An Online Pricing Mechanism for Mobile Crowdsensing Data Markets

Zhenzhe Zheng, Yanqing Peng, Fan Wu*, Shaojie Tang¶, and Guihai Chen
{zhengzhenzhe,wu-fan,gchen}@sjtu.edu.cn,yqpeng@foxmail.com,tangshaojie@gmail.com
Shanghai Key Laboratory of Scalable Computing and Systems, Shanghai Jiao Tong University, China
¶Department of Information Systems, University of Texas at Dallas, USA

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. To build a practical mobile crowd-sensed data market, we have to consider three major design challenges: data uncertainty, economic-robustness (arbitrage-freeness in particular), revenue maximization. By jointly considering the design challenges, we propose a novel online query-bAsed cRowd-sensEd daTa pricing mEchanism, namely ARETE, to determine the trading price of crowd-sensed data. Our theoretical analysis shows that ARETE guarantees both arbitrage-freeness and a constant competitive ratio in terms of revenue maximization. We have evaluated ARETE on a real-world sensory data set collected by Intel Berkeley lab. Evaluation results show that ARETE outperforms the state-of-the-art pricing mechanisms, and achieves around 90% of the optimal revenue.

CCS CONCEPTS

• **Networks** → **Network economics**; • **Theory of computation** → *Computational pricing and auctions*;

KEYWORDS

Data Marketplace, Mobile Crowdsensing, Online Pricing

1 INTRODUCTION

As a significant business reality, data trading has attracted increasing attentions and focuses. For example, Xignite [37] sells financial data, Gnip [17] vends data from social networks, and Factual [16] trades geographic data. Potential data consumers might be Nasdaq [28] for financial data, Instagram [22] for social data, and Here [20] for location trace data. To support these online data transactions, several marketplace services have emerged, *e.g.*, Azure Data Marketplace [3], Infochimps [21], and Dataexchange [13]. 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 [4, 24], 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 [26] to vehicular networks for traffic monitoring and prediction [41], these deployments generate tremendous volumes of valuable but uncertain numeric sensing 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 [8]. 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 raw 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 [33] and Thingful [32], 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 three 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 [9], 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 [25]. 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 in the data market [8]. Researchers have proposed several model-based methods to manage sensing data in the past decades [9, 14, 31]. However, due to the various formats of mobile sensing data and the complex correlation among data, it is not possible 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

*F. Wu is the corresponding author.

This work was supported in part by the State Key Development Program for Basic Research of China (973 project 2014CB340303). The work of Z. Zheng was also supported by Google PhD Fellowship and Microsoft Asia PhD Fellowship. This work was also supported in part by China NSF grant 61672348, 61672353, 61422208, and 61472252, in part by Shanghai Science and Technology fund 15220721300, and in part by the Scientific Research Foundation for the Returned Overseas Chinese Scholars. The opinions, findings, and conclusions expressed in this paper are those of the authors and do not necessarily reflect the views of the funding agencies or the government.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

Mobihoc '17, July 10-14, 2017, Chennai, India
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-4912-3/17/07…$15.00
https://doi.org/http://dx.doi.org/10.1145/3084041.3084044

fixed prices for the whole or parts of the data set [3, 12]. 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 [4, 24], 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-free* [24] 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 revenue maximization with incomplete information. 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. Such a cost structure makes existing cost-based pricing mechanisms unsuitable for data trading. Thus, the value-based pricing mechanisms are more attractable for data trading. However, in online marketing system, the valuations and arrival sequences of data consumers are unknown to the data vendor. Thus, the data vendor has to determine the price of data with incomplete information. The optimization on revenue 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.

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 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, including 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, to achieve arbitrage-freeness and a constant competitive ratio. Specifically, for each of data commodities, ARETE generates multiple *versions* with different accuracy levels to extract revenue from data consumers in different market segments, and determines the trading prices of the versions by dynamically learning the valuations of data consumers. 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 revenue, 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 revenue maximization.

• Second, we begin with considering a basic setting, in which data consumers only ask single-data queries, and design ARETE, including a versioning mechanism and an online pricing mechanism. We further extend ARETE to adapt to other data query scenarios. We prove that ARETE achieves both arbitrage-freeness and a constant competitive ratio in terms of revenue maximization.

