Analyzing Big Smart Metering Data Towards Differentiated User Services: A Sublinear Approach

Erte Pan, *Student Member, IEEE*, Dan Wang, *Senior Member, IEEE*, and Zhu Han, *Fellow, IEEE*

Abstract—With the advances of the information and communications technology, and smart meters in particular, fine grained user electricity usage of households is available for analyzing electricity usage behaviors. The information makes it possible for utility companies to provide differentiated user services from the time-of-use perspective, i.e., different pricing for users based upon when and how users consume power. In this paper, we present a methodology on differentiated user services based on extracted characteristic consumer load shapes (usage profiles as a function of time) from a large smart meter data set. We identify distinct user subgroups based upon their actual historic usage patterns, which are represented by the proposed electricity usage distributions. Since the big electricity user data cover millions of users and for each user the data are multi-dimensional and in fine-time granularity, we thus propose a sublinear algorithm to make the computation of the differentiated user service model efficient. The algorithm requests an input of only a small portion of users, and a sublinear amount of the electricity data from each of these selected users. We prove that the algorithm provides performance guarantees. Our simulated evaluation demonstrates the effectiveness of our algorithm and the differentiating user service model.

Index Terms—Smart meter, load profile, classification, sublinear algorithm

NOMENCLATURE

δ_1	confidence parameter for AlgoPercent()
δ_2	confidence parameter for DisTest()
ϵ_1	error bound parameter for AlgoPercent()
ϵ_2	error bound parameter for DisTest()
$\mathbf{p}^i$	expectation of $P_i\{\mathbf{x}\}$
$\mathcal{S}$	scale indicator set representing time instants
$f_m(\mathbf{p}^m)$	bill charge for typical mth type user
m_1	number of users to be sampled in AlgoPercent()
m_2	number of data points to be sampled in DisTest()
$P_i\{\mathbf{x}\}$	benchmark distribution for ith type of users
$z_{m,n}$	binary class indicator for nth user and mth type

1 INTRODUCTION

Facing the increasing concern on energy conservation all over the world, the power grids are currently undergoing substantial changes and upgrades. One important objective of the future grids is to provide a more customized electricity supply and pricing approach that is suitable for different types of users. In many current markets, electricity is charged only by the amount of electricity used; without considering the time pattern a user consumes electricity or discriminating the peak or off-peak hours. We call this current pricing a *fixed-price* service.

Such a fixed-price service has been adopted for decades. The development of smart meters makes it possible for the utility company to analyze users' behaviors, and therefore has the chance to offer load shape based pricing, referred as *differentiated user services*, i.e., pricing approaches that differ based upon when and how users consume electricity. With differentiated user services, the users can benefit from more choices to control their energy consumption and manage its cost. The utility company can also achieve better demand side management brought by these user behavior-oriented services and enjoy cost reduction when purchasing power in the peak hour from independent power providers in the wholesale power market.

Supported by the UH Electric Power Analytics Consortium, we investigate such differentiating user service model in this paper. We first conduct a trace study on user electricity usage. Multi-dimensional data of time and usage are collected by the smart meters installed for about 2 millions of households around Houston Area every 15 minutes for three years. Our initial analysis on a small portion of the data set reveals that users have different electricity usage patterns. This forms the basis showing that differentiated user services are possible. We also observe that the power usage behaviors of users are most represented by their electricity usage distributions. To characterize the electricity usage distributions of different set of users, we develop benchmark distributions. We then formulate a differentiating user service model for a utility company. The model establishes a theoretical profit computation for the utility

- *E. Pan and Z. Han are with the Department of Electrical and Computer Engineering, University of Houston, Houston, TX 77004. E-mail: {epan, zhan2}@uh.edu.*
- *D. Wang is with the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, KL, Hong Kong. E-mail: csdwang@comp.polyu.edu.hk.*

Manuscript received 9 Oct. 2015; revised 1 July 2016; accepted 7 Aug. 2016. Date of publication 17 Aug. 2016; date of current version 28 Oct. 2016. Recommended for acceptance by C. Phua. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TBDATA.2016.2599924

company. Such computation is based on the benchmark distributions and their associated different cost of power at peak versus non-peak times. The complexity comes from the big data side: for each user, there is a large amount of historical data; meanwhile, there are a numerous number of users in the real smart meter dataset. To efficiently solve the big data problem, we propose a sublinear algorithm for fast computation of the differentiating user service model. Our algorithm computes quality-guaranteed answers using a subset of data. More specifically, our algorithm only needs to input a small portion of users, and for each of these users, only a sublinear number of their electricity data. As a result, we substantially reduce the amount of data needed; and the computation of the differentiating user service model becomes possible. We prove that our computed results have performance guarantees. In particular, the results fall in the user defined error bounds with high confidence.

In summary, the main contributions of this paper are: we propose a differentiated user services model that computes the theoretical profit gain for the utility company based on different types of users; such differentiating user service model is based on analyzing a large amount of real world smart meter data. Successful implementation of the differentiating user service model faces a big data problem brought by the size of the underlying data set. We propose a novel algorithm which only processes a sublinear amount of data. We prove that our algorithm provides performance guarantees.

The remaining part of the paper proceeds as follows. In Section 2, we discuss the related work. In Section 3, we present a data trace study and clarify how we characterize the users by their electricity distributions. Section 4 is devoted to the differentiating user service model. We also show that the complexity to compute the model is non-trivial. We then develop a novel sublinear algorithm in Section 5 and prove its performance bounds. In Section 6, we evaluate our algorithms through simulation, and finally, we conclude our paper in Section 7.

2 Related Work

The emergence of smart meters [1] allows utility companies to understand the electricity usage of users in fine-granularity. Smart meters are usually integrated in the advanced metering infrastructure (AMI) [2] which also consolidates communications, software applications and data exchange interfaces. Ultimately, these become part of the smart grid networks [3], as illustrated in Fig. 1. There are good references, e.g., [4], on the problems and applications of smart grid [5].

Many studies investigate user behaviors and pricing strategies that deviate from the fixed price strategy. A fuzzy C-means clustering is investigated in [6] to disaggregate and learn energy consumption patterns from smart meter data. Day-ahead prices, user reactions and dynamic adjustments are studied in [7]. A model is developed in [8] to characterize the dynamic evolution of supply, demand, and market clearing prices under real-time pricing. A model with only one supplier and multiple users is studied in [9].

Big data is a popular studied topic recently [10]. The challenge comes from volume of the data and the variety of the data. The work in [11] describes a benchmark toolkit called IoTAbench for IoT Big Data scenarios. Authors in [12] examine smart meter analytics from a software performance perspective. Among many approaches, the sublinear algorithm [13] is a new paradigm to solve various big data problems. The essence of sublinear algorithm is to use a small portion of the data to compute results with guarantees. More specifically, the output of the sublinear algorithm is an approximation to the optimal result. As compared to approximation algorithms, which implicitly indicates that the approximation succeeds for 100 percent times, sublinear algorithms output an approximation with a $(1-\delta)$ percent (e.g., 95 percent) confidence to succeed. Such sacrifice in confidence makes sublinear possible.

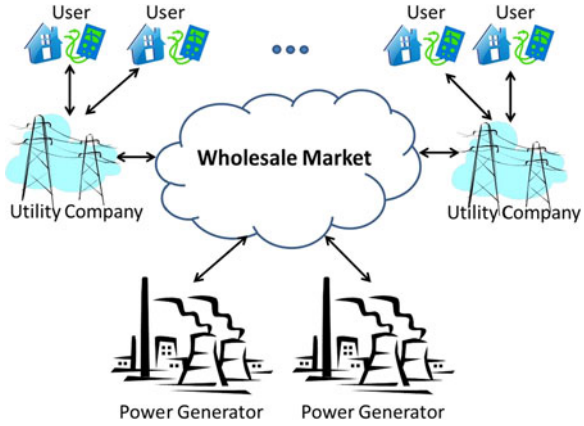

Fig. 1. Advanced metering infrastructure (AMI) system in smart grid.

