This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2900243, IEEE Transactions on Mobile Computing

ARETE: On Designing Joint Online Pricing and Reward Sharing Mechanisms for Mobile Data Markets

Zhenzhe Zheng, *Student Member, IEEE,* Yanqing Peng, Fan Wu, *Member, IEEE,* Shaojie Tang, *Member, IEEE,* and Guihai Chen, *Senior Member, IEEE*

Abstract—Although data has become an important kind of commercial goods, there are few appropriate online platforms to facilitate the trading of mobile crowd-sensed data so far. In this paper, we present the first architecture of mobile crowd-sensed data market, and conduct an in-depth study of the design problem of online data pricing and reward sharing. To build a practical mobile crowd-sensed data market, we have to consider four major design challenges: data uncertainty, economic-robustness (arbitrage-freeness in particular), profit maximization, and fair reward sharing. By jointly considering the design challenges, we propose an online query-bAsed cRowd-sensEd daTa pricing mEchanism, namely ARETE-PR, to determine the trading price of crowd-sensed data. Our theoretical analysis shows that ARETE-PR guarantees both arbitrage-freeness and a constant competitive ratio in terms of profit maximization. Based on some fairness criterions, we further design a reward sharing scheme, namely ARETE-SH, which is closely coupled with ARETE-PR, to incentivize data providers to contribute data. We have evaluated ARETE on a real-world sensory data set collected by Intel Berkeley lab. Evaluation results show that ARETE-PR outperforms the state-of-the-art pricing mechanisms, and achieves around 90% of the optimal revenue. ARETE-SH distributes the reward among data providers in a fair way.

Index Terms—Data Marketplace, Online Pricing, Profit Maximization, Shapley Value.

1 Introduction

As a significant business reality, data trading has attracted increasing attentions and focuses. For example, Xignite [60] sells financial data, Gnip [28] vends data from social networks, and Factual [27] trades geographic data. Potential data consumers might be Nasdaq [45] for financial data, Instagram [36] for social data, and Here [33] for location trace data. To support these online data transactions, several marketplace services have emerged, *e.g.*, Azure Data Marketplace [4], Infochimps [35], and Dataexchange [22]. These marketplace services offer centralized platforms, where data vendors can upload and sell their data, and data consumers can discover and purchase the data needed.

Although a few works have appeared to study the trading of structured and relational data [6], [40], mobile crowd-sensed data trading has not been fully explored in either industry or academia. Ranging from wireless sensor networks that monitor large wildlife environment [44] to vehicular networks for traffic monitoring and prediction [65], these deployments generate tremendous volumes of valuable but uncertain numeric sensed data. Due to lack of effective ways for data exchange, the mobile crowd-sensed data is currently used only by their operators for their own purposes. Such status has significantly suppressed market demand for mobile crowd-sensed data [11]. On one hand, data owners are willing to share their data for profits. On the other hand, data consumers, such as researchers, analysts, and application developers, would like to pay for data services built upon the acquired data. Therefore, it is highly needed to build an open data marketplace to enable mobile crowd-sensed data trading, and to boost data economy underlying the ubiquitous mobile data. Several open platforms, such as Thingspeak [55] and Thingful [54], have recently emerged for mobile data sharing on the Web, but none of them have deployed a practical data trading platform.

To design a flexible and practical mobile crowd-sensed data market, we have to cope with four major challenges. The first major challenge comes from the uncertainty of mobile crowd-sensed data, which makes it difficult to define the trading format of crowd-sensed data. The mobile data is normally noisy and imprecise [15], making it improper to directly feed raw data into data market. Furthermore, we can discover rich semantic information behind the raw data by aggregating data from multiple dimensions and domains [43]. Therefore, instead of directly selling raw data, the data vendor should design a statistical model to describe the raw data, and then provide semantically rich data services [11]. Researchers have proposed several model-based

F. Wu is the corresponding author.

- *This work was supported in part by the National Key R&D Program of China 2018YFB1004703, in part by China NSF grant 61672348 and 61672353, in part by Supported by the Open Project Program of the State Key Laboratory of Mathematical Engineering and Advanced Computing 2018A09, and in part by Alibaba Group through Alibaba Innovation Research Program, and In part by Initiative Postdocs Supporting Program. The opinions, findings, conclusions, and recommendations expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies or the government.*

- *Z. Zheng, Y. Peng, F. Wu, and G. Chen are with the Department of Computer Science and Engineering, Shanghai Key Laboratory of Scalable Computing and Systems, Shanghai Jiao Tong University, China. E-mails: zhengzhenzhe@sjtu.edu.cn; yqpeng@foxmail.com; {fwu, gchen}@cs.sjtu.edu.cn. S. Tang is with Department of Information Systems, University of Texas at Dallas, USA. E-mail: tangshaojie@gmail.com*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2900243, IEEE Transactions on Mobile Computing

methods to manage sensed data in the past decades [15], [24], [52]. However, due to the various formats of mobile sensed data and the complex correlation among data, it is difficult to select a universal and concise statistical model for all types of crowd-sensed data trading.

The second challenge is on designing flexible data pricing mechanisms with economic robustness guarantee. The pricing strategy currently used to sell data is simplistic, *i.e.*, the data vendor sets fixed prices for the whole or parts of the data set [4], [19]. This inflexible approach not only forces the data vendor to anticipate possible data subsets that data consumers might be interested in, but also drives the data consumers to purchase a superset of the data in need. To this end, a fine-grained data trading format, particularly, query-based data pricing [6], [40], is more suitable for data trading. In the data market with query-based data pricing mechanisms, data consumers can purchase ad-hoc queries over the whole data set, and thus have the flexibility to buy the data they exactly need. While providing convenience for data trading, this flexible data pricing mechanism can expose obscure arbitrage problems, in which a cunning data consumer may infer the answer of an expensive query from a set of cheaper queries. Thus, the data pricing mechanism should satisfy the property of *arbitrage-freeness* [40] to resist such manipulation behaviour. This introduces heavy burden on the design of data pricing mechanisms due to the complex arbitrage behaviour.

The third challenge is on profit maximization with incomplete information. The profit of a data vendor is the difference between data trading revenue and data acquisition cost. The problem of profit maximization can be decomposed into revenue maximization and data acquisition minimization. Data can be considered as one kind of information goods, which have a substantial initial investment cost, but tend to induce negligible marginal cost for reproduction. To minimize the data acquisition cost, which can be considered as the initial investment cost, we need to solve a submodular covering problem, which is a NP hard problem in general [38], [59]. For the revenue maximization, such a cost structure makes existing cost-based pricing mechanisms unsuitable, and the value-based pricing mechanisms are more attractable for data trading. However, in online data markets, data consumers may have diverse valuations even for the same data commodity. The data vendor may not know the valuation (and the valuation distribution) and the arrival sequences of data consumers. Thus, the data vendor has to determine the price of data with incomplete information. The optimization on profit maximization needs to take both the new cost structure and the lack of information into account, which inevitably doubles the difficulty in the design of data pricing mechanisms.

The last but not least challenge is on designing efficient reward sharing scheme aligned with fairness criterions. In data markets, the data vendor would provide some rewards for data providers to compensate their sensing costs, and to incentivize them to contribute large amount of high quality data [61]. In mobile crowdsensing system, the platform only compensates data providers for their incurred sensing cost, and hoards the revenue extracted from later data usage. In data markets, this is unfair to data providers, as the data commodities are generated based on the raw data contributed by data providers. We augment the basic reward with a bonus reward, which is a portion of revenue from data trading. Considering that the data providers may submit data with heterogeneous quality, the bonus reward sharing scheme design should be aligned with fairness criterions. However, the traditional reward sharing scheme that simultaneously satisfies the basic fairness axioms: *efficiency*, *symmetry*, *dummy*, and *additivity* (Please refer to Section 5.3 for definitions.), normally incurs high time and space complexity [50].

In this paper, we conduct an in-depth study on the problem of market design for mobile crowd-sensed data trading. First, we adopt a powerful statistical model, *i.e.*, Gaussian Process, to capture the uncertainty of numeric mobile sense data, and regard the resulting aggregated distributions as trading commodities in the data market. Based on this statistical model, we design a fine-grained query interface, containing three basic types of query formats, such that data consumers can obtain needed information through issuing ad-hoc queries. Second, we propose a query-based data pricing mechanism, namely ARETE-PR, to achieve arbitrage-freeness and a constant competitive ratio. Specifically, for each of data commodities, ARETE-PR generates multiple *versions* with different accuracy levels to extract revenue from data consumers in different market segments, and determines the trading prices of the data commodities by dynamically learning the valuations of data consumers. Third, we further design a reward sharing scheme, ARETE-SH, to efficiently calculate the Shapley value [50] for each data provider with the guideline of the four fairness axioms. To the best of our knowledge, we are the first to analyze the market structure of mobile crowd-sensed data trading, and propose an online pricing mechanism to facilitate this new kind of data business.

We summarize our contributions as follows.

- First, we present a marketplace for mobile crowd-sensed data trading, in which the data vendor can offer data services upon acquired raw data to obtain profit, and data consumers can purchase data services through issuing ad-hoc queries. We conduct a thorough analysis on the market structure of mobile crowd-sensed data trading, and examine the problems of profit maximization.
- Second, we begin with considering a basic setting, in which data consumers only ask single-data queries, and design ARETE-PR, including a versioning mechanism and an online pricing mechanism. We further extend ARETE-PR to adapt to other data query scenarios. We prove that ARETE-PR achieves both arbitrage-freeness and a constant competitive ratio in terms of profit maximization.
- Third, we formulate the problem of reward sharing as a coalitional game, and represent such reward sharing game by a Marginal-Contribution-Networks scheme [34]. With this concise representation scheme, we propose ARETE-SH to compute the Shapley value of the game in polynomial time, achieving the four fairness axioms.
- Finally, we evaluate the performance of ARETE with a real-world sensory data set. The evaluation results show that ARETE outperforms the state-of-the-art pricing mechanisms, and approaches the optimal fixed price revenue. The evaluation results also demonstrate that ARETE-SH can fairly distribute the rewards among data providers, and has

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2900243, IEEE Transactions on Mobile Computing

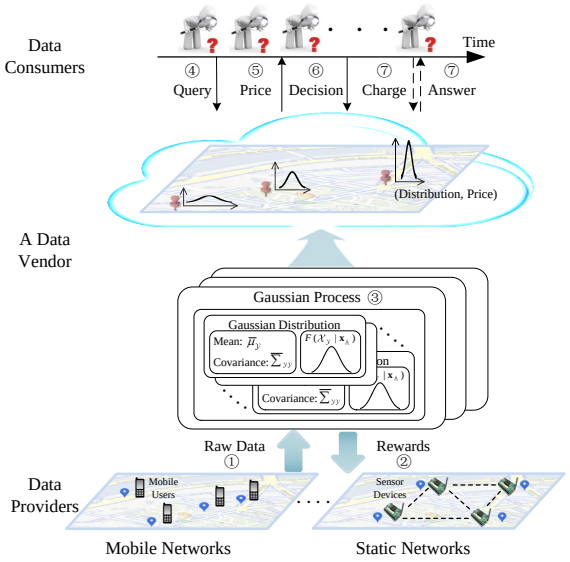


Fig. 1. A Mobile Crowd-Sensed Data Market.

a profound impact on the revenue of data trading in a long term.

The rest of this paper is organized as follows. In Section 2, we present system model and problem formulation. In Section 3, we propose a version-based online pricing mechanism, namely ARETE-PR. We extend ARETE-PR to support diverse query formats in Section 4. In Section 5, we formulate the problem of reward sharing as a coalitional game, and compute the Shapley value of the game. The evaluation results are presented in Section 6. In Section 7, we review related work. We conclude the paper in Section 8.

2 Preliminaries

In this section, we describe system model for mobile crowd-sensed data trading, and formally state the problems of profit maximization and reward sharing.

2.1 System Model

As illustrated by Figure 1, we consider a mobile crowd-sensed data marketplace with three major entities: a set of data providers, a data vendor, and a set of data consumers. In mobile crowd-sensing applications, the data vendor acquires raw data by employing data providers, such as sensor devices and mobile phone users, in a monitoring region, and wants to make profits from providing data services upon the collected data (Step ①). The data vendor would provide some rewards to incentivize data providers to report data (Step ②). Since the raw data is normally incomplete, imprecise, and erroneous, the data vendor needs to build statistical models to filter the raw data, and present a model-based query interface for data consumers (Step ③). The data consumers arrive at the data market sequentially, and request for data services through issuing ad-hoc queries over the statistical models (Step ④). The data vendor determines appropriate prices for data services in a principled way (Step ⑤). Upon receiving declared prices, the data consumer makes a purchasing decision (Step ⑥). If the data consumer accepts this price, she receives the answers of the queries, and pays for the price (Step ⑦). We introduce a set of major notations to define the crowd-sensed data market.

Data Providers: In a monitoring region Θ, the data vendor employs a set of m data providers to collect mobile data. Let $\mathbb{A} = \{a_1, a_2, \cdots, a_m\}$ denote the locations of the data providers, and vector $\mathbf{x}_{\mathbb{A}} = (x_1, x_2, \cdots, x_m)$ denote the real-time observations collected by data providers. We assume these observations are from authentic data sources, and data providers would not maliciously generate fake data from some distribution of data. As data providers consume their physical resources to collect data, the data vendor would like to distribute some monetary rewards to compensate their efforts, and incentivize them to contribute high quality data, which is similar to the incentive design in mobile crowdsensing systems [32], [61]. Data is one kind of digital goods, and can be repeatedly sold to a large number of data consumers, producing high revenue of data trading. As data commodities are generated based on the raw data contributed by data providers, data providers also have rights to share a portion of data trading revenue. Thus, in crowd-sensed data markets, the reward for data providers comes from two components: basic reward and bonus reward. The data provider $a_i \in \mathbb{A}$ would receive a basic reward $\bar{\phi}$, a fixed amount of money, if her observation x_i is used to generate data commodities. Based on the market value of data commodities, each data provider $a_i \in \mathbb{A}$ could also obtain a bonus reward ϕ_i, a portion of revenue from data trading. We assume the data vendor would share τ percentage of total revenue with data providers after negotiating with data providers.[1]

Statistical Model: Due to the unreliability of sensing devices and the fragility of data communication links, the mobile data is normally incomplete, imprecise, and erroneous. Furthermore, the sensed data is collected at some selected locations, and cannot fully represent the continuous feature of the physical environment. In addition, the sensed data may be correlated in multiple dimensions, *e.g.*, the temperatures of geographically proximate locations are likely to change synchronously. Such correlation information can be leveraged to provide rich semantic data services. Therefore, the data vendor needs to deploy a statistical model to filter the noise and erroneousness of raw data, infer the data at the locations where no data providers are employed, and describe the correlation of sensed data in multiple dimensions. In such cases, regression techniques can be used to handle the noise in raw data and to perform inference[2]. Although linear regression can draw good inferences, it cannot quantify the uncertainty of these inferences, which is critical to the price determination of data in markets. We use a powerful regression technique *Gaussian Process* [18], [58], which is a generalization of linear regression, and has been widely used as to model numerical sensor data [24], [26], to perform inferences, and to cope with the uncertainty quantification in the process of inferences. Choosing Gaussian

1. The determination for the parameter τ is beyond the scope of this paper, and such process can be modeled as a bargaining game [46] between the data vendor and data providers.

2. We can also use classical data clearing schemes [16], [48] to detect and correct the corrupt and inaccuracy raw data, which would reduce the noise of input data to the statistical model and improve the accuracy of inference.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2900243, IEEE Transactions on Mobile Computing

process as the statistical model for numerical crowd-sensed data also provides several advantages for data trading. For example, we can regard conditional Gaussian distributions as data commodities, and generate different versions of the data commodity by selecting different set of locations to observe. We can also define the accuracy of data commodities as the posterior variance of conditional distribution. We will show the details of these parts in the following discussion.