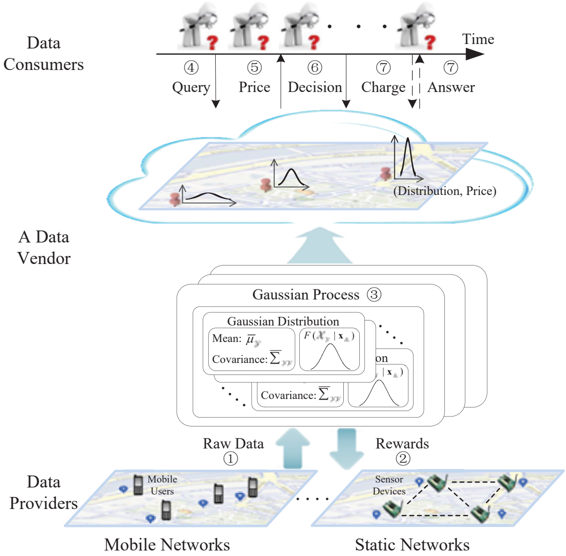

Data
Consumers
Time
④
Query
⑤
Price
⑥
Decision
⑦
Charge
⑦
Answer
(Distribution, Price)
A Data
Vendor
Gaussian Process ③
Gaussian Distribution
Mean: $\overline{\mu}_{\mathcal{Y}}$
Covariance: $\overline{\Sigma}_{\mathcal{YY}}$
$F(\mathcal{X}_{\mathcal{Y}} \mid \mathbf{x}_{\mathcal{A}})$
on
$\mid \mathbf{x}_{\mathcal{A}})$
Covariance: $\overline{\Sigma}_{\mathcal{YY}}$
Raw Data
①
Rewards
②
Data
Providers
Mobile
Users
Sensor
Devices
Mobile Networks
Static Networks

Figure 1: A Mobile Crowd-Sensed Data Market.

• 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 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. We extend ARETE to support diverse query formats in Section 4. The evaluation results are presented in Section 5. In Section 6, we review related work. We conclude the paper in Section 7.

2 PRELIMINARIES

In this section, we formally describe system model for mobile crowd-sensed data trading, and the problem of revenue maximization.

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 observations collected by the data providers. For convenience of discussion, we assume that each data provider only contributes one piece of data.

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. In addition, the sensing data is collected at some selected locations, and cannot fully represent the continuous feature of the physical environment. Therefore, the data vendor needs to filter the noisy raw data, and to infer the data at the locations where no data providers are employed. In such cases, regression techniques can be used to handle the noise in raw data and to perform inference. 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 adopt a powerful regression technique *Gaussian Process* [11, 36], which is a generalization of linear regression, and has been widely used as to model numerical sensing data [14, 15], to perform inferences, and to cope with the uncertainty quantification in the process of inferences.[1]

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 Σ. Let $\boldsymbol{\mu}_Y$ and Σ_{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:

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

where $\mathbf{x}_Y$ is a vector of possible values of random variables $\boldsymbol{\mathcal{X}}_Y$, $|\Sigma|$ is the determinant of matrix Σ, and Σ^{-1} is the inverse matrix of Σ. Under the Gaussian Process model, we can infer the data at any set of locations $Y \subseteq \Theta$ (even there are no observations at these locations), condition on the observations $\mathbf{x}_{\mathbb{A}}$. The resulting distribution $f_{\boldsymbol{\mathcal{X}}_Y|\boldsymbol{\mathcal{X}}_{\mathbb{A}}}(\mathbf{x}_Y|\mathbf{x}_{\mathbb{A}})$ is a conditional multivariate Gaussian distribution, whose posterior mean vector $\bar{\boldsymbol{\mu}}_Y$ and posterior covariance matrix $\overline{\Sigma}_{YY}$ can be expressed as:

$$\bar{\boldsymbol{\mu}}_Y = \boldsymbol{\mu}_Y + \Sigma_{Y\mathbb{A}}\Sigma_{\mathbb{A}\mathbb{A}}^{-1}(\mathbf{x}_{\mathbb{A}} - \boldsymbol{\mu}_{\mathbb{A}}), \quad (1)$$

$$\overline{\Sigma}_{YY} = \Sigma_{YY} - \Sigma_{Y\mathbb{A}}\Sigma_{\mathbb{A}\mathbb{A}}^{-1}\Sigma_{\mathbb{A}Y}, \quad (2)$$

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 public knowledge. Thus, the posterior covariance matrix $\overline{\Sigma}_{YY}$, which is independent on the actual observations $\mathbf{x}_{\mathbb{A}}$, is publicly known.

Data Commodity: In crowd-sensed data market, we define the data commodity for trading as the conditional Gaussian distributions $f_{\boldsymbol{\mathcal{X}}_Y|\boldsymbol{\mathcal{X}}_{\mathbb{A}}}(\mathbf{x}_Y|\mathbf{x}_{\mathbb{A}})$.[2] 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 selects a finite set of random variables at several locations, sometimes called 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 basic 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 covariance matrices are public knowledge, so the valuable information of a data commodity is its mean vector. Furthermore, by Equation (1), the mean of a data commodity Y is actually the vector of the means of the basic data commodities in it. 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 $[\underline{\mathbf{a}}_i, \overline{\mathbf{a}}_i]$.