Sublinear algorithms enjoy many studies from the theoretical point of view. Given a big data trace, sublinear algorithms have been developed to check the quantile of the data [14], whether the data stream is periodic, etc. One study related to our paper is [15], where sublinear algorithms are developed to check whether two distributions are close with certain confidence parameters. However, the algorithm is not suitable for our user classification task due to its inherent nature that the confidence parameters remain undetermined under some conditions. Hence, we propose our novel sublinear algorithm to overcome this drawback.

3 The User Electricity Usage Behavior and a Data Trace Study

In this section, we first propose a model to characterize user electricity usage patterns. We then use real user data to validate our model.

In this paper, we classify users according to their electricity usage distribution. A distribution in this paper is defined as a probability density function of a continuous/discrete random variable, which describes the likelihood for this random variable to take on a given value. We admit that there are many ways to characterize a user; for example, the total or average electricity he/she consumes in a month, the peak hour electricity usage in week days, etc. We believe the electricity usage distribution can more accurately characterize a user because a distribution provides a full spectrum of the user electricity usage.

Formally, let set $\mathcal{S}$ be an element set from the set $\mathcal{T}$ indicating time instants. In other words, $\mathcal{T}$ is the set of daily sampling frequency or sampling time mark at different scales. In this paper, $\mathcal{T} = \{t_1, t_2, t_3\}$ with $t_1 = \{0, 15, 30, \ldots, 1{,}425\,\text{min}\}$,

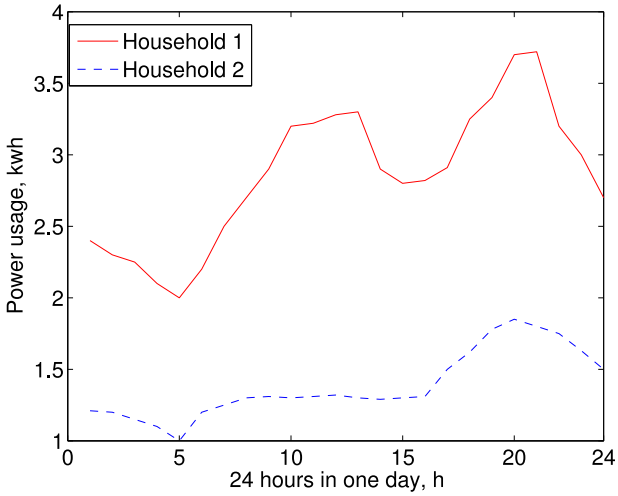

Fig. 2. Illustrative example of different average daily usage patterns of two benchmark distributions.

$t_2 = \{0, 1, \ldots, 23\,\text{hour}\}$ and $t_3 = \{1\,\text{day}\}$. Let $\mathbf{x}$ be a multi-dimensional random variable representing the daily electricity usage of a user, such that $\mathbf{x} \in \Re^{D(\mathcal{S})}$, $D(\mathcal{S}) \in \mathbb{Z}$ and $\mathcal{S} \in \mathcal{T}$, where $D(\mathcal{S})$ is the dimension of $\mathbf{x}$ at the scale $\mathcal{S}$ with $D(\mathcal{S}) = Card(\mathcal{S})$. $\mathcal{S}$ is called the scale indicator. Based on the daily usage pattern $\mathbf{x}$ of the user, we define the feature referred as "usage distribution" to characterize a user's behavior from the statistical point of view as the following:

Definition 1. *The electricity usage distribution $P\{\mathbf{x}\}$ at scale $\mathcal{S}$ is a distribution on the daily electricity usage $\mathbf{x}$, where $\mathbf{x} \in \Re^{D(\mathcal{S})}$.*

Given the historical recordings of one-dimensional random variable $\mathbf{x}$, the electricity usage distribution $P\{\mathbf{x}\}$ can be approximated using the empirical distribution/histogram. For multi-variate case where $D(\mathcal{S}) > 1$, $P\{\mathbf{x}\}$ can be estimated by assuming a certain structure (for instance Multi-variate Gaussian distribution) and fitting the parameters based on the dataset. From our trace data, we find that though the exact electricity usage of each household differs, many households share the same distribution. This becomes the basis for our classification. To represent the electricity usage distribution of each category, we choose to use a benchmark distribution. Formally, a benchmark distribution is defined as:

Definition 2. *A benchmark distribution is an electricity usage distribution $P\{\mathbf{x}\}$ which corresponds to the prototype of a group of users sharing similar electricity usage patterns. It has the expectation $\mathbf{p} = E\{\mathbf{x}\}$ with fixed values derived from real data statistics, satisfying $\mathbf{p} \in \Re^{D(\mathcal{S})}$, $D(\mathcal{S}) \in \mathbb{Z}$ and $\mathcal{S} \in \mathcal{T}$, such that each of $\mathbf{p}^i$, $i = 1, \ldots, D(\mathcal{S})$ is a fixed value.*

We now validate the electricity usage distribution, and the benchmark distribution. In Fig. 2, we show an illustration of different average daily usage patterns of two benchmark distributions at scale $S = t_2$. As can be seen, there are differences in peak hours and peak usage between the two benchmark distributions. In our real data analysis, it is also discovered that even though some users' average daily behaviors are similar (i.e., similar peak hours and peak usage), their usage distributions are quite different. At scale $\mathcal{S} = t_2 = \{0, 1, \ldots, 23\ \text{hour}\}$, through the histogram analysis on real data variate-wisely, it is revealed that each of the 24 variates conforms approximately to a Gaussian distribution. The users shared with similar average daily behaviors may end up with close Gaussian means but different variances in each dimension. All these demonstrate that using distribution to classify users is reasonable. In order to classify users by their electricity usage distributions, we take advantage of the benchmark distributions that are predefined by the utility company. We will detail the method in the next section.

4 Differentiated User Services: The Model and Big Data Challenge

4.1 An Overview

Many works nowadays study pricing strategies for utility companies [16], [17]. The overall model can be abstracted as follows. For the following discussion, let us assume that one objective of a power marketer is to maximize its potential profit from its customer base. The profit equals its revenue minus its cost. The revenue of a utility company depends on its *pricing strategy* (sometimes also called as pricing coefficient), i.e., the unit price for electricity. For a fixed price strategy, the unit price for a unit electricity at any time, for any household is fixed. Newly proposed pricing strategies have dynamic unit price according to different situations. For example, the pricing strategy in [16] is implemented as a quadratic cost function of energy usage at each hour. This kind of increasing, convex cost function is reported by the literature to be a suitable model for thermal generators; and the pricing strategy in [17] is expressed as piecewise linear functions whose pricing coefficients stay as different constant values in different time intervals within 24 hours.

In this paper, we introduce a differentiating user service model that seeks to classify customers based upon their characteristic usage as defined by historic usage distribution. To operationalize this model, the utility must 1) classify different users (by setting benchmark distributions); and 2) set different pricing coefficients for different users. Clearly, both factors influence the revenue of the utility company. Here, we try to limit the scope of our study where we assume that these two factors are given. These two factors can be determined by certain optimization criteria [18], gaming with other utility companies [19], or other external concerns of the utility company [1].

With these two factors, we present a differentiating user service model for the utility company to compute its revenue and profit in Section 4.2. We observe that such computation is itself non-trivial because it involves big data. We analyze the data complexity of this model in Section 4.3. In Section 5, we will develop sublinear algorithms that can efficiently conduct such computation.

