced. Pricing data, however, is far from trivial. Agarwal et al. [1] summarize four properties making data a unique asset. First, data can be replicated at zero marginal cost. Second, the value of data is inherently combinatorial. Third, the value of data varies widely among different buyers. Last, the usefulness of data lies in the value of information derived from it, which is difficult to verify a priori. Due to those properties, pricing models for physical goods cannot be directly applied or straightforwardly extended to data products, and thus new principles, theories, and methods need to be developed.

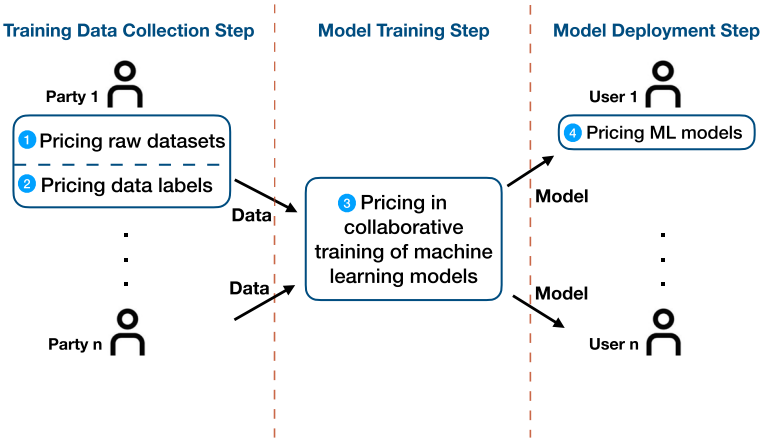


Fig. 1 Steps and pricing tasks in machine learning pipelines

Based on an extensive survey on existing research, we identify and focus on three steps in data and model supply tasks in the manufacturing pipeline of machine learning models [36]. The steps and their corresponding tasks are illustrated in Fig. 1. In the step of training data collection, raw data are collected and the associated labels are annotated. We review the research on pricing for raw data sets and data labels. In the step of collaborative training of machine learning models, we investigate how to price different participants' contributions through their data. In the model deployment step, we overview pricing machine learning models for end users. We focus on the four pricing tasks in machine learning pipelines as follows.

- *Pricing raw data sets* To build a machine learning model, the first step is collecting training data. Monetizing and trading raw data sets provide people with a convenient and efficient way to acquire a large amount of training data. A key challenge in pricing raw data sets is how to set the price reflecting the usefulness of a data set. Moreover, pricing models may be optimized toward different objectives, such as revenue maximization, arbitrage-freeness, and truthfulness. Achieving those optimization goals introduces additional challenges in the design and implementation of pricing models.
- *Pricing data labels* In the training data collection step, in addition to collecting raw data, obtaining data labels is critical. Crowdsourcing is a popular way for this purpose [112]. Unfortunately, spammers may commit no efforts in their assigned tasks and produce random answers, which leads to data sets of poor quality. Thus, a key challenge in pricing data labels is how to estimate label accuracy and compensate crowdworkers correspondingly, such that they are motivated and driven to invest high efforts and report accurate data labels [112]. This task only appears in machine learning pipelines where supervised machine learning models are produced.
- *Revenue allocation in collaborative machine learning* Collaborative machine learning is an emerging paradigm, where multiple data owners collaboratively train machine learning models on their aggregate data, and share the revenues of using/selling these models. Data sets from different owners may have different contributions to the learned models. Evenly distributing the revenues is not fair to data owners, particularly for those who contribute more valuable data, and thus may discourage future collaborations. To this end, a key challenge is how to fairly reward data owners' contributions.

– *Pricing machine learning models* Machine learning as a service (MLaaS) [18, 109] is a rapidly growing industry. Customers may purchase well-trained machine learning models or build models on top of those well-trained rather than building models from scratch by themselves. For example, one may use Google prediction API to classify an image for only $0.0015 [18]. While machine learning models and raw data sets share a series of common ideas in pricing, the pricing models of raw data sets cannot be trivially adapted to price machine learning models. How to version machine learning models and avoid arbitrage among multiple versions is a key challenge in this task.

The four tasks are related to each other. They share some core ideas, that is, linking prices of data products to their utilities to customers. But as the tasks have different application scenarios and pricing goals, they are solved by orthogonal techniques.

The existing models in the first two tasks aim at pricing training data sets with absolute utility functions, that is, the utility of a data product only depends on the properties of the product. One important difference between the first two tasks is about the utility functions. The utility (e.g., accuracy) of data labels is very hard to compute due to the lack of ground-truth verifications. The third task evaluates the utility of a data set by its marginal contribution to a machine learning model. Thus, the utility of a data set also depends on the utility of other data sets used to jointly build the model. The existing methods in the last task also employ absolute utility functions. But as machine learning models and data sets have different properties, new pricing models are developed.

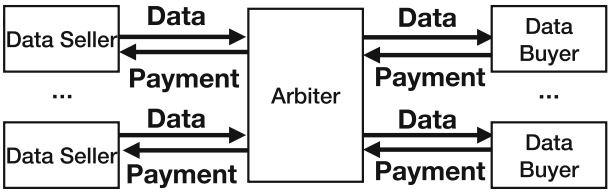
The four tasks are connected when machine learning models and data sets are priced in an end-to-end manner. On the one hand, the price of a machine learning model limits the budget of training data procurement and the revenue that can be split among data owners [1]. On the other hand, the costs of data procurement and model training also influence the selling price of machine learning models, as they are part of the manufacturing cost [67]. Figure 1 shows the connections of the four tasks.

There are some previous surveys related to data pricing [35, 63, 120]. This article covers a substantially deeper and more focused scope than those. In this article, we try to present a comprehensive survey on data pricing in machine learning pipelines. Very recently, Pei [86] presents a survey connecting economics, digital product pricing, and data product pricing. He identifies a series of desirable properties in data pricing and reviews the techniques achieving those properties. But Pei [86] does not focus on machine learning pipeline and does not cover the studies of pricing data labels.

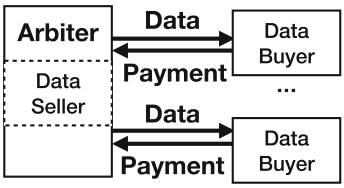
The rest of this survey is organized as follows. Section 2 reviews basic concepts and essential principles in data product pricing. Section 3 reviews pricing raw data sets. Pricing data labels is discussed in Sect. 4. In Sect. 5, we review the recent progress in revenue allocation in collaborative machine learning. Section 6 is about how to price machine learning models. In Sect. 7, we discuss data and model marketplaces in practice and compare data pricing models in action. We conclude this survey and discuss some future directions in Sect. 8.

2 Data marketplaces and pricing

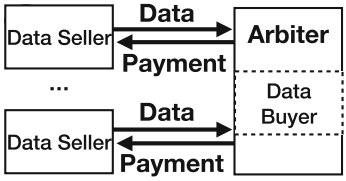
In this section, we briefly review data marketplaces and pricing in general. We first discuss the basic structures of data marketplaces. Then, we discuss some major pricing strategies in general. Third, we discuss different types of data markets in terms of competition and dominance. Last, we discuss the desiderata of data pricing.



(a) General data marketplace.



(b) Sell-side marketplace.



(c) Buy-side marketplace.

Fig. 2 Architectures of data marketplace

2.1 Data marketplaces

A data marketplace is a platform that allows people to buy and sell data products [95]. Some examples of data marketplaces include Dawex [25], Snowflake data marketplace [104], and BDEX [6]. Muschalle et al. [76] identify seven categories of participants in data marketplaces, namely analysts, application vendors, data processing algorithm developers, data providers, consultants, licensing and certification entities, and data market owners.

Figure 2a shows the conceptual architecture of data marketplaces. A data marketplace mainly consists of three major entities, namely data sellers, an arbiter (also known as data vendor [95] and data broker [84]), and data buyers. Data sellers own data products and are willing to share those products with the arbiter in exchange for rewards. Data buyers want to obtain data products to solve their problems. The function of an arbiter is to facilitate transactions between data sellers and data buyers. The arbiter collects data products from data sellers and sells them to data buyers. After collecting the payments from buyers, the arbiter distributes the payments to data sellers. In general, arbiters are modeled as non-profit participants in data marketplaces.

Some studies simplify the architecture of data marketplaces to sell-side marketplaces and buy-side marketplaces. A sell-side marketplace [120], as shown in Fig. 2b, has a single data provider and multiple data buyers. In a sell-side marketplace, the arbiter is operated by a monopoly data seller to sell the single seller's data products. In literature, sell-side marketplaces are considered by pricing models of both general data sets [43] and specific types of data products, such as XML documents [108] and data queries on a relational database [29].

A buy-side marketplace [120], as shown in Fig. 2c, has multiple data providers and a single consumer/data buyer. In a buy-side marketplace, the arbiter is operated by the single data buyer for purchasing data products from providers. Buy-side marketplaces are considered in many existing studies [26, 37, 49]. For instance, de Alfaro et al. [26] study a buy-side marketplace, where a single consumer pays crowdsource workers for labeling the single buyer's data set.

2.2 Pricing strategies

Many pricing strategies have been developed in pricing theory. Cost-based pricing, customer value-based pricing, and competition-based pricing are three important categories [27].

Cost-based pricing considers that the price of a product is determined by adding a specific amount of markup to the cost. This strategy is adopted in personal data pricing, where the cost is the total privacy compensation to data owners [84]. A disadvantage of the cost-based pricing strategy is that it only considers internal factors in determining the selling price. External factors, such as competition and demands, are not included [71].

Customer value-based pricing determines the price of a product primarily based on how much the target customers believe a product is worth [27]. To apply customer value-based pricing, a seller needs to estimate customers' demands for a product through their willingness and affordability [71]. Customer value-based pricing is the most popularly used strategy for data pricing.

Competition-based pricing determines the price of a product strategically based on competitors' price levels and behavior expectations [27]. Game theory provides a powerful tool to implement the strategy. In a non-cooperative game, every seller is selfish and sets the price that maximizes the seller's profit independently [71]. The competition result, that is the asking price of each seller, is the Nash equilibrium [80].

There are some other major pricing strategies in literature [8, 31, 48, 78, 82], such as operation-oriented pricing, revenue-oriented pricing, and relationship-oriented pricing. The remarkably rich body of studies in economics and marketing research on pricing tactics is far beyond the scope and capacity of this survey.

2.3 Four types of data markets

Similar to physical goods, the prices of data products are also influenced by the dominance and diversity of supplies and demands in the market.

Fricker and Maksimov [35] identify four types of data markets. First, in a *monopoly*, a supplier holds enough market power to set prices to maximize profits. Second, in an *oligopoly*, a small number of suppliers dominate the market. Third, in *strong competition markets*, individual suppliers do not have enough market powers to set profit-maximizing prices, and prices tend to align with marginal costs. Last, in a *monopsony*, a single buyer controls the market as the only consumer of products provided by sellers.

Most studies assume explicitly or implicitly a monopoly (monopsony) market structure where the data seller (data buyer) does not care about competing with others. Data pricing in oligopoly market is considered by Balasubramanian et al. [5]. Jiang et al. [51] study a perfect competition market where participants can directly trade with each other.

2.4 Desiderata of data pricing

There are some desiderata preferred by most pricing models. In this section, we briefly review the six desiderata suggested by Pei [86]. In addition, we complement the existing study by an important desideratum, effort elicitation.

Truthfulness Truthfulness is an important economic property of robust markets [123]. In a truthful market, all participants are selfish and only offer prices that maximize their utility values. Participants may have their own valuations on the same product, but a truthful

market guarantees that for each participant, offering the real valuation is an individual's best strategy. In other words, no participants will lie about their valuations. Truthfulness simplifies all participants' strategies and ensures basic market fairness [34].

Reverse auction is a common tool to implement truthful data markets. In a reverse auction, N sellers $D = \{s_1, \ldots, s_N\}$ compete for a buyer's deal by submitting their asking prices $\{b_1, \ldots, b_N\}$. An auction mechanism takes as input the submitted bids, selects a subset of sellers as winners, and determines the payment p_i to each winner s_i, where $p_i \geq b_i$. In a truthful reverse auction, the best strategy (dominant strategy) for a seller s_i to maximize the expected utility is submitting the individual's real valuation, no matter what others submit.

In his seminal paper on optimal mechanism design, Myerson [77] shows that a sealed-bid reverse auction mechanism is truthful if and only if (1) the selection rule is monotone, that is, if a seller s_i wins the auction by bidding b_i, it also wins by bidding $b_i' \leq b_i$; and (2) each winner is paid the critical value, that is, seller s_i would not win the auction if s_i bids higher than this value.

Revenue maximization Revenue maximization is a strategy to increase a seller's customer base by having low prices. This strategy is widely adopted by sellers in an emerging market to build market share and reputations. For traditional physical goods, the curves of marginal cost are U-shaped with respect to manufacturing level. The revenue of a seller is maximized when the manufacturing level is set such that the marginal revenue is zero [11]. Since data products can be re-produced at almost zero costs [1], the revenue maximization techniques for data products and physical products are quite different [86].

Fairness In some scenarios, sellers need to cooperatively participate in a transaction. A data market is fair to the contributors in a coalition if the revenue generated by the coalition is fairly divided among the sellers.

Suppose a set of sellers $D = \{s_1, \ldots, s_N\}$ cooperatively participate in a transaction that leads to a payment v. Shapley [99] lays out four axioms for a fair allocation.

- *Balance* The payment v should be fully distributed to the sellers in D.
- *Symmetry* Sellers making the same contribution to the payment should be paid the same. For a set of sellers S and two additional sellers s and s', if $S \cup \{s\}$ and $S \cup \{s'\}$ lead to the same payment, sellers s and s' should get the same payment.
- *Zero element* If a seller's data do not contribute to the payment of any coalitions, the seller should receive no payment.
- *Additivity* If the data of a group of sellers can be used for two tasks t_1 and t_2 with payments v_1 and v_2, respectively, then the payment to solve both tasks $t_1 + t_2$ should be $v_1 + v_2$.

It is proved that Shapley value $\psi(s)$ is the unique allocation method that satisfies the four axioms, which is defined as the average marginal contribution of s_i to all possible subsets of sellers $S \subseteq D \backslash \{s_i\}$

$$\psi(s) = \frac{1}{N} \sum_{S \subseteq D \backslash \{s\}} \frac{\mathcal{U}(S \cup \{s\}) - \mathcal{U}(S)}{\binom{N-1}{|S|}}, \tag{1}$$

where $\mathcal{U}(\cdot)$ is the utility function [99]. For example, in the context of collaborative machine learning, $\mathcal{U}(S)$ is the performance score of the machine learning model trained on the data sets of S, such as precision.

Equation 1 can be rewritten to

$$\psi(s) = \frac{1}{N!} \sum_{\pi \in \prod(D)} (\mathcal{U}(P_s^\pi \cup \{s\}) - \mathcal{U}(P_s^\pi))), \tag{2}$$

where $\pi \in \prod(D)$ is a permutation of sellers and P_s^π is the set of sellers that precede seller s in π.

The fact that Shapley value uniquely possesses Shapley fairness, combined with its flexibility to support different utility functions, makes it a popular tool to implement fair data marketplaces.

Arbitrage-free pricing Arbitrage is the activities that take advantage of price differences between multiple markets. In a data marketplace, a data seller may offer multiple versions of products. As a consequence, a critical concern is that a data buyer may circumvent the advertised price of a product through buying a bundle of cheaper ones, which negatively affects the seller's revenue. For example, consider a data seller selling noisy queries to the seller's database [84, 86], and the seller perturbs each query answer independently with random noise. An answer with a variance of 5 is sold at $5 and with a variance of 1 is sold at $50. A data buyer wants to obtain an answer of variance 1. The buyer can purchase the cheaper answer 5 times and compute their average. Since the noises are added independently, the aggregated average has variance 1. Thus, the customer saves $25 by arbitrage. A desirable pricing function should guarantee that no arbitrage is possible, in which case we call it arbitrage-free.