Confidence Level: Each data consumer $b_i \in \mathbb{B}$ reports an error bound ϵ_i and a confidence level η_i, representing the acceptable accuracy of the queried data commodity. The confidence level of the data commodity Y_i with an error bound ϵ_i is defined as:

$$CL(Y_i, \epsilon_i) \triangleq F(\boldsymbol{x}_{Y_i} \in \mathbf{B}(\bar{\boldsymbol{\mu}}_{Y_i}, \epsilon_i)), \quad (3)$$

where $\mathbf{B}(\bar{\boldsymbol{\mu}}_{Y_i}, \epsilon_i)$ represents the Euclidean ball with a center at $\bar{\boldsymbol{\mu}}_{Y_i}$ and a radius ϵ_i. We note that confidence level $CL(Y_i, \epsilon_i)$ is in direct proportion to the determinant of posterior covariance matrices $|\overline{\Sigma}_{Y_iY_i}|$ [27]. We can obtain the approximation results using numerical integration procedures. The data commodity Y_i satisfies the required confidence level of the data consumer b_i if $CL(Y_i, \epsilon_i) \geq \eta_i$.

[1] It is not possible to propose a universal statistical model to describe all types of sensing data. In this work, we focus on numerical sensing data, such as temperature, humidity, light, voltage, and etc.

[2] The possible privacy leakage of data providers and the potential violation of data copyright can be some other reasons to trade data services rather than raw data in data market.

Data Charging: Considering that the data commodity with different confidence levels should have different prices, the data vendor offers a discount $d_i \in (0, 1]$ for each data consumer $b_i \in \mathbb{B}$ according to her required confidence level (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 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.

2.2 Problem Formulation

In this paper, we consider one important problem in mobile crowd-sensed data market: *Revenue Maximization.*

The goal of data vendor is to maximize obtained revenue, which is defined as the sum of the charges for data customers that purchase data commodities, *i.e.*, $C \triangleq \sum_{b_i \in \mathbb{B}: v_i > c_i} c_i$. In contrast, 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 2.1 (Arbitrage-free Data Pricing Mechanism). Wh-enever 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 revenue maximization in mobile crowd-sensed data market: The data vendor dynamically determines the charge $\mathbf{c}$ (by calculating the basic prices $\mathbf{p}$ and discount factor d_i) for data consumers $\mathbb{B}$, without knowing the data consumers' arrival sequence and private valuation vector $\mathbf{v}$, such that the resulting data pricing mechanism achieves good competitive ratio and the property of arbitrage-freeness.

3 ONLINE DATA PRICING

In this section, we propose ARETE, which is a version-based online posted-pricing mechanism for mobile crowd-sensed data market. ARETE consists of two components: a versioning mechanism and an online pricing mechanism. The versioning mechanism generates multiple versions for a data commodity to satisfy the diverse confidence levels of data consumers. The online pricing mechanism determines the basic price for each basic data commodity with the goal of revenue maximization.

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 for one selected basic data commodity. We further extend ARETE to adapt to the other types of query in Section 4.

3.1 Versioning

In ARETE, we regard the conditional Gaussian distribution $f(x_y|\mathbf{x}_{\mathcal{A}})$ generated by the observations $\mathbf{x}_{\mathcal{A}}$ from some data providers $\mathcal{A} \subseteq \mathbb{A}$ as a version of the basic data commodity $y \in \mathbb{Y}$,[3] and use conditional entropy to quantify the accuracy of the version. The conditional entropy of the Gaussian distribution $f_{\mathcal{X}_y|\boldsymbol{\mathcal{X}}_{\mathbb{A}}}(x_y|\mathbf{x}_{\mathbb{A}})$ is:

$$\begin{aligned} H(\mathcal{X}_y|\boldsymbol{\mathcal{X}}_{\mathbb{A}}) &\triangleq -\int f(x_y, \mathbf{x}_{\mathcal{A}}) \log f(x_y|\mathbf{x}_{\mathcal{A}})\, dx_y\, d\mathbf{x}_{\mathcal{A}} \\ &= \frac{1}{2} \log \left(2\pi e \bar{\sigma}_y^2\right), \end{aligned} \tag{4}$$

where $\bar{\sigma}_y^2$ is the posterior variance of the distribution $f(x_y|\mathbf{x}_{\mathcal{A}})$. The conditional entropy can be calculated in a closed form using Equation (2).