4.2 The Differentiating User Service Model

We now formally present our differentiated user services model. Our differentiating user service model has three elements: 1) A set of benchmark distributions $\{P_1\{\mathbf{x}\}, P_2\{\mathbf{x}\}, \ldots, P_i\{\mathbf{x}\}\}, i = 1, \ldots, L$ with corresponding expectations $\{\mathbf{p}^1, \mathbf{p}^2, \ldots, \mathbf{p}^i\}$ where L is the total types of users. These expectations are used for the utility company to specify a pricing plan; 2) pricing coefficients: the unit pricing

rate ($dollar/kw \times h$) for different types of users; and 3) power cost coefficients: the utility company needs to pay for buying the electricity from the power plants. We denote the user type indicator as the binary scalar $z_{m,n}$: $z_{m,n} = 1$ if the nth user is in the mth category with benchmark distribution $P_m\{\mathbf{x}\}$ and expectation $\mathbf{p}^m$; $z_{m,n} = 0$ otherwise. The user type indicator $z_{m,n}$ can be obtained via classification based on the electricity usage distribution $P\{\mathbf{x}_m\}$ of the mth user and the given benchmark distributions, which will be elaborated in the next section. Assume now the user type indicators are known, we have the total bill gain G of the utility company as

$$G = \sum_{n=1}^{N} \sum_{m=1}^{L} z_{m,n} f_m(\mathbf{p}^m), \tag{1}$$

where variable N stands for the number of users and function $f_m(\mathbf{p}^m)$ stands for the bill charge for the typical mth category/type user. For simplicity and consistency, we will refer to the mth user type as type $\mathbf{p}^m$ for the rest of this paper.

To ease the presentation, we simplify our differentiating user service model and set $L = 2$, i.e., we only consider two types of users in this paper. For the $L > 2$ case, the differentiating user service model can be easily extended in a similar way. The total bill gain of the utility company is then

$$G = \sum_{n=1}^{N} (z_{1,n} f_1(\mathbf{p}^1) + z_{2,n} f_2(\mathbf{p}^2)). \tag{2}$$

To transform (2) into a concise form, we have

$$G = \alpha \cdot N \cdot f_1(\mathbf{p}^1) + \beta \cdot N \cdot f_2(\mathbf{p}^2), \tag{3}$$

where the coefficients α and β indicate the percentage of type $\mathbf{p}^1$ and type $\mathbf{p}^2$ users among the total population, respectively. As can be seen, coefficients α and β are functions of user type indicators $z_{m,n}$. Hence, α and β are dependent on the classification results based on the electricity usage distributions $P\{\mathbf{x}_m\}$ of each user and the given benchmark distributions.

Given two types of users, we accordingly model two types of pricing plans for the individual bill, referred as the flat plan and the dynamic plan, respectively

$$\text{flat plan: } \tilde{f}_F(\mathbf{p}^1) = c_f \sum_{i=1}^{D(S)} p_i^1, \tag{4}$$

$$\text{dynamic plan: } \tilde{f}_D(\mathbf{p}^2) = c_p \sum_{i \in \mathbf{P}} p_i^2 + c_o \sum_{j \in \mathbf{P}^c} p_j^2. \tag{5}$$

Vectors $\mathbf{p}^1$ and $\mathbf{p}^2$ are the associated expectations of benchmark distributions $P_1\{\mathbf{x}\}$ and $P_2\{\mathbf{x}\}$, respectively. The fixed coefficient c_f represents the pricing rate $dollar/kw \times h$ for the flat plan, and c_p and c_o regulate the pricing rate for the dynamic cluster where the peak usage and off-peak usage are charged, respectively. The set $\mathcal{P}$ is a set of peak-hour time mark at the scale $\mathcal{S}$ ($\mathcal{S} \neq t_3$) and $\mathcal{P}^c$ is the complement of $\mathcal{P}$ which satisfies $\mathcal{P}^c = \mathcal{S} - \mathcal{P}$.

Considering the fact that some $\mathbf{p}^2$ type users may actually prefer the flat plan or some $\mathbf{p}^1$ type users incline to accept the dynamic plan, we denote the fixed probability factor a_f as the probability of $\mathbf{p}^1$-type choosing the flat plan, and a_d as the probability of $\mathbf{p}^1$-type choosing the dynamic plan. Also, we denote b_f as the probability of $\mathbf{p}^2$-type choosing the flat plan and b_d as the probability of $\mathbf{p}^2$-type choosing the dynamic plan. There are many other approaches to model this kind of users' reaction or preference towards the provided various pricing schemes: for example, authors in [20] model the users' reaction as binary values conditioned on some constraints; in [7], the users' preference is expressed in the functional form with tunable parameters. Since modeling users' preference is not the focus of this paper, we use the fixed-real-valued parameters instead. Hence, the complete individual bill for a $\mathbf{p}^1$-type user is

$$f_1(\mathbf{p}^1) = a_f \tilde{f}_F(\mathbf{p}^1) + a_d \tilde{f}_D(\mathbf{p}^1), \quad a_f + a_d = 1. \tag{6}$$

The complete individual bill for a $\mathbf{p}^2$-type user is

$$f_2(\mathbf{p}^2) = b_f \tilde{f}_F(\mathbf{p}^2) + b_d \tilde{f}_D(\mathbf{p}^2), \quad b_f + b_d = 1. \tag{7}$$

We also investigate the expense that the utility company incurred on buying the power from the power plants. Since the electricity market charges differently at peak hours and off-peak hours, we model the expense as

$$E = \sum_{i=1}^{N} \left(a_p \sum_{j \in \mathcal{P}} x_{ij} + a_o \sum_{j \notin \mathcal{P}} x_{ij} \right). \tag{8}$$

The fixed coefficients a_p and a_o represent the pricing rates in the unit of $dollar/kw \times h$ corresponding to the peak usage and off-peak usage, respectively. The value x_{ij} indicates the electricity consumption of the ith input user data point at time dimension j. So far, we have fully developed the expression of the net profit $\hat{G}$ as

$$\begin{aligned} \hat{G} = G - E \; &= \alpha \cdot N \cdot f_1(\mathbf{p}^1) + \beta \cdot N \cdot f_2(\mathbf{p}^2) \\ &\quad - \sum_{i=1}^{N} \left(a_p \sum_{j \in \mathcal{P}} x_{ij} + a_o \sum_{j \notin \mathcal{P}} x_{ij} \right). \end{aligned} \tag{9}$$

As stated before in this section, the percentage coefficients α and β are the output results of user classification, which will be further elaborated in the next section. The rest parameters $\{\mathbf{p}^1, \mathbf{p}^2\}$ and $\theta = \{N, c_f, c_p, c_o, a_f, a_d, b_f, b_d, a_p, a_o\}$ are determined by the utility company.

4.3 Model Analysis and the Big Data Challenge

We now look into the computation of the differentiating user service model. The expected profit comes from two ends: 1) the expected number of users belonging to each groups, and 2) within each group, the expected number of users adopting differentiated user services and the expected number of users remaining in the fixed-price services. In our model, the percentage of different groups of users, α and β, is computed at the first step as an output of classifying users. The expected total profit of a utility company is then calculated with given pricing coefficients and power cost coefficients, the two key parameters in our model. We are particularly interested in the percentage values because they can be utilized for quick estimation of the bill income

TABLE 1
Examples of the Amount of Data Need to Be Processed in GB Unit

(m_u, m_d)	$(2 \times 10^5, 8{,}760)$	$(2 \times 10^6, 8{,}760)$	$(2 \times 10^6, 17{,}520)$
Data(GB)	14.016	140.16	280.32

without bothering to calculate each user's power consumption in the big data setting. Moreover, the percentage values serve as the feedback indicator from users. By comparing the percentage values of different years, the utility company can gather the feedback information on how the users adjust their usage behaviors so that the company may update the current pricing plans in the dynamic fashion.