Privacy-preservation Privacy protection during the transactions of data raises more and more concerns. In data marketplaces, the privacy of buyers, sellers, and involved third parties are highly vulnerable, and might be disclosed in many different ways [86]. Many different solutions have been proposed for privacy protection in data markets [34, 47, 62]. In this survey, we focus on the studies along the line of privacy compensation [62, 83, 84], which investigate how to provide compensations for the privacy disclosure of data owners. For the purpose of privacy protection, sensitive data sets are usually traded with injected random noise [83]. A data set with less random noise is more accurate, but may leak more privacy and thus more compensations should be made to the data owner.

Computational efficiency The numbers of transactions, sellers and buyers may be huge in a data marketplace. Therefore, it is a fundamental requirement for a pricing model to compute prices efficiently with respect to a large number of goods and participants. Prices should be computed in polynomial time with respect to the number of participants [1] or the number of data products [17]. In some application scenarios, however, it takes exponential time to compute the pricing functions with desirable properties, such as Shapley fairness [37], arbitrage-freeness [58], and revenue maximization [17]. For example, Koutris et al. [58] show that computing arbitrage-free prices of join queries on a relational database is in general NP-hard. How to efficiently determine prices with desirable properties presents technical challenges.

Effort elicitation In addition to the above six desiderata, here we propose a new one, effort elicitation.

In a data marketplace, a data buyer may purchase training data labels via crowdsourcing. Crowdworkers are presented with unlabeled data instances (for instance, images) and are

asked to provide labels (for instance, a binary label indicating whether or not the image contains pedestrians). A major challenge in label collection is to ensure that workers invest their efforts and provide accurate answers. A poorly designed pricing model may result in labels with very low quality [96]. For example, if each task has a fixed price, an obvious strategy that maximizes a worker's profit is to just provide arbitrary answers without even solving the tasks [112]. Many techniques have been developed to post-process noisy answers in order to improve their quality. However, when the inputs to these algorithms are highly erroneous, it is difficult to guarantee that the processed answers will be reliable enough for downstream machine learning tasks [96]. In order to avoid the troubles of "garbage in, garbage out," a desirable approach is to design proper rewards for crowdsourcing tasks that incent workers to invest efforts and provide higher quality answers [112].

3 Pricing raw data sets

In this section, we review the existing studies focusing on pricing raw data sets. The existing studies consider four types of scenarios. The most traditional methods price data sets as indivisible units and do not consider supplier competitions. The intrinsic properties of data sets, such as volumes, are factors determining prices. In the second scenario, how to price indivisible data sets in a competitive market is studied. In the third scenario, data consumers can purchase just a fraction of an entire data set, which is more flexible to consumers but may have the issue of arbitrage. The last scenario addresses pricing personal data by privacy compensation.

3.1 Pricing general data

Machine learning and statistical models are vulnerable to poor quality training data, and thus, high-quality data are valuable to data buyers [112]. Pricing data sets based on quality becomes a natural choice.

Heckman et al. [43] identify a list of factors to assess the quality of a data set, such as age of data, accuracy of data, and volume of data. A linear model is proposed to set the price of a data set as

$$\text{price} = \text{Fixed cost} + \sum_i w_i \cdot \text{factor}_i .$$

Estimating the model parameters w_i is a difficult task, as many data sets may not have public prices associated with them. A more comprehensive list of quality criteria is proposed in [106].

Yu and Zhang [117] study the problem of trading multiple versions of a data set, constructed by different data quality factors. They assume customers' demands and maximum acceptable prices of different versions are public. A bi-level programming model is established to address the problem. At the first level, the data seller determines versions and their prices to maximize the total revenue. At the second level, a group of buyers select data products to maximize their utilities. Solving the bi-level programming model is NP-hard. Yu and Zhang [117] propose a heuristic genetic algorithm to approach it numerically.

3.2 Pricing crowdsensing data

Crowdsensing is a powerful tool to quickly and cheaply obtain vast amounts of training data for machine learning models [112, 122]. In a crowdsensing marketplace, a task requester initiates a data collection task and compensates participating workers according to their reported costs. As workers may exaggerate their costs, pricing models should incentivize workers to truthfully reveal their costs.

Yang et al. [115] design a reverse auction mechanism for mobile sensing data, that is truthful, individually rational, and profitable. A pricing model is truthful if all sellers truthfully report their data collection costs. A model is individually rational if all sellers have non-negative net profits, and profitable if the data buyer has non-negative net profits. The authors assume that a buyer has a set $\Gamma = \{\tau_1, \ldots, \tau_n\}$ of sensing tasks, where each task τ_i has a value v_i to the buyer. Each seller s_i chooses a subset of tasks $\Gamma_i \subseteq \Gamma$ and has a private cost c_i for performing the tasks. Seller s_i decides a price b_i for the sensed data and submits the task-bid pair (Γ_i, b_i) to the buyer. After collecting all bids, the buyer selects a subset of sellers S as winners and determines the payment p_i to each winner s_i.

The proposed auction mechanism, MSensing, selects winners S in a greedy manner. Starting with $S = \emptyset$, it iteratively chooses the seller that brings the largest non-negative net marginal profit. Each winner $s_i \in S$ is paid the critical value p_i of s_i, that is, seller s_i would not win the auction if s_i bids higher than p_i. Specifically, MSensing runs the winner selection algorithm over users $S' = U \backslash \{s_i\}$. The payment p_i is the largest price s_i can bid, such that s_i can replace a user in S'. Please note that $p_i \geq b_i$, this is because due to incomplete cost information, the buyer provides extra compensations to sellers on top of their bids to motivate them to reveal actual costs. MSensing satisfies Myerson's characterization of truthful auction mechanisms [77].

The follow-up work by Jin et al. [52] considers the situation where a data buyer has a data quality requirement Q_j for each sensing task t_j. The authors propose a Vickrey–Clarke–Groves mechanism [4] like truthful reverse combinatorial auction. They assume that the data quality q_i of each seller s_i is public and q_i is the same for all sensing tasks. The authors first consider the scenario where each seller only bids for one bundle of sensing tasks Γ_i. The auction winners S must satisfy the quality requirement for each task t_j, that is, $\sum_{s_i \in S, \text{ if } t_j \in \Gamma_i} q_i \geq Q_j$. The objective of the auction is to maximize the total utility of the buyer and the sellers. The authors prove that winner determination under the setting is NP-hard and propose a greedy winner selection algorithm with a guaranteed approximation ratio to the optimal total utility. Each winner is paid by the winner's critical payment. The authors further study the total utility maximization problem in a more general scenario, where each seller can bid for multiple bundles of tasks. They propose an iterative descending algorithm that achieves close-to-optimal total utility. However, the auction is not truthful.

Koutsopoulos [60] considers a similar setting as Jin et al. [52] do, but assumes that a data buyer has only one sensing task. The author proposes a truthful reverse auction that minimizes the expected cost of the buyer while guaranteeing the data quality requirement. The author assumes that the data buyer has prior knowledge about the distribution of each seller s_i's unit participation cost c_i. The units of participation x_i of s_i is a positive real value indicating how much data is purchased from s_i. Given the sellers' bids, the data buyer determines the auction winners and their participation units by solving a linear programming model, which minimizes the total expected payment under the data quality constraint. Critical payments are made to the selected winners. All sellers bidding truthfully forms a Bayesian Nash equilibrium [61].

3.3 Pricing data queries

Query-based pricing models tailor the purchase of data to users' needs. Customers can purchase their interested parts of a data set through data queries, and are charged according to their issued queries. While such a marketplace mechanism provides greater flexibility to buyers, a less carefully designed pricing model may open the loophole for arbitrage, which allows buyers to obtain a query result in a cost less than the advertised prices.

Given a database D and a multi-set of query bundles $\mathbf{S} = \{\mathbf{Q}_1, \ldots, \mathbf{Q}_m\}$, a query bundle $\mathbf{Q}$ is determined by $\mathbf{S}$, if the answer to $\mathbf{Q}$ can be computed only from the answers to the query bundles in $\mathbf{S}$. A pricing function is arbitrage-free if the advertised price $\pi(\mathbf{Q}) \leq \sum_{i=1}^{m} \pi(\mathbf{Q_i})$, that is, the answer to a query bundle $\mathbf{Q}$ cannot be obtained more cheaply from an alternative set of query bundles.

The first formal framework for arbitrage-free query-based data pricing was introduced by Koutris et al. [58]. The major idea is that a data seller can first specify the prices of a few views $\mathbf{V}$ over a database, and then the price of a query bundle $\mathbf{Q}$ is decided algorithmically. Theoretically, the authors show that if there are no arbitrage situations among the views in $\mathbf{V}$, there exists a unique arbitrage-free and discount-free pricing function $\pi(\mathbf{Q})$. Specifically, $\pi(\mathbf{Q})$ is the total price of the cheapest subset of $\mathbf{V}$ that determines $\mathbf{Q}$, which is found by query determinacy [79]. They also show the complexity of evaluating the price functions. Unfortunately, the pricing model is NP-hard for a large class of practical queries. They develop polynomial time algorithms for specific classes of conjunctive queries, chain queries, and cyclic queries.

Subsequently, Koutris et al. [59] develop a prototype pricing system, QueryMarket, based on the idea [58]. They formulate the pricing model as an integer linear program (ILP) with the objective to minimize the total cost of purchased views $\mathbf{V}_p$. The purchased views $\mathbf{V}_p$ must satisfy the following requirements. For a tuple t in the query answer $\mathbf{Q}(D)$, there must exist a subset of views in $\mathbf{V}_p$ that can produce t and for each relation R in $\mathbf{Q}$, at least one view on R should be purchased. For a tuple t not in $\mathbf{Q}(D)$, there must exist a subset of views in $\mathbf{V}_p$ that can indicate $t \notin \mathbf{Q}(D)$. Although the pricing problem in the setting is in general NP-hard, QueryMarket shows that a large class of queries can be priced in practice, albeit for small data sets. To handle the case that a query $\mathbf{Q}$ may require databases from multiple sellers, they introduce a revenue sharing policy among sellers. Specifically, each seller gets a share of the query price $\pi(\mathbf{Q})$, which is proportional to the maximum revenue that the seller can get among all minimum-cost solutions to the ILP.

The problem of designing arbitrage-free pricing models for linear aggregation queries is studied by Li et al. [62]. Given a data set of n real values $\mathbf{x} = \langle x_1, \ldots, x_n \rangle$, a linear query over $\mathbf{x}$ is a real-valued vector $\mathbf{q} = \langle w_1, \ldots, w_n \rangle$, and the answer is $\mathbf{q}(\mathbf{x}) = \sum_{i=1} w_i x_i$. The authors propose a marketplace, where a data buyer can purchase a single linear query $\mathbf{q}$ with a variance constraint v defined by the buyer. The query $\mathbf{Q} = (\mathbf{q}, v)$ is answered by an unbiased estimator of $\mathbf{q}(\mathbf{x})$ with a variance smaller than or equal to v. The authors first develop a proposition that the pricing function π cannot decrease faster than $\frac{1}{v}$, that is, $\pi(\mathbf{q}, v) = \Omega(\frac{1}{v})$. Then, they propose a family of arbitrage-free pricing functions, $\pi(\mathbf{q}, v) = \frac{f^2(\mathbf{q})}{v}$, where the function $f(\cdot)$ is semi-norm. Last, they provide a general framework to synthesize new arbitrage-free pricing functions from the existing ones. For any arbitrage-free pricing functions $\pi_1, \ldots, \pi_k$, the pricing function $\pi(\mathbf{Q}) = f(\pi_1(\mathbf{Q}), \ldots, \pi_k(\mathbf{Q}))$ is also arbitrage-free if $f(\cdot)$ is a subadditive and non-decreasing function. A comprehensive list of celebrated arbitrage-free pricing functions are listed by Niu et al. [84]. In addition to synthesized pricing functions, Li et al. [62] also study a similar view-based pricing framework as Koutris et al.

[58] do. By adapting the theoretical results in [58], the authors show that the view-based pricing model for linear aggregation queries is NP-hard.

Lin and Kifer [65] study arbitrage-free pricing for general data queries. They propose three pricing schemes, namely instance-independent pricing, up-front dependent pricing, and delayed pricing. The authors further summarize five forms of arbitrages, namely price-based arbitrage, separate account arbitrage, post-processing arbitrage, serendipitous arbitrage, and almost-certain arbitrage. The authors point out that the model by Koutris et al. [58] has pricing-based arbitrage, that is, the computed prices may leak information about D. Theoretically, they propose an instance-independent pricing function and a delayed pricing function that are arbitrage-free across all forms. The major idea is to tackle the pricing problem from a probabilistic view. Queries that are more likely to reveal the true database instance are priced higher.

In the same vein, Deep and Koutris [28] characterize the structure of pricing functions with respect to information arbitrage and bundle arbitrage, where information arbitrage covers both post-processing arbitrage and serendipitous arbitrage defined by Lin and Kifer [65]. For both instance-independent pricing and answer-dependent pricing of a query, an arbitrage-free pricing function should be monotone and subadditive with respect to the amount of information revealed by asking the query. Several examples of arbitrage-free pricing functions are presented, including the weighted coverage function and the Shannon entropy function.

Deep and Koutris [29] later implement the theoretical framework [28] into a real time pricing system, QIRANA, which computes the price of a query bundle $\mathbf{Q}$ from the view of uncertainty reduction. They assume that a buyer is facing a set of all possible database instances S with the same schema as the true database instance D. After receiving the query answer $E = \mathbf{Q}(D)$, the buyer can rule out some database instances $D_i \in S$ that cannot be D by checking whether $\mathbf{Q}(D_i) = E$. A query bundle that eliminates more database instances is priced higher, as it reveals more information about D. The authors propose an arbitrage-free answer-dependent pricing function, which assigns a weight w_i to each database $D_i \in S$, and computes the price of a query bundle by

$$\pi(\mathbf{Q}) = \sum_{i \in \{i \mid D_i \in S\}, \mathbf{Q}(D) \neq \mathbf{Q}(D_i)} w_i . \tag{3}$$

By default, the same weight $w_i = \frac{P}{|S|}$ is assigned to each possible database instance D_i, where P is a parameter set by the data owner. The data owner can also provide QIRANA with some example query bundles and their corresponding prices. Then, QIRANA will automatically learn instance weights w_i from the given examples by solving an entropy maximization problem. Choosing S to be the complete set of possible database instances leads to a $\#P$-hard problem. To make the pricing function tractable, QIRANA uses a random sample of database instances as S.

Chawla et al. [16] extend the pricing function in Eq. 3 to maximize seller revenue. They consider the setting that the supply is unlimited and the buyers are single-minded, that is, a buyer only wants to buy a single query bundle $\mathbf{Q}$. A buyer will purchase $\mathbf{Q}$ if the advertised price $\pi(\mathbf{Q})$ is smaller than or equal to the buyer's valuation $v_{\mathbf{Q}}$. The authors take a training data set consisting of some query bundles and their customer valuations. Three pricing schemes are investigated. The major idea of the pricing schemes is that, according to Eq. 3, a query can be priced as a bundle of items (database instances). Uniform bundle pricing sets the same price for all query bundles. Item pricing sets the price of a query bundle using Eq. 3, where the weights w_i are learned from the training data. XOS pricing learns k weights $w_i^1, \ldots, w_i^k$ for each item D_i and sets the price of $\mathbf{Q}$ as $\pi(\mathbf{Q}) = \max_{j=1}^{k} \sum_{i \in \{i \mid D_i \in S\}, \mathbf{Q}(D) \neq \mathbf{Q}(D_i)} w_i^j$.

Theoretically, the approximation rate of each pricing scheme to the optimal revenue is studied. Although XOS pricing scheme enjoys the best approximation rate, the authors show that item pricing usually achieves larger revenue in practice.