Algorithm 1: Versioning Mechanism

```
Input: The number of versions $T$; An accuracy vector $\mathbf{h}$; A scale
        parameter $\lambda$.
Output: A vector of selected data providers $\boldsymbol{\mathcal{A}}$; A vector of discount
        factors $\mathbf{d}$.
1  $t \leftarrow 0$;  $\boldsymbol{\mathcal{A}} \leftarrow \boldsymbol{\varnothing}$;  $A \leftarrow \varnothing$;
2  while $t \leq T$ do
3      $a^* \leftarrow \arg\min_{a_i \in \mathbb{A} \setminus A} H(\mathcal{X}_y|\boldsymbol{\mathcal{X}}_A \cup \mathcal{X}_{a_i})$;
4      $A \leftarrow A \bigcup \{a^*\}$;
5      if $H(\mathcal{X}_y|\boldsymbol{\mathcal{X}}_A \cup \mathcal{X}_{a_i}) \leq h_t$ then
6          $\mathcal{A}_t \leftarrow A$; $\boldsymbol{\mathcal{A}} \leftarrow \mathcal{A}_t$; $t \leftarrow t + 1$;
7  $\sigma_{y|\mathcal{A}_T} \leftarrow \sigma_y - \Sigma_{y\mathcal{A}_T} \Sigma^{-1}_{\mathcal{A}_T\mathcal{A}_T} \Sigma_{\mathcal{A}_T y}$;
8  for $t = 1$ to $T$ do
9      $\sigma_{y|\mathcal{A}_t} \leftarrow \sigma_y - \Sigma_{y\mathcal{A}_t} \Sigma^{-1}_{\mathcal{A}_t\mathcal{A}_t} \Sigma_{\mathcal{A}_t y}$;
10     $f_1(x) = f_{\mathcal{X}_y|\boldsymbol{\mathcal{X}}_{\mathcal{A}_T}}(x_y|\mathbf{x}_{\mathcal{A}_T})$;
11     $f_2(x) = f_{\mathcal{X}_y|\boldsymbol{\mathcal{X}}_{\mathcal{A}_t}}(x_y|\mathbf{x}_{\mathcal{A}_t})$;
12     $\widehat{D}(f_1||f_2) \leftarrow \frac{1}{2}\left(\log \frac{\sigma^2_{y|\mathcal{A}_t}}{\sigma^2_{y|\mathcal{A}_T}} + \frac{\sigma^2_{y|\mathcal{A}_T}}{\sigma^2_{y|\mathcal{A}_t}} - 1\right)$;
13     $d_t \leftarrow e^{-\lambda \widehat{D}(f_1||f_2)}$;
14 return $\boldsymbol{\mathcal{A}}$, $\boldsymbol{d}$;
```

By using the standard market research techniques, such as surveys, the data vendor can determine the number of versions T and the corresponding accuracy vector $\mathbf{h} = (h_1, h_2, \cdots, h_T)$, meaning that the conditional entropy of the tth version should be less than h_t.[4] In general, we assume that $h_{t_1} > h_{t_2}$, for $1 \leq t_1 < t_2 \leq T$. We use $\mathcal{A}_t \subseteq \mathbb{A}$ to denote the data providers recruited to generate the tth version. The data vendor always wants to employ less data providers to achieve the accuracy requirements of the versions, due to the high cost of recruiting large number of data providers.

We 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 conditional entropy satisfies the accuracy requirement of certain version. Formally, our goal is to select the next data provider a_i that minimizes $H(\mathcal{X}_y|\boldsymbol{\mathcal{X}}_A \cup \mathcal{X}_{a_i})$, where A is the set of currently selected data providers. We break the tie following a random rule (Lines 3 to 4). If the new conditional entropy $H(\mathcal{X}_y|\boldsymbol{\mathcal{X}}_A)$ is less

[3] 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 a highly accurate data commodity.

[4] Determining the number of versions and the accuracy vector is beyond the scope of this paper, and will be discussed in our future work. Several previous works [30, 34] shed light on possible solutions for this problem.

than the accuracy of the tth version h_t, we generate this version by setting $\mathcal{A}_t$ as the current data provider set A (Lines 5 to 6).

The remaining issue is to determine discount factor for each version. We set the discount factor of a version proportional to its distance to the full version, *i.e.*, the distribution $f_{\mathcal{X}_y|\mathcal{X}_{\mathcal{A}_T}}(x_y|\mathbf{x}_{\mathcal{A}_T})$, and fix 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 [10]. Specifically, the relative entropy between the full version $f_1(x) = f_{\mathcal{X}_y|\mathcal{X}_{\mathcal{A}_T}}(x_y|\mathbf{x}_{\mathcal{A}_T})$ and the tth version $f_2(x) = f_{\mathcal{X}_y|\mathcal{X}_{\mathcal{A}_t}}(x_y|\mathbf{x}_{\mathcal{A}_t})$ is

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

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 (5) may not reflect such property, because the relative entropy depends on both the mean and variance. As shown in Equation (4), the accuracy of a version 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

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

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

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

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 $\sigma_{y|\mathcal{A}_T}$ of the full version $f(y|\mathcal{A}_t)$ in Line 7. For the tth version, we calculate its variance $\sigma_{y|\mathcal{A}_t}$ in Line 9, and the corresponding distance and discount factor according to Equation (6) and Equation (7), respectively (Lines 12 to 13).