To compute α and β, we need to classify users by comparing the electricity usage distribution of each user to the benchmark distribution. To simplify the notation for given two user types in total, we use binary variable z_i as the user type indicator for the ith user instead of previously used $z_{m,n}$. Hence, the computation is calculated as $\alpha = \frac{1}{N}\sum_{i=1}^{N} z_i$ and $\beta = \frac{1}{N}\sum_{i=1}^{N}(1 - z_i) = 1 - \alpha$. z_i is determined via user classification

$$z_i = \begin{cases} 1, & Dis(P\{\mathbf{x}_i\}, P_1\{\mathbf{x}_i\}) \leq Dis(P\{\mathbf{x}_i\}, P_2\{\mathbf{x}_i\}), \\ 0, & Dis(P\{\mathbf{x}_i\}, P_1\{\mathbf{x}_i\}) > Dis(P\{\mathbf{x}_i\}, P_2\{\mathbf{x}_i\}). \end{cases} \quad (10)$$

$P_1\{\mathbf{x}_i\}$ and $P_2\{\mathbf{x}_i\}$ represent the benchmark distributions of $\mathbf{p}^1$-type and $\mathbf{p}^2$-type, respectively. The function $Dis(\cdot)$ is the closeness measure between two distributions, and we choose the function $Dis(\cdot)$ to be the L-2 distance[1] [21] between the two given distributions. We are particularly interested in discovering the percentage for the reason that once the new pricing plans are offered to all the users, the utility company may learn the feedback of user reaction by analyzing the current percentage values and compare them to the historical ones. In this way, the percentage values provide the utility company with the guidance on adjusting pricing.

Note that a straightforward computation of (9) needs to evaluate each user; and for each user to compute the L_2 distance of his/her distribution and the benchmark distribution, the computation needs to access each data of the user. In Table 1, we show a few illustrative examples of the amount of data need to be processed, given the number of users m_u, and the number of data points m_d of numerical discrete electricity usage distribution each user has at scale $S = t_2 = \{0, 1, \ldots, 23 \text{ hour}\}$.

We would like to remind that, as discussed in Section 4.1, such computation needs to be executed many times if it is part of an optimization where different benchmark distributions, and different price coefficients are evaluated. We thus develop a much more efficient computation approach through a novel sublinear algorithm in next section.

5 The Problem and Sublinear Algorithms

5.1 The Problem and Algorithm Sketch

Our objective is to know the percentages of $\mathbf{p}^1$ and $\mathbf{p}^2$ so that we can compute the total income of the utility company. We have shown in the previous section that the complexity is high. We also note that the complexity comes from the big data collecting complexity, not the computational complexity. To this end, we propose a novel sublinear algorithm where we substantially reduce the amount of sampled data needed in computation and obtain quality outputs. We first formally define the algorithm quality we use in this paper.

Definition 3 Algorithm Quality. *We measure accuracy in terms of the absolute deviation of the computed answer $\hat{a}$ from the exact answer a. We assume that such deviation is less than ϵ. In addition, this deviation does not exceed in most cases; for example, only δ percent such deviation exceeds ϵ and δ is small. More precisely, we would like to have that $Pr[|\hat{a} - a| \geq \epsilon] \leq \delta$. Here we refer to ϵ as the accuracy parameter and δ as the confidence parameter.*

We now present the sketch of our algorithm development. Our objective is that given the quality parameters ϵ, δ, we use a subset of data to compute α, β and we guarantee the results are within ϵ, δ.

In our algorithm development, we split the output quality parameters ϵ and δ into ϵ_1, δ_1, and ϵ_2, δ_2. We first develop two sub algorithms AlgoPercent() and AlgoDist(), each of which is itself a sublinear algorithm. AlgoPercent() samples a subset of users, and for each user use his/her full electricity data. It guarantees ϵ_1, δ_1. AlgoDist() applies to a single user and sample a subset of his/her electricity data. It guarantees ϵ_2, δ_2. Finally we develop an overall algorithm for distributed service model computing AlgoDSMC() that call AlgoPercent() and AlgoDist() as sub functions. AlgoDSMC () guarantees ϵ and δ. In what follows, Section 5.2 develops AlgoPercent(), Section 5.3 develops AlgoDist(), and Section 5.4 develops AlgoDSMC().

5.2 Sublinear on Percentage

The objective is to use a small portion of users to determine the percentage of $\mathbf{p}^1$-type and $\mathbf{p}^2$-type users. Since our proposed algorithm does not require the information of all input users, we refer to this property as "sublinear on percentage". Our proposed algorithm is referred as AlgoPercent(), taking ϵ_1 and δ_1 as the parameter, and all the user data $\mathbf{X}$ as the input. ϵ_1 specifies an error bound for the output estimated $\hat{a}$, while δ_1 means indicates the confidence/probability of success that the error bound can be maintained. Instead of computing over the total N users, AlgoPercent() first sub-samples $m_1 > -\frac{\log \delta_1}{2\epsilon_1^2}$ users. The user type classification is then performed on each one of the m_1 users to obtain the percentage of $\mathbf{p}^1$ and $\mathbf{p}^2$ users, respectively. AlgoPercent() is summarized in the Algorithm 1.

Algorithm 1. $AlgoPercent(\mathbf{X}, \epsilon_1, \delta_1)$

Sub-sample m_1 out of N users.
Perform user classification and compute $\hat{a}$ as illustrated in (10).

Recall that z_i is the user type indicator defined in Section 4.3. We assume that z_i are independent and define $Y = \sum_{i=1}^{N} z_i$, where N is the total number of users. Assume the percentage of $\mathbf{p}^1$-user among the population is α. We have $\alpha = \frac{1}{N}E[Y]$. Since z_i are all independent, $E[z_i] = \alpha$. Let

1. Other distance measures can be chosen for the closeness measure $Dis(\cdot)$ as well. However, we choose L-2 distance for fast computation reason.

$\Lambda = \sum_{i=1}^{m_1} z_i$, where m_1 is the total number of users we sampled. Let $\overline{\Lambda} = \frac{1}{m_1}\Lambda$. The next proposition says that the expectation of the sampled set $\overline{\Lambda}$ is the same as the expectation of all set. Using these notations, we then have the following proposition associated with AlgoPercent(). It is straightforward to see that $\overline{\Lambda}$ is the unbiased estimator of α. Based on this, we show that $m_1 > -\frac{\log \delta_1}{2\epsilon_1^2}$ is the constraint that is required to maintain the ϵ_1 error bound.

Proposition 1. *Given ϵ_1, δ_1, to guarantee that we have a probability of $1-\delta_1$ success that the percentage of $\mathbf{p}^1$-type users will not deviate from the true α for more than ϵ_1, the number of users we need to sample must be at least $-\frac{\log \delta_1}{2\epsilon_1^2}$.*

Proof. By the Hoeffding Inequality, which provides an upper bound on the probability that the sum of random variables deviates from its expected value, we have

$$Pr[\overline{\Lambda} - E[\overline{\Lambda}] \leq -\epsilon_1] \leq e^{-2\epsilon_1^2 m_1},$$

$$Pr[\overline{\Lambda} - E[\overline{\Lambda}] \geq \epsilon_1] \leq e^{-2\epsilon_1^2 m_1},$$

$$Pr[|\overline{\Lambda} - E[\overline{\Lambda}]| > \epsilon_1] \leq e^{-2\epsilon_1^2 m_1}.$$

Therefore, recall that $\overline{\Lambda}$ is the unbiased estimator of α, we have

$$Pr[|\overline{\Lambda} - \alpha| > \epsilon_1] = Pr[|\overline{\Lambda} - E[\overline{\Lambda}]| > \epsilon_1] \leq e^{-2\epsilon_1^2 m_1}.$$

To make sure that $e^{-2\epsilon_1^2 m_1} < \delta_1$, we need to have $m_1 > -\frac{\log \delta_1}{2\epsilon_1^2}$. □

5.3 Sublinear on Distribution

In this section, we will introduce one existent sublinear algorithm, based on which our proposed modified sublinear method will be elaborated afterwards. Sublinear algorithms can be regarded as one branch of approximation algorithms with confidence guarantees. The term "sublinear" is traditionally interpreted as an algorithm runs in sublinear time or complexity. However, in this paper, we point out that "sublinear" can also be used to indicate the algorithm uses $o(N)$ samples in space, where N is the total number of input elements.