Miao et al. [75] study the problem of pricing selection-projection-natural join queries over incomplete databases. An arbitrage-free pricing function is proposed based on the idea of data provenance, which describes the origins of a piece of data and its processing history [10, 107]. Let t be a tuple in a query answer $\mathbf{Q}(D)$. The lineage $L(t, D)$ of t is defined as the set of tuples in the database D that contribute to t. The authors assume that each tuple t has a base price $p(t)$. The price of $\mathbf{Q}$ is set to the weighted aggregation of the costs of all tuples in $M(\mathbf{Q}, D) = \cup_{t \in \mathbf{Q}(D)} L(t, D)$. Specifically, $\pi^{UCA}(\mathbf{Q}) = \sum_{i \in \{i | t_i \in M(\mathbf{Q}, D)\}} \mu_i \, p(t_i)$, where μ_i is the percentage of attributes of t_i that are not missing. The authors also propose an answer quality aware pricing function, $\pi^{QUCA}(\mathbf{Q}) = \frac{\Delta}{n} \pi^{UCA}(\mathbf{Q}) \kappa(\mathbf{Q}, D)$, where $\kappa(\mathbf{Q}, D)$ is the answer quality and Δ is a constant. However, π^{QUCA} is not arbitrage-free.

Purchasing data is usually not a one-shot deal. A customer may purchase multiple queries from the same data seller. A history-aware pricing function will not charge the customer twice for already purchased information. QueryMarket [59] tracks the purchased views of a customer and avoids charging those views when pricing future queries of the customer. Both [29] and [75] support history-aware pricing in the same vein as [59]. One drawback of these history-based approaches is that the seller must provide reliable storage to keep users' query history [111].

Upadhyaya et al. [111] propose an optimal history-aware pricing function, that is, a buyer is only charged once for purchased data. The key idea is to allow buyers to ask for refunds of already purchased data. In their setting, a query is priced according to its output size. The seller computes an identifier (coupon) for each tuple in the query answer $\mathbf{Q}(D)$. Both $\mathbf{Q}(D)$ and the corresponding coupons are sent to the buyer. If the buyer receives the same tuple t from two queries, the buyer can ask for a refund of t by presenting the two coupons associated with t in the two corresponding queries. To prevent buyers from borrowing coupons from others and receiving unconscionable refunds, each coupon is uniquely associated with a buyer. By tracking coupon status, the data seller guarantees that each coupon will be used only once. However, the pricing function has no arbitrage-free guarantee [29].

3.4 Privacy compensation

Machine learning models in many areas, like recommendation systems [21] and personalized medical treatments [72], require a large amount of personal data. However, trading and sharing personal data may leak the privacy of data providers. Therefore, how to measure and properly compensate data providers for their privacy loss is an important concern in designing marketplaces of personal data.

Differential privacy [32] is a mathematical framework rigorously providing privacy protection and plays an essential role in personal data pricing. Following the principle of differential privacy, random noises are injected into a data set, such that data buyers can learn useful information about the whole data set but cannot learn specifics accurately about an individual. The magnitude of random noise impacts data providers' privacy loss and the data price. A data set with less injected random noise may leak more privacy and is priced higher. Pricing models of personal data routinely adopt cost-plus pricing strategy, where sellers first compensate data providers for their privacy loss, and then scale up the total privacy compensation to determine the price for data buyers [84].

Ghosh and Roth [38] initiate the study of pricing privacy by auction. They propose a truthful marketplace to sell single counting queries on binary data. In their settings, a data seller has a data set consisting of personal data $d_i \in \{0, 1\}$ of individual i. The data seller sells an estimator $\widehat{s}$ of the sum $s = \sum_i d_i$ and compensates data providers for their privacy loss. Under the framework of differential privacy, the authors treat privacy as a commodity to be traded. In particular, if a provider's data is used in an $\epsilon-$differentially private manner, ϵ privacy units should be purchased from the provider. Thus, the privacy compensation problem can be transformed into variants of multi-unit reverse auction. The authors assume that each data provider i has a privacy cost function

$$c_i(\epsilon) = v_i * \epsilon, \tag{4}$$

representing the cost for using the data in an ϵ-differentially private manner, where v_i is the unit privacy cost of i. In an auction, data providers are asked to submit their asking prices b_i for the use of their data. Ghosh and Roth [38] consider two situations. In the first situation, a buyer has an accuracy requirement on $\widehat{s}$, that is, $\Pr[|\widehat{s} - s| \geq k] \leq \frac{1}{3}$. The authors establish an observation that they only need to purchase data from m individuals and use them in an ϵ-differential privacy manner, where m and ϵ only depend on the accuracy goal. It's shown that the classic Vickrey–Clarke–Groves auction minimizes the buyer's payment and guarantees the accuracy goal. The major idea is to select m individuals with the cheapest bids and provide each winner with a uniform compensation $\epsilon \cdot b$, where b is the $(m + 1)$-th smallest bid. In the second situation, a buyer has a budget constraint and wants to maximize the accuracy of $\hat{s}$. The authors propose a greedy-based approximation algorithm to solve the problem.

The value of personal data and privacy valuation may be correlated. For example, a patient may assign a higher price to the patient's medical report than the healthy people ask for. Ghosh and Roth [38] show a negative result that in the situations having such correlations, no individually rational direct mechanism can protect privacy.

In a follow-up study, Dandekar et al. [21] consider the scenario of selling linear aggregate queries $\mathbf{q} = \langle w_1, \ldots, w_n \rangle$ over real-valued personal data $D = \langle d_1, \ldots, d_n \rangle$. They assume data providers have the same privacy cost function as Eq. 4, and propose a truthful reverse auction mechanism to maximize the accuracy of estimators for budget-constraint buyers. The error of an estimator $\widehat{s}$ of the true answer $s = \sum_i w_i d_i$ is its squared error $(\widehat{s} - s)^2$. It is shown that an $\widehat{s}$ computed from more providers with large corresponding weights in $\mathbf{q}$ is more accurate. Therefore, the problem is transformed into a knapsack reverse auction [103] that maximizes the total weights of the selected providers under budget constraints. Specifically, the authors treat the budget as the capacity of the knapsack, the privacy cost of a data entry d_i as its weight in the knapsack, and w_i as the value of d_i. A greedy-based algorithm with an approximation ratio of 5 is proposed to solve the problem.

The aforementioned studies [21, 38] assume that data buyers can purchase an arbitrary amount of privacy from each data provider. However, a conservative individual may not want to sell the individual's data if the privacy loss is too large. Nget et al. [83] study the same problem as Dandekar et al. [21] do in a more realistic situation, that is, an individual i can refuse to participate in an estimator if the privacy loss of i is larger than a threshold ϵ_i. They assume that the privacy cost function of each data provider is public and propose a heuristic method to determine query price. The model first randomly samples a subset of data providers. Then, it uses the data from each sampled individual i in an ϵ_i-differentially private manner and computes the compensations correspondingly. If the total compensation is larger than the budget, the model decreases the differential privacy levels of the high cost providers, such that the budget goal is met. Last, they generate the perturbed query answers

by personalized differential privacy [54], which guarantees the differential privacy for each selected individual i. They repeat the above steps several times and return the perturbed answer with the smallest squared error.

Later, Zhang et al. [121] propose a truthful personal data marketplace, where each data provider i can specify the personal maximum tolerable privacy loss ϵ_i. They first show that the accuracy of query answers is proportional to the total amount of purchased privacy. Under the assumption that the distributions of privacy costs of all individuals are public, they design a variant of Bayesian optimal knapsack procurement [33], which maximizes the expected total purchased privacy under the constraint of a data buyer's expected budget. The authors solve the problem by adopting the algorithm in [33]. The noisy query answer is generated using personalized differential privacy [54], which guarantees ϵ_i-differential privacy for each selected individual i.

The models proposed by Dandekar et al. [21, 38] may be attacked by arbitrage. Li et al. [62] consider the situation where a data buyer has a variance constraint v on the purchased noisy query answers. They assume that the privacy costs of individuals are public, and propose a theoretical framework for assigning arbitrage-free prices to linear aggregate queries $\mathbf{q}$. A perturbed answer is generated from the true answer by adding Laplace noise with the expectation 0 and variance $\sqrt{\frac{v}{2}}$. Measured by differential privacy, the privacy loss of an individual i is upper-bounded by $\epsilon = \frac{w}{\sqrt{\frac{v}{2}}}$ if the individual is involved in the query, and 0 otherwise, where w is the largest absolute weight in $\mathbf{q}$. Several privacy compensation functions are proposed, such as $p_i(\epsilon) = c_i\epsilon$, where c_i is the unit privacy cost of individual i. The price of a query is the sum of the privacy compensations, which is proved to be arbitrage-free.

Li et al. [62] only compensate individuals involved in queries. However, as two individuals' data may be correlated, the privacy of a not-involved individual may be leaked due to the revelation of the other individual's data. To fairly compensate individuals for their privacy, Niu et al. [84] extend the model by Li et al. [62] and propose a pricing model that is arbitrage-free and dependency fair. Dependent fairness requires that a data provider should receive a privacy compensation as long as some data of other providers that is correlated to the data of this provider is involved in a query. Employing dependent differential privacy [66], the privacy loss of a data provider i caused by a query is upper-bounded by $\epsilon_i = \frac{ds_i}{\sqrt{\frac{v}{2}}}$, where ds_i is the dependent sensitivity of the query at provider i's data. The authors propose a bottom-up mechanism and a top-down mechanism to determine privacy compensations and query prices. The bottom-up mechanism computes compensations in the same way as Li et al. [62] do and determines query prices as a multiple of the total compensations. The top-down mechanism first determines the query price using a user-defined arbitrage-free pricing function and spares some fraction of a buyer's payment for privacy compensation. Each data provider receives a division of the compensation proportional to the provider's privacy loss.

All of the privacy compensation methods discussed above assume a trustworthy platform/agent to trade data providers' privacy with data buyers. Data providers, however, cannot control the usage of their own data.

In this concern, Jin et al. [53] develop a truthful crowdsensing marketplace, where data owners can determine how much privacy to disclose. In their marketplace, obfuscated geo-locations of data owners are traded by auctions. Data owners first inject random noise to their data based on their own privacy preferences. Then, each data owner bids with the cost as well as the mean and variance of the injected random noise. The buyer determines auction winners to maximize data accuracy with respect to the buyer's budget constraint. The authors show

that the optimization problem is NP-hard and develop a greedy heuristic solution. The major idea is to iteratively select data owners that bring the largest marginal utility contributions until the budget is used up.

In this section, we review representative pricing models of raw data sets in four types of scenarios, where different desiderata are considered. A limitation of the discussed pricing models is that data sets are priced without considering their down-stream applications. Fernandez et al. [34] argue that the value of a data set to customers is usually task dependent and cannot be evaluated by the intrinsic properties of the data set alone. As the pricing models of raw data sets are agnostic to the down-stream applications of raw data sets, these pricing models can be used in machine learning pipelines of building both supervised and unsupervised machine learning models.

4 Pricing data labels

Crowdsourcing is a popular method for collecting large-scale labeled training data for machine learning tasks [96]. Unfortunately, crowdsourced data often suffers from quality issues. This is mainly due to the existence of lazy and spamming workers, who submit low-quality labels. Those workers can be discouraged from participating in the tasks by rewarding them with a performance-based payment [88]. However, due to a lack of ground-truth verification of the collected labels, how to evaluate label quality and price the labels correspondingly is a challenging task. In this section, we review two types of label pricing models, which are designed to motivate workers to exert efforts and submit accurate data labels.

4.1 Gold task-based pricing models

A gold task is one for which the answer is known to the data buyer a priori. Gold tasks can be uniformly mixed at random within the tasks for workers to evaluate workers' performance, which determines the payments to workers. Since workers cannot distinguish gold tasks from others, this strategy can motivate workers to provide accurate labels.

Shah and Zhou [96] consider a crowdsourcing setup where workers perform binary labeling tasks. The authors propose a multiplicative pricing model using gold tasks. The model allows a worker to skip an assigned task if the worker is not confident about the answer. The total payment to a worker u is computed based on u's performance on the answered tasks. The workers are selfish and want to maximize their individual expected payments. The authors assume that each worker has a private belief $\Pr(y_t = l)$ about how likely the true label y_t of a task t is l. The pricing model is designed to incentivize workers to only report high-confidence labels with beliefs greater than a threshold p. The total reward starts at β. For each correct answer in the gold tasks, the reward will be multiplied by $\frac{1}{p}$. However, if any of these gold tasks are answered incorrectly, the reward will drop to zero, that is,

$$\pi(u) = \beta \cdot \frac{1}{p^c} \cdot \mathbf{1}(r = 0), \tag{5}$$

where $\mathbf{1}(\cdot)$ is an indicator function and c and r are the number of correct and wrong answers, respectively. This pricing model motivates workers to only answer tasks that they are sufficiently confident about. The pricing model is incentive compatible, that is, a worker receives the maximum expected payment if and only if the worker exerts efforts to report accurate labels. The pricing model also satisfies the "no-free-lunch" axiom, that is, workers who only

provide wrong answers will receive no payments. In their setting, the proposed method is the unique incentive compatible model that satisfies the "no-free-lunch" axiom.

Shah et al. [98] further generalize the model [96] to multi-label tasks. For each task, a worker can submit multiple answers $\widehat{Y}$ that the worker believes is most likely to be correct. This multi-selection system provides workers more flexibility to express their beliefs, which can use the expertise of workers with partial knowledge more effectively than single-selection systems. The authors assume that the workers' beliefs for any label being the true label for a task lie in the set $\{0\} \cup (p, 1]$, where p is fixed and known. The authors want to encourage workers to only report the set of labels with positive beliefs. The reward of a worker for a gold task is $(1-p)^{(|\widehat{Y}|-1)}$ if one of the worker's answers is correct and 0 if otherwise. The total payment to a worker is determined by the product of the worker's rewards on all gold tasks.

In a later study, Shah and Zhou [97] propose a two-stage multiplicative pricing model to motivate workers to self-correct their answers. In the first stage, a worker answers the assigned tasks. In the second stage, if the worker's answer to a task t does not agree with the answer from the peer workers, the worker has an opportunity to change the answer. The worker u receives a high reward for a gold task t if the initial answer to the task is correct, a low reward if the updated answer is correct, and 0 reward if the final answer is wrong. The total payment is determined by the product of the worker's rewards from gold tasks. Theoretically, the authors prove that the proposed method is the unique incentive compatible model that satisfies the no-free-lunch axiom. Empirically, they show in a simulation that the self-correction setting can significantly improve the data quality compared to the standard single-stage settings.

To reduce the variance in payoffs, the aforementioned methods [96–98] require each worker to solve a sufficient number of gold tasks. This leads to a waste of procurement budget, as the answers to the gold tasks are already known.

de Alfaro et al. [26] address the limitation by combining the ideas from peer prediction and gold tasks. They arrange the workers in a hierarchy, where every worker shares one common task with each of its children. A few gold tasks are used to incentivize high efforts from the workers at the top level of the hierarchy. Assuming these workers exert sufficient efforts to provide high quality answers, their answers can be used as pseudo gold tasks for workers in the second layer, who can in turn provide pseudo gold tasks for the next level, and so forth. A worker will be punished if the worker does not agree with the parent on the task shared between them. As the workers at the top level are evaluated by the true gold tasks, they are evaluated more accurately than the other workers, which is not fair to workers at lower layers.

The follow-up work by Goel and Faltings [40] considers fair payment among workers; that is, the expected reward of a worker is directly proportional to the accuracy of the worker's answers and independent of the strategy and proficiency of the worker's random peers. The key idea is to estimate the proficiency of workers, which is the probability that a worker can solve the tasks correctly. Goel and Faltings [40] start by estimating the proficiency of a small group of workers with gold tasks. Then, the answers by the small group of workers to non-gold tasks are used as contributed gold tasks, where the workers' proficiencies are used as the trustworthy degree of those tasks. The contributed gold tasks are used to estimate the proficiency of more workers. Finally, the payoff of each worker is proportional to the worker's estimated proficiency, such that workers with good proficiency receive high payments. The model guarantees that exerting high efforts to provide accurate labels is a dominant strategy for each worker.