3.2 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 $1 \leq k \leq K$ 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 $1 \leq k \leq K$. 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

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

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 $\mathbf{1}$.

Algorithm 2: Online Pricing Mechanism

```
Input: Reals: $\alpha \in (0,1]$, $\beta > 0$, $\gamma \in (0,1]$; The $i$th data consumer $b_i$;
       A vector of discount factors $\mathbf{d}$; The highest valuation $\delta$; 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
   $g(k) = \frac{\Delta}{(1+\beta)^{K-k+1}}$, $\Delta = \frac{1-\frac{1}{1+\beta}}{1-\left(\frac{1}{1+\beta}\right)^K}$;
3  Suppose the selected price is $\hat{p}_{k_i}$;
4  Choose the lowest version $t_i$ that satisfies the required confidence
   level $\eta_i$ 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      $c_i(k_i) \leftarrow c_i$;
8  else
9      $c_i(k_i) \leftarrow 0$;
10 foreach $k = 1$ to $K$ do
11     if $k = k_i$ then
12         $\hat{c}_i(k) \leftarrow \frac{\gamma\Delta}{\delta}\frac{c_i(k)}{\hat{f}_i(k)}$;    $w_i(k) \leftarrow w_{i-1}(k) \times (1+\alpha)^{\hat{c}_i(k)}$;
13     else
14         $\hat{c}_i(k) \leftarrow 0$;    $w_i(k) \leftarrow w_{i-1}(k)$;
15 return $c_i$;
```

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

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

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 [7]. However, considering that different candidate prices can produce different amount of revenue, we adopt a geometric distribution as the exploitation distribution, *i.e.*,

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

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 $\hat{p}_k/g(k) = O\left((1+\beta)^{k-1}(1+\beta)^{K-k+1}\right) = O(\delta)$, 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).

After calculating the confidence level of each version using Equation (3), we can select the lowest version t_i, that satisfies the required confidence level 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 $[0, 1]$.

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

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 5.

3.3 Analysis

We analyze the competitive ratio of ARETE in this subsection. In ARETE, 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. We leave the detailed proof to our technical report [1].

Lemma 3.1. *ARETE loses only a $(1+\beta)$ factor in rounding down the optimal price to one of the prices from $\hat{\boldsymbol{p}}$.*

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

Lemma 3.2. *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:*

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

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

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

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

$$
\begin{aligned}
&\log\frac{W_n}{W_0} = \sum_{i=1}^{n} \log\frac{W_i}{W_{i-1}} \le \sum_{i=1}^{n}\left(1 + \alpha\frac{\sum_{k=1}^{K} w_{i-1}(k)\hat{c}_i(k)}{W_{i-1}}\right) \\
&\le \sum_{i=1}^{n} \alpha\frac{\sum_{k=1}^{K} w_{i-1}(k)\hat{c}_i(k)}{W_{i-1}} = \alpha\sum_{i=1}^{n}\sum_{k=1}^{K} f_i(k)\hat{c}_i(k) \\
&= \alpha\mathbf{f}_i \cdot \hat{\boldsymbol{c}}_i. \qquad (11)
\end{aligned}
$$

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

$$
\log\frac{W_n}{W_0} \ge \sum_{i=1}^{n} \hat{c}_i(k) \log(1+\alpha) - \log K. \qquad (12)
$$

Combining Equations (11) and (12), we get

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

We have completed the proof. □

By Lemma 3.1, Lemma 3.2 and an appropriate choice of parameters α, β and γ, we can obtain the following competitive ratio for ARETE.

Theorem 3.3. *Given a real value ϵ, there exists a constant θ, such that for any valuation sequences $\boldsymbol{v}$ with optimal revenue $OPT \ge \theta\delta \log\log\delta$, ARETE is $(1+\epsilon)$-competitive.*

[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 confidence levels to seek less payments. The policy of selecting the lowest version enforces data consumers to truthfully report their required confidence levels.

Proof. Using Lemma 3.2 and the properties of ARETE, we show the lower bound of revenue $\sum_{i=1}^{n} c_i(k_i)$ for any selected basic price sequence $\hat{\boldsymbol{p}} = (\hat{p}_{k_1}, \hat{p}_{k_2}, \cdots, \hat{p}_{k_n})$.