We associate a random variable $\mathcal{X}_y$ with each location $y \in \Theta$, and a set of random variables $\boldsymbol{\mathcal{X}}_Y$ with a set of locations $Y \subseteq \Theta$, representing the possible data at the corresponding locations. We can specify the Gaussian Process model with a mean function $\boldsymbol{\mu}$, and a symmetric and positive-definite covariance function $\boldsymbol{\Sigma}$. Let $\boldsymbol{\mu}_Y$ and $\boldsymbol{\Sigma}_{YY}$ denote the mean vector and the covariance matrix for a set of random variables $\boldsymbol{\mathcal{X}}_Y \subseteq \boldsymbol{\mathcal{X}}_\Theta$, respectively. In Gaussian Process, the joint distribution over the corresponding set of random variables $\boldsymbol{\mathcal{X}}_Y \subseteq \boldsymbol{\mathcal{X}}_\Theta$ is a multivariate Gaussian distribution, and the probability density function is:

$$f(\mathbf{x}_Y) = \frac{1}{(2\pi)^{|Y|/2}|\boldsymbol{\Sigma}_{YY}|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}_Y-\boldsymbol{\mu}_Y)^T\boldsymbol{\Sigma}_{YY}^{-1}(\mathbf{x}_Y-\boldsymbol{\mu}_Y)},$$

where $\mathbf{x}_Y$ is a vector of possible values of random variables $\boldsymbol{\mathcal{X}}_Y$, $|\boldsymbol{\Sigma}|$ is the determinant of matrix $\boldsymbol{\Sigma}$, and $\boldsymbol{\Sigma}^{-1}$ is the inverse matrix of $\boldsymbol{\Sigma}$. Under Gaussian Process model, we can infer the data at any location $y \subseteq \Theta$ (even there is no sensor deployed at this location) based on the observations $\mathbf{x}_\mathbb{A}$. The resulting distribution $f_{\mathcal{X}_y|\boldsymbol{\mathcal{X}}_\mathbb{A}}(x_y|\mathbf{x}_\mathbb{A})$ is a conditional univariate Gaussian distribution, whose posterior mean $\bar{\mu}_y$ and posterior variance $\bar{\sigma}_y^2$ can be expressed as:

$$\bar{\mu}_y = \mu_y + \boldsymbol{\Sigma}_{y\mathbb{A}}\boldsymbol{\Sigma}_{\mathbb{A}\mathbb{A}}^{-1}(\mathbf{x}_\mathbb{A} - \boldsymbol{\mu}_\mathbb{A}), \quad (1)$$

$$\bar{\sigma}_y^2 = \sigma_y^2 - \boldsymbol{\Sigma}_{y\mathbb{A}}\boldsymbol{\Sigma}_{\mathbb{A}\mathbb{A}}^{-1}\boldsymbol{\Sigma}_{\mathbb{A}y}, \quad (2)$$

In data market, the data vendor obtains revenue by providing data services based on the collected raw data $\mathbf{x}_\mathbb{A}$. The other information, such as the parameters of the statistical model, is common knowledge. Thus, the posterior variance $\bar{\sigma}_y^2$, which is independent on the actual observations $\mathbf{x}_\mathbb{A}$, is publicly known.

Data Commodity: In crowd-sensed data market, we define data commodity for trading as conditional Gaussian distributions $f_{\boldsymbol{\mathcal{X}}_Y|\boldsymbol{\mathcal{X}}_\mathbb{A}}(\mathbf{x}_Y|\mathbf{x}_\mathbb{A})$, which can be considered as a type of data service. In addition to the noise and erroneousness of raw data, the possible privacy leakage [62] and the potential violation of data copyright [13] are other two concerns to directly trade raw data in data markets. We call the distribution $f_{\mathcal{X}_y|\boldsymbol{\mathcal{X}}_\mathbb{A}}(x_y|\mathbf{x}_\mathbb{A})$ of a single random variable $\mathcal{X}_y$ as a *basic* data commodity. Considering that the possible locations of the monitoring region are infinite, the data vendor would select a finite set of random variables at several locations, known as Point of Interests (PoIs), to approximately describe the environmental phenomenon of the whole region Θ. We denote the set of these PoIs by $\mathbb{Y} = \{1, 2, \cdots, l\}$. For notational convenience, we will use $Y \subseteq \mathbb{Y}$ to index the data commodity $f_{\boldsymbol{\mathcal{X}}_Y|\boldsymbol{\mathcal{X}}_\mathbb{A}}(\mathbf{x}_Y|\mathbf{x}_\mathbb{A})$ in the following discussion.

The data vendor assigns a price p_y to each basic data commodity $y \in \mathbb{Y}$. We denote all the basic prices by a vector $\boldsymbol{p} = (p_1, p_2, \cdots, p_l)$. We will discuss the determination of the basic prices in Section 3. As mentioned above, the variance information is public knowledge, so the valuable information of a data commodity is its mean vector. Furthermore, the mean of a data commodity $f_{\boldsymbol{\mathcal{X}}_Y|\boldsymbol{\mathcal{X}}_\mathbb{A}}(\mathbf{x}_Y|\mathbf{x}_\mathbb{A})$ is actually the vector of the means of the contained basic data commodities $f_{\mathcal{X}_y|\boldsymbol{\mathcal{X}}_\mathbb{A}}(x_y|\mathbf{x}_\mathbb{A})$, $y \in Y$. Based on this fact, we set the price of a data commodity $Y \subseteq \mathbb{Y}$ as the sum of the basic prices of the basic data commodities in Y, *i.e.*, $p_Y = \sum_{y\in Y} p_y$.

Data Consumers: The n data consumers, denoted by $\mathbb{B} = \{b_1, b_2, \cdots, b_n\}$, arrive at the marketplace in a certain sequence. Each data consumer b_i issues a query about a data commodity $Y_i \subseteq \mathbb{Y}$, and has a private valuation v_i for the query. For the convenience of analysis, we normalize the valuations into the range $[1, \delta]$. We denote the valuations of all the data consumers by $\mathbf{v} = (v_1, v_2, \cdots, v_n)$. We consider the following types of query in this paper:

- *Single-Data Query:* A data consumer b_i is interested in the (inferential) data at a single location $y_i \in \mathbb{Y}$, *i.e.*, the (posterior) mean $\bar{\mu}_{y_i}$ of the basic data commodity y_i.
- *Multi-Data Query:* A data consumer b_i wants to know the (inferential) data of a certain region $Y_i \subseteq \mathbb{Y}$, *i.e.*, the (posterior) mean vector $\bar{\boldsymbol{\mu}}_{Y_i}$ of the data commodity Y_i. We assume that the maximum dimension of all the queried data commodities is a constant κ, *i.e.*, $\kappa = \max_{b_i\in\mathbb{B}} |Y_i|$.
- *Range Query:* A data consumer b_i asks for the probability that the data at the region $Y_i \subseteq \mathbb{Y}$ belongs to a range $[\underline{\mathbf{a}}_i, \overline{\mathbf{a}}_i]$.

Data Accuracy: We define the accuracy of data commodity $Y \in \mathbb{Y}$ as the average posterior variance of the contained basic data commodities, *i.e.*, $\frac{\sum_{y\in Y}\bar{\sigma}_y^2}{|Y|}$, which has been widely used to measure the performance of statistical inference over sensed data [30], [41]. Such criterion is easy to explain to data consumers, and can be verified by evaluating Equation (2) with the public knowledge of the covariance function of Gaussian model and the locations of data providers[3]. Furthermore, with this accuracy criterion, the data vendor can measure the accuracy of data providers' data by evaluating their location information, resisting their manipulation on data accuracy. This is very important to the reward sharing process, as the reward is related to data provider's contribution to the accuracy improvement during data commodity generation. Due to diverse applications for the purchased data, data consumers may have different accuracy requirements for data commodities. Each data consumer $b_i \in \mathbb{B}$ submits an accuracy threshold ϵ_i for her queried data commodity Y_i. The data commodity Y_i satisfies the accuracy requirement of data consumer b_i if the average posterior variance is less than the threshold ϵ_i, *i.e.*, $\frac{\sum_{y\in Y_i}\bar{\sigma}_y^2}{|Y_i|} \leq \epsilon_i$.

Data Charging: Considering that the data commodity with different accuracy requirements should have different prices, the data vendor offers a discount $d_i \in (0, 1]$ for the data consumer $b_i \in \mathbb{B}$ with low accuracy requirement (Please refer to Section 3 for the determination of the discount factor.). Thus, the charge for the data consumer b_i's query about the data commodity Y_i is $c_i = p_{Y_i} \times d_i$. If data consumer b_i's valuation v_i is higher than c_i, she would

3. We can introduce privacy-preserving and verifiable mechanisms [14] to evaluate the location information, and still protect the privacy of data providers.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2900243, IEEE Transactions on Mobile Computing

TABLE 1
FREQUENTLY USED NOTATIONS

Notation	Remark
$\mathbb{A},\ x_i$	Set of data providers and data observation.
$\bar{\phi},\ \bar{\phi}_i$	Basic reward and bonus reward for data provider i.
$\mathcal{X}_y, \boldsymbol{\mathcal{X}}_Y$	Random variable(s) with location(s) y or Y.
$\mu_y,\ \boldsymbol{\mu}_Y,\ \sigma_y^2,\ \boldsymbol{\Sigma}_{YY}$	Mean or mean vector and variance or covariance matrix of random variable(s) $\mathcal{X}_y$ or $\boldsymbol{\mathcal{X}}_Y$.
$\bar{\mu}_y,\ \bar{\sigma}_y^2$	Posterior mean and posterior variance of random variable $\mathcal{X}_y$.
$f_{\boldsymbol{\mathcal{X}}_Y\|\boldsymbol{\mathcal{X}}_{\mathbb{A}}}(\mathbf{x}_Y\|\mathbf{x}_{\mathbb{A}})$,	Conditional Gaussian distributions.
$\mathbb{Y}$	Set of PoIs.
$\boldsymbol{p}, p_y$	Vector of basic prices, basic price.
$\mathbb{B}, b_i$	Set of data consumers, data consumer.
$\mathbf{v}, v_i$	Vector of data consumers' valuations, valuation.
δ	Upper bound of valuation.
κ	Maximum dimension of data commodities.
ϵ_i, d_i, c_i	Data accuracy requirement, discount factor, charge.
Ψ, C	Profit, revenue.
τ	The portion of revenue for sharing.
$\mathcal{A}_i$	Set of data providers to generate the ith version.
$V(\mathcal{A})$	Variance reduction.
$\Delta_j(\mathcal{A})$	Marginal variance reduction for data provider j.
α, β, γ	Parameters of online mechanism.

purchase the query, and pay the charge; otherwise, she leaves and pays nothing. We use vector $\boldsymbol{c} = (c_1, c_2, \cdots, c_n)$ to denote the charges of all data consumers.

We list the frequently used notations in Table 1.

2.2 Problem Formulation

In this paper, we consider two closely related problems in the mobile crowd-sensed data market: *Profit Maximization* and *Reward Sharing*.

Profit Maximization: The goal of data vendor is to maximize the profit from data trading, which is defined as the difference between the revenue and the data acquisition cost. The total revenue from data trading is the sum of the charges for data customers that purchase data commodities, *i.e.*, $C \triangleq \sum_{b_i \in \mathbb{B}: v_i > c_i} c_i$. The data acquisition cost is the total rewards distributed to incentivize data providers, *i.e.*, $\tau \times C + \bar{\phi} \times M$, where M is the number of observed data. Thus, the profit of the data vendor is $\Psi \triangleq (1-\tau) \times C - \bar{\phi} \times M$. As in previous papers [10], we will use competitive analysis to investigate the performance of online pricing mechanism. We here give the formal definition of $(1+\epsilon)$-competitive online data pricing mechanism.

Definition 1 ($(1+\epsilon)$-Competitive Data Pricing Mechanism)**.** *A data pricing mechanism is $(1+\epsilon)$-competitive if the ratio between the profit of the optimal offline mechanism and the profit of the online mechanism is $(1+\epsilon)$.*

The optimal offline mechanism selects a single fixed price for each (basic) data commodity with the posterior knowledge of the valuations of all data consumers. The optimal revenue for each data commodity is given by $C^* = p^* \times n_{p^*}$, where p^* is the optimal price, and n_p is the number of data consumers with values larger than p. This optimal offline counterpart is widely used in performance analysis of online learning algorithms [10], [12].

In contrast to the goal of the data vendor, the selfish data consumers always tend to purchase their desired query results with lower charges. For example, the data consumers can indirectly infer the answer of an expensive query by buying a set of cheaper queries. The data pricing mechanism should be robust enough to resist such arbitrage behaviours. We define an arbitrage-free data pricing mechanism as follows.

Definition 2 (Arbitrage-free Data Pricing Mechanism)**.** *Whenever a query q can be entirely answered by a query bundle $\{q_1, q_2, \cdots, q_k\}$, an arbitrage-free data pricing mechanism must satisfy that $c(q) \leq \sum_{i=1}^{k} c(q_k)$, where $c(q)$ denotes the charge for the query q.*

We now formally present the problem of profit maximization in mobile crowd-sensed data markets: the data vendor dynamically selects data providers to generate qualified data commodities, and determines the charge **c** (by calculating the basic prices **p** and discount factor **d**) for data consumers $\mathbb{B}$, such that the resulting data pricing mechanism achieves a good approximation ratio in terms of profit maximization and the property of arbitrage-freeness.

Reward Sharing: In data markets, the data commodities are generated by aggregating the collected raw data from data providers. As data can be copied with a negligible marginal cost, the data can extract high revenue from the market by repeatedly selling to a large number of data consumers. Thus, in addition to the basic reward, the data vendor should also share a portion of revenue with data providers to further incentivize them to contribute high quality data. To determine the basic reward, the data vendor uses the criterion of the number of data providers, as she wants to minimize the total basic reward. Considering that the data providers might submit data with heterogeneous qualities and then have different contribution levels to generate data commodities and the revenue of data trading, we should use the criterion of contribution levels for reward sharing, guaranteeing the fair axioms. In data markets, another important and critical issue for the data vendor is the incentive design for data providers: how to generate the qualified data commodities with the minimum basic reward, and fairly distribute the total bonus reward $\tau \times C$ among the data providers $\mathbb{A}$, given their heterogeneous contribution levels?

3 ONLINE DATA PRICING

In this section, we propose ARETE-PR, which is a version-based online posted-pricing mechanism for mobile crowd-sensed data market. ARETE-PR consists of two components: a versioning mechanism and an online pricing mechanism. The versioning mechanism efficiently selects a set of data providers to generate a qualified version of data commodity for data consumer, minimizing the data acquisition cost. The online pricing mechanism dynamically determines the price for each basic data commodity with the goal of revenue maximization. The versioning mechanism and pricing mechanism jointly maximize the profit of data trading.

We begin with a simple but classical setting, in which data consumers only issue single-data queries. In this case, we can consider the price determination for each of basic data commodities independently, and discuss the design of ARETE-PR for one selected basic data commodity. We further extend ARETE-PR to adapt to the other types of query in Section 4.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2900243, IEEE Transactions on Mobile Computing

3.1 Design Rationale

Under the cost structure of information (a fixed cost of production but negligible marginal costs of duplication), the price of data should be linked to the valuations of data consumers rather than data production costs. Furthermore, data consumers have diverse accuracy requirements over data commodities. Considering the new cost structure of data and the diverse accuracy requirements of data consumers, we propose a valuation-based data pricing mechanism coupled with a versioning technique for mobile crowd-sensed data trading. Specifically, we partition a data commodity into multiple versions with different accuracies and prices, and provide the qualified version and an appropriate price for each arrived data consumer. The challenging problem here is how to efficiently select data providers to generate the qualified version with a minimum data acquisition cost. We also need to determine the discount factor for each version. Observing that the accuracy, *i.e.*, the variance reduction is a submodular function with respective to the set of selected data providers, we can formulate the process of versioning as a problem of submodular covering, and propose a greedy selection algorithm with performance guarantee. Furthermore, we set the price of each version as the basic price of the full version multiplying a discounting factor, which is proportional to the "distance" of the corresponding version to the full version. We modify the concept of *relative entropy*, a nature metric of distribution difference, to measure this distance.