4.2 Peer prediction-based pricing models

Peer prediction-based pricing model can incentivize efforts and accurate data labels without access to gold tasks. Those models take advantage of the stochastic correlation of answers to the same tasks, and set up a game among workers, called a mechanism in game theory. The game is designed such that workers who exert effort in solving the tasks can achieve high expected rewards, whereas spammers providing random answers on average receive no payments. A pricing model is incentive compatible if it admits exerting high efforts and truthful reporting as an equilibrium.

Dasgupta and Ghosh [22] initiate the study of effort elicitation and propose the DG model to price binary labels. A data buyer assigns a set of data labeling tasks to a group of workers, such that each task is labeled by multiple workers and each worker labels multiple tasks. They assume that a worker u_i either invests no effort and thus provides a random label, or invests full effort with a cost c_i and provides a true label with probability p_i. Here, p_i is called the proficiency of u_i. The workers are self-interested, who want to maximize their payoffs.

The DG model pays a worker u_i on an assigned task t based on how surprisingly u_i's report is consistent with that of the peer worker u_p. Denote by $\widehat{y}$ and $\widehat{y}_p$ the answers from u_i and u_p to a task, respectively. The model pays u_i with a constant reward subtracting the probability $\Pr(u_i, u_p)$ that u_i and u_p have the same answer to a random task, that is,

$$\pi(u_i, t) = \beta \cdot (\mathbf{1}(\widehat{y} = \widehat{y}_p) - \Pr(u_i, u_p)), \tag{6}$$

where β is a non-negative payment scaling parameter that is chosen to cover workers' effort costs and $\Pr(u_i, u_p)$ is approximated from the submitted labels. The total payment to a worker u_i is the sum of u_i's payment for each task.

The pricing model incentivizes efforts, as the expected payment for spammers who do not solve their tasks and report random/constant labels is exactly zero. Under the assumption that the proficiency of all workers are better than random guess, it is shown that the DG model is incentive compatible. Even though the pricing model also has non-informative equilibria, such as all workers reporting the same label, those equilibria are less profitable to the workers, and thus are not attractive to the workers.

In a multi-label situation, two labels l_1 and l_2 may be positively correlated. Shnayder et al. [100] show that under the DG model [22], workers can achieve more profits by misreporting l_1 by l_2. The correlated agreement (CA) mechanism [100] extends the DG model to multi-label tasks. In the CA mechanism, knowledge about label correlation is required. A label correlation matrix Δ is learned from workers' submissions, where an element $\Delta_{i,j} = \Pr(l_i, l_j) - \Pr(l_i)\Pr(l_j)$ is the correlation degree between labels l_i and l_j. Denote by $S(\cdot)$ the sign function of Δ, that is, $S(l_i, l_j) = 1$ if $\Delta_{i,j} > 0$, and 0 otherwise. A worker u will be rewarded for a task t if u's report is positively correlated with that of peer u_p. To penalize the case where all workers blindly report the same label, a worker u will be penalized if u is likely to be consistent with worker u_p on random tasks. In particular, the payment to worker u for reporting $\hat{y}$ is

$$\pi(u, t) = \beta \cdot (S(\widehat{y}, \widehat{y}_p) - S(\widehat{y}_a, \widehat{y}_b)),$$

where $\hat{y}_p$ is the answer to task t by worker u_p, $\widehat{y}_a$ is the answer to a random task by worker u, and $\widehat{y}_b$ is the answer to another random task by worker u_p. When the number of tasks is large, such that label correlations Δ can be accurately learned, the CA mechanism is incentive compatible with the highest payment. However, the mechanism fails if two labels l_1 and l_2 are not distinguishable with respect to $S(\cdot)$, that is, $\forall l_i \in Y$, $S(l_1, l_i) = S(l_2, l_i)$. In this situation, workers may misreport l_1 by l_2 and still receive the same payoffs.

Radanovic et al. [88] provide complementary theoretical results on pricing multi-label tasks. They assume that the labels only have limited correlations, that is, $\Pr(o_p = l_2 | o = l_1) < \Pr(o_p = l_2 | o = l_2)$, where o and o_p are the observed labels of worker u and worker u_p, respectively. The mechanism pays the report $\widehat{y}$ by worker u on a task t by

$$\pi(u, t) = \frac{\mathbf{1}(\widehat{y} = \widehat{y}_p)}{\mathrm{R}(\widehat{y})} - 1,$$

where $\mathrm{R}(\widehat{y})$ is the empirical frequency of $\widehat{y}$, which is computed from all submissions. It is shown that exerting high efforts and truthful reporting is strictly more profitable than any other equilibria. However, their assumptions on label correlations may not hold in some applications [100].

The aforementioned methods [22, 88, 100] require that each task must be completed by at least two workers, which leads to duplicate answers, and thus does not use the crowd efficiently. For a setting with binary labels, Liu and Chen [68] propose to learn a classifier $\mathcal{M}$ from workers' reports, and use the classifier's predictions $\mathcal{M}(t)$ as peer reports. Since workers' submitted labels are noisy, the classifier is trained by the techniques of learning with noisy labels [81]. Specifically, they first estimate the error rates of submitted labels. Then, the classifier is optimized by an error rate calibrated loss function $\varphi(\cdot)$ proposed by Natarajan et al. [81]. A report $\widehat{y}$ to a task t is priced based on $-\varphi(\mathcal{M}(t), \widehat{y})$, such that labels with large loss are priced lower. Under the assumption that $\mathcal{M}$ is better than random guess, exerting efforts to find the truth labels is the highest-paying equilibrium.

Liu and Chen [69] study the problem of sequential label collection, where labeling tasks are published in multiple rounds. In their settings, an accurately labeled task has a fixed reward to a data buyer, whereas a mistakenly labeled task has no value to a data buyer. They propose an incentive compatible pricing model that maximizes the expected utility for a data buyer, which is the difference between the total rewards and the total payment.

They develop a multi-armed bandit algorithm to extend the DG model [22], which dynamically adjusts the parameter β in Eq. 6. A larger β encourages more accurate labels but costs more money. As the bandit algorithm requires a static environment, this method may fail to learn the optimal β if adversarial workers adjust their strategies according to their interactions with the mechanism [41, 46]. Hu et al. [46] solve the problem by reinforcement learning, which is more robust to strategic behaviors of workers.

In practice, peer prediction-based models need to adjust payments to avoid negative payments. The adjustment may lead to an issue that spammers may receive positive and high rewards. Radanovic and Faltings [87] address the issue by proposing a reputation system PropeRBoost to adjust the payments. PropeRBoost publishes tasks to workers in multiple rounds, and computes a reputation score for each worker based on the worker's past submissions. In each round r, it first applies the DG model [22] to compute workers' payments, and then re-scales the payments by the reputations of the corresponding workers. It is shown that the average payment of a spammer converges to 0 as r approaches infinity.

In this section, we review gold task-based and peer prediction-based pricing models for data labels. The developed pricing models guarantee that exerting efforts to report accurate data labels is the most profitable strategy of all workers. A major concern of gold task-based methods is that these methods require a sufficient number of gold tasks to obtain good performance. In some scenarios, however, gold tasks are very expensive to obtain. For peer prediction-based methods, the existence of multiple equilibria is a major limitation, as workers may converge to an uninformative equilibrium, where workers do not exert full efforts [101].

5 Pricing in collaborative training of machine learning models

Collaborative machine learning is an appealing paradigm where multiple data owners collaboratively build high-quality machine learning models by contributing their data. As the data sets from different data owners may have different contributions to the trained machine learning models, data owners who contribute more valuable data should receive more rewards [102]. In this section, we review contribution evaluation and revenue allocation techniques in collaborative machine learning.

5.1 Revenue allocation by Shapley value

Shapley fairness is widely adopted as the foundation of fair revenue allocation in collaborative machine learning. It guarantees that each participant receives a payment proportional to the participant's marginal contribution to the performance of the trained machine learning model. The challenge in adopting Shapley value lies in its exponential computational cost.

Maleki et al. [73] tackle the efficiency issue of Shapley value by proposing a permutation sampling algorithm for bounded utility functions. By Eq. 2, the Shapley value of a seller is the marginal utility contribution averaged over all possible subsets of sellers, which can be estimated by sample mean. Denote by $\widehat{\psi}(s)$ an (ϵ, δ)-approximator of a seller's Shapley value, that is, $\Pr(|\widehat{\psi}(s) - \psi(s)| \leq \epsilon) \geq 1 - \delta$. To compute the estimators for all sellers, by Hoeffding's inequality [44], we need $O(\frac{2r^2N}{\epsilon^2} \log \frac{2N}{\delta})$ samples and evaluate the utility function $O(N^2 \log N)$ times, where N is the number of sellers and r is the range of the utility function. Evaluating the utility function itself, such as computing testing accuracy, is computationally expensive, as it requires training a machine learning model. Therefore, the method is not scalable to a large number of sellers.

Ghorbani and Zou [37] extend the Monte-Carlo method by Maleki et al. [73] to price individual data point in supervised learning, and propose truncated-based and gradient-based approximation methods. Their truncated-based method reduces the number of utility evaluations by ignoring coalitions of large size. The authors argue that it is sufficient to estimate Shapley values up to the intrinsic noise in the prediction performance $\mathcal{U}$ on the test data set, which can be measured as the bootstrap variance of $\mathcal{U}$. In addition, the performance change by adding one more training data point s to a large training data set S is ignorably small. Therefore, if the utility of S is close to the utility of the whole data set D, the marginal contribution of s to S can be regarded as 0 in practice, and thus its computation can be truncated. Their gradient-based method speeds up the evaluation of utility functions by reducing training time, where a model is trained with only one pass through the training data. They update the model by performing gradient descent on one data point s at a time and the marginal contribution of s is the change in the model performance. The two approximation methods introduce estimation bias into the approximated Shapley values and have no guarantees on the approximation error.

Jia et al. [50] propose two approximation algorithms with provable error bounds for Shapley value that significantly reduce the number of utility evaluations. The first algorithm adopts the idea of group testing in feature selection [124]. Denote by β_i a Boolean random variable indicating whether a seller s_i is in a random sample of sellers. A sampling distribution of $\beta_1, \ldots, \beta_N$ is designed such that the difference in Shapley values between a seller s_i and a seller s_j is

$$\psi(s_i) - \psi(s_j) = \frac{1}{N-1} \sum_{S \subseteq D \setminus \{s_i, s_j\}} \frac{\mathcal{U}(S \cup \{s_i\}) - \mathcal{U}(S \cup \{s_j\})}{\binom{N-2}{|S|}}$$
$$= \mathrm{E}[(\beta_i - \beta_j)\mathcal{U}(\beta_1, \ldots, \beta_N)],$$

where $\mathcal{U}(\beta_1, \ldots, \beta_N)$ is the utility evaluated on the appearing sellers and D are all sellers. The Shapley value of sellers can be derived from the estimated Shapley differences between all datum pairs by solving a feasibility problem. They demonstrate that the algorithm returns an (ϵ, δ)-approximation with $O(N(logN)^2)$ utility evaluations. The second algorithm is based on their observation that Shapley values are approximately sparse, that is, most values are around the mean. Exploiting this property, they apply the idea of sparse signal recovering in compressive sensing [89], and develop an algorithm that produces an (ϵ, δ)-approximation with only $O(N \log(\log(N)))$ utility evaluations.

Jia et al. [49] further discover that Shapley values for data points used in unweighted kNN classifiers can be computed exactly only in $O(N \log N)$ time. Given a testing point x_{test} with label y_{test}, they define the utility of a kNN classifier as the likelihood of y_{test}, that is,

$$\mathcal{U}(S) = \frac{1}{k} \sum_{i=1}^{\min(k,|S|)} \mathbf{1}(y_{\alpha_i(S)} = y_{\text{test}}),$$

where $\alpha_i(S)$ is the index of the training data that is the i-th closest to x_{test} in the set of data points S. The special utility function enables efficient computation of Shapley differences between two data points $x_{\alpha_i(S)}$ and $x_{\alpha_{i+1}(S)}$, that is,

$$\psi(x_{\alpha_i(S)}) - \psi(x_{\alpha_{i+1}(S)}) = \frac{\mathbf{1}(y_{\alpha_i(S)} = y_{\text{test}}) - \mathbf{1}(y_{\alpha_{i+1}(S)} = y_{\text{test}})}{k} \frac{\min(i,k)}{i}. \tag{7}$$

They start by computing $\psi(x_{\alpha_N(S)}) = \frac{\mathbf{1}(y_{\alpha_N(S)} = y_{\text{test}})}{N}$ and then exploiting Eq. 7 to recursively compute the Shapley values in the order of $x_{\alpha_N(S)}, \ldots, x_{\alpha_1(S)}$. They further develop an (ϵ, δ)-approximation algorithm based on Locality Sensitive Hashing [23] with only sub-linear complexity. The major idea is to only compute Shapley values for the retrieved $k^* = \max(k, \frac{1}{\epsilon})$ nearest neighbors of x_{test} and ignore the rest data points, as their Shapley values are too small. Moreover, they present a Monte-Carlo approximation algorithm with $O(\frac{N}{\epsilon^2} \log(k) \log(\frac{k}{\delta}))$ time complexity for weighted kNN classifiers.

The aforementioned studies [37, 49, 50] evaluate the utility of a model by its performance on a validation data set. Sim et al. [102] consider the situation where no validation data sets are available, and propose to use information gain on model parameters as the utility function. Denote by θ the model parameters. After training on data D, the information gain $\mathrm{IG}(\theta) = H(\theta) - H(\theta|D)$ is the reduction in the uncertainty of θ, where $H(\cdot)$ is the entropy function. In addition to Shapley fairness, three additional incentive conditions for revenue allocation are proposed, namely individual rationality, stability of the grand coalition, and group welfare. They also present p-Shapley fairness, which assigns a reward $\pi(s_i) = k\psi(s_i)^p$ to a seller s_i. By tuning parameter $p \in [0, 1]$, they can trade off between achieving different incentive conditions. Rather than monetary incentives, each participant receives a machine learning model as a reward. To realize different levels of rewards, the models are trained by injecting different levels of noise into training labels.

Federated Learning [7, 74] enables multiple decentralized participants to collaboratively train a machine learning model while keeping their training data locally. The data sets contributed by the participants are used in a sequential order determined by a central server. Evaluating participants' contributions using Shapley value incurs high communication costs

among the decentralized participants. Moreover, Shapley value neglects the order of data sources. To accommodate the challenges, Wang et al. [113] propose federated Shapley value. Denote by $\mathcal{U}(s_i + s_j)$ the utility of the model, which is trained on s_i's data first, then on s_j's data. Let I_t be the set of selected participants in round t of the federated learning process. The federated Shapley value of participant s_i at round t is defined as follows.

$$\psi_t(s_i) = \begin{cases} \frac{1}{|I_t|} \sum_{S \subseteq I_t \setminus \{s_i\}} \frac{\mathcal{U}(I_{1:t-1} + (S \cup \{s_i\})) - \mathcal{U}(I_{1:t-1} + S)}{\binom{|I_t|-1}{|S|}} & \text{if } s_i \in I_t \\ 0 & \text{if } s_i \notin I_t \end{cases} \tag{8}$$

The federated Shapley value of s_i is $\psi(s_i) = \sum_{t=1}^{T} \psi(s_i)$, where T is the total rounds in federated learning. The authors show that federated Shapley values satisfy the balance and additivity axioms of Shapley fairness. The other two axioms, symmetry and zero element, are satisfied in each round. They extend the permutation sampling and group testing approximation methods [50] to compute federated Shapley values.