The objective is that for each user, we use a sublinear number of samples from its electricity data distribution to determine whether the user belongs to the category of $\mathbf{p}^1$ or $\mathbf{p}^2$. We call this algorithm AlgoDist(). The distribution of the user behavior is denoted as $P\{x\}$ which is a 24-dimensional distribution at the scale $S = t_2$. We provide the heuristic sampling method to deal with the 24-dimensional distributions. In details, the user distribution is compared with the benchmark distribution by AlgoDist() on one dimension at a time for totally 24 times (corresponding to 24 hours per day). If the distribution passes the closeness test by the AlgoDist() for at least 12 times, it is then regarded as close to the compared benchmark distribution. As a result, the user is classified as the compared type. Since there are two defined types of users, the proposed AlgoDist() only compares the testing user distribution with the $\mathbf{p}^1$-type benchmark. If the closeness test fails, the user is automatically classified as the other type.

The distribution of one dimensional random variable can be expressed as the histogram in the discretized fashion, with the bin size denoted as δ. We denote the set $\mathcal{SP} = \{1\delta, \ldots, n\delta\}$ as the sample space of the distribution. We assume that the separated distribution under the investigation is discrete distribution over the n elements in the set $\mathcal{SP}$. Moreover, the distribution is assumed to be represented by a vector $\mathbf{p} = (p_1, \ldots, p_n)$ where p_i is the probability of sampling the ith element in the set $\mathcal{SP}$. Given a benchmark distribution in the probability vector form as $\mathbf{q} = (q_1, \ldots, q_n)$, we want to test whether the distribution $\mathbf{p}$ is close enough to $\mathbf{q}$ in the L_2-distance. The traditional way is to compute the whole distribution in the L_2-distance. However, it suffers from the heavy computation which is unacceptable in the big data analysis. Inspired by [15], we propose a novel modified sublinear algorithm based on the original sublinear algorithm. The key idea of the original sublinear algorithm is that by using the sampling, we can repeatedly draw a much smaller portion of all the users. Within the smaller portion of users, each distribution of user behavior is again repeatedly sampled in a smaller portion of the entire distribution. This kind of sampling is also applied to the benchmark distribution at the same time. Two metrics are proposed to measure the closeness between the distributions: 1) the collision probability is defined as the probability that a sample from each of $\mathbf{p}$ and $\mathbf{q}$ yields the same element and is equal to $\mathbf{p} \times \mathbf{q}$; 2) the self-collision of $\mathbf{p}$ and that of $\mathbf{q}$ are defined similarly as $\mathbf{p} \times \mathbf{p}$ and $\mathbf{q} \times \mathbf{q}$, respectively. The complete $DistTest(\mathbf{p}, \mathbf{q}, m_2, \epsilon_2, \delta_2)$ is a realization from the original sublinear method [15] and summarized as in Algorithm 2 (the related proof can be found in [15]). Note that in our application, the error parameter ϵ_2 serves as the classification criteria: if the L_2-distance two testing distributions are within ϵ_2, the testing user is classified as the $\mathbf{p}^1$ type; otherwise, the testing user is classified into the $\mathbf{p}^2$ type.

From [15], it is proved that the error and confidence factors of DistTest() are guaranteed by the following theorem:

Theorem 1. *Given ϵ_2, δ_2 and distributions $\mathbf{p}$ and $\mathbf{q}$, the DistTest () of testing closeness passes with the probability at least $1-\delta_2$ if $||\mathbf{p}-\mathbf{q}|| \leq \epsilon_2/2$ while it passes with the probability less than δ_2 if $||\mathbf{p}-\mathbf{q}|| > \epsilon_2$. The running time of the DistTest() is $O(\epsilon_2^{-4} log(1/\delta_2))$.*

Algorithm 2. $DistTest(\mathbf{p}, \mathbf{q}, m_2, \epsilon_2, \delta_2)$ Based on L_2-Distance Test

for $i = 1, 2, \ldots, O(log(1/\delta_2))$ **do**
  Let F_p = a set of m_2 samples from $\mathbf{p}$.
  Let F_q = a set of m_2 samples from $\mathbf{q}$.
  Let r_p be the number of pairwise self-collisions in F_p.
  Let r_q be the number of pairwise self-collisions in F_q.
  Let Q_p = a set of m_2 samples from $\mathbf{p}$.
  Let Q_q = a set of m_2 samples from $\mathbf{q}$.
  Let s_{pq} be the number of collisions between Q_p and Q_q.
  Denote $r = \frac{2m_2}{m_2-1}(r_p + r_q)$.
  Denote $t = 2s_{pq}$.
  if $r - t > m_2^2\epsilon_2^2/2$ **then**
    then reject, i.e., consider the two distributions are different.
Reject if the majority of the iterations reject, accept otherwise.

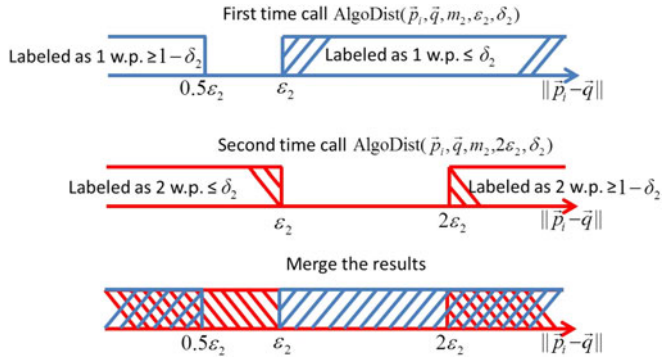

Fig. 3. Illustration of the underlying philosophy of designing Algorithm 3.

The main drawback of directly employing the original sublinear algorithm in classification is that the confidence of the classification output remains undetermined when the L_2-distance of the testing distribution $\mathbf{p}$ and the benchmark distribution $\mathbf{q}$ is truly in the interval $[\epsilon_2/2, \epsilon_2]$ according to Theorem 1. Based on the original sublinear algorithm $DistTest(\mathbf{p}, \mathbf{q}, m_2, \epsilon_2, \delta_2)$ as indicated in the Algorithm 2, we propose a novel modified sublinear algorithm to overcome this drawback and give the complete confidence estimates. Follow the previous assumptions, let $\mathbf{p}^i$ be the power usage distribution corresponding to the ith user, $\forall i = 1, 2, \ldots, N$. Suppose the classification by comparing a set of user electricity usage distributions $\{\mathbf{p}^i\}$ with $\mathbf{q}$ outputs two labels with the label 1 indicating user type (class) 1 whose $\mathbf{p}^i$ is close to $\mathbf{q}$; the label 2 indicating user type (class) 2 whose $\mathbf{p}^i$ is away to $\mathbf{q}$. We call the proposed modified sublinear algorithm $AlgoDist()$ and summarize it as in Algorithm 3. The underlying philosophy of designing Algorithm 3 is to call DistTest() twice but with different parameters ϵ_2 and $2\epsilon_2$, respectively, which help to get rid of the interval of undetermined confidence. Each time we call DistTest(), the classified labels of the output are retained partially. Both results are then merged with some treatment to the overlapped labels, in order to obtain a complete and consistent labeled result. The idea is illustrated in Fig. 3.

Algorithm 3. Modified Sublinear Algorithm $AlgoDist(\mathbf{p}^i, \mathbf{q}, m_2, \epsilon_2, \delta_2)$ Based on $DistTest()$

for $i = 1, 2, \ldots, N$ **do**

$Step1$: Employ $AlgoDist(\mathbf{p}^i, \mathbf{q}, m_2, \epsilon_2, \delta_2)$ and obtain the classification results as $\{LabelSet1\}$.