Yet, another critical problem of designing online data pricing mechanism is the determination of basic prices. The most challenging part is that both valuations and arrival sequence of data consumers are unknown to the data vendor. The data vendor needs an online mechanism to dynamically learn the valuation information of data consumers, and sets a near-optimal basic price to maximize the revenue. We determine the basic prices by making a trade-off between "exploitation" and "exploration" to data consumers' valuations. On one hand, if the data vendor exclusively chooses the candidate price that she believes is the best (exploitation), she may fail to discover one of the other candidate prices that actually has a higher revenue in the long term. On the other hand, if she spends too much time trying out all the candidate prices to learn the valuations of data consumers (exploration), she may fail to choose the price that is good enough to obtain a high total revenue in time. Therefore, for each of the arrived data consumers, we select a price following a mixed distribution, which is a combination of an exploitation distribution and an exploration distribution. Based on the response of the data consumer to the chosen price, we update the mixed distribution in a principle way, to guide the selection of candidate prices in the following transactions.

3.2 Versioning

In ARETE-PR, we regard the conditional Gaussian distribution $f(x_y|\mathbf{x}_{\mathcal{A}_i})$ generated by the observations $\mathbf{x}_{\mathcal{A}_i}$ from the selected data providers $\mathcal{A}_i \subseteq \mathbb{A}$ as a version of the basic data commodity $y \in \mathbb{Y}$,[4] which satisfies the accuracy requirement of the arrived data consumer b_i if $\bar{\sigma}_y^2 \leq \epsilon_i$. Here, we use $\mathcal{A}_i$ to denote the data providers recruited to generate the version for data consumer b_i. Using the posterior covariance in Equation (2), we can further express this constraint as

$$\mathbf{\Sigma}_{y\mathcal{A}_i}\mathbf{\Sigma}_{\mathcal{A}_i\mathcal{A}_i}^{-1}\mathbf{\Sigma}_{\mathcal{A}_i y} \geq \boldsymbol{\sigma}_y^2 - \epsilon_i. \tag{3}$$

We call the left hand side of the above inequality as *variance reduction* $V(\mathcal{A}_i)$ due to observing data from the selected data providers $\mathcal{A}_i$, *i.e.*, $V(\mathcal{A}_i) \triangleq \mathbf{\Sigma}_{y\mathcal{A}_i}\mathbf{\Sigma}_{\mathcal{A}_i\mathcal{A}_i}^{-1}\mathbf{\Sigma}_{\mathcal{A}_i y}$. We assume the original variance $\boldsymbol{\sigma}_y^2$ is a constant variance. As the data vendor has to pay a fixed basic reward for each selected data provider, she always wants to recruit less data providers to achieve the accuracy requirements of data consumers, minimizing the total basic reward. We now can formulate the problem of basic reward minimization for the version generation as follows

Problem: *Basic Reward Minimization*
Objective: Minimize $\bar{\phi} \times |\mathcal{A}_i|$
Subject to:

$$V(\mathcal{A}_i) \geq \boldsymbol{\sigma}_y^2 - \epsilon_i, \quad \mathcal{A}_i \subseteq \mathbb{A}. \tag{4}$$

It can be shown that the variance reduction function $V(\mathcal{A})$ is a monotonic submodular function with respective to the set of selected data providers $\mathcal{A}$ [20], [41]. In addition, the objective function is modular. Thus, the problem of basic reward minimization is a submodular covering problem and is NP-hard [38], [59]. Greedy algorithm has been recognized as an efficient approximation approach for submodular optimization [41], [59]. We present a greedy algorithm for the selection of data providers, and analyze the approximation ratio for such greedy algorithm.

We now present the principle of greedy versioning mechanism in Algorithm 1 step by step. The versioning algorithm greedily adds the most "informative" data provider following a sequence, until the current posterior variance satisfies the accuracy requirement of the data consumer. Formally, our goal is to select the next data provider a_j that maximizes the marginal variance reduction $\Delta_j(A) \triangleq V(A \cup \{a_j\}) - V(A)$. where A is the set of currently selected data providers. We break the tie following a random rule (Lines 2 to 6). If the new posterior variance $\bar{\sigma}_y^2$ is less than the accuracy threshold ϵ_i, we set $\mathcal{A}_i$ as the current data provider set A (Line 7). From the result in [59], we have the following performance guarantee for the greedy versioning algorithm.

Theorem 1. *For the problem of basic reward minimization, the greedy versioning algorithm can achieve the approximation ratio of* $1 + ln(\Delta_{max}/\Delta_{min})$, *where* Δ_{max} *and* Δ_{min} *are the maximum marginal variance reduction and minimum marginal variance reduction of only selecting one single data provider, respectively*, i.e., $\Delta_{max} \triangleq \max_{a_i \in \mathbb{A}} \Delta_i(\varnothing)$ *and* $\Delta_{min} \triangleq \min_{a_i \in \mathbb{A}} \Delta_i(\varnothing)$.

The remaining issue is to determine discount factor for the generated version. We set the discount factor of a version proportional to its distance to the full version, *i.e.*,

4. For mobile crowd-sensed data, there are many possible versioning strategies, *e.g.*, aggregating different amounts of raw data to generate versions, which is adopted in this paper, or artificially adding the noises of different levels into an accurate data commodity.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2900243, IEEE Transactions on Mobile Computing

Algorithm 1: Versioning Mechanism

Input: The ith data consumer b_i; The queried data commodity y; The accuracy requirement ϵ_i; A scale parameter λ.

Output: A set of selected data providers $\mathcal{A}_i$; A discount factor d_i.

1 $A \leftarrow \varnothing$;
2 **while** $V(A) < \sigma_y^2 - \epsilon_i$ **do**
3 **foreach** $a_j \in \mathbb{A} \backslash A$ **do**
4 $\Delta_j(A) \leftarrow V(A \cup \{a_j\}) - V(A)$;
5 $a^* \leftarrow \arg\max_{a_j \in \mathbb{A}\backslash A} \Delta_j(A)$;
6 $A \leftarrow A \bigcup \{a^*\}$;
7 $\mathcal{A}_i \leftarrow A$;
8 $\bar{\sigma}^2_{y|\mathbb{A}} \leftarrow \sigma_y^2 - \Sigma_{y\mathbb{A}} \Sigma_{\mathbb{A}\mathbb{A}}^{-1} \Sigma_{\mathbb{A}y}$;
9 $\bar{\sigma}^2_{y|\mathcal{A}_t} \leftarrow \sigma_y^2 - \Sigma_{y\mathcal{A}_i} \Sigma_{\mathcal{A}_i\mathcal{A}_i}^{-1} \Sigma_{\mathcal{A}_i y}$;
10 $f_1(x) = f_{\mathcal{X}_y|\boldsymbol{\mathcal{X}}_{\mathbb{A}}}(x_y|\mathbf{x}_{\mathbb{A}})$; $f_2(x) = f_{\mathcal{X}_y|\boldsymbol{\mathcal{X}}_{\mathcal{A}_i}}(x_y|\mathbf{x}_{\mathcal{A}_i})$;
11 $\widehat{D}(f_1||f_2) \leftarrow \frac{1}{2}\left(\log \frac{\bar{\sigma}^2_{y|\mathcal{A}_i}}{\bar{\sigma}^2_{y|\mathbb{A}}} + \frac{\bar{\sigma}^2_{y|\mathbb{A}}}{\bar{\sigma}^2_{y|\mathcal{A}_i}} - 1\right)$;
12 $d_i \leftarrow e^{-\lambda \widehat{D}(f_1||f_2)}$;
13 **return** $\mathcal{A}_i$, d_i;

the distribution $f_{\mathcal{X}_y|\boldsymbol{\mathcal{X}}_{\mathbb{A}}}(x_y|\mathbf{x}_{\mathbb{A}})$, and normalize the discount factor for the full version as 1. The concept of *relative entropy*, or *Kullback-Leibler distance*, is a measure of the distance between two distributions [17]. Specifically, the relative entropy between the full version $f_1(x) = f_{\mathcal{X}_y|\boldsymbol{\mathcal{X}}_{\mathbb{A}}}(x_y|\mathbf{x}_{\mathbb{A}})$ and the generated version $f_2(x) = f_{\mathcal{X}_y|\boldsymbol{\mathcal{X}}_{\mathcal{A}_i}}(x_y|\mathbf{x}_{\mathcal{A}_i})$ is

$$D(f_1||f_2) \triangleq \int f_1(x) \log \frac{f_1(x)}{f_2(x)}\, dx = \frac{1}{2}\left(\log \frac{\sigma_2^2}{\sigma_1^2} + \frac{\sigma_1^2 + (\mu_1 - \mu_2)^2}{\sigma_2^2} - 1\right). \quad (5)$$

The relative entropy is nonnegative and is equal to zero if and only if $f_1 = f_2$. Intuitively, a version with a lower accuracy should be "farther" from the full version. However, the distance calculated by Equation (3) may not reflect such property, because the relative entropy depends on both the mean and variance. The accuracy of a data commodity only rests on its variance. Inspired by this, we modify the relative entropy by ignoring the mean terms, and regard it as the distance between two versions

$$\widehat{D}(f_1||f_2) = \frac{1}{2}\left(\log \frac{\sigma_2^2}{\sigma_1^2} + \frac{\sigma_1^2}{\sigma_2^2} - 1\right). \quad (6)$$

Considering that discount factor should lie in the range $[0, 1]$, we define the discount factor for the version as:

$$d_i \triangleq e^{-\lambda \widehat{D}(f_1||f_2)}, \quad (7)$$

where λ is a scale parameter.

We give the detailed steps to calculate the discount factor for each version in Algorithm 1. We calculate the variance $\bar{\sigma}^2_{y|\mathbb{A}}$ of the full version $f(y|\mathcal{A}_i)$ in Line 8. For the generated version, we calculate its variance $\bar{\sigma}^2_{y|\mathcal{A}_i}$ in Line 9, and the corresponding distance and discount factor according to Equation (6) and Equation (7), respectively (Lines 11 to 12).

We use a simple example to illustrate the ideas of the versioning mechanism in Figure 2. Suppose there are three

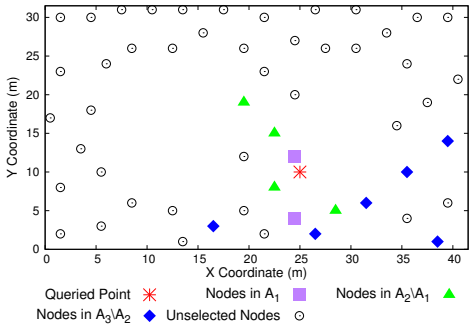


Fig. 2. Versioning results of the data commodity at location (25, 10).

data consumers issuing data queries at the location $(25, 10)$. Their accuracy requirements are $\epsilon_1 = 38.94$, $\epsilon_2 = 14.32$ and $\epsilon_3 = 9.60$, respectively. We show the set of data providers selected by the versioning mechanism in Figure 2. From this result, we observe that the data providers, neighboring the queried point, have a high probability to be selected, because they are more informative to the queried point. At the same time, the versioning algorithm might ignore some data providers, although they are in the vicinity of the queried point, because their marginal entropy is relatively small given the currently selected data providers. We set the scale parameter λ in Equation (7) as 2.77 to adjust the discount factors to appropriate values. Under this setting, we calculate the corresponding discount factors for the three versions as $\mathbf{d} = (0.36, 0.85, 1)$.

3.3 Online Pricing

We now describe the detailed principle of online pricing mechanism in Algorithm 2. For each arrived data consumer, we select the basic price from a vector of candidate discrete prices $\hat{\boldsymbol{p}} = (\hat{p}_1, \hat{p}_2, \cdots, \hat{p}_K)$, where $\hat{p}_k = (1+\beta)^{k-1}$ for any $1 \le k \le K$ and $\beta > 0$. Since the upper bound of valuation is δ, we have $K = \lfloor \log_{1+\beta} \delta \rfloor + 1$. Let $c_i(k)$ be the revenue attained by setting price $\hat{p}_k$ for the ith data consumer b_i. We initially set $c_0(k)$ to be zero for any $1 \le k \le K$. Given a parameter $\alpha \in (0, 1]$, we define a weight $w_i(k)$ for the price $\hat{p}_k$ in the ith transaction as

$$w_i(k) \triangleq (1+\alpha)^{\sum_{j=1}^{i} c_j(k)}, \quad (8)$$

which is an exponential weight function, denoting the performances of the candidate prices in the previous transactions. The candidate price with a large weight should have a high probability to be chosen as a basic price in the following transactions. We denote the weight vector for all candidate prices in the ith transaction by $\mathbf{w}_i = (w_i(1), w_i(2), \cdots, w_i(K))$, and initially set $\mathbf{w}_0$ to be $\mathbf{1}$.

For the ith arrived data consumer $b_i \in \mathbb{B}$, Algorithm 2 selects a candidate price $\hat{p}_k$ following the distribution $\hat{f}_i(k)$, which is a combination of an exploitation distribution and an exploration distribution (Line 2). On one hand, we try to exploit the currently expected best price to gain a high revenue, and define the exploitation distribution as

$$f_i(k) \triangleq \frac{w_{i-1}(k)}{\sum_{j=1}^{K} w_{i-1}(j)}, \quad \forall\, 1 \le k \le K. \quad (9)$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2900243, IEEE Transactions on Mobile Computing

Algorithm 2: Online Pricing Mechanism

Input: Reals: $\alpha \in (0,1], \beta > 0, \gamma \in (0,1]$; The ith data consumer b_i; A vector of discount factors $\mathbf{d}$; The highest valuation δ; The number of candidate prices K; A vector of candidate prices $\hat{\boldsymbol{p}}$; A weight vector $\mathbf{w}_{i-1}$.

Output: The charge c_i for data consumer b_i.

1 $\mathbf{c}_i \leftarrow \mathbf{0}$;

2 Select the candidate price as $\hat{p}_k$ following the probability: $\hat{f}_i(k) \leftarrow (1-\gamma) f_i(k) + \gamma g(k)$, where $f_i(k) = \frac{w_{i-1}(k)}{\sum_{j=1}^{K} w_{i-1}(j)}$ and $g(k) = \frac{\Delta}{(1+\beta)^{K-k}},\ \Delta = \frac{1-\frac{1}{1+\beta}}{1-\left(\frac{1}{1+\beta}\right)^K}$;

3 Suppose the selected price is $\hat{p}_{k_i}$;

4 Choose the lowest version that satisfies the accuracy requirement η_i of data consumer b_i, and set her discount factor $\hat{d}_i \leftarrow d_{t_i}$;

5 $c_i \leftarrow \hat{p}_{k_i} \times \hat{d}_i$;

6 **if** *Data consumer b_i accepts the charge c_i* **then**

7 $\quad c_i(k_i) \leftarrow c_i$;

8 **else**

9 $\quad c_i(k_i) \leftarrow 0$;

10 **foreach** $k = 1$ *to* K **do**

11 $\quad$ **if** $k = k_i$ **then**

12 $\quad\quad \hat{c}_i(k) \leftarrow \frac{\gamma\Delta}{\delta}\frac{c_i(k)}{\hat{f}_i(k)}$; $w_i(k) \leftarrow w_{i-1}(k) \times (1+\alpha)^{\hat{c}_i(k)}$;

13 $\quad$ **else**

14 $\quad\quad \hat{c}_i(k) \leftarrow 0$; $\quad w_i(k) \leftarrow w_{i-1}(k)$;

15 **return** c_i;

On the other hand, since some candidate prices may obtain a low revenue at first, but receive a high revenue later, we also apply an exploration distribution to find the ultimate optimal price in long terms. Thus, we further assign each candidate price $\hat{p}_k$ an exploration probability distribution. A classical exploration distribution is uniform distribution, which assigns each of the candidate prices the same probability [10]. However, considering that different candidate prices can produce different amount of revenue, we adopt a geometric distribution as the exploitation distribution, *i.e.*,

$$g(k) \triangleq \frac{1}{1-\left(\frac{1}{1+\beta}\right)^K}\frac{1-\frac{1}{1+\beta}}{(1+\beta)^{K-k}},\ \forall\, 1 \le k \le K. \quad (10)$$

To simplify notation, we set $\Delta = \frac{1-\frac{1}{1+\beta}}{1-\left(\frac{1}{1+\beta}\right)^K}$. Since the kth candidate price is $\hat{p}_k = (1+\beta)^{k-1}$, such exploration distribution ensures that $\hat{p}_k/g(k) = O\left((1+\beta)^{k-1}(1+\beta)^{K-k}\right) = O(\delta)$, which is a useful property for the competitive ratio analysis. Let $\hat{p}_{k_i}$ denote the selected price for data consumer b_i following the combined distribution $\hat{f}_i(k)$ (Line 3).