Participants in federated learning spend some costs for contributing their data sets, such as privacy cost [45] and energy costs [56]. Yu et al. [118] propose a fair revenue allocation mechanism for federated learning that jointly considers the costs and contributions of participants. At round t, each participant s_i has a public cost $c_i(t)$ and receives a reward $\pi_i(t)$. The regret $r_i(t)$ of s_i is a function of the difference between the total cost and total reward of s_i. A large value of $r_i(t)$ indicates that s_i is not well compensated for the costs incurred to s_i. The authors argue that the payments of participants at each round should achieve contribution fairness and regret fairness. Contribution fairness requires that the payment $\pi_i(t)$ and the Shapley value $\psi_t(s_i)$ of each participant s_i should be positively correlated, that is, $\sum_i \pi_i(t)\psi_t(s_i)$ should be maximized. Regret fairness requires that the participants should have similar regrets, that is, the difference of the regrets among participants should be minimized. The payments of participants are determined by solving an optimization problem with respect to a budget constraint. Theoretically, they show that the time-averaged regret of participants is upper-bounded by a constant value as $t \to \infty$.

Shapley value is vulnerable to data-replication attacks. A data provider may replicate his/her data with zero cost and acts as an additional provider to get extra unconscionable rewards. Agarwal et al. [1] address the issue by penalizing similar data sets to disincentivize replication, that is, the replication-robust Shapley value is defined as

$$\psi_r(s_i) = \psi(s_i) e^{-\lambda \sum_{s_j \in D \setminus \{s_i\}} \mathrm{SM}(s_i, s_j)},$$

where SM is a similarity metric and λ is a constant. However, the proposed replication-robust Shapley value no longer satisfies the balance axiom in Shapley fairness.

Han et al. [42] study the replication attack in data markets with submodular utility functions. They show that the total reward received by an attacker increases monotonically with respect to the number of the attacker's replications. They discover that the extra reward to the attacker mainly comes from the marginal contributions to small seller groups by the attacker's replication. To fix the issue, the authors propose to down-weigh those contributions when computing Shapley values. Their method guarantees that attackers receive smaller rewards with more replications.

Ohrimenko et al. [85] design a replication robust collaborative data market, where each participant is asked to pay a participation fee. This method discourages replication, as the extra reward received by an attacker cannot cover the attacker's participation cost.

5.2 Other revenue allocation methods

There are some other revenue allocation methods in collaborative machine learning other than Shapley value.

Leave-one-out [20] is a commonly used method to evaluate data importance. It compares the performance of a model trained on the full data set with the performance trained on the full set minus one point. The performance drop is defined as the value of the data point, that is, $\pi(s_i) = \mathcal{U}(D) - \mathcal{U}(D \backslash \{s_i\})$. Leave-one-out is often approximated by influence function [20, 57], which measures how the model changes as the weight of a training point is changed without retraining the model. Richardson et al. [93] apply an influence function to reward participants in federated learning for their contributed data points. It is shown that the pricing model is incentive compatible. Applying influence functions to price data points are also investigated in [50, 92]. Compared with Shapley value, leave-one-out methods, in general, are more efficient as they do not require model retraining. However, leave-one-out methods may not accurately assess the values of data points. The methods may assign a low value to one of the two exactly equivalent data points, regardless of how important the datum is, as high performance may still be achieved by including the other datum [116].

Yan and Procaccia [114] design a data pricing model based on *core* [39], which is a celebrated revenue allocation solution in cooperative game theory. The solution seeks to achieve maximum stability of how participants team up with each other. Core requires that the total reward of each coalition S should be at least equal to the utility $\mathcal{U}(S)$, that is, $\forall S \subseteq D, \sum_{s_i \in S} \pi(s_i) \geq \mathcal{U}(S)$, where $\pi(s_i)$ is the reward of participant s_i and D is the set of all participants. When such a reward cannot be achieved, *least core* relaxes the constraints by allowing a minimum difference ϵ between the utility of S and the total reward of S. In particular, least core computes the payment to each participant by solving the following linear program.

$$\begin{aligned} &\min \epsilon \\ &\text{s.t.} \sum_{s_i \in D} \pi(s_i) = \mathcal{U}(D), \\ &\quad \sum_{s_i \in S} \pi(s_i) + \epsilon \geq \mathcal{U}(S) \ \forall S \subseteq D. \end{aligned} \tag{9}$$

The number of constraints in Eq. 9 grows exponentially with respect to the number of participants. Maleki et al. [114] tackle the efficiency issue by proposing a Monte Carlo approximation algorithm with guaranteed approximation errors. Their approximation method samples a relatively small number of coalitions and solves Eq. 9 on the sampled coalitions. If Eq. 9 has multiple solutions, the solution with the smallest l_2-norm is chosen. Their revenue allocation satisfies the balance, symmetry, and zero element axioms of Shapley fairness.

Yoon et al. [116] propose a reinforcement learning algorithm to value data points. They learn a data value estimator that estimates data values and selects the most valuable samples to train a target classifier. They jointly learn the data value estimator and the corresponding classifier, which enables the classifier and the data value estimator to improve the performance of each other. However, this method cannot guarantee fair revenue distribution among participants.

Most of the existing revenue allocation methods are developed in the settings that supervised machine learning models are jointly trained. The participants are rewarded based on the contributions of their data sets to the utility of the jointly trained machine learning model. To adapt existing pricing models to scenarios where unsupervised machine learning models

are jointly trained, the major challenge is to develop a utility function that participants can all agree on. For some traditional unsupervised machine learning models, there are some widely accepted performance metrics that can serve as the utility functions. For example, Silhouette Coefficient [94] and Calinski–Harabasz index [12] are widely used to evaluate the performance of clustering algorithms when ground-truth clusters are unknown. However, developing a utility function for some unsupervised models, such as pre-trained deep language models [9, 30], may be challenging, as they are evaluated differently in many down-stream machine learning tasks.

In this section, we review pricing models in collaborative training of machine learning models. The major idea is to price each participant's data set based on its contribution to the performance of the jointly trained machine learning model. Shapley value-based methods guarantee fair revenue distribution among participants, but suffer from poor computational efficiency and scalability. Some alternative methods [114, 116] enjoy better efficiency or coalition stability, but lose fairness guarantee.

6 Pricing machine learning models

Machine learning models are needed in many different applications and scenarios. Rather than building machine learning models from scratch, many users and companies turn to purchase well-trained machine learning models, due to their lack of expertise and computation resources [17, 119]. In this section, we review pricing models for machine learning models and discuss the differences between pricing machine learning models and raw data sets.

6.1 Pricing models

Pricing machine learning models is an emerging research area. To the best of our knowledge, the existing studies mainly focus on arbitrage-free and revenue maximization pricing.

Chen et al. [17] propose an arbitrage-free and revenue maximization machine learning model marketplace. In their setting, a model owner sells multiple versions of a machine learning model to different buyers. The seller first trains an optimal model on the whole raw data set. Then, the seller produces different versions of the optimal model by adding Gaussian noises with different variances to the parameters of the optimal model. The expected error rates of the generated model instances are monotonically increasing with respect to the variance of the injected noise. An arbitrage-free pricing function guarantees that a buyer cannot derive a high performance model by paying less. Under their mechanism, a pricing function is arbitrage-free if and only if the function is monotone and subadditive with respect to the inverse of the noise variance. Unfortunately, their pricing model only works for machine learning models trained with strictly convex objective functions.

Chen et al. [17] further study revenue maximization in pricing machine learning models with respect to the demands and valuations of a set of buyers. They show that determining the optimal prices is coNP-hard. To overcome the computational hardness, they relax the subadditive constraints $\pi(x+y) \leq \pi(x)+\pi(y)$ by $\frac{\widehat{\pi}(x)}{x} \leq \frac{\widehat{\pi}(y)}{y}$, where $x \leq y$ and $\widehat{\pi}$ is an approximation of the optimal pricing function π. They show that $\widehat{\pi}$ is arbitrage-free and $\forall x > 0, \pi(x)/2 \leq \widehat{\pi}(x) \leq \pi(x)$. They propose a dynamic programming algorithm to compute $\widehat{\pi}$ in $O(n^2)$ time, where n is the number of model versions.

Liu et al. [67] present an end-to-end model marketplace, which jointly considers data owners' privacy costs and model buyers' demands. A broker collects data from data owners,

and produces multiple versions of a machine learning model for sale with different subsets of training data and different differential privacy levels ϵ. The revenues are fully distributed to data owners. Objective perturbation [15] is used to train models with required differential privacy levels, which injects quantified random noise into the objective function of a model. Each data owner s_i requests a minimum compensation for using the owner's data to train a model with ϵ-differential privacy, that is

$$\pi(s_i, \epsilon) = b_i \cdot c_i(\epsilon),$$

where b_i is proportional to the Shapley value of s_i with regard to all sellers' data sets and $c_i(\epsilon)$ is the privacy cost of s_i. A desirable pricing model should guarantee revenue maximization, arbitrage-freeness with respect to differential privacy levels, and covers the compensations to data owners. Computing the optimal pricing function is coNP-hard, and thus they propose a dynamic programming algorithm to solve the problem approximately. A limitation of the pricing model is that it cannot adjust prices with respect to dynamic customer demands, which may limit the broker's revenue.

Agarwal et al. [1] consider an online auction for machine learning model market, which is truthful and revenue maximizing. They assume buyers come one at a time, and each wants to purchase a machine learning model for the buyer's prediction task. Denote by $\mathcal{G}(\widehat{Y}_i, Y_i)$ the quality of a model's prediction $\widehat{Y}_i$ on buyer i's validation data set Y_i. The reward that buyer i receives from the model is $\mu_i \cdot \mathcal{G}(\widehat{Y}_i, Y_i)$, where μ_i is buyer i's private valuation on unit performance. Denote by p_i and b_i, respectively, the asking price of the broker and the bid of buyer i for unit performance. The broker produces a noisy machine learning model for buyer i based on the price difference $p_i - b_i$. Specifically, the model is trained on a data set with quantified injected random noise, such that the model's performance is degraded proportionally to $p_i - b_i$. Buyer i is charged by a function $RF(p_i, b_i, Y_i)$, which is designed following Myerson's payment function rule [77]. The utility that buyer i receives by bidding b_i is

$$\mathcal{U}(b_i) = \mu_i \cdot \mathcal{G}(\widehat{Y}_i, Y_i) - RF(p_i, b_i, Y_i),$$

where $\widehat{Y}_i$ is the prediction of the returned noisy model. It is shown that truthfully bidding the buyer's valuation μ_i can maximize buyer i's utility. The authors apply a multiplicative weights method [3] to compute the price p_i from historical revenues. They show that the pricing mechanism achieves maximum revenue.

6.2 Pricing raw data products versus machine learning models

At a high level, pricing machine learning models and raw data sets share a series of common desiderata and techniques. But their pricing models are essentially different from each other on at least four aspects.

First, the pricing units of machine learning models are often well defined and fixed. A machine learning model is usually priced and sold as a whole. Customers can purchase either a machine learning model or the usage of a machine learning model via API calls, where each call has a fixed price. In contrast, a raw data set can be consumed in multiple granularities. For example, a customer may be interested in the sales information of American customers in the last year. Another customer, however, may want to purchase the sales information during the Christmas season. Such flexibility makes it easier to version raw data products, and enables more flexible pricing mechanisms. For example, according to how much information is revealed, different prices can be assigned to different queries on the same database [29].

Second, versioning in model markets is harder than that in data markets. As data sets have strong and flexible aggregate-ability, different versions of a data set can be easily produced by aggregating along different dimensions. Producing different versions of a machine learning model requires more sophisticated techniques [17], since it is challenging to accurately control the differences between multiple versions.

Third, the value of raw data sets to customers is generally harder to measure than that of machine learning models. Often, raw data sets are used to train machine learning models. The ultimate value of a data set dependents not only on its intrinsic properties but also on the specific task that the data set is used for and the analyzing methods [34]. Therefore, it is usually hard for customers to understand the value of a data set. Many machine learning models are designed for specific tasks and are directly used by people to support decision making [1]. It is easier for people to verify and understand the value of such machine learning models. For example, customers can value a classification model based on its prediction accuracy.

Last, preventing arbitrage is usually harder in model market than in raw data market. As shown by Tramér et al. [109], Yu et al. [119], machine learning models may be stolen by adversaries via a reasonable number of API calls. A customer with a large number of query instances may first purchase some predictions from a target machine learning model. Then, the customer can train a local model with near-equivalent outputs as the target model and use the local model to predict the remaining query instances with almost no cost.

In this section, we review pricing machine learning models. We first revisit arbitrage-free and revenue maximization pricing models. Then, we discuss several major differences between machine learning model products and raw data set products, including pricing units, versioning, arbitrage prevention, and customer valuation.

7 Data pricing in practice

Good progress has been made on pricing data products. There are, however, still very limited studies about how to apply the theoretical data pricing models in real-world applications. In this section, we first introduce some real-world data marketplaces and their data pricing models. Then, we provide guidelines for data pricing practitioners to choose proper pricing models for their application scenarios.

7.1 Data pricing models in industry

In recent years, many data marketplaces have emerged, where raw and processed data sets, the services of data labeling, and machine learning models are traded [105]. Six pricing models are commonly used in the real-world data marketplaces [86, 105], namely free data, flat fee tariff model, pay-per-use model, package pricing, two-part tariff model, and freemium. Data marketplaces operated by authorities or non-profit organizations usually provide raw data sets free of charge. The pay-per-use pricing model sets a price for each consumed unit of data products, and charges data buyers based on their consumptions. In the package pricing model, a data buyer can purchase a certain amount of data or API calls for a fixed price. In the flat fee tariff (subscription) pricing model, data buyers pay a fixed price to get unlimited access to a data product for a certain period of time. The two-part tariff model charges customers a fixed basic fee plus an additional fee per unit consumed. Last, in the freemium model, a data buyer can obtain basic access to data products for free and pay for advanced features.

The UCI machine learning repository [91] and Kaggle [55] are data marketplaces offering free data sets for some specific machine learning tasks. Till January 2022, the UCI machine learning repository and Kaggle provide 622 and 125,771 free data sets, respectively.

Datarade [24] is one of the largest commercial data marketplaces in the world, where data buyers can purchase commercial data sets from the Datarade authorized data vendors. The data sets on sale extend across several industries and sectors, such as real estate data, consumer data, and mobile application data. Datarade supports the pay-per-use pricing model, the package pricing model, and the flat fee tariff pricing model. For instance, Datastream, a US data vendor on the marketplace, sells a single query to its identity matching data set for $1.9, a package of one thousand queries to the data set at a discount price of $19, and unrestricted monthly access to the data set for a subscription fee of $28,500 [64].

The CARUSO dataplace [13] is a sell-side marketplace [120], which is operated by a single data seller. The data seller first collects in-vehicle data from participating cars, such as battery voltage, coolant temperature, and check control messages, and then sells the data to data buyers. The marketplace employs the two-part tariff model, that is, data buyers are charged a fixed membership fee plus additional fees for their consumed packages of data items, which may range between 0.30 euros and 8.50 euros per package.

REKLAIM [90] is a buy-side marketplace [120], where the single data buyer pays individuals for sharing their personal data, such as identity information, browsing history, and financial data. In REKLAIM, the flat fee tariff pricing model is used. The data buyer pays different monthly rewards to the participating individuals who share different types of information.

In addition to raw data sets, obtaining data labels is also critical in building machine learning models. Amazon Mechanical Turk [110] and Appen [2] are leading crowdsourcing marketplaces for this purpose, where the two-part tariff pricing model is adopted. A data buyer first publishes a group of data labeling tasks, where each task is associated with a fixed reward determined by the data buyer. The reward is commonly set between 1 and 3 cents per label/judgment. To motivate workers to exert efforts and submit accurate labels, after verifying the collected data labels, the data buyer is recommended to pay bonus rewards to the workers submitting high-quality data labels.