$Step2$: Employ $AlgoDist(\mathbf{p}^i, \mathbf{q}, m_2, 2\epsilon_2, \delta_2)$ and obtain the classification results as $\{LabelSet2\}$.

$Step3$: Keep the labeled 1 in $\{LabelSet1\}$ and reject all the labeled 2.

$Step4$: Keep the labeled 2 in $\{LabelSet2\}$ and reject all the labeled 1.

$Step5$: Combine the retained labels into $\{LabelSet3\}$; If the same user is both labeled as 1 in $\{LabelSet1\}$ and labeled as 2 in $\{LabelSet2\}$, his/her label is randomly determined as either 1 or 2 in $\{LabelSet3\}$.

$Step6$: Output $\{LabelSet3\}$ as the final classification results.

The algorithm accuracy of $AlgoDist()$, i.e., the classification accuracy, is given by the following Lemma 1:

Lemma 1. *Given ϵ_2, δ_2 and distributions $\{\mathbf{p}^i\}$ and $\mathbf{q}$, the AlgoDist() of classifying users is based on the L_2-distance criteria: label user as 1 if $||\mathbf{p}^i - \mathbf{q}|| \leq \epsilon_2$; label user as 2 if $||\mathbf{p}^i - \mathbf{q}|| > \epsilon_2$. And the classification accuracy is at least $1 - 2\delta_2$. In addition, Pr[labeled as 1|true 1] $\geq (1 - 2\delta_2)$ and Pr[labeled as 2|true 2] $\geq (1 - 2\delta_2)$.*

Lemma 1 essentially further develops the discussion in Theorem 1. In Theorem 1, the implicit underlying user group with $\epsilon_2/2 \leq ||\mathbf{p} - \mathbf{q}|| \leq \epsilon_2$ is not discussed. However, in Lemma 1, all users are taken into consideration. The proof of Lemma 1 is given in the appendix.

5.4 The Overall Algorithm

In this section, we derive the overall algorithm quality ϵ, δ of AlgoDSMC() and its sufficient condition. Since the overall objective is to estimate α, it is straightforward to see that the overall algorithm quality is $\epsilon = \epsilon_1$ and $\delta = \delta_1$. Given the parameters, $\epsilon_1, \delta_1, \epsilon_2, \delta_2$, for the sub-algorithms, ϵ_1 and δ_1 are passed to the AlgoPercent(ϵ_1, δ_1). Within the AlgoPercent (ϵ_1, δ_1), the small number of users, m_1 is sampled from the entire input users. For each one of the m_1 users, the AlgoDist($P\{\mathbf{x}_i\}$, $\mathbf{p}^1$, m_2, ϵ_2, δ_2) is called where $P\{\mathbf{x}\}$ corresponds to the distribution of the testing one user out of the m_1 users.

In AlgoPercent(ϵ_1, δ_1) with given error and confidence parameters, we compute that the sub-sampling number of users needs to be at least $m_1 = -\frac{\log \delta}{2\epsilon_1^2}$, under the assumption that AlgoDist($P\{\mathbf{x}_i\}$, $\mathbf{p}^1$, m_2, ϵ_2, δ_2) gives 100 percent correct classifications. However, given the error and confidence parameters ϵ_2, δ_2, the AlgoDist() again utilizes the sublinear algorithm and may misclassify a $\mathbf{p}^1$ user into $\mathbf{p}^2$-type. This means that AlgoDist() will not give 100 percent correct classifications, which nullifies the previous assumption made when analyzing AlgoPercent(ϵ_1, δ_1) due to the cascading relationship between AlgoPercent() and AlgoDist(). Then we have to modify the parameter settings of AlgoPercent() by taking the property of AlgoDist() into consideration. As a result, the subsample number m_1 has to be modified in order to maintain $\epsilon = \epsilon_1, \delta = \delta_1$.

We derive the constraint for the subsample number m, in order to make the probability of the failure of the overall algorithm, $Pr[|\hat{\alpha} - \alpha^*| \geq \epsilon]$, bounded by δ.

Theorem 2. *Given ϵ, δ for the overall algorithm quality, ϵ_2, δ_2 for AlgoDist() and suppose δ_2 is small enough such that $\epsilon > 6\delta_2$, to guarantee that we have a probability of $1 - \delta$ success that the percentage of $\mathbf{p}^1$-type users will not deviate from the true α for more than ϵ,* i.e., *$Pr[|\hat{\alpha} - \alpha| \geq \epsilon] = Pr[|\overline{\Lambda} - \alpha| \geq \epsilon] \leq \delta$, the number of users we need to sample must be at least $m \geq -\frac{\log \delta}{2(\epsilon - 6\delta_2)^2}$.*

Proof. Denote $p_{11} = Pr[x \in \omega_1 | x \in \psi_1]$ as the probability that user x is truly label 1 user and has been classified as label 1. Similarly, define $p_{21} = Pr[x \in \omega_1 | x \in \psi_2]$ as the probability that user x is truly label 2 user and has been classified as label 1. By Lemma 1, we have $1 - p_{11} \leq 2\delta_2$ and $p_{21} = 1 - p_{22} \leq 2\delta_2$. Follow the discussion in Section 5.2 and Lemma 1, we have

$$E[\overline{\Lambda}] = \frac{1}{N}\left(\sum_{i=1}^{N_1} p_{11} \cdot 1 + \sum_{j=1}^{N_2} p_{21} \cdot 1\right) = \alpha p_{11} + (1 - \alpha) p_{21}.$$

Therefore,

$$|\overline{\Lambda} - E[\overline{\Lambda}]| = |\overline{\Lambda} - \alpha + \alpha(1 - p_{11} + p_{21}) - p_{21}|,$$

$$|\overline{\Lambda} - E[\overline{\Lambda}]| \geq |\overline{\Lambda} - \alpha| - |\alpha(1 - p_{11} + p_{21})| - |p_{21}|,$$

$$|\overline{\Lambda} - E[\overline{\Lambda}]| \geq |\overline{\Lambda} - \alpha| - \alpha|1 - p_{11}| - \alpha|p_{21}| - |p_{21}|,$$

$$|\overline{\Lambda} - E[\overline{\Lambda}]| \geq |\overline{\Lambda} - \alpha| - \alpha \cdot 2\delta_2 - \alpha \cdot 2\delta_2 - 2\delta_2,$$

$$|\overline{\Lambda} - E[\overline{\Lambda}]| \geq |\overline{\Lambda} - \alpha| - 6\delta_2.$$

Hence, given $|\overline{\Lambda} - \alpha| \geq \epsilon$, it is sufficient to say that $|\overline{\Lambda} - E[\overline{\Lambda}]| \geq \epsilon - 6\delta_2$, which implies

$$Pr[|\overline{\Lambda} - \alpha| \geq \epsilon] \leq Pr[|\overline{\Lambda} - E[\overline{\Lambda}]| \geq \epsilon - 6\delta_2].$$

Denote $\tilde{\epsilon} =\geq \epsilon - 6\delta_2 > 0$, by the Hoeffding Inequality, we have

$$Pr[|\overline{\Lambda} - E[\overline{\Lambda}]| \geq \tilde{\epsilon}] \leq e^{-2\tilde{\epsilon}^2 m}.$$

To ensure that $e^{-2\tilde{\epsilon}^2 m} \leq \delta$, we need to have

$$m \geq -\frac{\log \delta}{2\tilde{\epsilon}^2}.$$

That is,

$$m \geq -\frac{\log \delta}{2(\epsilon - 6\delta_2)^2}.$$

If this holds and using $\overline{\Lambda}$ as the estimator of α, then

$$Pr[|\hat{\alpha} - \alpha| \geq \epsilon] = Pr[|\overline{\Lambda} - \alpha| \geq \epsilon] \leq \delta.$$

□

5.5 Extension to Multiple User Types

Given that there are many types of users in real residential consumption data, we address the extension of current methods on two user types to multiple user types that are larger than two classes in this section. The idea is similar to the one-versus-rest method widely utilized in multi-class classification problems. Recall that we perform user classification according to (10) for the two user types case. Given $L > 2$ as the number of total user types, we estimate the population proportion $\alpha_k, k = 1, \ldots, L$ for each one of the user types in turn. Accordingly, the classification rules in (10) have been modified as

$$z_k = \begin{cases} 1, & Dis(P\{\mathbf{x}_i\}, P_k\{\mathbf{x}_i\}) = \min_j Dis(P\{\mathbf{x}_i\}, P_j\{\mathbf{x}_i\}), \\ 0, & \text{otherwise}, \end{cases} \tag{11}$$

where j takes values as $j = 1, \ldots, L$. Through this manipulation, it is straightforward to see that the multi-type classification problem has been transformed in the original two user types problem (i.e., one user type as the current under testing kth user type and the other one as the combination of all the rest user types), which can be solved by the same method as previously described.