We efficiently select the smallest set of data providers to generate the lowest version that satisfies the required accuracy requirement of the data consumer b_i.[5] The discount factor $\hat{d}_i$ to data consumer b_i is the corresponding discount factor d_{t_i} for version t_i returned by Algorithm 1 (Line 4). The charge for data consumer b_i then is $c_i = \hat{p}_{k_i} \times \hat{d}_i$ (Line 5).

According to the data consumer's purchasing decision, we receive a revenue $c_i(k_i) \in \{0, c_i\}$ of the chosen price $\hat{p}_{k_i}$. In the posted pricing setting, we cannot observe the revenue generated by the other candidate prices. So we set $c_i(k) = 0$ for any $k \neq k_i$ (Lines 6 to 9). Based on this revenue vector $\mathbf{c}_i = (c_i(1), c_i(2), \cdots, c_i(K))$, we generate a virtual revenue vector $\hat{\boldsymbol{c}}_i = (\hat{c}_i(1), \hat{c}_i(2), \cdots, \hat{c}_i(K))$, and use it to update the weights of candidate prices. We calculate this virtual revenue vector by distinguishing the two cases:

▷ For the chosen price $\hat{p}_{k_i}$, we set the virtual revenue $\hat{c}_i(k_i)$ to be $\frac{\gamma\Delta}{\delta}\frac{c_i(k)}{\hat{f}_i(k)}$.

▷ For the other prices $\hat{p}_k,\ k \neq k_i$, we set $\hat{c}_i(k)$ to be zero.

We update the weight vector $\mathbf{w}_i$ using Equation (8) with virtual revenue vector $\hat{\boldsymbol{c}}_i$ (Lines 10 to 14). We have the following two properties for this virtual revenue vector $\hat{\boldsymbol{c}}_i$, which is heavily used in the analysis of competitive ratio in next section.

► The expected virtual revenue (with respective to the selection distribution $\hat{f}_i(k)$) for any candidate price $\hat{p}_k$ is proportional to the actual revenue of the price $c_i(k)$, *i.e.*,

$$\begin{aligned}\mathbf{E}[\hat{c}_i(k)] &= \mathbf{E}\left[\hat{c}_i(k) | (\hat{p}_{k_1}, \hat{p}_{k_2}, \cdots, \hat{p}_{k_{i-1}})\right] \\ &= \mathbf{E}\left[\hat{f}_i(k) \times \frac{\gamma\Delta}{\delta}\frac{c_i(k)}{\hat{f}_i(k)} + (1-\hat{f}_i(k)) \times 0\right] \\ &= \frac{\gamma\Delta}{\delta} c_i(k).\end{aligned}$$

► The virtual revenue $\hat{c}_i(k)$ is in the range $[0,1]$.

$$\begin{aligned}\hat{c}_i(k) &= \frac{\gamma\Delta}{\delta}\frac{c_i(k)}{\hat{f}_i(k)} \le \frac{\gamma\Delta}{\delta}\frac{c_i(k) \times (1+\beta)^{K-k}}{\gamma\Delta} \\ &= \frac{(1+\beta)^{k-1} \times (1+\beta)^{K-k}}{\delta} \le 1.\end{aligned}$$

We remark that the data vendor can dynamically tune the parameters α, β, γ in Algorithm 2 to adapt to different market settings. Specifically, the parameter α represents the weights of candidate prices in exploitation process (*i.e.*, a larger α indicates that we heavily exploit the candidate prices with good performance in previous transactions.). The parameter γ denotes the trade-off between the exploitation and exploration (*i.e.*, a smaller γ represents a higher degree of exploitation.). For example, the data vendor can set a large α and a small γ to actively exploit the collected valuation knowledge, when the data providers' valuations follow a normal distribution. In contrast, when the data providers' valuations come from a uniform distribution, the data vendor can set a low α and a high γ to achieve good performance. The parameter β reflects the trade-off between revenue maximization and computational complexity, *i.e.*, a larger β, implying more candidate prices to choose, can extract a larger revenue but incurs a higher computational overhead. We design experiments to evaluate the effects of these parameters in Section 6.

We finally illustrate this online pricing algorithm by an example. For simplicity, suppose we only provide the full version of the data commodity, and the parameters are $\alpha = 1$, $\beta = 1$ and $\gamma = 2/3$. We set the upper bound

5. Although the data vendor can choose high versions for data consumers to extract much revenue, this would incur market anarchy: data consumers would strategically report low accuracy requirement to seek less payments. The policy of selecting the lowest version enforces data consumers to truthfully report their required data accuracy requirement.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2900243, IEEE Transactions on Mobile Computing

of valuation δ to be 2. According to these parameters, we will only have $K = 2$ candidate prices with values 0 and 1 respectively. Recall that both prices have weight 1 initially. Therefore, they both have probability $1/2$ in the first exploration distribution f_1. Furthermore, we can calculate by Equation (10) that the first exploitation distribution is $g_1(1) = 1/3$ and $g_1(2) = 2/3$. With $\gamma = 2/3$, our final distribution $\hat{f}_i$ will be $\frac{2}{3}f_i + \frac{1}{3}g_i$, which is $\hat{f}_i(1) = 4/9$ and $\hat{f}_i(2) = 5/9$. Now suppose for the first consumer, we sampled $k_1 = 1$ from this distribution. In this case, the charge for the consumer will be $p_1 = 2^0 = 1$. Assume that the consumer accepts the charge. In this case, the revenues for these two prices are $c_1(1) = 1$ and $c_1(2) = 0$. For p_1, we will update $w_2(1) = 2^{0.4} = 1.3$; but for p_2, $w_2(2)$ will still remain to be 1. As a result, the exploration distribution f_2 will be biased towards 1 in the second round, i.e., we will prefer choosing p_1 for the second consumer.

3.4 Analysis

We analyze the competitive ratio of ARETE-PR in this subsection. We use Ψ^* and $\hat{\Psi}$ to denote the optimal profit and approximate profit achieved by ARETE-PR, respectively. Similarly, C^* and $\widehat{C}$ denote the optimal revenue and approximate revenue, respectively. We have the similar meaning for notations M^* and $\widehat{M}$. According to Theorem 1, we have the following performance guarantee for the greedy versioning algorithm

$$\frac{M^*}{\widehat{M}} \geq 1 + \ln \frac{\Delta_{max}}{\Delta_{min}}. \tag{11}$$

We now analyze the competitive ratio of the online pricing mechanism. In the online pricing mechanism, we only consider a vector of discrete candidate prices $\hat{\boldsymbol{p}}$, while ignoring the other possible values in $[1, \delta]$. We show that the attained revenue does not lose much under this restriction.

Lemma 1. *The online pricing mechanism loses a $(1+\beta)$ factor in rounding down the optimal price to one of the prices from $\hat{\boldsymbol{p}}$.*

Proof. Let n_p denote the number of consumers whose valuations are greater than p, *i.e.*, $n_p = |\{b_i \in \mathbb{B} | v_i \geq p\}|$. The revenue of the optimal fixed price p^* is $C^* = p^* \times n_{p^*}$. For the optimal price p^*, there exists some index $k \in [1, K]$ such that $(1+\beta)^{k-1} \leq p^* \leq (1+\beta)^k$. Let C^*_β represent the revenue of the optimal fixed price mechanism, where the candidate prices are restricted in the the discrete price vector $\hat{\boldsymbol{p}}$. We can have:

$$\begin{aligned} C^*_\beta &\geq (1+\beta)^{k-1} \times n_{(1+\beta)^{k-1}} \geq (1+\beta)^{k-1} \times n_{p^*} \\ &\geq \frac{p^*}{(1+\beta)} \times n_{p^*} = \frac{1}{(1+\beta)} \times C^*. \end{aligned}$$

The second inequality comes from the fact that decreasing the fixed price from p^* to $(1+\beta)^{k-1}$ does not reduce the number of sales to data consumers. □

We then show another useful lemma for the competitive ratio analysis.

Lemma 2. *For any parameter $\alpha > 0$, any sequence of virtual revenue vectors $\hat{\boldsymbol{c}}_1, \hat{\boldsymbol{c}}_2, \cdots, \hat{\boldsymbol{c}}_n$, and the exploitation distribution vectors $\boldsymbol{f}_i = (f_i(1), f_i(2), \cdots, f_i(K))$, we have:*

$$\sum_{i=1}^{n} \boldsymbol{f}_i \cdot \hat{\boldsymbol{c}}_i \geq \frac{\sum_{i=1}^{n} \hat{c}_i(k) \log(1+\alpha) - \log K}{\alpha}, \quad \forall 1 \leq k \leq K.$$

Proof. Let $W_i = \sum_{k=1}^{K} w_i(k)$ for any $1 \leq i \leq n$. Since the virtual revenue $\hat{c}_i(k)$ is in the range $[0, 1]$, we can get the following equations.

$$\begin{aligned} \frac{W_i}{W_{i-1}} &= \sum_{k=1}^{K} \frac{w_{i-1}(k)(1+\alpha)^{\hat{c}_i(k)}}{W_{i-1}} \leq \sum_{k=1}^{K} \frac{w_{i-1}(k)(1+\alpha\hat{c}_i(k))}{W_{i-1}} \\ &= 1 + \alpha \frac{\sum_{k=1}^{K} w_{i-1}(k)\hat{c}_i(k)}{W_{i-1}}, \end{aligned}$$

where for the inequality we used the fact that for $x \in [0, 1]$, $(1+\alpha)^x \leq 1 + \alpha x$. Thus,

$$\begin{aligned} &\log \frac{W_n}{W_0} = \sum_{i=1}^{n} \log \frac{W_i}{W_{i-1}} \leq \sum_{i=1}^{n} \left(1 + \alpha \frac{\sum_{k=1}^{K} w_{i-1}(k)\hat{c}_i(k)}{W_{i-1}}\right) \\ &\leq \sum_{i=1}^{n} \alpha \frac{\sum_{k=1}^{K} w_{i-1}(k)\hat{c}_i(k)}{W_{i-1}} = \alpha \sum_{i=1}^{n} \sum_{k=1}^{K} f_i(k)\hat{c}_i(k) \\ &= \alpha \mathbf{f}_i \cdot \hat{\mathbf{c}}_i. \end{aligned} \tag{12}$$

Since $W_n \geq w_n(k) = (1+\alpha)^{\sum_{i=1}^{n} \hat{c}_i(k)}$ for any $1 \leq k \leq K$, and $W_0 = K$, we have

$$\log \frac{W_n}{W_0} \geq \sum_{i=1}^{n} \hat{c}_i(k) \log(1+\alpha) - \log K. \tag{13}$$

Combining Equations (12) and (13), we get

$$\mathbf{f}_i \cdot \hat{\mathbf{c}}_i \geq \frac{\sum_{i=1}^{n} \hat{c}_i(k) \log(1+\alpha) - \log K}{\alpha}.$$

We have completed the proof. □

By Lemma 1, Lemma 2 and an appropriate choice of parameters α, β and γ, we can obtain the following competitive ratio for the online pricing mechanism.

Theorem 2. *Given a real value ϵ, there exists a constant θ, such that for any valuation sequences $\boldsymbol{v}$ with optimal revenue $C^* \geq \theta\delta \log\log\delta$, the online pricing mechanism is $(1+\epsilon)$-competitive.*

Proof. Using Lemma 2 and the properties of the online pricing mechanism, we show the lower bound of revenue $\sum_{i=1}^{n} c_i(k_i)$ for any selected basic price sequence $\hat{\boldsymbol{p}} = (\hat{p}_{k_1}, \hat{p}_{k_2}, \cdots, \hat{p}_{k_n})$.

$$\begin{aligned} &\sum_{i=1}^{n} c_i(k_i) = \frac{\delta}{\gamma\Delta} \sum_{i=1}^{n} \hat{f}_i(k_i)\hat{c}_i(k_i) \\ &= \frac{\delta}{\gamma\Delta} \sum_{i=1}^{n} \left[(1-\gamma) f_i(k_i)\hat{c}_i(k_i) + \gamma \frac{\Delta}{(1+\beta)^{K-k_i+1}} \hat{c}_i(k_i)\right] \\ &\geq \frac{(1-\gamma)\delta}{\gamma\Delta} \sum_{i=1}^{n} f_i(k_i)\hat{c}_i(k_i) = \frac{(1-\gamma)\delta}{\gamma\Delta} \sum_{i=1}^{n} \mathbf{f}_i \cdot \hat{\mathbf{c}}_i \\ &\geq \frac{(1-\gamma)\delta}{\gamma\Delta\alpha} \left(\sum_{i=1}^{n} \hat{c}_i(k) \log(1+\alpha) - \log K\right). \end{aligned}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2900243, IEEE Transactions on Mobile Computing

We next take the expectation of both sides of the above equation with respect to distribution $\hat{\boldsymbol{p}}$. Having $\mathbf{E}[\hat{c}_i(k)] = \frac{\gamma\Delta}{\delta} c_i(k)$ for each $\hat{c}_i(k)$, we can get:

$$\begin{aligned}
&\mathbf{E}\left[\sum_{i=1}^{n} c_i(k_i)\right] \\
\geq\ & \frac{(1-\gamma)\delta}{\gamma\Delta\alpha}\left[\frac{\gamma\Delta}{\delta} \times \sum_{i=1}^{n} c_i(k)\log(1+\alpha) - \log K\right] \\
=\ & \frac{(1-\gamma)\log(1+\alpha)}{\alpha}\sum_{i=1}^{n} c_i(k) - \frac{(1-\gamma)\delta\log K}{\gamma\Delta\alpha} \\
\geq\ & (1-\gamma-\frac{\alpha}{2})C^*_\beta - \frac{\delta\log\log\delta}{\gamma\Delta\alpha} \\
\geq\ & \frac{(1-\gamma-\frac{\alpha}{2})}{(1+\beta)}C^* - \frac{\delta\log\log\delta}{\gamma\Delta\alpha}.
\end{aligned}$$

In the third equality, we select the optimal fixed price from $\hat{\boldsymbol{p}}$, and thus $\max_k\{\sum_{i=1}^{n} c_i(k)\} = C^*_\beta$. The third equality follows from that $\log(1+\alpha) \geq \alpha - \frac{\alpha^2}{2}$ for any $\alpha > 0$. By Lemma 1, the last inequality holds. By choosing appropriate parameters α, β and γ, we prove the theorem. □

We have proven that the online pricing mechanism achieves a constant competitive ratio when the optimal revenue is larger than $O(\delta\log\log\delta)$. The following theorem shows that any online pricing algorithm that achieves a constant ratio, must have an additive constant term $\Omega(\delta)$. Designing an online pricing algorithm with a tight lower bound is our future work.