Machine learning as a service (MLaaS) [18, 109] is a rapidly growing industry. Google Cloud [19] sells the API access to its well-trained machine learning models. The freemium and the package pricing models are adopted to price the API calls. For instance, a customer of Google cloud can obtain 1000 free API calls to an image classifier per month, and pay $1.5 for each additional 1000 API calls.

7.2 A quick summary and comparisons of data pricing models

Different data pricing models have their particular suitable application scenarios. This suggests that the data pricing practitioners need to pick the data pricing models on a case-by-case basis for their settings at hand. Whether a pricing model is suitable to a setting often depends on multiple factors, including the practitioner's pricing task, the type of data products to be priced, the practitioner's optimization goals, and the theoretical assumptions that the pricing model relies on.

As discussed in Sect. 1, we focus on four data pricing tasks in machine learning pipelines, namely pricing raw data sets, pricing data labels, revenue allocation in collaborative machine learning, and pricing machine learning models. We summarize all of the representative data pricing models for the four data pricing tasks in Tables 1, 2, 3, and 4, respectively. The four

Table 1 The representative data pricing models of raw data sets

Product	Objectives	References
General data	No specific optimization goals	[43]
	(1) Revenue maximization	[117]
Sensing data	(1) Truthfulness; (2) Individual rationality; (3) Profitability	[115]
	(1) Truthfulness; (2) Social welfare maximization	[52]
	(1) Truthfulness (2) Buyer's cost minimization	[60]
Chain queries, conjunctive queries, and cyclic queries	(1) Arbitrage-freeness; (2) Discount-freeness	[58]
Conjunctive queries	(1) Arbitrage-freeness; (2) Discount-freeness; (3) History-awareness; (4) Fairness	[59]
General data queries	(1) Arbitrage-freeness	[28], [65]
	(1) Arbitrage-freeness; (2) History-awareness	[29]
	(1) Arbitrage-freeness; (2) Revenue maximization	[16]
Selection-projection-natural join queries over incomplete databases	(1) Arbitrage-freeness; (2) History-awareness	[75]
Selection queries	(1) History-awareness	[111]
Counting queries on binary data	(1) Privacy compensation; (2) Truthfulness; (3) Buyer's cost minimization	[38]
Linear aggregation queries	(1) Privacy compensation; (2) Query accuracy maximization under budget constraint	[21]
	(1) Privacy compensation; (2) Personalized maximum tolerable privacy loss; (3) Query accuracy maximization under budget constraint	[83]
	(1) Privacy compensation; (2) Truthfulness; (3) Personalized maximum tolerable privacy loss; (4) Query accuracy maximization under budget constraint	[121]
	(1) Privacy compensation; (2) Arbitrage-freeness	[62]
	(1) Privacy compensation; (2) Arbitrage-freeness; (3) Dependency fairness	[84]
Geo-location data	(1) Privacy compensation; (2) Data accuracy maximization under budget constraint	[53]

tables can be used as a starting point in helping data pricing practitioners to choose the right pricing models for their settings.

Tables 1, 2, 3, and 4 have three columns. The first column displays the data products that can be priced by the model. Pricing models with the value "general data" in the column means that the models have no restrictions on the types of data sets to be priced. For instance, the pricing models can be applied to image data, time series data, video data, etc. The second column shows the optimization goals of the pricing models. A pricing model may be optimized toward multiple objectives. The last column shows the references to the data pricing models.

Table 2 The representative data pricing models of data labels

Product	Objectives	References
Binary-label tasks	(1) Effort elicitation with gold tasks; (2) No-free-lunch axiom	[96]
	(1) Effort elicitation without gold tasks	[22]
	(1) Effort elicitation without gold tasks; (2) Reducing duplicate answers	[68]
	(1) Effort elicitation without gold tasks; (2) Buyer's utility maximization	[46, 69]
	(1) Effort elicitation without gold tasks; (2) Spammers' rewards minimization	[87]
Multi-label tasks	(1) Effort elicitation without gold tasks	[88, 100]
	(1) Effort elicitation with gold tasks; (2) No-free-lunch axiom	[98]
	(1) Effort elicitation with gold tasks; (2) No-free-lunch axiom; (3) Motivating workers to self-correct answers	[97]
	(1) Effort elicitation with gold tasks; (2) Improving scalability of gold tasks; (3) Fairness	[40]
Binary-label tasks and numeric-label tasks	(1) Effort elicitation with gold tasks; (2) Improving scalability of gold tasks	[26]

Table 3 The representative data pricing models of collaborative model training

Product	Objectives	References
Data sets for general ML models	(1) Shapley fairness	[37, 50, 73]
	(1) Shapley fairness; (2) Individual rationality; (3) Stability of the grand coalition; (4) Group welfare maximization	[102]
	(1) Shapley fairness; (2) Data replication-robustness	[1, 42, 85]
	(1) Coalition stability	[114]
	(1) Improving the performance of the collaboratively trained model	[116]
Data sets for kNN classifiers	(1) Shapley fairness	[49]
Data sets for federated learning	(1) Shapley fairness	[113]
	(1) Shapley fairness; (2) Regret fairness	[118]
	(1) Truthfulness	[93]

To find an appropriate data pricing model, a practitioner can match the pricing task, the type of data products, and the optimization goals in his/her setting with the ones in Tables 1, 2, 3, and 4. Most of the pricing models shown in Tables 3 and 4 are developed in the settings that classifiers are trained. As discussed in Sect. 5, we can adapt those models to the scenarios where unsupervised machine learning models are trained and traded by developing proper utility functions.

Table 4 The representative data pricing models of machine learning models

Product	Objectives	References
ML models	(1) Arbitrage-freeness; (2) Revenue maximization	[17]
	(1) Arbitrage-freeness, revenue maximization; (2) Privacy compensation	[67]
	(1) Truthfulness; (2) Revenue maximization	[1]

Please note that many data pricing models are developed under some assumptions on their application scenarios. For example, the pricing model of revenue allocation proposed by Maleki et al. [73] requires that the utility function of the collaboratively trained machine learning model is bounded. Therefore, before applying a selected pricing model, the practitioner also needs to carefully verify whether the assumptions of the pricing model hold in the application setting. If the assumptions do not hold, the pricing model may not function as expected and thus needs to be changed or refined.

In this section, we discuss how to perform data pricing in practice. We first review the pricing models adopted by some representative data marketplaces in industry. Then, we provide guidelines for the data pricing practitioners to choose proper pricing models for their settings.

8 Conclusions and future directions

In this paper, we survey data pricing in end-to-end machine learning pipelines. We consider three important steps in machine learning pipelines where pricing may be substantially involved, namely raw data collection and labeling, collaborative training machine learning models, and machine learning model marketplaces. We systematically review representative studies in those steps, discuss the pricing principles and review the existing methods. End-to-end machine learning pipelines are playing a more and more important role in the current big data and AI economics era. To the best of our knowledge, this is the first survey on data pricing in machine learning pipelines.

Data pricing is still in its early stage. There are many research challenges for future works. We list some of them here.

First, the existing studies focus on designing proper rewarding models in each separate stage of machine learning pipelines. There is a lack of systematic study of an end-to-end revenue allocation solution. As presented in our survey, the manufacturing process of machine learning models involves multiple parties, including data owners, data processors, machine learning model designers, and other possible participants. Each party provides value-added contributions at one stage of the pipeline and receives a reward. A natural question is how to allocate manufacturing budgets among different parties. To answer the question, we need a mechanism to measure and compare the contributions of different parties in different stages. We also need a system that can dynamically adjust the budget allocations in response to the changes in supply and demand.

Second, almost all pricing models of collaborative model training formulate revenue allocation as a cooperative game, and use Shapley value to carry out the allocation. They justify

the usage of Shapley value through the four axioms, namely balance, symmetry, zero element, and additivity. However, Yan and Procaccia [114] argue that the necessity of additivity for data valuation is debatable. Except for the additivity axiom, many other celebrated allocation solutions in cooperative game theory can also satisfy the other three axioms. Compared with Shapley value, the other solutions have their advantages and limitations. For example, normalized Banzhaf value [14] computes the payment to each player as the player's average marginal contribution toward all coalitions of other players. Even though normalized Banzhaf value does not satisfy the additivity axiom, it is more robust to data replication attacks than Shapley value [42]. In a marketplace where robustness is more important than additivity, normalized Banzhaf value is more preferable than Shapley value. Different types of data marketplaces may have different goals [34], and thus require different axioms. Therefore, we need a better understanding about the necessary axioms in different marketplaces and explore revenue allocation solutions in specific marketplaces.

Third, fine-grained data procurement for machine learning tasks is not fully explored. In practice, data sets from two sellers may have similar or overlapping parts. A data buyer with a limited budget may not want to purchase many similar data points, as the diversity of training data sets is critical to the performance of machine learning models [34]. Query-based pricing models [58] allow data buyers to only purchase their interested parts of a data set. However, the existing query-based pricing models are only designed for relational data sets in monopoly markets. Supporting query-based pricing in marketplaces of general data sets with competing sellers brings new challenges and opportunities. For example, it is interesting for data buyers to explore how to distribute their budgets among data sellers to maximize the utility of purchased data sets. For data sellers, it is important to assign prices to different parts of their data sets based on supply and demand, such that the data sellers and their data sets can remain competitive in the market.

Last, rigorous evaluation methods for data pricing models need to be developed. Many existing pricing models are only evaluated in oversimplified experimental environments, where many assumptions are made on the behaviors of market participants. A theoretically sound model, however, may not work in practice, as some model assumptions may break. For example, in a real-world market, participants can have adversarial, ignorant, or coalition-building behaviors. However, the effects of those behaviors on the performance of pricing models are largely dismissed in detailed analysis. Therefore, as suggested by Fernandez et al. [34], a simulation platform that can simulate different behaviors of market participants should be developed. The platform can help us study the advantages and limitations of pricing models in target environments, and choose the best one to deploy.

References

1. Agarwal A, Dahleh MA, Sarkar T (2019) A marketplace for data: an algorithmic solution. In: Karlin A, Immorlica N, Johari R (eds) Proceedings of the 2019 ACM conference on economics and computation, EC 2019, Phoenix, AZ, USA, June 24–28, 2019. ACM, pp 701–726. https://doi.org/10.1145/3328526.3329589
2. Appen (2022) Appen. https://appen.com. Accessed 04 Jan 2022
3. Arora S, Hazan E, Kale S (2012) The multiplicative weights update method: a meta-algorithm and applications. Theory Comput 8(1):121–164. https://doi.org/10.4086/toc.2012.v008a006
4. Ausubel LM, Milgrom P et al (2006) The lovely but lonely Vickrey auction. Comb Auctions 17:22–26
5. Balasubramanian S, Bhattacharya S, Krishnan VV (2015) Pricing information goods: a strategic analysis of the selling and pay-per-use mechanisms. Mark Sci 34(2):218–234
6. BDEX (2021) Bdex. https://www.bdex.com. Accessed 09 May 2021

7. Brendan M, Daniel R (2017) Federated learning: collaborative machine learning without centralized training data. Google AI Blog. https://ai.googleblog.com/2017/04/federated-learning-collaborative.html. Accessed 02 Jul 2021
8. Brennan R, Canning L, Mcdowell R (2013) Business-to-business marketing. Sage Publications, Thousand Oaks. https://doi.org/10.4135/9781446276518
9. Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler DM, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, Amodei D (2020) Language models are few-shot learners. In: Larochelle H, Ranzato M, Hadsell R, Balcan M, Lin H (eds) Advances in neural information processing systems 33: annual conference on neural information processing systems 2020, NeurIPS 2020, December 6–12, 2020, virtual. https://proceedings.neurips.cc/paper/2020/hash/1457c0d6bfcb4967418bfb8ac142f64a-Abstract.html
10. Buneman P, Tan WC (2007) Provenance in databases. In: Chan CY, Ooi BC, Zhou A (eds) Proceedings of the ACM SIGMOD international conference on management of data, Beijing, China, June 12–14, 2007. ACM, pp 1171–1173. https://doi.org/10.1145/1247480.1247646
11. Burkett JP (2006) Microeconomics: optimization, experiments, and behavior. OUP Catalogue. Oxford University Press, New York
12. Caliński T, Harabasz J (1974) A dendrite method for cluster analysis. Commun Stat Theory Methods 3(1):1–27
13. Caruso (2022) Caruso dataplace. https://www.caruso-dataplace.com/pricing/. Accessed 04 Jan 2022
14. Chalkiadakis G, Elkind E, Wooldridge MJ (2011) Computational aspects of cooperative game theory. Synthesis lectures on artificial intelligence and machine learning. Morgan & Claypool Publishers, San Rafael. https://doi.org/10.2200/S00355ED1V01Y201107AIM016
15. Chaudhuri K, Monteleoni C, Sarwate AD (2011) Differentially private empirical risk minimization. J Mach Learn Res 12:1069–1109
16. Chawla S, Deep S, Koutris P, Teng Y (2019) Revenue maximization for query pricing. Proc VLDB Endow 13(1):1–14. https://doi.org/10.14778/3357377.3357378
17. Chen L, Koutris P, Kumar A (2019) Towards model-based pricing for machine learning in a data marketplace. In: Boncz PA, Manegold S, Ailamaki A, Deshpande A, Kraska T (eds) Proceedings of the 2019 international conference on management of data, SIGMOD conference 2019, Amsterdam, The Netherlands, June 30–July 5, 2019. ACM, pp 1535–1552. https://doi.org/10.1145/3299869.3300078
18. Chen L, Zaharia M, Zou JY (2020) Frugalml: How to use ML prediction APIs more accurately and cheaply. In: Advances in neural information processing systems, vol 33
19. Cloud G (2022) Google cloud. https://cloud.google.com/products/ai. Accessed 04 Jan 2022
20. Cook RD, Weisberg S (1980) Characterizations of an empirical influence function for detecting influential cases in regression. Technometrics 22(4):495–508. https://doi.org/10.1080/00401706.1980.10486199
21. Dandekar P, Fawaz N, Ioannidis S (2012) Privacy auctions for recommender systems. In: Goldberg PW (ed) Internet and network economics—8th international workshop, WINE 2012, Liverpool, UK, December 10–12, 2012. Proceedings, lecture notes in computer science, vol 7695. Springer, pp 309–322. https://doi.org/10.1007/978-3-642-35311-6_23
22. Dasgupta A, Ghosh A (2013) Crowdsourced judgement elicitation with endogenous proficiency. In: Schwabe D, Almeida VAF, Glaser H, Baeza-Yates R, Moon SB (eds) 22nd International world wide web conference, WWW '13, Rio de Janeiro, Brazil, May 13–17, 2013. International World Wide Web Conferences Steering Committee/ACM, pp 319–330. https://doi.org/10.1145/2488388.2488417
23. Datar M, Immorlica N, Indyk P, Mirrokni VS (2004) Locality-sensitive hashing scheme based on p-stable distributions. In: Snoeyink J, Boissonnat J (eds) Proceedings of the 20th ACM symposium on computational geometry, Brooklyn, New York, USA, June 8–11, 2004. ACM, pp 253–262. https://doi.org/10.1145/997817.997857
24. Datarade (2022) Datarade data marketplace. https://datarade.ai. Accessed 04 Jan 2022
25. Dawex (2021) Dawex. https://www.dawex.com/en/. Accessed 09 May 2021
26. de Alfaro L, Faella M, Polychronopoulos V, Shavlovsky M (2016) Incentives for truthful evaluations. CoRR arXiv:1608.07886
27. De Toni D, Milan GS, Saciloto EB, Larentis F (2017) Pricing strategies and levels and their impact on corporate profitability. Revista de Administração (São Paulo) 52(2):120–133
28. Deep S, Koutris P (2017a) The design of arbitrage-free data pricing schemes. In: Benedikt M, Orsi G (eds) 20th International conference on database theory, ICDT 2017, March 21–24, 2017, Venice, Italy, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, LIPIcs, vol 68, pp 12:1–12:18. https://doi.org/10.4230/LIPIcs.ICDT.2017.12
29. Deep S, Koutris P (2017b) QIRANA: a framework for scalable query pricing. In: Salihoglu S, Zhou W, Chirkova R, Yang J, Suciu D (eds) Proceedings of the 2017 ACM international conference on