6 Evaluation

In this section, we evaluate our proposed differentiating user service model and associated algorithms. The evaluation is done in a desktop computer equipped with eight central processing units of Intel(R) Core(TM) i7-4770 CPU of 3.40 GHz. The software used for evaluation is Matlab. Due to the nondisclosure agreement, all results are computed from the simulated data that are generated according to the real data analysis. The proposed algorithms can be directly applied to real data without modification. Note that our sublinear algorithm has already provided a theoretical bound. We thus primarily investigate the relationship among the number of data we need to process, the errors and confidence interval. More specifically we evaluate:

1) The relations among the number of sub-samples m, m_2 and the error ϵ and confidence δ;
2) The proposed algorithm accurately estimates of α value within an acceptable error bound;
3) The differentiated user services model performs better than the traditional single rate fixed-price approach; In addition, analyzing the impact brought by the factor α and total user number N;
4) The proposed sublinear algorithm significantly reduces the computation load.

We evaluate the proposed differentiated user services model and the sublinear algorithm at scale $S = t_2 = \{0, 1, \ldots, 23 \text{ hour}\}$. The evaluation data set is simulated based on the real data analysis and similar to the generation procedure of the benchmark distributions. According to the data trace study in Section 3, we simulate the dataset based on Gaussian distributions. Specifically, we first generate the benchmark distribution of $\mathbf{p}^1$-type user similarly as the Household one plotted in Fig. 2 in Section 3. The total number of users is set to $N = 100{,}000$. α is varying from 0.1 to 0.8. The usage distribution of one $\mathbf{p}^1$ user is generated in this way: 1) each dimension of $\mathbf{p}^1$ (recall that $\mathbf{p}^1$ is the corresponding expectation of the benchmark distribution of $\mathbf{p}^1$-type, as defined in Section 4.2) is added with a random Gaussian noise drawn from $Gauss(0, 0.5)$, resulting in a noisy vector $\tilde{p}^1$; 2) generate a sequence of random variables $\tilde{k}_i^1, i = 1, \ldots, 24$ that conforms to the Gaussian distribution $Gauss(\tilde{p}_i^1, 0.1)$ respectively; 3) form the vector $\tilde{y} = (\tilde{k}_1^1, \ldots, \tilde{k}_{24}^1)$ as one instance of daily usage that belongs to the $\mathbf{p}^1$-type user; 4) repeat 2) and 3) for 365 times and obtain the set of vectors $\{\tilde{y}\}$ that represents the usage distribution of the $\mathbf{p}^1$ user. Finally, we repeat the procedure from 1) to 4) for $\alpha \times N$ times and obtain the data points of $\mathbf{p}^1$-type users. The $\mathbf{p}^2$-type users are simulated in the same fashion except that the Gaussian noise conforms to $Gauss(0, 0.1)$ and their user number is $\beta \times N$.

For objective 1), we use the data set generated with $N = 100{,}000$ and $\alpha = 0.7$. The parameters $\epsilon_1 = 0.05$, $\delta_1 = 0.05$, $\delta_2 = 0.005$ and $\epsilon_2 = 0.5$ are fixed. Then we vary m_2 for $m = 2{,}000, 3{,}000, 10{,}000, 15{,}000$. We define the estimation error as the absolute value of the difference between the estimated $\hat{\alpha}$ and the true α. By inputting the data set and the parameters into our proposed sublinear algorithm, we obtain the results shown in Fig. 4. As can be seen, as the sub-sample number m_2 grows larger, the estimation error

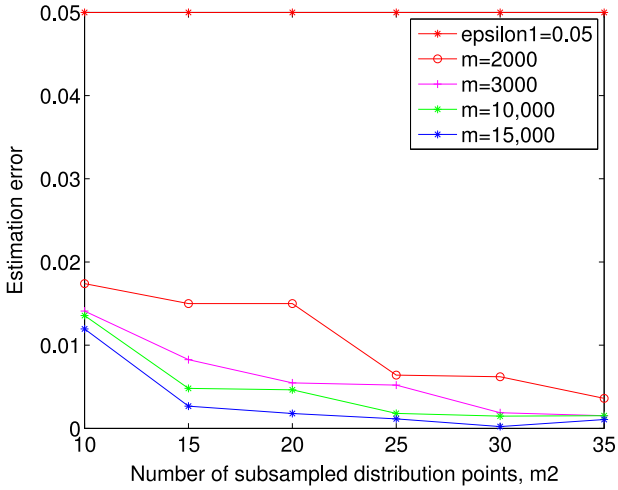

Fig. 4. Estimation errors $|\hat{\alpha} - \alpha|$ versus sub-sampling number m_2 from the entire distribution.

generally becomes smaller, which is consistent with the spirit of the sublinear methods: the more we sub-sample, the more precise results we will obtain. However, the speed of ameliorating the result to become closer to the true value is much slower than the increase speed of the required sub-samples: if we want to further reduce the error that is already small, we need to pay much more price, i.e., giving a much larger step of increments for m_2.

For objective 2), the total number of users is set to $N = 100,000$. α is varying from 0.1 to 0.8. We fix the parameters for the AlgoPercent() and AlgoDist() as: $\epsilon_1 = 0.05$, $\epsilon_2 = 0.5$, $\delta_1 = 0.05$, $\delta_2 = 0.005$, $m = 50,000$, $m_2 = 60$. Notice that under this parameter setting, the overall algorithm quality is $\epsilon = \epsilon_1 = 0.05$ and $\delta = \delta_1 = 0.05$. The data sets are then input into our overall algorithm. The estimated results are shown in Fig. 5. As can be seen, our algorithm estimates the α values precisely within the error bounds throughout all the simulated values. The maximum absolute error percentage is 1.42 percent, and the minimum absolute error percentage is 0.10 percent. And the subsamples used are only 50 percent of the total users. Moreover, we have tested our proposed L2 distance for closeness measurement against other possible solutions for closeness test, such as

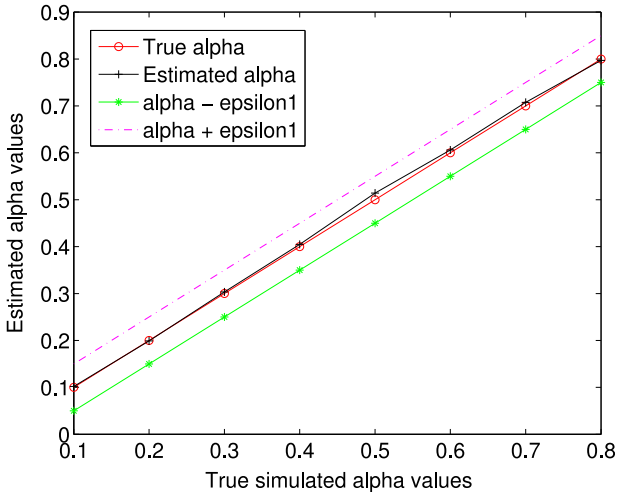

Fig. 5. Estimated α values versus simulated true α values.