Theorem 3. *There is no constant-competitive online pricing algorithm for all valuation sequences with $C^* \geq o(\delta)$.*

Proof. We can state the theorem in another way: suppose APX is an online algorithm with a constant competitive ratio c, *i.e.*, for all valuation sequence $\mathbf{v}$, $APX(\mathbf{v}) \geq C^*(\mathbf{v})/c - f(\delta)$. Then, we must have $f(\delta) = \Omega(\delta)$. This statement directly implies the claim we make in the theorem, and we now prove that $f(\delta) \geq \delta/(\eta\eta_1)$, where $\eta = 2c$ and $\eta_1 = 2\eta^{\eta-1}$.

We assume that the valuation sequence contains only one valuation. Let $Pr[a,b]$ denote the probability that mechanism APX sets the sales price in the range $[a,b]$. We prove the result by distinguishing two cases.

• Suppose it is the case that $Pr[1,\delta/\eta_1] \leq 1/\eta$. Then, if the valuation is δ/η_1, the online algorithm's expected revenue is at most $APX(\mathbf{v}) = \delta/(\eta_1\eta)$ while the optimal result is $C^*(\mathbf{v}) = \delta/\eta_1$. Therefore, we have: $f(\delta) \geq C^*(\mathbf{v})/c - APX(\mathbf{v}) \geq \delta/(\eta_1 c) - \delta/(\eta_1\eta) = \delta/(\eta_1\eta)$.

• In the case that $Pr[1,\delta/\eta_1] > 1/\eta$, we define the series L_t as follows, $L_0 = 0$ and $L_{t+1} = \delta/\eta_1 + L_t$. We can get $L_{t+1} = \delta/\eta_1 + \delta\eta/\eta_1 + \cdots + \delta\eta^t/\eta_1$. By definition of η and η_1, we have $L_k \leq \delta$. Combining that $Pr[0,\delta/\eta_1] > 1/\eta$, there must exist some interval $(L_t, L_{t+1}] \subseteq [1,\delta]$ such that $Pr(L_t, L_{t+1}] \leq 1/\eta$. Suppose the valuation is L_{t+1}. In this case, the online algorithm's expected revenue is at most $APX(\mathbf{v}) = L_t + L_{t+1}/\eta$, while the optimal result is $C^*(\mathbf{v}) = L_{t+1}$. Therefore, we have $f(\delta) \geq C^*(\mathbf{v}) - APX(\mathbf{v}) \geq L_{t+1}/c - (L_t + L_{t+1}/\eta) = L_{t+1}/\eta - L_t$. Plugging in the definition of L_{t+1}, we can get that $f(\delta) \geq \delta/(\eta\eta_1)$.

From the above analysis of two cases, we can conclude that $f(\delta) \geq \delta/(\eta\eta_1)$, and thus our claim holds. □

From the above analysis, we have the following performance guarantee for the online pricing mechanism under the condition that $C^* \geq \theta\delta\log\log\delta$:

$$\frac{C^*}{\widehat{C}} \leq 1 + \epsilon. \tag{14}$$

We now can prove the competitive ratio of ARETE-PR in terms of profit maximization.

Theorem 4. *For the problem of profit maximization in crowdsensed data markets, ARETE-PR can achieve the competitive ratio of $(1+\epsilon)$.*

Proof. We can assume that the performance loss from revenue maximization is less than the performance loss from basic reward minimization, *i.e.*, $(1+\epsilon) \leq 1 + \ln\frac{\Delta_{max}}{\Delta_{min}}$, as ϵ. We also assume that both optimal profit and approximate profit are non-negative, *i.e.*, $(1-\tau)\times C^* - \bar{\phi}\times M^* \geq 0$ and $(1-\tau)\times\widehat{C} - \bar{\phi}\times\widehat{M} \geq 0$. From equations (11) and (14), we then have

$$\frac{C^*}{\widehat{C}} \leq (1+\epsilon) \leq 1 + \ln\frac{\Delta_{max}}{\Delta_{min}} \leq \frac{M^*}{\widehat{M}}.$$

Furthermore, we can verify that for any positive numbers a, b, c and d with $a - c \geq 0$ and $b - d \geq 0$, if $a/b \leq c/d$, then we have $(a-c)/(b-d) \leq a/b$. Based on these observations, the approximation ratio of ARETE-PR satisfies the following relation:

$$\begin{aligned}
\frac{\Psi^*}{\widehat{\Psi}} &= \frac{(1-\tau)\times C^* - \bar{\phi}\times M^*}{(1-\tau)\times\widehat{C} - \bar{\phi}\times\widehat{M}} \\
&\leq \frac{C^*}{\widehat{C}} \leq (1+\epsilon).
\end{aligned} \tag{15}$$

Therefore, our theorem holds. □

4 Adaption to Other Query Types

In this section, we extend ARETE-PR to support multi-data query and range query scenarios.

4.1 Multi-Data Query

We can formulate the pricing problem for multi-data query as an unlimited-supply combinatorial posted-price auction with single-minded data consumers. A single-minded data consumer is interested in only a single data commodity, and has no valuation for all the other data commodities[6]. As we have discussed in Section 2.1, the price of a data commodity $Y \subseteq \mathbb{Y}$ is the sum of the prices of the basic data commodity in it, *i.e.*, $p_Y = \sum_{y\in Y} p_y$.

The extended ARETE-PR also consists of two components: versioning mechanism and pricing mechanism. We show that the versioning mechanism in ARETE-PR can be modified slightly to provide the version generation in the multi-data query scenario. Based on the pricing algorithm

6. In contrast, a multi-minded data consumer requests for multiple data commodities, and has different private valuations for different commodities. The multi-minded data consumers have powerful strategic behaviors to manipulate the online pricing mechanisms. The related works about the multi-arm bandit problem in strategic setting [2], [5] shed light on designing online pricing mechanism to resist the complex strategic behaviors of multi-minded data consumers. We reserve the detailed discussion to our future work.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2900243, IEEE Transactions on Mobile Computing

Algorithm 3: Pricing Mechanism for Multi-Data Query

Input: A set of random basic data commodity $\mathbb{Y}_1$; A data consumer b_i; A data commodity Y_i; A discount factor vector $\mathbf{d}_{Y_i}$; A weight vector $\mathbf{W}$.

Output: The charge c_i for the data consumer b_i.

1 $c_i \leftarrow 0$;
2 **if** $|Y_i \bigcap \mathbb{Y}_1| = 1$ **then**
3 $\quad y \leftarrow Y_i \bigcap \mathbb{Y}_1$;
4 $\quad c_i \leftarrow OPM_y(b_i, d_{Y_i}, \mathbf{W}_y)$;
5 **else**
6 $\quad$ Ignore the data consumer b_i;
7 **return** c_i

in original ARETE-PR, we design an online randomized pricing mechanism for multi-data query, and analyze its competitive ratio.

Versioning Mechanism In multi-data query scenario, the accuracy of a data commodity $Y \subseteq \mathbb{Y}$ is its average posterior variance $\frac{\sum_{y\in Y}\bar{\sigma}_y^2}{|Y|}$ after observing data from the selected data providers $\mathcal{A}_i$. The data commodity satisfies the accuracy requirement of the data consumer $b_i \in \mathbb{B}$ when $\frac{\sum_{y\in Y}\bar{\sigma}_y^2}{|Y|} \leq \epsilon_i$, which can be further expressed as

$$\sum_{y\in Y} \mathbf{\Sigma}_{y\mathcal{A}_i}\mathbf{\Sigma}_{\mathcal{A}_i\mathcal{A}_i}^{-1}\mathbf{\Sigma}_{\mathcal{A}_i y} \geq \sum_{y\in Y}\boldsymbol{\sigma}_y^2 - |Y| \times \epsilon_i.$$

The sum of submodular functions is also a submodular function. Similarly, we can formulate the process of data provider selection as a submodular covering problem, and the greedy versioning mechanism in ARETE-PR can be applied to the scenario of multi-data query . To determine the discount factors of different versions in multi-data query scenario, we extend relative entropy between the full version $f_1(x) = f_{\boldsymbol{\mathcal{X}}_Y|\boldsymbol{\mathcal{X}}_{\mathcal{A}_T}}(\mathbf{x}_Y|\mathbf{x}_{\mathcal{A}_T})$ and the tth version $f_2(x) = f_{\boldsymbol{\mathcal{X}}_Y|\boldsymbol{\mathcal{X}}_{\mathcal{A}_t}}(\mathbf{x}_Y|\mathbf{x}_{\mathcal{A}_t})$ to multivariate Gaussian distribution scenario, and define the revised relative entropy as

$$\widehat{D}\left(f_1||f_2\right) \triangleq \frac{1}{2}\left(\log\frac{|\mathbf{\Sigma}_2|}{|\mathbf{\Sigma}_1|} + tr(\mathbf{\Sigma}_2^{-1}\mathbf{\Sigma}_1) - |Y|\right),$$

where $tr(\mathbf{\Sigma})$ is the trace of matrix $\mathbf{\Sigma}$. We use this relative entropy to determine the discount factor for each version. Using the new concepts of accuracy and relative entropy $\widehat{D}\left(f_1||f_2\right)$, we can extend the versioning mechanism in ARETE-PR to multi-data query scenario, achieving the same performance guarantee.

Theorem 5. *For the problem of basic reward minimization in multi-data query scenario, the greedy versioning mechanism still achieves the approximation ratio of* $1 + \ln\frac{\Delta_{max}}{\Delta_{min}}$.

Online Pricing Mechanism Algorithm 3 presents the pseudo-code of online pricing mechanism for multi-data query. We reduce the online randomized pricing mechanism for multi-data query into multiple pricing mechanisms for single-data query in original ARETE-PR, *i.e.*, Algorithm 2. We describe this reduction in the following procedure.

Step 1: We first randomly partition the basic data commodities $\mathbb{Y}$ into two sets: $\mathbb{Y}_1$ and $\mathbb{Y}_2$, by placing each basic data commodity into $\mathbb{Y}_1$ with probability $\frac{1}{\kappa}$, where κ is the maximum size of the required data commodities, *i.e.*, $\kappa = \max_{b_i\in\mathbb{B}}|Y_i|$.

Step 2: We ignore data consumers, who want zero or more than one basic data commodity in $\mathbb{Y}_1$, and only consider the data consumers who want exactly one data commodity in $\mathbb{Y}_1$. We denote this type of data consumers by $\mathbb{B}_1 = \left\{b_i \in \mathbb{B} \middle| |Y_i \bigcap \mathbb{Y}_1| = 1\right\}$.

Step 3: We then set the prices of the basic data commodities in $\mathbb{Y}_2$ as zero, and effectively set the prices of the basic data commodities in $\mathbb{Y}_1$ with respect to the data consumers $\mathbb{B}_1$. Given a qualified data consumer b_i with $Y_i \cap \mathbb{Y} = y$, a discount factor vector $\mathbf{d}_{Y_i}$, and a weight vector $\mathbf{W}_y$, the Online Pricing Mechanism (abbreviated as OPM_y) for single-data query can determine the price for the basic data commodity y and the charge for the data consumer b_i (Line 3 to 4). The discount factor vector $\mathbf{d}_{Y_i}$ for Y_i is calculated by versioning mechanism. All the other parameters for the algorithm OPM_y are the same for all the basic data commodities, and we omit them here.

We show that this extended online pricing mechanism also achieves sub-optimal revenue.

Theorem 6. *Given a real value ϵ, there exists a constant θ such that for any valuation sequences with optimal revenue $C^* \geq l \times \theta \times \delta \times \log\log\delta$, the extended online pricing mechanism is $(1+\epsilon)$-competitive.*

Proof. We use $\boldsymbol{p}^* = (p_{y_1}^*, p_{y_2}^*, \cdots, p_{y_l}^*)$ to denote the optimal basic price vector for the basic data commodity $\mathbb{Y}$ in multi-data query scenario, and C^* to denote the optimal revenue achieved by $\boldsymbol{p}^*$. Let $C_{i,j}^*$ denote the revenue made by selling data commodity y_i to data consumer b_j with price $p_{y_i}^*$, and thus $C_{i,j}^* \in \left\{0, p_{y_i}^* \times d_j\right\}$ and $C^* = \sum_{i=1}^{l}\sum_{j=1}^{n} C_{i,j}^*$. Define a indicator variable $X_{i,j} = 1$ if the data commodity $y_i \in \mathbb{Y}_1$ and $b_j \in \mathbb{B}_1$; otherwise $X_{i,j} = 0$. We have

$$\mathbf{E}\left[X_{i,j}\right] = \mathbf{Pr}\left[y_i \in \mathbb{Y}_1, b_j \in \mathbb{B}_1\right] \geq \frac{1}{\kappa}\left(1 - \frac{1}{\kappa}\right)^{\kappa-1}.$$

We first show the relation between C^* and the quantity $\mathbf{E}\left[\sum_{y_i\in\mathbb{Y}_1}\sum_{b_j\in\mathbb{B}_1} C_{i,j}^*\right]$.

$$\begin{aligned}
&\mathbf{E}\left[\sum_{y_i\in\mathbb{Y}_1}\sum_{b_j\in\mathbb{B}_1} C_{i,j}^*\right] = \mathbf{E}\left[\sum_{i=1}^{l}\sum_{j=1}^{n} X_{i,j}C_{i,j}^*\right] \\
= &\sum_{i=1}^{l}\sum_{j=1}^{n}\mathbf{E}\left[X_{i,j}\right]C_{i,j}^* \geq \frac{1}{\kappa}\left(1-\frac{1}{\kappa}\right)^{\kappa-1}C^* \\
\geq &\frac{C^*}{\kappa e}.
\end{aligned}$$

We next analyze the expected revenue achieved by Algorithm 3. We can view Algorithm 3 as performing $|\mathbb{Y}_1|$ separate online pricing algorithms for single-data query. Let C_i^* denote the optimal revenue using a fixed basic price for $y_i \in \mathbb{Y}_1$ We note that the revenue C_i^* is at least $\sum_{b_j\in\mathbb{B}_1} C_{i,j}^*$, because setting prices of the basic data commodities in $\mathbb{Y}_2$ to be zero can increase the number of sales to data consumers in $\mathbb{B}_1$. By Theorem 2, the expected revenue of the online pricing mechanism OPM_{y_i} for a single data commodity

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2900243, IEEE Transactions on Mobile Computing

$y_i \in \mathbb{Y}_1$ will be at least $(1+\varepsilon)\, C_i^* - O\left(\theta \times \delta \times \log\log\delta\right)$. Therefore, given a randomized set of basic data commodities $\mathbb{Y}_1$, the revenue achieved by Algorithm 3 is at least:

$$\sum_{y_i \in \mathbb{Y}_1} \Big((1+\varepsilon) C_i^* - O(\theta \times \delta \times \log\log\delta) \Big).$$

Taking the expectation of the above equation with respect to the randomized generation of set $\mathbb{Y}_1$, we can get:

$$\begin{aligned} & \mathbf{E}\left[\sum_{y_i \in \mathbb{Y}_1} \Big((1+\varepsilon) C_i^* - O(\theta \times \delta \times \log\log\delta) \Big) \right] \\ \geq \quad & \mathbf{E}\Bigg[(1+\varepsilon) \sum_{y_i \in \mathbb{Y}_1} \sum_{b_j \in \mathbb{B}_1} C_{i,j}^* \\ & - |\mathbb{Y}_1|\, O\left(\theta \times \delta \times \log\log\delta\right) \Bigg] \\ \geq \quad & \frac{1+\varepsilon}{\kappa e} C^* - O\left(\frac{l}{\kappa} \times \theta \times \delta \log\log\delta \right). \end{aligned}$$

Similarly, by selecting appropriate parameters α, β and γ and assuming that k is a constant, we can get the results. □

Using the similar analytical technique in Theorem 4, we can have the following result for the extended ARETE-PR mechanism.