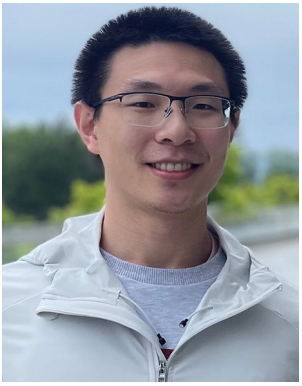
management of data, SIGMOD conference 2017, Chicago, IL, USA, May 14–19, 2017. ACM, pp 699–713. https://doi.org/10.1145/3035918.3064017
30. Devlin J, Chang M, Lee K, Toutanova K (2019) BERT: pre-training of deep bidirectional transformers for language understanding. In: Burstein J, Doran C, Solorio T (eds) Proceedings of the 2019 conference of the North American chapter of the Association for Computational Linguistics: human language technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2–7, 2019, vol 1 (long and short papers). Association for Computational Linguistics, pp 4171–4186. https://doi.org/10.18653/v1/n19-1423
31. Dibb S, Simkin L, Pride WM, Ferrell O (2005) Marketing: concepts and strategies, 5th edn. Houghton Mifflin, Abingdon
32. Dwork C (2008) Differential privacy: a survey of results. In: Agrawal M, Du D, Duan Z, Li A (eds) Theory and applications of models of computation, 5th international conference, TAMC 2008, Xi'an, China, April 25–29, 2008. Proceedings, lecture notes in computer science, vol 4978. Springer, pp 1–19. https://doi.org/10.1007/978-3-540-79228-4_1
33. Ensthaler L, Giebe T (2014) Bayesian optimal knapsack procurement. Eur J Oper Res 234(3):774–779. https://doi.org/10.1016/j.ejor.2013.09.031
34. Fernandez RC, Subramaniam P, Franklin MJ (2020) Data market platforms: trading data assets to solve data problems. Proc VLDB Endow 13(12):1933–1947. https://doi.org/10.14778/3407790.3407800
35. Fricker SA, Maksimov YV (2017) Pricing of data products in data marketplaces. In: Ojala A, Holmström Olsson H, Werder K (eds) Software business. Springer, Cham, pp 49–66
36. Fung C, Beschastnikh I (2019) Brokered agreements in multi-party machine learning. In: Proceedings of the 10th ACM SIGOPS Asia-Pacific workshop on systems, pp 69–75
37. Ghorbani A, Zou JY (2019) Data shapley: equitable valuation of data for machine learning. In: Chaudhuri K, Salakhutdinov R (eds) Proceedings of the 36th international conference on machine learning, ICML 2019, 9–15 June 2019, Long Beach, California, USA, PMLR, proceedings of machine learning research, vol 97, pp 2242–2251. http://proceedings.mlr.press/v97/ghorbani19c.html
38. Ghosh A, Roth A (2011) Selling privacy at auction. In: Shoham Y, Chen Y, Roughgarden T (eds) Proceedings 12th ACM conference on electronic commerce (EC-2011), San Jose, CA, USA, June 5–9, 2011. ACM, pp 199–208. https://doi.org/10.1145/1993574.1993605
39. Gillies DB (1959) Solutions to general non-zero-sum games. Contrib Theory Games 4:47–85
40. Goel N, Faltings B (2019) Deep Bayesian trust: a dominant and fair incentive mechanism for crowd. In: The thirty-third AAAI conference on artificial intelligence, AAAI 2019, the thirty-first innovative applications of artificial intelligence conference, IAAI 2019, the ninth AAAI symposium on educational advances in artificial intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27–February 1, 2019. AAAI Press, pp 1996–2003. https://doi.org/10.1609/aaai.v33i01.33011996
41. Gur Y, Zeevi AJ, Besbes O (2014) Stochastic multi-armed-bandit problem with non-stationary rewards. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ (eds) Advances in neural information processing systems 27: annual conference on neural information processing systems 2014, December 8–13 2014, Montreal, Quebec, Canada, pp 199–207. https://proceedings.neurips.cc/paper/2014/hash/903ce9225fca3e988c2af215d4e544d3-Abstract.html
42. Han D, Tople S, Rogers A, Wooldridge MJ, Ohrimenko O, Tschiatschek S (2020) Replication-robust payoff-allocation with applications in machine learning marketplaces. CoRR arXiv:2006.14583
43. Heckman JR, Boehmer EL, Peters EH, Davaloo M, Kurup NG (2015) A pricing model for data markets. iConference 2015 proceedings
44. Hoeffding W (1994) Probability inequalities for sums of bounded random variables. In: The collected works of Wassily Hoeffding. Springer, pp 409–426
45. Hu R, Gong Y (2020) Trading data for learning: incentive mechanism for on-device federated learning. In: IEEE global communications conference, GLOBECOM 2020, virtual event, Taiwan, December 7-11, 2020. IEEE, pp 1–6. https://doi.org/10.1109/GLOBECOM42002.2020.9322475
46. Hu Z, Liang Y, Zhang J, Li Z, Liu Y (2018) Inference aided reinforcement learning for incentive mechanism design in crowdsourcing. In: Bengio S, Wallach HM, Larochelle H, Grauman K, Cesa-Bianchi N, Garnett R (eds) Advances in neural information processing systems 31: annual conference on neural information processing systems 2018, NeurIPS 2018, December 3–8, 2018, Montréal, Canada, pp 5512–5522. https://proceedings.neurips.cc/paper/2018/hash/f2e43fa3400d826df4195a9ac70dca62-Abstract.html
47. Hynes N, Dao D, Yan D, Cheng R, Song D (2018) A demonstration of sterling: a privacy-preserving data marketplace. Proc VLDB Endow 11(12):2086–2089. https://doi.org/10.14778/3229863.3236266
48. Irvin G (1978) Modern cost-benefit methods. Macmillan Publishers Limited, London. https://doi.org/10.1007/978-1-349-15912-3

49. Jia R, Dao D, Wang B, Hubis FA, Gürel NM, Li B, Zhang C, Spanos CJ, Song D (2019) Efficient task-specific data valuation for nearest neighbor algorithms. Proc VLDB Endow 12(11):1610–1623. https://doi.org/10.14778/3342263.3342637
50. Jia R, Dao D, Wang B, Hubis FA, Hynes N, Gürel NM, Li B, Zhang C, Song D, Spanos CJ (2019) Towards efficient data valuation based on the Shapley value. In: Chaudhuri K, Sugiyama M (eds) The 22nd international conference on artificial intelligence and statistics, AISTATS 2019, 16–18 April 2019, Naha, Okinawa, Japan, PMLR, proceedings of machine learning research, vol 89, pp 1167–1176. http://proceedings.mlr.press/v89/jia19a.html
51. Jiang C, Gao L, Duan L, Huang J (2015) Economics of peer-to-peer mobile crowdsensing. In: 2015 IEEE global communications conference (GLOBECOM). IEEE, pp 1–6
52. Jin H, Su L, Chen D, Nahrstedt K, Xu J (2015) Quality of information aware incentive mechanisms for mobile crowd sensing systems. In: Shen SX, Sun Y, Chen J, Zhang J, Zussman G (eds) Proceedings of the 16th ACM international symposium on mobile ad hoc networking and computing, MobiHoc 2015, Hangzhou, China, June 22–25, 2015. ACM, pp 167–176. https://doi.org/10.1145/2746285.2746310
53. Jin W, Xiao M, Li M, Guo L (2019) If you do not care about it, sell it: trading location privacy in mobile crowd sensing. In: 2019 IEEE conference on computer communications, INFOCOM 2019, Paris, France, April 29–May 2, 2019. IEEE, pp 1045–1053. https://doi.org/10.1109/INFOCOM.2019.8737457
54. Jorgensen Z, Yu T, Cormode G (2015) Conservative or liberal? personalized differential privacy. In: Gehrke J, Lehner W, Shim K, Cha SK, Lohman GM (eds) 31st IEEE international conference on data engineering, ICDE 2015, Seoul, South Korea, April 13–17, 2015. IEEE Computer Society, pp 1023–1034. https://doi.org/10.1109/ICDE.2015.7113353
55. Kaggle (2022) Kaggle. https://www.kaggle.com/datasets. Accessed 04 Jan 2022
56. Kang J, Xiong Z, Niyato D, Xie S, Zhang J (2019) Incentive mechanism for reliable federated learning: a joint optimization approach to combining reputation and contract theory. IEEE Internet Things J 6(6):10700–10714. https://doi.org/10.1109/JIOT.2019.2940820
57. Koh PW, Liang P (2017) Understanding black-box predictions via influence functions. In: Precup D, Teh YW (eds) Proceedings of the 34th international conference on machine learning, ICML 2017, Sydney, NSW, Australia, 6–11 August 2017, PMLR, proceedings of machine learning research, vol 70, pp 1885–1894. http://proceedings.mlr.press/v70/koh17a.html
58. Koutris P, Upadhyaya P, Balazinska M, Howe B, Suciu D (2012) Query-based data pricing. In: Benedikt M, Krötzsch M, Lenzerini M (eds) Proceedings of the 31st ACM SIGMOD-SIGACT-SIGART symposium on principles of database systems, PODS 2012, Scottsdale, AZ, USA, May 20–24, 2012. ACM, pp 167–178. https://doi.org/10.1145/2213556.2213582
59. Koutris P, Upadhyaya P, Balazinska M, Howe B, Suciu D (2013) Toward practical query pricing with querymarket. In: Ross KA, Srivastava D, Papadias D (eds) Proceedings of the ACM SIGMOD international conference on management of data, SIGMOD 2013, New York, NY, USA, June 22–27, 2013. ACM, pp 613–624. https://doi.org/10.1145/2463676.2465335
60. Koutsopoulos I (2013) Optimal incentive-driven design of participatory sensing systems. In: Proceedings of the IEEE INFOCOM 2013, Turin, Italy, April 14–19, 2013. IEEE, pp 1402–1410. https://doi.org/10.1109/INFCOM.2013.6566934
61. Leyton-Brown K, Shoham Y (2008) Essentials of game theory: a concise multidisciplinary introduction. Synthesis lectures on artificial intelligence and machine learning. Morgan & Claypool Publishers, San Rafael. https://doi.org/10.2200/S00108ED1V01Y200802AIM003
62. Li C, Li DY, Miklau G, Suciu D (2013) A theory of pricing private data. In: Tan W, Guerrini G, Catania B, Gounaris A (eds) Joint 2013 EDBT/ICDT conferences, ICDT '13 proceedings, Genoa, Italy, March 18–22, 2013. ACM, pp 33–44. https://doi.org/10.1145/2448496.2448502
63. Liang F, Yu W, An D, Yang Q, Fu X, Zhao W (2018) A survey on big data market: pricing, trading and protection. IEEE Access 6:15132–15154
64. Lightsight (2022) Datastream identity dataset. https://datarade.ai/data-products/us-super-mobile2. Accessed 04 Jan 2022
65. Lin B, Kifer D (2014) On arbitrage-free pricing for general data queries. Proc VLDB Endow 7(9):757–768. https://doi.org/10.14778/2732939.2732948
66. Liu C, Chakraborty S, Mittal P (2016) Dependence makes you vulnberable: differential privacy under dependent tuples. In: 23rd annual network and distributed system security symposium, NDSS 2016, San Diego, California, USA, February 21–24, 2016. The Internet Society. http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2017/09/dependence-makes-you-vulnerable-differential-privacy-under-dependent-tuples.pdf
67. Liu J, Lou J, Liu J, Xiong L, Pei J, Sun J (2021) Dealer: an end-to-end model marketplace with differential privacy. Proc VLDB Endow 14(6):957–969. https://doi.org/10.14778/3447689.3447700

68. Liu Y, Chen Y (2017a) Machine-learning aided peer prediction. In: Daskalakis C, Babaioff M, Moulin H (eds) Proceedings of the 2017 ACM conference on economics and computation, EC '17, Cambridge, MA, USA, June 26–30, 2017. ACM, pp 63–80. https://doi.org/10.1145/3033274.3085126
69. Liu Y, Chen Y (2017) Sequential peer prediction: learning to elicit effort using posted prices. In: Singh SP, Markovitch S (eds) Proceedings of the thirty-first AAAI conference on artificial intelligence, February 4–9, 2017, San Francisco, California, USA. AAAI Press, pp 607–613. http://aaai.org/ocs/index.php/AAAI/AAAI17/paper/view/14970
70. Louis C (2020) Roundup of machine learning forecasts and market estimates, 2020. Forbes. https://www.forbes.com/sites/louiscolumbus/2020/01/19/roundup-of-machine-learning-forecasts-and-market-estimates-2020. Accessed 28 Jun 2021
71. Luong NC, Hoang DT, Wang P, Niyato D, Kim DI, Han Z (2016) Data collection and wireless communication in internet of things (IoT) using economic analysis and pricing models: a survey. IEEE Commun Surv Tutor 18(4):2546–2590
72. Ma L, Zhang C, Wang Y, Ruan W, Wang J, Tang W, Ma X, Gao X, Gao J (2020) Concare: personalized clinical feature embedding via capturing the healthcare context. In: The thirty-fourth AAAI conference on artificial intelligence, AAAI 2020, the thirty-second innovative applications of artificial intelligence conference, IAAI 2020, the tenth AAAI symposium on educational advances in artificial intelligence, EAAI 2020, New York, NY, USA, February 7–12, 2020. AAAI Press, pp 833–840. https://aaai.org/ojs/index.php/AAAI/article/view/5428
73. Maleki S, Tran-Thanh L, Hines G, Rahwan T, Rogers A (2013) Bounding the estimation error of sampling-based Shapley value approximation with/without stratifying. CoRR arXiv:1306.4265
74. McMahan B, Moore E, Ramage D, Hampson S, y Arcas BA (2017) Communication-efficient learning of deep networks from decentralized data. In: Singh A, Zhu XJ (eds) Proceedings of the 20th international conference on artificial intelligence and statistics, AISTATS 2017, 20–22 April 2017, Fort Lauderdale, FL, USA, PMLR, proceedings of machine learning research, vol 54, pp 1273–1282. http://proceedings.mlr.press/v54/mcmahan17a.html
75. Miao X, Gao Y, Chen L, Peng H, Yin J, Li Q (2020) Towards query pricing on incomplete data. IEEE Trans Knowl Data Eng
76. Muschalle A, Stahl F, Löser A, Vossen G (2012) Pricing approaches for data markets. In: International workshop on business intelligence for the real-time enterprise. Springer, pp 129–144
77. Myerson RB (1981) Optimal auction design. Math Oper Res 6(1):58–73
78. Nagle TT, Hogan J (2010) The strategy and tactics of pricing: a guide to growing more profitably. Prentice Hall, Hoboken
79. Nash A, Segoufin L, Vianu V (2007) Determinacy and rewriting of conjunctive queries using views: a progress report. In: Schwentick T, Suciu D (eds) Database theory—ICDT 2007, 11th international conference, Barcelona, Spain, January 10–12, 2007, Proceedings, lecture notes in computer science, vol 4353. Springer, pp 59–73. https://doi.org/10.1007/11965893_5
80. Nash JF (1950) Equilibrium points in n-person games. Proc Natl Acad Sci 36:48–49
81. Natarajan N, Dhillon IS, Ravikumar P, Tewari A (2013) Learning with noisy labels. In: Burges CJC, Bottou L, Ghahramani Z, Weinberger KQ (eds) Advances in neural information processing systems 26: 27th annual conference on neural information processing systems 2013. Proceedings of a meeting held December 5–8, 2013, Lake Tahoe, Nevada, United States, pp 1196–1204. https://proceedings.neurips.cc/paper/2013/hash/3871bd64012152bfb53fdf04b401193f-Abstract.html
82. Neumeier M (2015) The brand flip: why customers now run companies-and how to profit from it. New Riders, San Francisco
83. Nget R, Cao Y, Yoshikawa M (2017) How to balance privacy and money through pricing mechanism in personal data market. In: Degenhardt J, Kallumadi S, de Rijke M, Si L, Trotman A, Xu Y (eds) Proceedings of the SIGIR 2017 workshop On eCommerce co-located with the 40th international ACM SIGIR conference on research and development in information retrieval, eCOM@SIGIR 2017, Tokyo, Japan, August 11, 2017. CEUR-WS.org, CEUR workshop proceedings, vol 2311. http://ceur-ws.org/Vol-2311/paper_15.pdf
84. Niu C, Zheng Z, Wu F, Tang S, Gao X, Chen G (2018) Unlocking the value of privacy: trading aggregate statistics over private correlated data. In: Guo Y, Farooq F (eds) Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, KDD 2018, London, UK, August 19–23, 2018. ACM, pp 2031–2040. https://doi.org/10.1145/3219819.3220013
85. Ohrimenko O, Tople S, Tschiatschek S (2019) Collaborative machine learning markets with data-replication-robust payments. CoRR arXiv:1911.09052
86. Pei J (2020) A survey on data pricing: from economics to data science. IEEE Trans Knowl Data Eng. https://doi.org/10.1109/TKDE.2020.3045927