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

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:

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

In the third equality, we select the optimal fixed price from $\hat{\boldsymbol{p}}$, and thus $\max_k\{\sum_{i=1}^{n} c_i(k)\} = OPT_\beta$. The third equality follows from that $\log(1+\alpha) \geq \alpha - \frac{\alpha^2}{2}$ for any $\alpha > 0$. By Lemma 3.1, the last inequality holds. By choosing appropriate parameters α, β and γ, we prove the theorem. □

We have proven that ARETE 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 3.4. *There is no constant-competitive online pricing algorithm for all valuation sequences with $OPT \geq o(\delta)$.*

Due to the limitation of space, we leave the proof of Theorem 3.4 to our technical report [1].

4 ADAPTION TO OTHER QUERY TYPES

In this section, we extend ARETE to support multi-data query formats, and leave the extension to range query formats to our technical report [1], due to space limitation.

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. 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.*, $p_Y = \sum_{y\in Y} p_y$.

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 $b_i$.
$c_i \leftarrow 0$;
if $|Y_i \bigcap \mathbb{Y}_1| = 1$ then
    $y \leftarrow Y_i \bigcap \mathbb{Y}_1$;
    $c_i \leftarrow OPM_y(b_i, d_{Y_i}, \mathbf{W}_y)$;
else
    Ignore the data consumer $b_i$;
return $c_i$
```

The extended ARETE also consists of two components: versioning mechanism and pricing mechanism. We show that the versioning mechanism in ARETE can be modified slightly to provide the version generation in the multi-data query scenario. Based on the pricing algorithm in original ARETE, we design an online randomized pricing mechanism for multi-data query, and analyze its competitive ratio.

Versioning Mechanism In multi-data query scenario, we define the conditional entropy of a commodity $Y \subseteq \mathbb{Y}$ as:

$$\begin{aligned}
H(\boldsymbol{\mathcal{X}}_Y|\boldsymbol{\mathcal{X}}_{\mathbb{A}}) &\triangleq -\int f(\mathbf{x}_Y, \mathbf{x}_{\mathcal{A}})\log f(\mathbf{x}_Y|\mathbf{x}_{\mathcal{A}})\,d\mathbf{x}_Y\,d\mathbf{x}_{\mathcal{A}}\\
&= \frac{1}{2}\log\left((2\pi e)^{|Y|}|\overline{\Sigma}_{YY}|\right),
\end{aligned}$$

where $|\Sigma|$ is the determinant of matrix Σ. We use this conditional entropy as a criterion to generate versions. In this case, the revised 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 $f_2(x) = f_{\boldsymbol{\mathcal{X}}_Y|\boldsymbol{\mathcal{X}}_{\mathcal{A}_t}}(\mathbf{x}_Y|\mathbf{x}_{\mathcal{A}_t})$ is also extended to the multivariate Gaussian distribution scenario, and is defined as

$$\widehat{D}\left(f_1||f_2\right) \triangleq \frac{1}{2}\left(\log\frac{|\Sigma_2|}{|\Sigma_1|} + tr(\Sigma_2^{-1}\Sigma_1) - |Y|\right),$$

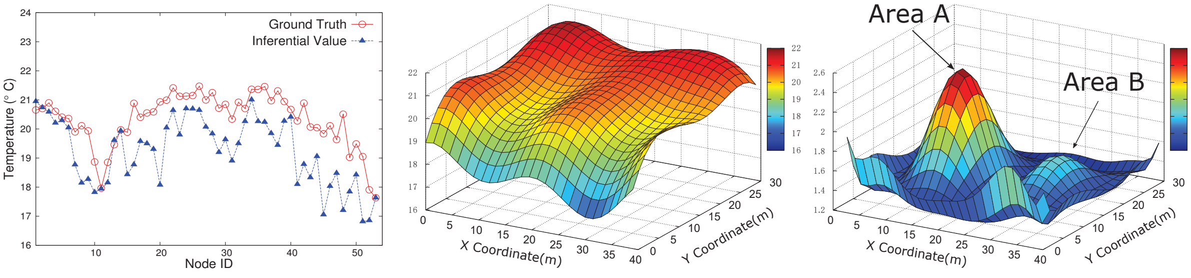
where $tr(\Sigma)$ is the trace of matrix Σ. We use this relative entropy to determine the discount factor for each version. Using the new conditional entropy $H(\boldsymbol{\mathcal{X}}_Y|\boldsymbol{\mathcal{X}}_{\mathbb{A}})$ and relative entropy $\widehat{D}\left(f_1||f_2\right)$, we can extend the versioning mechanism in ARETE (Algorithm 1) to the multi-data query scenario.

Online Pricing Mechanism Algorithm 3 presents the pseudocode of online pricing mechanism for multi-data query scenario. We reduce the online randomized pricing mechanism for multi-data query into multiple pricing mechanisms for single-data query in original ARETE, *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



(a) Inferential Value vs Ground Truth.

(b) Inferential Temperature.

(c) Variance of Temperature Inference.

Figure 2: Posterior mean and posterior variance of the temperature Gaussian Process estimated using all sensors.

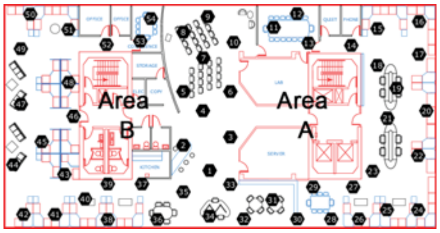


Figure 3: Sensor network deployment with 54 nodes in one selected lab.

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 the extended ARETE also achieves sub-optimal revenue. Due to the limitation of space, we reserve the detailed proof to our technical report [1].

THEOREM 4.1. *Given a real value ϵ, there exists a constant θ such that for any valuation sequences with optimal revenue $OPT \geq l \times \theta \times \delta \times \log\log\delta$, the extended ARETE is $(1+\epsilon)$-competitive with respect to the optimal fixed price revenue.*