Theorem 7. *For the problem of profit maximization in multi-data query scenario, the extended ARETE-PR mechanism still achieves the competitive ratio of* $1+\epsilon$.

4.2 Range Query

In the case of range query, a data consumer wants to know the probability that a data commodity belongs to a specific range. For example, data consumers may be interested in whether monitoring environmental parameters, such as temperature, concentration of carbon dioxide, exceed some thresholds. The above mechanisms for single-data query and multi-data query can be easily extended to support range query. The versioning mechanisms remain the same, while in the pricing mechanisms, *i.e.*, Algorithm 2 and Algorithm 3, we multiply the final price by another discount factor $d_r = \frac{1}{2|Y|}$. This is because data consumers can know the posterior mean $\bar{\boldsymbol{\mu}}_Y$ of the data commodity Y by performing $2|Y|$ range queries. More specifically, data consumers can learn the mean of each basic data commodity $y \in Y$ by asking two range queries: $F(\mathcal{X}_y \in [-\infty, a_1])$ and $F(\mathcal{X}_y \in [-\infty, a_2])$. This can be done by looking up the standardized normal distribution table. As the mean of data commodity Y is the vector of the mean of the basic data commodity in Y, data consumers only need to ask $2|Y|$ similar queries to learn the mean of the data commodity Y. We can show that this modified online pricing mechanism for range query still achieves a constant approximation ratio. The proofs are similar as that in Theorem 4 and Theorem 7. In the interest of space, we omit the proof.

Finally, we show that ARETE-PR is arbitrage-free for different types of queries.

Theorem 8. *ARETE-PR is an arbitrage-free data pricing mechanism.*

Proof. We say a query q is "determined" by a query bundle $\{q_1, q_2, \cdots, q_k\}$ when the query q can be answered by the query bundle. We prove that ARETE-PR can resist arbitrage behaviours in both single-data query and multi-data query.

▷ In the single-data query case, the query q_1 with a low data accuracy is determined by the query q_2 with a high data accuracy. According to our versioning rule in Algorithm 2, the version used to answer the query q_1 is not higher than that used to answer q_2. Since the version with a lower accuracy has a large discount factor, the discount offered to the query q_1 is not less than that offers to q_2. Therefore, the charge to q_1 is always not less than the charge to q_2.

▷ In the multi-data query case, the multi-data query q over the data commodity Y is determined by the single-data query bundle $\{q_1, q_2, \cdots, q_{|Y|}\}$, where q_y is a single-data query over a basic data commodity y in Y. In extended ARETE-PR, we set the price of the data commodity Y as the sum of the basic prices of the basic commodities in Y. Thus, no arbitrage behaviours exist in this query scenario.

▷ In the range query case, the data query q over a data commodity Y is determined by the $2 \times |Y|$ different range queries over Y. In ARETE-PR, we set the charge of each range query as the charge of q multiplying a discount factor $\frac{1}{2\times|Y|}$. Therefore, the charge to q is equal to the sum of the charges to the range queries. In this case, ARETE-PR also satisfies the property of arbitrage-free. □

5 Reward Sharing

In this section, we design a reward sharing scheme, namely ARETE-SH, to fairly distribute the total bonus rewards among data providers. We start from formulating the problem of reward sharing as a coalitional game based on the versioning mechanism in ARETE-PR, which significantly reduces the space complexity of the classical representation form of the game. We then use a concise scheme, marginal contribution networks [34] to capture the substitutability among coalitions, further reducing the complexity of game representation. Finally, we design a computationally efficient algorithm to exactly calculate the Shapley value [50] of the reward sharing game, achieving four basic fairness axioms.

Since the total bonus rewards for data providers are simply the sum of rewards they obtain from different data commodities, we examine the reward sharing design for a specific data commodity in the following discussion.

5.1 Cooperative Game for Reward Sharing

Considering that data providers collaborate to generate data commodities, we model the interaction among data providers with the tool of cooperative game theory. In data markets, data providers could be connected with each other via certain kinds of networks, and are able to form small group and deviate from the ground coalition if the reward is not distributed in a fair way. For example, in recently emerging blockchain-based IoT data markets [21], [23], [37], [51], data providers are connected via a distributed network. As all the data trading information, including the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2900243, IEEE Transactions on Mobile Computing

reward received by each data provider, are published on the blockchain, data providers could be aware of the unfairness if the rewards are not well divided. The success of data market heavily relies on recruiting enough data providers to contribute high quality data. Considering the large volume of users and the quick speed of information spreading in social network, the data vendor could launch data acquisition campaign over social network. Data providers in social network can form a team to achieve competitive advantage and complete complex data acquisition tasks efficiently [42], [49]. Therefore, it is nature to adopt cooperative game theory to describe the behaviors of these groups in social network.

We model the problem of reward sharing in a data market as a coalitional game with m data providers $\mathbb{A} = \{a_1, a_2, \cdots, a_m\}$ and a reward vector $\mathbf{r} = (r_1, r_2, \cdots, r_n)$, $R \triangleq \sum_{i=1}^{n} r_i$, where r_i is the exclusive bonus reward for the set of data providers, who could provide the qualified version of data commodity for the data consumer $b_i \in \mathbb{B}$. The reward r_i is a certain percentage of the revenue c_i generated by ARETE-PR from data trading, *i.e.*, $r_i \triangleq \tau \times c_i$, where the specific value of τ can be determined by the negotiation between the data vendor and data providers in a bargaining game [46]. We call any nonempty subset of data providers $\mathcal{A} \subseteq \mathbb{A}$ a *coalition*. In general, there are exponential number of coalitions that can generate the qualified versions, which satisfies the accuracy requirement of the data consumer b_i. This will take space exponential in the number of data providers to describe the reward sharing game. We reduce the space complexity by using the versioning algorithm (Algorithm 1) to define the qualified coalitions for reward sharing. We call the coalitions that are selected by the versioning algorithm as *basic coalitions*. As the versioning mechanism randomly picks one data provider when multiple candidate data providers have the same marginal variance reduction, there may exist multiple eligible basic coalitions that have the same cardinality and satisfy the accuracy requirements of data consumers. We denote these e_i "equivalent" basic coalitions for the ith version by a collection $\hat{\mathcal{A}}_i = \{\mathcal{A}_i^1, \mathcal{A}_i^2, \cdots, \mathcal{A}_i^{e_i}\}$, where $|\mathcal{A}_i^j| = |\mathcal{A}_i^k|$ and $V(\mathcal{A}_i^j) \geq \sigma_y^2 - \epsilon_i, V(\mathcal{A}_i^k) \geq \sigma_y^2 - \epsilon_i$, for any $1 \leq j, k \leq e_i$. We represent the basic coalitions for all versions by vector $\boldsymbol{\mathcal{A}} = (\hat{\mathcal{A}}_1, \hat{\mathcal{A}}_2, \cdots, \hat{\mathcal{A}}_n)$. The data consumers are reordered such that $\epsilon_1 > \epsilon_2 > \cdots > \epsilon_n$. According to the greedy selection rule of the versioning algorithm, we can observe that for any basic coalition $\mathcal{A}_{i_1} \in \hat{\mathcal{A}}_{i_1}$ of version i_1 and a lower version i_2, $1 \leq i_2 < i_1$, there always exists a basic coalition $\mathcal{A}_{i_2} \in \hat{\mathcal{A}}_{i_2}$ for version i_2 such that $\mathcal{A}_{i_2} \subset \mathcal{A}_{i_1}$. We say a coalition $\mathcal{A}$ can generate the ith version and is eligible for sharing the reward r_i, only if the coalition $\mathcal{A}$ contains one of the basic coalition $\mathcal{A}_i^j$ from $\hat{\mathcal{A}}_i$.

By these notations, we can formally define the coalitional game for reward sharing.

Definition 3. *The reward sharing game can be represented by the pair $(\mathbb{A}, W)$, where*

▶ *$\mathbb{A}$ is the set of data providers and*

▶ *$W : 2^{\mathbb{A}} \mapsto \mathbb{R}$ is a worth function that maps each coalition of data providers $\mathcal{A} \subseteq \mathbb{A}$ to a real-valued reward*, i.e., $W(\mathcal{A}) = \sum_{i=1}^{i^*} r_i$, *where i^* is the highest version that the coalition $\mathcal{A}$ can generate*, i.e., $i^* = \arg\max_{1 \leq j \leq n} \mathcal{A} \supseteq \mathcal{A}_i^j$.

We assume that the reward of a coalition can be freely distributed among its members, which is known as the transferable utility assumption. The space complexity is still exponential in the number of data providers if we directly express the above reward sharing game. Observing that basic coalitions in collection $\hat{\mathcal{A}}_i$ are substitutable, we can use a compact representation scheme, marginal contribution networks [34], to capture this feature and efficiently describe the reward sharing game in next subsection.

5.2 Marginal Contribution Networks

The basic idea behind marginal contribution networks (MC-Nets) is to represent coalitional games using a set of *rules*, which have the following syntactic form: $Pattern \rightarrow Reward$. The $Pattern$ is a conjunction of data providers, including two types of literals: *positive literals* and *negative literals*. We use the negative literals to represent the absence of certain data providers, which are useful for expressing substitutability. Formally, we express the $Pattern$ with m_p positive literals and m_n negative literals as

$$\{a_1 \wedge a_2 \wedge \cdots \wedge a_{m_p} \wedge \neg\bar{a}_1 \wedge \neg\bar{a}_2 \wedge \cdots \wedge \neg\bar{a}_{m_n}\}.$$

We say that a rule applies to a coalition $\mathcal{A}$, if $\mathcal{A}$ meets the requirement of the $Pattern$, *i.e.*, $\{a_i\}_{i=1}^{m_p} \in \mathcal{A}$ and $\{\bar{a}_i\}_{i=1}^{m_n} \notin \mathcal{A}$. The reward of a coalition is defined to be the sum over the reward of all the rules that apply to the coalition.

We now use MC-Nets to represent the reward sharing game in Definition 3, and show the corresponding pseudo-code in Algorithm 4 (Lines 2 to 9). As the reward r_i will be counted only once for the reward of the coalition that contains multiple basic coalitions from $\hat{\mathcal{A}}_i$, we need to capture the substitutability among the basic coalitions in $\hat{\mathcal{A}}_i$. The coalitional game for sharing the reward r_i of the tth version can be represented as the following rules:

$$\begin{aligned} \{\mathcal{A}_i^1\} &\rightarrow r_i \\ \{\mathcal{A}_i^2 \wedge \neg\bar{\mathcal{A}}_i^1\} &\rightarrow r_i \\ &\vdots \\ \left\{\mathcal{A}_i^{e_i} \wedge \neg\bar{\mathcal{A}}_i^{e_i-1} \wedge \neg\bar{\mathcal{A}}_i^{e_i-2} \wedge \cdots \wedge \neg\bar{\mathcal{A}}_i^1\right\} &\rightarrow r_i \end{aligned}$$

In the jth rule, the positive literals are $\mathcal{A}_i^j$, and the negative literals are data providers in $\bigcup_{k=1}^{j-1} \bar{\mathcal{A}}_i^k$, where $\bar{\mathcal{A}}_j^k = \mathcal{A}_i^k \backslash \mathcal{A}_i^j$ (Lines 4 to 6). The entire game for reward sharing can then be built up from the set of rules for all versions (Lines 7 to 9). This expressive representation scheme fully describes the reward sharing game from Definition 3, and reduces the space requirement to $O(ne^*)$, where n is the number of versions (also the number of data consumers) and e^* is the maximum equivalent basic coalitions for one version, *i.e.*, $e^* = max_{1 \leq i \leq n} e_i$.

5.3 Computing the Shapley Value

We first briefly introduce the concept of Shapley value, which is a powerful result for cooperative game proven by Shapley in 1953 [50]. We use ϕ_i to denote the Shapley value for data provider $i \in \mathbb{A}$. The Shapley value is the unique way to distribute the grand reward among data providers that satisfies four fairness axioms:

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2900243, IEEE Transactions on Mobile Computing

Efficiency (EFF): The sum of the share of all data providers is the grand reward, *i.e.*, $\sum_{i\in\mathbb{A}}\phi_i = W(\mathbb{A}) = R$.

Symmetry (SYM): If data providers i and j are interchangeable, *i.e.*, $W(\mathcal{A}\cup\{i\}) = W(\mathcal{A}\cup\{j\}), \forall\mathcal{A}\subseteq\mathbb{A}\backslash\{i,j\}$, then their Shapley values are equal, *i.e.*, $\phi_i = \phi_j$.

Dummy (DUM): If data provider i is a dummy data provider, *i.e.*, her marginal contributions to all coalition $\mathcal{A}$ are the same, then $\phi_i = W(\{i\})$.

Additivity (ADD): For any two coalitional games V and W defined over the same set of data providers $\mathbb{A}$, $\phi_i(V+W) = \phi_i(V) + \phi_i(W)$ for all $i\in\mathbb{A}$, where the game $V+W$ is defined as $(V+W)(\mathcal{A}) = V(\mathcal{A}) + W(\mathcal{A})$ for all $\mathcal{A}\subseteq\mathbb{A}$.

The Shapley value to data provider i is the average marginal contribution of i over all possible permutations of the data providers, and can be calculated by:

$$\phi_i = \sum_{\mathcal{A}\subseteq\mathbb{A}\backslash\{i\}} \frac{|\mathcal{A}|!(|\mathbb{A}|-|\mathcal{A}|-1)!}{|\mathbb{A}|!}(W(\mathcal{A}\cup\{i\}) - W(\mathcal{A})).$$

Given the MC-Nets of the reward sharing game, we can design ARETE-SH, a simple and efficient algorithm to compute the Shapley value of the game. Specifically, we first compute the Shapley value of data providers in each rule by considering each rule as a separate game. The final Shapley value of each data provider is the sum of the Shapley values she obtains in all rules. The following lemma from [34] demonstrates that this "divide and conquer" scheme correctly computes the Shapley value of data providers in the reward sharing coalition game.

Lemma 3. *The Shapley value of a data provider in reward sharing game is equal to the sum of the Shapley value over each rule in MC-Nets.*

We now compute the Shapley value of data providers in each rule, and show the corresponding pseudo-code in Algorithm 4 (Lines 10 to 19). We separate the analysis into two scenarios: one for rules with only positive literals, and the other for rules with both positive and negative literals.

In the rules with only positive literals, the positive literals in the rule are indistinguishable from each other. By the *Efficiency* axiom and *Symmetry* axiom, the Shapley value of each positive literals in the rule is r/m_p, where r is the reward of the rule, and m_p is the number of positive literals in the rule (Lines 12 to 14).

For the rules that have mixed literals, we further consider the positive literals (Lines 16 to 17) and negative literals (Lines 18 to 19), separately. A positive literal a_i has non-zero marginal contribution only in the permutation that a_i appears after the rest of the positive literals but before any of the negative literals. Therefore, the Shapley value for the positive literal a_i in the rule with m_p positive literals and m_n negative literals is

$$\phi_i = \frac{(m_p-1)!m_n!}{(m_p+m_n)!}r = \frac{r}{m_p\binom{m_p+m_n}{m_n}}. \tag{16}$$

The negative literal a_j has a non-zero marginal contribution, if all positive literals come before the literal a_j, and a_j is the first among the negative literals. Thus, we have

$$\phi_j = \frac{m_p!(m_n-1)!}{(m_p+m_n)!}(-r) = \frac{-r}{m_n\binom{m_p+m_n}{m_p}}. \tag{17}$$

Algorithm 4: Reward Sharing Mechanism

Input: A basic coalition vector $\boldsymbol{\mathcal{A}}$; Reward vector $\mathbf{r}$.
Output: Reward vector for data providers $\boldsymbol{\Phi}$.