87. Radanovic G, Faltings B (2016) Learning to scale payments in crowdsourcing with properboost. In: Ghosh A, Lease M (eds) Proceedings of the fourth AAAI conference on human computation and crowdsourcing, HCOMP 2016, 30 October–3 November, 2016, Austin, Texas, USA. AAAI Press, pp 179–188. http://aaai.org/ocs/index.php/HCOMP/HCOMP16/paper/view/14033
88. Radanovic G, Faltings B, Jurca R (2016) Incentives for effort in crowdsourcing using the peer truth serum. ACM Trans Intell Syst Technol 7(4):48:1-48:28. https://doi.org/10.1145/2856102
89. Rauhut H (2010) Compressive sensing and structured random matrices. Theor Found Numer Methods Sparse Recovery 9:1–92
90. REKLAIM (2022) Reklaim. https://www.reklaimyours.com. Accessed 04 Jan 2022
91. Repository UML (2022) UCI machine learning repository. https://archive.ics.uci.edu/ml/index.php. Accessed 04 Jan 2022
92. Richardson A, Filos-Ratsikas A, Faltings B (2019) Rewarding high-quality data via influence functions. CoRR arXiv:1908.11598
93. Richardson A, Filos-Ratsikas A, Faltings B (2020) Budget-bounded incentives for federated learning. In: Yang Q, Fan L, Yu H (eds) Federated learning—privacy and incentive, lecture notes in computer science, vol 12500. Springer, Berlin, pp 176–188. https://doi.org/10.1007/978-3-030-63076-8_13
94. Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65
95. Schomm F, Stahl F, Vossen G (2013) Marketplaces for data: an initial survey. ACM SIGMOD Rec 42(1):15–26
96. Shah NB, Zhou D (2015) Double or nothing: multiplicative incentive mechanisms for crowdsourcing. In: Cortes C, Lawrence ND, Lee DD, Sugiyama M, Garnett R (eds) Advances in neural information processing systems 28: annual conference on neural information processing systems 2015, December 7–12, 2015, Montreal, Quebec, Canada, pp 1–9. https://proceedings.neurips.cc/paper/2015/hash/c81e728d9d4c2f636f067f89cc14862c-Abstract.html
97. Shah NB, Zhou D (2016) No oops, you won't do it again: mechanisms for self-correction in crowdsourcing. In: Balcan M, Weinberger KQ (eds) Proceedings of the 33nd international conference on machine learning, ICML 2016, New York City, NY, USA, June 19–24, 2016. JMLR.org, JMLR workshop and conference proceedings, vol 48, pp 1–10. http://proceedings.mlr.press/v48/shaha16.html
98. Shah NB, Zhou D, Peres Y (2015) Approval voting and incentives in crowdsourcing. In: Bach FR, Blei DM (eds) Proceedings of the 32nd international conference on machine learning, ICML 2015, Lille, France, 6–11 July 2015. JMLR.org, JMLR workshop and conference proceedings, vol 37, pp 10–19. http://proceedings.mlr.press/v37/shaha15.html
99. Shapley LS (1953) A value for n-person games. Contrib Theory Games 2:307–317
100. Shnayder V, Agarwal A, Frongillo RM, Parkes DC (2016) Informed truthfulness in multi-task peer prediction. In: Conitzer V, Bergemann D, Chen Y (eds) Proceedings of the 2016 ACM conference on economics and computation, EC '16, Maastricht, The Netherlands, July 24–28, 2016. ACM, pp 179–196. https://doi.org/10.1145/2940716.2940790
101. Shnayder V, Frongillo RM, Parkes DC (2016b) Measuring performance of peer prediction mechanisms using replicator dynamics. In: Kambhampati S (ed) Proceedings of the twenty-fifth international joint conference on artificial intelligence, IJCAI 2016, New York, NY, USA, 9–15 July 2016. IJCAI/AAAI Press, pp 2611–2617. http://www.ijcai.org/Abstract/16/371
102. Sim RHL, Zhang Y, Chan MC, Low BKH (2020) Collaborative machine learning with incentive-aware model rewards. In: Proceedings of the 37th international conference on machine learning, ICML 2020, 13–18 July 2020, virtual event, PMLR, proceedings of machine learning research, vol 119, pp 8927–8936. http://proceedings.mlr.press/v119/sim20a.html
103. Singer Y (2010) Budget feasible mechanisms. In: 51th annual IEEE symposium on foundations of computer science, FOCS 2010, October 23–26, 2010, Las Vegas, Nevada, USA. IEEE Computer Society, pp 765–774. https://doi.org/10.1109/FOCS.2010.78
104. Snowflake (2021) Snowflake data marketplace. https://www.snowflake.com/data-marketplace/. Accessed 09 May 2021
105. Spiekermann M (2019) Data marketplaces: trends and monetisation of data goods. Intereconomics 54(4):208–216
106. Stahl F, Vossen G (2016) Data quality scores for pricing on data marketplaces. In: Nguyen NT, Trawinski B, Fujita H, Hong T (eds) Intelligent information and database systems—8th Asian conference, ACIIDS 2016, Da Nang, Vietnam, March 14–16, 2016, proceedings, part I, lecture notes in computer science, vol 9621. Springer, pp 215–224. https://doi.org/10.1007/978-3-662-49381-6_21
107. Tang R, Wu H, Bao Z, Bressan S, Valduriez P (2013) The price is right—models and algorithms for pricing data. In: Decker H, Lhotská L, Link S, Basl J, Tjoa AM (eds) Database and expert systems applications—24th international conference, DEXA 2013, Prague, Czech Republic, August 26–29, 2013. Proceedings,

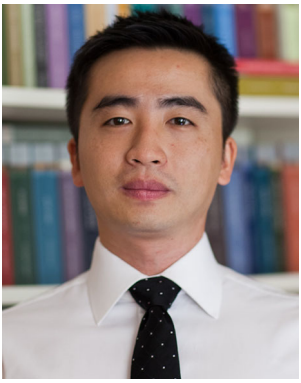
part II, lecture notes in computer science, vol 8056. Springer, pp 380–394. https://doi.org/10.1007/978-3-642-40173-2_31
108. Tang R, Amarilli A, Senellart P, Bressan S (2014) Get a sample for a discount—sampling-based XML data pricing. In: Decker H, Lhotská L, Link S, Spies M, Wagner RR (eds) Database and expert systems applications—25th international conference, DEXA 2014, Munich, Germany, September 1–4, 2014. Proceedings, part I, lecture notes in computer science, vol 8644. Springer, pp 20–34. https://doi.org/10.1007/978-3-319-10073-9_3
109. Tramèr F, Zhang F, Juels A, Reiter MK, Ristenpart T (2016) Stealing machine learning models via prediction APIs. In: Holz T, Savage S (eds) 25th USENIX security symposium, USENIX Security 16, Austin, TX, USA, August 10–12, 2016. USENIX Association, pp 601–618. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/tramer
110. Turk AM (2022) Amazon mechanical turk. https://www.mturk.com. Accessed 04 Jan 2022
111. Upadhyaya P, Balazinska M, Suciu D (2016) Price-optimal querying with data APIs. Proc VLDB Endow 9(14):1695–1706. https://doi.org/10.14778/3007328.3007335
112. Vaughan JW (2017) Making better use of the crowd: how crowdsourcing can advance machine learning research. J Mach Learn Res 18:193:1-193:46
113. Wang T, Rausch J, Zhang C, Jia R, Song D (2020) A principled approach to data valuation for federated learning. In: Yang Q, Fan L, Yu H (eds) Federated learning—privacy and incentive, lecture notes in computer science, vol 12500. Springer, Berlin, pp 153–167. https://doi.org/10.1007/978-3-030-63076-8_11
114. Yan T, Procaccia AD (2021) If you like Shapley then you'll love the core. In: Thirty-fifth AAAI conference on artificial intelligence, AAAI 2021, thirty-third conference on innovative applications of artificial intelligence, IAAI 2021, the eleventh symposium on educational advances in artificial intelligence, EAAI 2021, virtual event, February 2–9, 2021. AAAI Press, pp 5751–5759. https://ojs.aaai.org/index.php/AAAI/article/view/16721
115. Yang D, Xue G, Fang X, Tang J (2012) Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing. In: Akan ÖB, Ekici E, Qiu L, Snoeren AC (eds) The 18th annual international conference on mobile computing and networking, Mobicom'12, Istanbul, Turkey, August 22–26, 2012. ACM, pp 173–184. https://doi.org/10.1145/2348543.2348567
116. Yoon J, Arik SÖ, Pfister T (2020) Data valuation using reinforcement learning. In: Proceedings of the 37th international conference on machine learning, ICML 2020, 13–18 July 2020, virtual event, PMLR, proceedings of machine learning research, vol 119, pp 10842–10851. http://proceedings.mlr.press/v119/yoon20a.html
117. Yu H, Zhang M (2017) Data pricing strategy based on data quality. Comput Ind Eng 112:1–10. https://doi.org/10.1016/j.cie.2017.08.008
118. Yu H, Liu Z, Liu Y, Chen T, Cong M, Weng X, Niyato D, Yang Q (2020) A sustainable incentive scheme for federated learning. IEEE Intell Syst 35(4):58–69. https://doi.org/10.1109/MIS.2020.2987774
119. Yu H, Yang K, Zhang T, Tsai Y, Ho T, Jin Y (2020b) Cloudleak: large-scale deep learning models stealing through adversarial examples. In: 27th annual network and distributed system security symposium, NDSS 2020, San Diego, California, USA, February 23–26, 2020. The Internet Society. https://www.ndss-symposium.org/ndss-paper/cloudleak-large-scale-deep-learning-models-stealing-through-adversarial-examples/
120. Zhang M, Beltran F (2020) A survey of data pricing methods. SSRN
121. Zhang M, Beltrán F, Liu J (2020) Selling data at an auction under privacy constraints. In: Adams RP, Gogate V (eds) Proceedings of the thirty-sixth conference on uncertainty in artificial intelligence, UAI 2020, virtual online, August 3–6, 2020, proceedings of machine learning research, vol 124. AUAI Press, pp 669–678. http://proceedings.mlr.press/v124/zhang20b.html
122. Zhang X, Yang Z, Sun W, Liu Y, Tang S, Xing K, Mao X (2016) Incentives for mobile crowd sensing: a survey. IEEE Commun Surv Tutor 18(1):54–67. https://doi.org/10.1109/COMST.2015.2415528
123. Zhou X, Zheng H (2009) Trust: a general framework for truthful double spectrum auctions. In: IEEE INFOCOM 2009. IEEE, pp 999–1007
124. Zhou Y, Porwal U, Zhang C, Ngo HQ, Nguyen L, Ré C, Govindaraju V (2014) Parallel feature selection inspired by group testing. In: Ghahramani Z, Welling M, Cortes C, Lawrence ND, Weinberger KQ (eds) Advances in neural information processing systems 27: annual conference on neural information processing systems 2014, December 8–13 2014, Montreal, Quebec, Canada, pp 3554–3562. https://proceedings.neurips.cc/paper/2014/hash/fb8feff253bb6c834deb61ec76baa893-Abstract.html



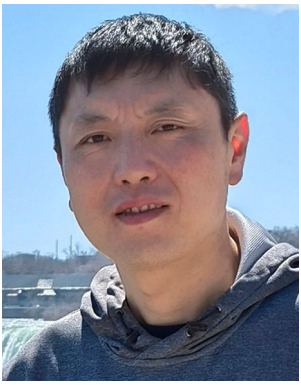
Zicun Cong is currently a PhD student at the School of Computing Science, Simon Fraser University, Canada. His research interests lie in machine learning and data mining, with an emphasis on data pricing and trustworthy artificial intelligence. He has worked extensively on interpreting the internal mechanisms of complex machine learning and statistical models. Currently, he is focusing on designing efficient, scalable, and interpretable algorithms for data and model pricing.

Xuan Luo is a PhD student in the Computing Science Department at Simon Fraser University, supervised by Professor Jian Pei. She received her master's degree in the University of British Columbia in 2020, where she was advised by Professor Victor C.M. Leung. Her research interest lies in fair data valuation in databases, especially applications of Shapley value for data pricing or data importance in database. Previously, she worked for years as a software engineer in the database departments in companies like Amazon.

Jian Pei is a Professor in the School of Computing Science at Simon Fraser University. He is a leading researcher in the general areas of data science, big data, data mining, and database systems. He is recognized as a Fellow of the Royal Society of Canada (Canada's National Academy), the Canadian Academy of Engineering, the Association of Computing Machinery (ACM), and the Institute of Electrical and Electronics Engineers (IEEE). Since 2000, he has published one textbook, two monographs, and over 300 research papers in refereed journals and conferences, which have been cited extensively by others. His research has generated remarkable impact substantially beyond academia. He also demonstrated outstanding professional leadership in many academic organizations and activities. He was the editor-in-chief of the IEEE Transactions of Knowledge and Data Engineering (TKDE) in 2013–2016, the chair of the Special Interest Group on Knowledge Discovery in Data (SIGKDD) of the Association for Computing Machinery (ACM) in 2017–2021, and a general co-chair or program committee co-chair of many premier conferences. He maintains a wide spectrum of industry relations with both global and local industry partners. He received many prestigious awards, including the 2017 ACM SIGKDD Innovation Award, the 2015 ACM SIGKDD Service Award, the 2014 IEEE ICDM Research Contributions Award, the British Columbia Innovation Council 2005 Young Innovator Award, an NSERC 2008 Discovery Accelerator Supplements Award (2008), an IBM Faculty Award (2006), a KDD Best Application Paper Award (2008), an ICDE Influential Paper Award (2018), a PAKDD Best Paper Award (2014), and a PAKDD Most Influential Paper Award (2009).



Feida Zhu is currently a tenured associate professor in the School of Computing and Information Systems at Singapore Management University. His research interests include blockchain, data asset and AI governance, privacy-aware large-scale data mining and machine learning, graph/network mining, and social network analysis, with emphasis on their application to business, financial, and consumer innovation. Dr. Zhu has over 100 peer-reviewed research publications at top international venues with multiple Best Paper Awards. He won the Early Career Award of PAKDDÕ19 and is the General Co-Chair of ICDMÕ18 and KDDÕ21. Prof. Zhu obtained his PhD in Computer Science from the University of Illinois at Urbana-Champaign (UIUC) in 2009.



Yong Zhang currently is a Distinguished Researcher at Huawei Technologies Canada and leading the big data and intelligence platform laboratory at Vancouver research center. Prior to that, he was a postdoctoral research fellow at Stanford University, USA. His research interests include large-scale numerical optimization and machine learning. His research works have been published in top-tier journals and conferences.