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

THEOREM 4.2. *ARETE 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 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 confidence level is determined by the query q_2 with a high confidence level. According to our version selection 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 lower version 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, 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. □

5 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 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 2(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 [29]. After determining the mean and kernel functions, we can plot the posterior mean and posterior variance of the lab in Figure 2(b) and Figure 2(c), respectively, using Equation (1)

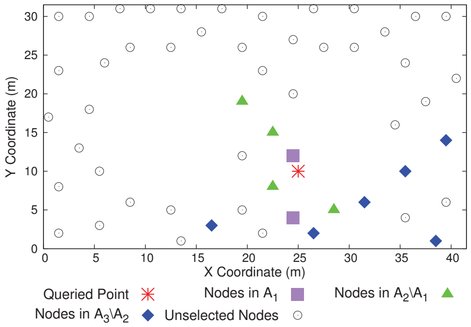


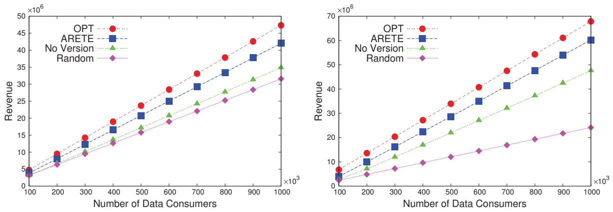
Figure 4: Versioning results of the data commodity at location (25, 10).

and Equation (2). Figure 2(b) shows the areas near the windows (y-coordinates lie near 0.) have lower inferential temperature. From Figure 2(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 error bound $\epsilon_i \in (0, 10]$ and an acceptable confidence level $\eta_i \in (0, 1]$ for each data consumer $b_i \in \mathbb{B}$. All the evaluation results are averaged over 200 runs.

5.1 Performance of ARETE

We implement ARETE, and compare its performance with three other pricing mechanisms: Optimal pricing mechanism (“OPT” for short), Random pricing mechanism (“Random” for short), and ARETE without versioning (“No Version” for short). In “OPT” mechanism, the valuations of all data consumers are known in advance, and thus we 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 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 3.1. 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 selected data commodity at location $(25, 10)$ in this set of evaluations.



(a) Uniform Distribution.

(b) Normal Distribution.

Figure 5: The revenue of ARETE under different valuation distributions.

Figure 4 shows the versioning result of the data commodity at location $(25, 10)$. The vector of conditional entropy for the three versions is $\mathbf{h} = (3.25, 2.75, 2.55)$. We recall that $\mathcal{A}_t$ denotes the set of data providers for the tth version, and $\mathcal{A}_{t+1} \backslash \mathcal{A}_t$ denotes the data providers that only stay in $\mathcal{A}_{t+1}$. In Equation (7), we set the scale parameter λ 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 $\mathbf{d} = (0.36, 0.85, 1)$. From Figure 4, we observe that the data providers, neighboring the queried point, have a high probability to be selected into versioning results, 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.

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 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 required confidence levels cannot afford the high price of the full version, and the data vendor loses much revenue from these data consumers. We observe that ARETE mechanism gains around 90% revenue of the “OPT” in both uniform and normal distribution. This demonstrates that ARETE 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 RELATED WORK

We briefly review the related works in this section.

Data Marketplace In the seminal paper of data trading [4], Balazinska *et al.* visioned the implications of emerging data markets, and discussed the potential research opportunities in this direction. Later, Koutris *et al.* [24] poi-nted 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 profit driven data acquisition in mobile crowd-sensed data market [39]. However, these previous works did not answer the fundamental question in data trading: how to determine the price for data? 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 [18]. Wang *et al.* designed CrowdAltas to automatically update maps based on people's GPS traces [35]. 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 [23, 38, 40]. However, currently, the operators collected and analyzed mobile 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 [6, 7, 19]. By exploiting the machine learning techniques in multi-armed bandit problem [2], Blum *et al.* [7] 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 [6] improved on the approximation results [7] 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 [5] proposed several approximation algorithms to achieve near-optimal revenue. In mobile crowd-sensed 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.