1 $Rule \leftarrow \varnothing$; $\quad \boldsymbol{\Phi} \leftarrow \mathbf{0}$;
2 **for** $t = 1$ *to* T **do**
3 **for** $i = 1$ *to* e_t **do**
4 **for** $j = i-1$ *to* 1 **do**
5 $\bar{\mathcal{A}}_t^j \leftarrow \mathcal{A}_t^j \backslash \mathcal{A}_t^i$;
6 $Pos \leftarrow \mathcal{A}_t^i$;
 $Neg \leftarrow \{\neg\bar{\mathcal{A}}_t^{i-1} \bigwedge \neg\bar{\mathcal{A}}_t^{i-2} \bigwedge \cdots \bigwedge \neg\bar{\mathcal{A}}_t^1\}$;
7 $Pattern \leftarrow Pos \bigwedge Neg$;
8 $Reward \leftarrow r_t$;
9 $Rule \leftarrow Rule \bigcup \{(Pattern \rightarrow Reward)\}$;
10 **foreach** $(Pattern_k \rightarrow Reward_k) \in Rule$ **do**
11 $m_p \leftarrow |Pos_k|$; $\quad m_n \leftarrow |Neg_k|$;
12 **if** $Pattern_k$ *is positive* **then**
13 **foreach** $a_i \in Pattern_k$ **do**
14 $\phi_i \leftarrow \phi_i + \frac{r_k}{|m_p+m_n|}$;
15 **if** $Pattern_k$ *is mixed* **then**
16 **foreach** $a_i \in Pos_k$ **do**
17 $\phi_i \leftarrow \phi_i + \frac{r_k}{m_p\binom{m_p+m_n}{m_n}}$;
18 **foreach** $a_i \in Neg_k$ **do**
19 $\phi_i \leftarrow \phi_i - \frac{r_k}{m_n\binom{m_p+m_n}{m_p}}$;
20 **return** $\boldsymbol{\Phi}$;

According to *Symmetry* axiom, all positive literals have the same value ϕ_i, and negative literals have the value of ϕ_j.

We can compute the Shapley value of a data provider in a given rule within constant time. There are at most $n \times e^*$ rules in the game, and thus the time complexity of Algorithm 4 is $O(mne^*)$.

6 Evaluation Results

In this section, we evaluate ARETE on a public real-world sensory data set.

Sensory Data Set. The data set we considered in our evaluations is the Intel sensed data set collected by Intel Berkeley lab between February 28th and April 5th, 2004. As shown in Figure 3, 54 Mica2Dot sensor nodes were deployed in the lab to collect multi-dimensional environment attributes, including temperature, humidity, light, voltage, and etc, in a real time manner. In our evaluations, we sample temperature measurements at 30 seconds intervals on 11 consecutive days (Starting Feb. 28th, 2004) in the lab with x-coordinate varying from 0m to 40.5m and y-coordinate varying from 0m to 31m. We set the upper right corner of the lab to be the origin with the coordinates $(0,0)$. We collect 11 data sets, randomly choose one of them as the data commodity, and use the remaining data sets to train the parameters of Gaussian Process model.

For choosing Gaussian Process as the statistical model, we have to know the mean and kernel functions. In our evaluations, we use regression techniques to estimate the mean function. We assume that the kernel function is

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2900243, IEEE Transactions on Mobile Computing

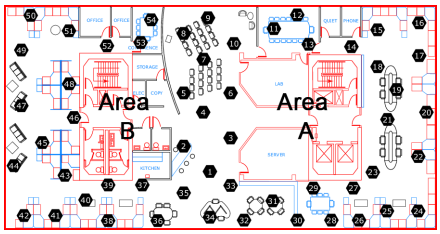


Fig. 3. Sensor network deployment with 54 nodes in one selected lab.

isotropic, which means that the covariance between two locations only depends on their corresponding distance. One canonical isotropic kernel function is Gaussian kernel function: $\mathcal{K}(a_1, a_2) = \sigma^2 \exp\left(-\frac{d(a_1,a_2)^2}{2l^2}\right)$, where $d(a_1, a_2)$ is the distance between locations a_1 and a_2. Using the training data sets, we can learn the parameters σ and l by cross-validation. In order to verify the efficient description of the isotropic kernel function for our data sets, we compare the empirical data of each sensor node with the readings inferred via the data from the other 53 sensors. As Figure 4(a) shows, for most sensor nodes (around 85%), the error of the inferential readings are within 10% of the ground truth. We note that ARETE is independent of specific kernel functions. For more complicated environment, we can adopt some general anisotropic kernel functions [47]. After determining the mean and kernel functions, we can plot the posterior mean and posterior variance of the lab in Figure 4(b) and Figure 4(c), respectively, using Equation (1) and Equation (2). Figure 4(b) shows the areas near the windows (y-coordinates lie near 0.) have lower inferential temperature. From Figure 4(c), we observe that area A and area B, located in the center of the lab, have higher posterior variances, because in these areas with few sensor nodes deployed, we lack enough relative data to confidently infer their readings.

Evaluation Setup. We introduce the setting of our evaluations. We regard the 54 sensor nodes as data providers in the context of data market. We create a finite mesh grid with mesh width 1m in the lab region, and obtain 1312 grid points, which are considered as basic data commodities. We emulate a large scale data market, in which the number of data consumers ranges from 10^5 to 10^6 with increment of 10^5. We consider two classical valuation distributions: Uniform distribution and Normal distribution, and set the maximum valuation of data consumers as $\delta = 256$. We randomly generate an accuracy requirement $\eta_i \in (0, 1]$ for each data consumer b_i. All the evaluation results are averaged over 200 runs.

6.1 Performance of ARETE-PR

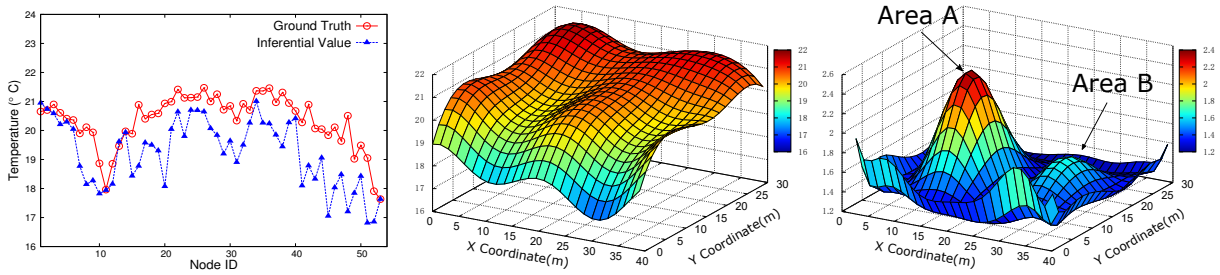
We implement ARETE-PR, and compare its performance with three other pricing mechanisms: Optimal pricing mechanism ("OPT" for short), Random pricing mechanism ("Random" for short), and ARETE-PR without versioning ("No Version" for short). In "OPT" mechanism, the valuation information and arrival sequence of all data consumers are known in advance, and the data vendor can calculate the off-line optimal revenue by setting a single fixed price. We note that the "OPT" is impractical as it requires the priori knowledge of data consumers' valuations, but can be served as a bench mark in our evaluations. In "Random" mechanism, we randomly select a price in $[1, \delta]$ as the charge for each data consumer's query. In order to investigate the impact of versioning mechanism on the data market's performance, we also consider the ARETE-PR without versioning, in which each data commodity only has the full version. Considering the computational overhead, we set β to be 0.1, which can capture at least 90% of optimal revenue by Lemma 1. Since α and β jointly determine the trade-off between exploration and exploitation, we fix α as 0.02, and adjust γ to examine the role of exploration and exploitation in different valuation distribution scenarios. When the valuations are drawn from normal distribution, we set $\gamma = 0.1$, and for uniform distribution, we set $\gamma = 0.35$. As we determine the price for data commodities independently, we only report the revenue of the data commodity at location $(25, 10)$ in this set of evaluations.

Figure 5 shows the revenue of different pricing mechanisms, when the valuations follow two different distributions. Generally, in both normal distribution and uniform distribution, ARETE-PR always outperforms the "Random" and "No Version" mechanisms, and approaches the results of "OPT". The "Random" mechanism does not take any advantage of the collected valuation information, and achieves the worst performance. This performance degradation is especially severe in normal distribution scenario, because the "Random" mechanism does not adopt the exploitation process, which can significantly improve the performance when the valuations densely locate in a certain small range. In "No Version" pricing mechanism, data consumers with low accuracy requirements cannot afford the high price of the full version, and the data vendor loses much revenue from these data consumers. We observe that ARETE-PR mechanism gains around 90% revenue of the "OPT" in both uniform and normal distribution. This demonstrates that ARETE-PR can adaptively learn the valuations of consumers, and set an appropriate price to obtain high revenue. From Figure 5, we can also see that the revenue increases linearly with respect to the number of data consumers. This is because data commodity is one kind of information goods and is unlimitedly supplied, and thus the data vendor can always gain revenue by selling more data commodities to more data consumers.

6.2 Performance of ARETE-SH

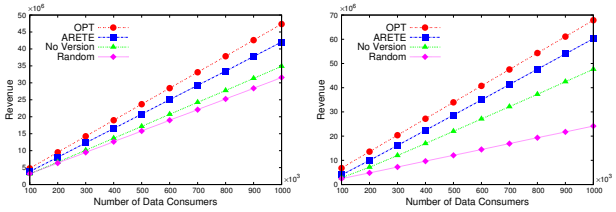
We now report the evaluation results of ARETE-SH. For each data commodity, we fix the number of corresponding data consumers as 10^5, and choose normal distribution as their valuation distributions. We first focus on sharing the reward from selling a single data commodity at a fixed location $(25, 10)$. In practice, it is complicated to design each version for each data consumer. Thus, the data vendor could pre-define several standard versions, and selects the lowest qualified version to the data consumer. As shown in Figure 2, we apply the versioning mechanism of ARETE to generate three basic coalition $\mathcal{A}_1$, $\mathcal{A}_2$ and $\mathcal{A}_3$. We assume that the reward for sharing is 80% of the revenue generated by the online pricing mechanism of ARETE. Thus, we can calculate the rewards for the three basic coalitions $(\mathcal{A}_1, \mathcal{A}_2, \mathcal{A}_3)$ as $(r_1, r_2, r_3) = (0.517 \times 10^6, 1.222 \times 10^6, 1.438 \times 10^6)$.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2900243, IEEE Transactions on Mobile Computing



(a) Inferential Value vs Ground Truth. (b) Inferential Temperature. (c) Variance of Temperature Inference.

Fig. 4. Posterior mean and posterior variance of the temperature Gaussian Process estimated using all sensors.



(a) Uniform Distribution. (b) Normal Distribution.

Fig. 5. The revenue of ARETE under different valuation distributions.

We randomly select three data providers with ID 8 from $\mathcal{A}_1$, ID 7 from $\mathcal{A}_2 \backslash \mathcal{A}_1$, and ID 11 from $\mathcal{A}_3 \backslash \mathcal{A}_2$. We plot their corresponding rewards in Figure 6(a), where the separation of the bars represents the source of the reward. Figure 6(a) shows that the data providers from the same $\mathcal{A}_i$ obtain the same reward from r_i, *e.g.*, data provider 8 and data provider 7, belonging to $\mathcal{A}_2$, obtain the same reward from r_2. This is because according to the principle of ARETE-SH, we equally share the reward r_i among the data providers in $\mathcal{A}_i$. From Figure 6(a), we can also see that the data providers from $\mathcal{A}_i$ obtain higher reward than the data providers from $\mathcal{A}_{i+1} \backslash \mathcal{A}_i$, *e.g.*, data provider 8 receives more rewards than data provider 7. The reason is that we have $\mathcal{A}_1 \subset \mathcal{A}_2 \subset \mathcal{A}_3$ from the versioning result, meaning that the data providers in $\mathcal{A}_i$ can obtain rewards from r_j, $j \geq i$. Compared with data providers in $\mathcal{A}_{i+1} \backslash \mathcal{A}_i$, data providers in $\mathcal{A}_i$ can get extra rewards from r_i. Thus, we can conclude that ARETE-SH equally distributes the reward r_i among data providers in $\mathcal{A}_i$, and the data providers with high variance reduction can receive more rewards, which demonstrates the fairness of ARETE-SH.

We now investigate the effect of data market demand on the reward sharing. We query on the data commodities in the whole area, and calculate the accumulated reward of each data provider. We first consider the unbiased demand setting, in which each data commodity is queried by the same number of data consumers. We further consider the biased demand scenario, in which data commodities located in the left side (x-coordinate lies in the range $[0, 20]$) receives more queries than those located in the right side. In Figure 6(b) and Figure 6(c), the radius of each circle represents the cumulative reward of data provider at the corresponding location. As Figure 6(b) shows, in the unbiased case, the data providers in the sparse area can attain higher rewards than those in the dense area. The reason is that ARETE-SH only shares the reward with the sets of data providers selected by the versioning mechanism in ARETE-PR. According to the selection criterion in greedy versioning mechanism, the data providers in the sparse area have high chances to be selected to generate the data commodities around them, as they provide more informative information to the generation of data commodities. From Figure 6(c), we can see that the data providers in the dense area can also obtain high reward if their located area (left side) has popular queries. This is because the data vendor can obtain large revenue from the data commodities with high market demands, and the total bonus reward in ARETE-SH is proportional to the revenue extracted from data trading.

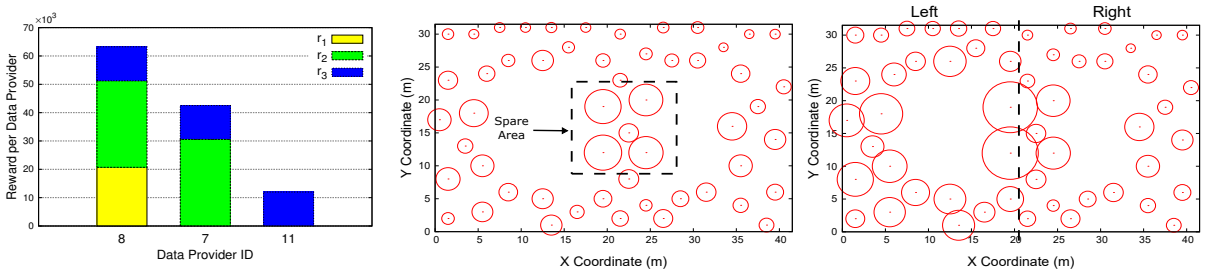
The evaluation results of ARETE-SH have a profound impact on the revenue of data trading in a long term: with the discriminative reward provided by ARETE-SH, the data acquisition scheme could automatically guide data providers to collect the data that is profitable in data markets. This positive impact of ARETE-SH on revenue is due to two critical design ideas in ARETE-SH: one is the criterion to select the qualified coalitions of data providers for reward sharing, and the other one is the proportion relation of reward to the revenue. The result in Figure 6(b) implies that the reward distributed by ARETE-SH would incentivize data providers to collect data for the areas with few data providers employed. This will improve the accuracies of data commodities in the sparse areas and attract the data consumers with high accuracy requirements, leading to high revenue for the data market. Without ARETE-SH scheme, the data commodities in these sparse areas cannot obtain revenue as they do not satisfy the accuracy requirements of these data consumers. Figure 6(c) indicates that ARETE-SH would steer data providers to collect data for the areas with high market demands, which would maintain the data commodities in these areas at a high accuracy level, and continuously extract revenue from the market.

7 Related Work

We briefly review the related works in this section.

Data Marketplace In the seminal paper of data trading [6], Balazinska *et al.* visioned the implications of emerging data markets, and discussed the potential research opportunities in this direction. Later, Koutris *et al.* [40] pointed out the inflexibility of current data pricing approaches, and proposed a query-based data pricing framework, which requires two important properties: *arbitrage-free* and *discount-free*. Recently, Zheng *et al.* studied the problem of

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2900243, IEEE Transactions on Mobile Computing



(a) Reward per Data Provider (b) Unbiased Query (c) Biased Query

Fig. 6. Performance of ARETE-SH.

profit driven data acquisition in mobile crowd-sensed data market [63]. However, these previous works did not answer the fundamental question in data trading: how to determine the price for data services? We tackle this open problem by designing a online pricing mechanism.

Mobile Crowdsensing: The ubiquitous mobile devices with powerful sensors have boosted the rapid growth of diverse mobile sensing applications in numerous contexts. For example, Gu *et al.* presented crowdsensing-based indoor localization system [29]. Wang *et al.* designed CrowdAltas to automatically update maps based on people's GPS traces [56]. The success of these applications highly depends on the supply of large amount of crowd-sensed data from crowds. Thus, researchers have proposed pricing mechanisms to incentivize workers to contribute their collected data [32], [39], [61], [64]. Kai *et al.* extended the traditional single-minded setting to multi-minded user model, in which users have different private costs for different tasks, and only perform a subset of the tasks [32]. The authors then designed an online pricing mechanism to incentive multi-minded users under the adversarial scenario. Mobile crowdsensing is a variance of crowdsourcing in mobile environment, the problem of incentive design has also been widely investigated for different types of crowdsourcing services. For example, Wen and Lin designed an optimal fee schedule to coordinate the incentive conflict between a crowdsourcing website and contest sponsors [57]. Their results imply that the widespread linear fee schedule is not optimal. Alelyani *et al.* adopted the machine learning techniques, such as topic modeling and NLP techniques, for price estimation, and proposed Context-Centric Pricing approach to support software crowdsourcing pricing [1].

The incentive design in mobile crowdsensing system is different from that in data markets. Data providers in data markets also incur sensing costs during data acquisition process, and the data vendor compensates these costs with a basic rewards, which is similar in mobile crwodsensing. The difference part is that the data vendor has to share a portion of revenue from data trading with data providers. In data markets, we augment the basic reward with a bound reward to offer incentive for data providers, and design an efficient algorithm to calculate the Shapley value, achieving the four fairness axioms. Currently, the operators collected and analyzed crowd-sensed data for their own application purposes. To break this barrier, we proposed a data market to facilitate the exchange and trading of crowd-sensed data, enabling the potential usage of mobile data in new sensing applications.

Online Pricing Mechanism: In this paper, we built a connection between data pricing design and online digital auction design [9], [10], [31]. By exploiting the machine learning techniques in multi-armed bandit problem [3], Blum *et al.* [10] proposed an online posted-price digital auction, achieving a constant competitive ratio with an additional loss term $O(\delta \log \delta \log\log \delta)$. Later, Blum and Hartline [9] improved on the approximation results [10] by reducing the additive loss term to $O(\delta \log\log \delta)$. As for online auctions with multiple unlimited items and single-minded buyers, Balcan and Blum [7] proposed several approximation algorithms to achieve near-optimal revenue. Balcan *et al.* [8] showed that single posted-price mechanisms can achieve sub-optimal revenue for the unlimited supply setting with multi-minded buyers. Without considering the strategic behaviours of buyers, the digital auction design can be reduced to algorithmic pricing problem, and several approximation pricing algorithms have been proposed in different scenarios [25], [53]. In mobile data markets, the trading data should be further partitioned into multiple versions to implement some levels of price discrimination, extracting revenue from different market segments. The major advantage of our work over the previous works is to model digital goods as divisible items, producing new challenges for online pricing mechanism design.

8 Conclusion

In this work, we have proposed the first data market prototype to enable mobile crowd-sensed data trading on the Web. We have built a Gaussian Process model to capture the uncertainty of mobile data, and provided three basic query interfaces for data consumers to extract their needed information from the statistical model. We have considered the problem of profit maximization, and proposed an online query-based data pricing mechanism, namely ARETE-PR, containing two major components: a versioning mechanism and an online pricing mechanism. ARETE-PR satisfies arbitrage-freeness, and achieves a constant competitive ratio. We have further designed a reward sharing scheme, namely ARETE-SH, to calculate the Shapley value for data providers. We have leveraged a real-world sensory data set to evaluate ARETE. The evaluation results show that ARETE outperforms the existing pricing mechanisms, and is almost as effective as the optimal fixed price mechanism. ARETE-SH can distribute the rewards in a fair manner.

REFERENCES

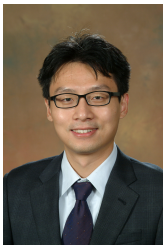
[1] T. Alelyani, K. Mao, and Y. Yang. Context-centric pricing: Early pricing models for software crowdsourcing tasks. In *PROMISE*, 2017.
[2] K. Amin, A. Rostamizadeh, and U. Syed. Learning prices for repeated auctions with strategic buyers. In *NIPS*. 2013.
[3] P. Auer, N. Cesa-Bianchi, Y. Freund, and R. E. Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *FOCS*, 1995.
[4] Azure data marketplace. http://www.infochimps.com/.
[5] M. Babaioff, Y. Sharma, and A. Slivkins. Characterizing truthful multi-armed bandit mechanisms: Extended abstract. In *EC*, 2009.
[6] M. Balazinska, B. Howe, and D. Suciu. Data markets in the cloud: An opportunity for the database community. In *VLDB*, 2011.
[7] M.-F. Balcan and A. Blum. Approximation algorithms and online mechanisms for item pricing. In *EC*, 2006.
[8] M.-F. Balcan, A. Blum, and Y. Mansour. Item pricing for revenue maximization. In *EC*, 2008.
[9] A. Blum and J. D. Hartline. Near-optimal online auctions. In *SODA*, 2005.
[10] A. Blum, V. Kumar, A. Rudra, and F. Wu. Online learning in online auctions. In *SODA*, 2003.
[11] J.-M. Bohli, C. Sorge, and D. Westhoff. Initial observations on economics, pricing, and penetration of the internet of things market. *SIGCOMM Computer Communication Review*, 39(2):50–55, 2009.
[12] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.
[13] R. K. Chellappa and S. Shivendu. Managing piracy: Pricing and sampling strategies for digital experience goods in vertically segmented markets. *Information Systems Research*, 16(4):400–417, 2005.
[14] Q. Chen, H. Hu, and J. Xu. Authenticating top-k queries in location-based services with confidentiality. *Journal Proceedings of the VLDB Endowment*, 7(1):49–60, 2013.
[15] R. Cheng, T. Emrich, H.-P. Kriegel, N. Mamoulis, M. Renz, G. Trajcevski, and A. Zufle. Managing uncertainty in spatial and spatio-temporal data. In *ICDE*, 2014.
[16] X. Chu, I. F. Ilyas, S. Krishnan, and J. Wang. Data cleaning: Overview and emerging challenges. In *SIGMOD*, 2016.
[17] T. M. Cover and J. A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012.
[18] N. Cressie. *Statistics for spatial data*. John Wiley & Sons, 2015.
[19] Customlists. http://www.customlists.net/.
[20] A. Das and D. Kempe. Algorithms for subset selection in linear regression. In *STOC*, 2008.
[21] Databroker dao. https://databrokerdao.com/.
[22] Dataexchange. http://new.thedataexchange.com/.
[23] Datum. https://datum.org/.
[24] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *VLDB*, 2004.
[25] D. E. Difallah, M. Catasta, G. Demartini, and P. Cudré-Mauroux. Scaling-up the crowd: Micro-task pricing schemes for worker retention and latency improvement. In *HCOMP*, 2014.
[26] W. Du, Z. Xing, M. Li, B. He, L. H. C. Chua, and H. Miao. Optimal sensor placement and measurement of wind for water quality studies in urban reservoirs. In *IPSN*, 2014.
[27] Factual. https://www.factual.com/.
[28] Gnip. https://gnip.com/.
[29] F. Gu, J. Niu, and L. Duan. Waipo: A fusion-based collaborative indoor localization system on smartphones. *IEEE/ACM Transactions on Networking*, 2017. DOI: 10.1109/TNET.2017.2680448.
[30] A. Guillory and J. A. Bilmes. Online submodular set cover, ranking, and repeated active learning. In *NIPS*. 2011.
[31] V. Guruswami, J. D. Hartline, A. R. Karlin, D. Kempe, C. Kenyon, and F. McSherry. On profit-maximizing envy-free pricing. In *SODA*, 2005.
[32] K. Han, Y. He, H. Tan, S. Tang, H. Huang, and J. Luo. Online pricing for mobile crowdsourcing with multi-mindeds users. In *Mobihoc*, 2017.
[33] Here. https://company.here.com/here/.
[34] S. Ieong and Y. Shoham. Marginal contribution nets: A compact representation scheme for coalitional games. In *EC*, 2005.
[35] Infochimps. http://www.infochimps.com/.
[36] Instagram. https://www.instagram.com/.
[37] Internet of Things Data Marketplace by IOTA. https://data.iota.org/.
[38] R. K. Iyer and J. A. Bilmes. Submodular optimization with submodular cover and submodular knapsack constraints. In *NIPS*, 2013.
[39] M. Karaliopoulos, I. Koutsopoulos, and M. Titsias. First learn then earn: Optimizing mobile crowdsensing campaigns through data-driven user profiling. In *MobiHoc*, 2016.
[40] P. Koutris, P. Upadhyaya, M. Balazinska, B. Howe, and D. Suciu. Toward practical query pricing with querymarket. In *SIGMOD*, 2013.
[41] A. Krause, H. B. McMahan, C. Guestrin, and A. Gupta. Robust submodular observation selection. *Journal of Machine Learning Research*, 9(Dec):2761–2801, 2008.
[42] T. Lappas, K. Liu, and E. Terzi. Finding a team of experts in social networks. In *KDD*, 2009.
[43] C. Meng, W. Jiang, Y. Li, J. Gao, L. Su, H. Ding, and Y. Cheng. Truth discovery on crowd sensing of correlated entities. In *SenSys*, 2015.
[44] L. Mo, Y. He, Y. Liu, J. Zhao, S.-J. Tang, X.-Y. Li, and G. Dai. Canopy closure estimates with greenorbs: Sustainable sensing in the forest. In *SenSys*, 2009.
[45] Nasdaq. http://www.nasdaq.com/.
[46] J. F. Nash. The bargaining problem. *Econometrica*, 18(2):155–162, 1950.
[47] D. J. Nott and W. T. Dunsmuir. Estimation of nonstationary spatial covariance structure. *Biometrika*, 89(4):819–829, 2002.
[48] E. Rahm and H. H. Do. Data cleaning: Problems and current approaches. *IEEE Data Eng. Bull.*, 23(4):3–13, 2000.
[49] M. Rokicki, S. Zerr, and S. Siersdorfer. Groupsourcing: Team competition designs for crowdsourcing. In *WWW*, 2015.
[50] L. Shapley. A value for n-person games, in H.W. Kuhn and A.W. Tucker, editors. *Contributions to the Theory of Games, volume II,Princeton University Press*, 1953.
[51] Streamr. https://www.streamr.com/.
[52] L. Sun, R. Cheng, D. W. Cheung, and J. Cheng. Mining uncertain data with probabilistic guarantees. In *KDD*, 2010.
[53] V. Syrgkanis and J. Gehrke. Pricing queries (approximately) optimally. Technical report, http://arxiv.org/abs/1508.05347, 2015.
[54] Thingful. https://thingful.net/.
[55] Thingspeak. https://thingspeak.com/.
[56] Y. Wang, X. Liu, H. Wei, G. Forman, C. Chen, and Y. Zhu. Crowdatlas: Self-updating maps for cloud and personal use. In *MobiSys*, 2013.
[57] Z. Wen and L. Lin. Pricing crowdsourcing services. In *LISS*, 2015.
[58] C. K. Williams and C. E. Rasmussen. *Gaussian Processes for Regression*. MIT, 1996.
[59] L. A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.
[60] Xignite. http://www.xignite.com/.
[61] D. Yang, G. Xue, X. Fang, and J. Tang. Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing. In *MobiCom*, 2012.
[62] L. Zhang, Y. Li, X. Xiao, X. Li, J. Wang, A. Zhou, and Q. Li. Crowdbuy: Privacy-friendly image dataset purchasing via crowdsourcing. In *INFOCOM*, 2018.
[63] Z. Zheng, Y. Peng, F. Wu, S. Tang, and G. Chen. Trading data in the crowd: Profit-driven data acquisition for mobile crowdsensing. *IEEE Journal on Selected Areas in Communications*, 35(2):486–501, 2017.
[64] Z. Zheng, Z. Yang, F. Wu, and G. Chen. Mechanism design for mobile crowdsensing with execution uncertainty. In *ICDCS*, 2017.
[65] P. Zhou, Y. Zheng, and M. Li. How long to wait?: Predicting bus arrival time with mobile phone based participatory sensing. In *MobiSys*, 2012.

Zhenzhe Zheng received the BE in Software Engineering from Xidian University, in 2012, and the MS degree and the PhD degree in computer science and engineering from Shanghai Jiao Tong University, in 2015 and 2018, respectively. He is now visiting the University of Illinois at Urbana-Champaign (UIUC) as a post doc researcher. His research interests include wireless networking and mobile computing, game theory and algorithm design. He is a student member of the ACM, IEEE, and CCF.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMC.2019.2900243, IEEE Transactions on Mobile Computing

Yanqing Peng is a Ph.D. candidate from School of Computing, University of Utah, USA. He received B.Eng. degree in Computer Science and Engineering from Shanghai Jiao Tong University in 2016. His research interests include wireless networking, datacenter networking, algorithmic game theory, and large-scale data management.

Fan Wu is a professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. He received the BS degree in Computer Science from Nanjing University, in 2004, and the PhD degree in Computer Science and Engineering from the State University of New York at Buffalo, in 2009. He has visited the University of Illinois at Urbana-Champaign (UIUC) as a Post Doc Research Associate. His research interests include wireless networking and mobile computing, algorithmic game theory and its applications, and privacy preservation. He has published more than 100 peer-reviewed papers in technical journals and conference proceedings. He is a recipient of the first class prize for Natural Science Award of China Ministry of Education, NSFC Excellent Young Scholars Program, ACM China Rising Star Award, CCF-Tencent "Rhinoceros bird" Outstanding Award, CCF-Intel Young Faculty Researcher Program Award, and Pujiang Scholar. He has served as the chair of CCF YOCSEF Shanghai, on the editorial board of Elsevier Computer Communications, and as the member of technical program committees of more than 60 academic conferences. For more information, please visit http://www.cs.sjtu.edu.cn/~fwu/.



Shaojie Tang is currently an assistant professor of Naveen Jindal School of Management at University of Texas at Dallas. He received the PhD degree in computer science from Illinois Institute of Technology, in 2012. His research interests include social networks, mobile commerce, game theory, e-business, and optimization. He received the Best Paper Awards in ACM MobiHoc 2014 and IEEE MASS 2013. He also received the ACM SIGMobile service award in 2014. Dr. Tang served in various positions (as chairs and TPC members) at numerous conferences, including ACM MobiHoc and IEEE ICNP. He is an editor for International Journal of Distributed Sensor Networks.

Guihai Chen earned the BS degree from Nanjing University, in 1984, the ME degree from Southeast University, in 1987, and the PhD degree from the University of Hong Kong, in 1997. He is a distinguished professor of Shanghai Jiaotong University, China. He had been invited as a visiting professor by many universities including Kyushu Institute of Technology, Japan, in 1998, University of Queensland, Australia, in 2000, and Wayne State University, USA during September 2001 to August 2003. He has a wide range of research interests with focus on sensor network, peer-to-peer computing, high-performance computer architecture and combinatorics. He has published more than 200 peer-reviewed papers, and more than 120 of them are in well-archived international journals such as IEEE Transactions on Parallel and Distributed Systems, Journal of Parallel and Distributed Computing, Wireless Network, The Computer Journal, International Journal of Foundations of Computer Science, and Performance Evaluation, and also in well-known conference proceedings such as HPCA, MOBIHOC, INFOCOM, ICNP, ICPP, IPDPS, and ICDCS.