7 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 revenue maximization, and proposed an online query-based data pricing mechanism, namely ARETE, containing two major components: a versioning mechanism and an online pricing mechanism. ARETE satisfies arbitrage-freeness, and achieves a constant competitive ratio. 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.

REFERENCES

[1] An online pricing mechanism for mobile crowdsensing data market. Technical report, https://drive.google.com/open?id=0BzPQ3WpemY1Va0J4dEhBbS02ems, 2017.

[2] 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.

[3] Azure data marketplace. http://www.infochimps.com/.

[4] M. Balazinska, B. Howe, and D. Suciu. Data markets in the cloud: An opportunity for the database community. In *VLDB*, 2011.

[5] M.-F. Balcan and A. Blum. Approximation algorithms and online mechanisms for item pricing. In *EC*, 2006.

[6] A. Blum and J. D. Hartline. Near-optimal online auctions. In *SODA*, 2005.

[7] A. Blum, V. Kumar, A. Rudra, and F. Wu. Online learning in online auctions. In *SODA*, 2003.

[8] 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.

[9] 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.

[10] T. M. Cover and J. A. Thomas. *Elements of information theory.* John Wiley & Sons, 2012.

[11] N. Cressie. *Statistics for spatial data.* John Wiley & Sons, 2015.

[12] Customlists. http://www.customlists.net/.

[13] Dataexchange. http://new.thedataexchange.com/.

[14] A. Deshpande, C. Guestrin, S. R. Madden, J. M. Hellerstein, and W. Hong. Model-driven data acquisition in sensor networks. In *VLDB*, 2004.

[15] 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.

[16] Factual. https://www.factual.com/.

[17] Gnip. https://gnip.com/.

[18] 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.

[19] V. Guruswami, J. D. Hartline, A. R. Karlin, D. Kempe, C. Kenyon, and F. McSherry. On profit-maximizing envy-free pricing. In *SODA*, 2005.

[20] Here. https://company.here.com/here/.

[21] Infochimps. http://www.infochimps.com/.

[22] Instagram. https://www.instagram.com/.

[23] M. Karaliopoulos, I. Koutsopoulos, and M. Titsias. First learn then earn: Optimizing mobile crowdsensing campaigns through data-driven user profiling. In *MobiHoc*, 2016.

[24] P. Koutris, P. Upadhyaya, M. Balazinska, B. Howe, and D. Suciu. Toward practical query pricing with querymarket. In *SIGMOD*, 2013.

[25] 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.

[26] 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.

[27] D. Monhor. A Chebyshev inequality for multivariate normal distribution. *Probability in the Engineering and Informational Sciences*, 21(02):289–300, 2007.

[28] Nasdaq. http://www.nasdaq.com/.

[29] D. J. Nott and W. T. Dunsmuir. Estimation of nonstationary spatial covariance structure. *Biometrika*, 89(4):819–829, 2002.

[30] A. Odlyzko. Paris metro pricing for the internet. In *EC*, 1999.

[31] L. Sun, R. Cheng, D. W. Cheung, and J. Cheng. Mining uncertain data with probabilistic guarantees. In *KDD*, 2010.

[32] Thingful. https://thingful.net/.

[33] Thingspeak. https://thingspeak.com/.

[34] V. Valancius, C. Lumezanu, N. Feamster, R. Johari, and V. V. Vazirani. How many tiers?: Pricing in the internet transit market. In *SIGCOMM*, 2011.

[35] 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.

[36] C. K. Williams and C. E. Rasmussen. *Gaussian Processes for Regression.* MIT, 1996.

[37] Xignite. http://www.xignite.com/.

[38] D. Yang, G. Xue, X. Fang, and J. Tang. Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing. In *MobiCom*, 2012.

[39] 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*, 2017. DOI: 10.1109/JSAC.2017.2659258.

[40] Z. Zheng, Z. Yang, F. Wu, and G. Chen. Mechanism design for mobile crowdsensing with execution uncertainty. In *ICDCS*, 2017.

[41] P. Zhou, Y. Zheng, and M. Li. How long to wait?: Predicting bus arrival time with mobile phone based participatory sensing. In *MobiSys*, 2012.