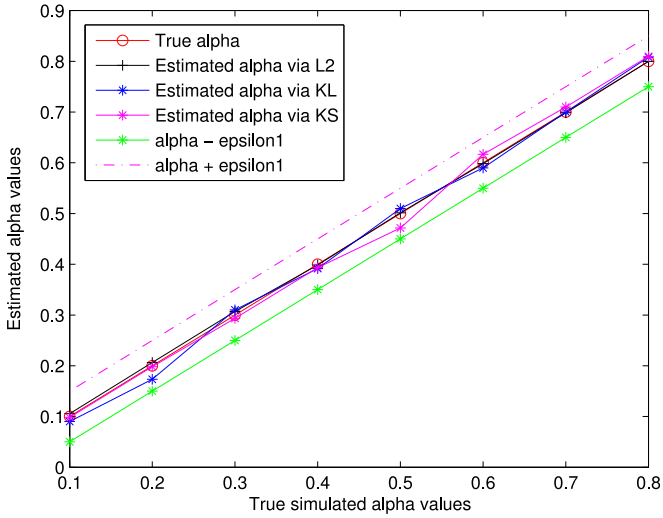

Fig. 6. Estimated α values versus simulated true α values via different closeness measurements: L2 distance, KL divergence, and KS tests.

the Kullback-Leibler (KL) divergence and Kolmogorov-Smirnoff (KS) tests. Using the same simulated dataset and the same parameters for the algorithms, we have obtained the results shown in Fig. 6. As can be seen, all the estimated α values using the three different measurements are within the error bounds, which demonstrates the suitability of using KL divergence and KS tests. However, it is worth to note that both KL divergence and KS tests require all input distribution data involved in the computation, while our proposed sublinear methods based on L2 distance only requires a small portion of the input data and hence, are more efficiency in the sense of computation.

In addition, we also test our proposed extended method for multiple user types with total number of user types $L = 3$. We set the proportions of 1st, 2nd and 3rd user types as α_1, α_2 and α_3, respectively. α_1 is varying from 0.1 to 0.8, α_2 is varying from 0.7 to 0, and α_3 is stayig 0.2. We simulate these three user populations and fix the same parameters ϵ_1, ϵ_2, δ_1, δ_2, m, and m_2 for the AlgoPercent() and AlgoDist() as used in two user types case. The estimated results are shown in Fig. 7. As can be seen, the estimated α values for all user types are within the error bounds and the absolute error is very small. This

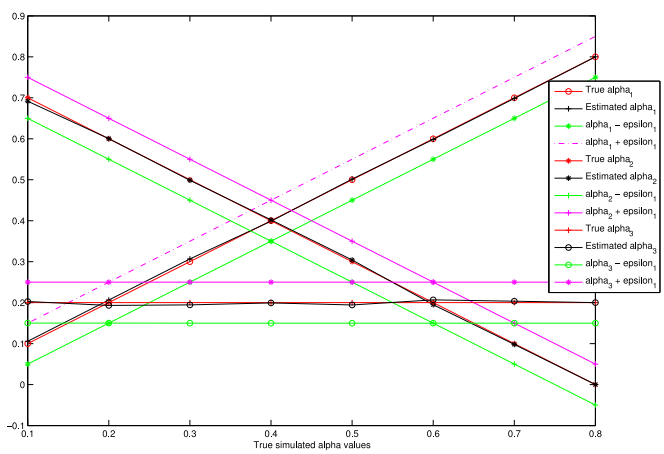

Fig. 7. Estimated α values versus simulated true α values for the three user types case.

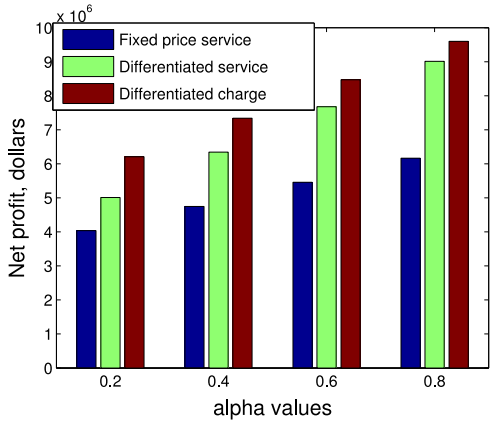

Fig. 8. Net profits from the differentiated user services versus net profits from non-differentiated user services with varying proportion of user types.

demonstrates the efficiency of our extension methods for multiple user types that are larger than 2.

For objective 3), we first investigate how the proportion of different types of users impact on the net profit. Using the data sets generated in the objective 1) with different α values, we set the parameters of the proposed differentiated user services model as: $N = 100,000$, $c_f = 1$, $c_p = 3$, $c_0 = 0.8$. $a_f = 0.1$, $a_d = 0.9$, $b_f = 0.8$, $b_d = 0.2$, $a_p = 2$ and $a_o = 0.5$. These parameters are chosen with a reference to the real electricity markets and bills. According to electricity usage patterns during the past years in Houston, the peak hour set is $\mathbf{P} = \{10, 11, 12, 18, 19, 20, 21 \text{ hour}\}$. The estimated α values are input into our differentiated user services model. To compare our model with the other two traditional pricing plans, we choose $c_f = 1$ to be the pricing factor that applies to all the users uniformly for the pricing plan referred as the fixed price service. We also choose $c_p = 3$ and $c_0 = 0.8$ for the plan that charges users uniformly but with varying price at peak and off-peak hours, referred as the differentiated charge. The experiment results are shown in Fig. 8, where three mechanisms of pricing are explained: (1) charging uniformly according to usage regardless of time/hour when the usage happens (referred as fixed price service); (2) charging according to usage but with different rates at peak versus off-peak hours (referred as differentiated charge); (3) charging according to different types of user (referred as our proposed differentiating user service). It can be seen that the proposed model favors over the $\mathbf{p}^1$-type users who generally use more power especially during the peak hours and bring more profits. The differentiated charge obtains the highest profits because it forces all the users to pay much more money at the peak hours, which is usually not suitable for the $\mathbf{p}^2$-type users. Fig. 8 indicates that: fixed price service is inefficient in the sense of static pricing; differentiated charge is inefficient in the sense of over charging (certain types of user would not accept this kind of service); differentiating user service is a reasonable pricing compared with the rest two. The profits and percentage analysis can be further utilized in the future to evolve into an advanced dynamic model with dynamic pricing factors, from which the reactions of users can be revealed.

TABLE 2
Comparison of the Amount of Data (GB Unit) Needed to Be Processed Between Direct Computation and Proposed Sublinear Algorithm with Different Parameter Settings of (m, m_2)

m, m_2	$m = N = 10^5$	(2,000, 10)	(2,000, 20)	(2,000, 35)
Data(GB)	7.008	0.041	0.081	0.142
m, m_2	(3,000, 10)	(3,000, 20)	(3,000, 30)	(2,000, 35)
Data(GB)	0.061	0.122	0.183	0.214
m, m_2	$(10^4, 10)$	$(10^4, 20)$	$(10^4, 30)$	$(10^4, 35)$
Data(GB)	0.204	0.407	0.610	0.712

TABLE 3
Minimum Amount of Data (GB Unit) Involved in the Computation of Proposed Sublinear Methods as a Function of $(\epsilon_1, \delta_1, \delta_2)$

$\epsilon_1, \delta_1, \delta_2$	(0.05,0.05,0.006)	$(0.05, 0.05, 0.001)$
Data(GB)	0.2402	0.0328
$\epsilon_1, \delta_1, \delta_2$	$(0.05, 0.005, 0.001)$	$(0.01, 0.05, 0.001)$
Data(GB)	0.0581	3.9732

For objective 4), we compare the computation load for direct computation with the proposed sublinear algorithm under some different parameter settings specified in the objective 1). Taking the repeating procedure in AlgoDist() into account, the overall amount of data needed to be processed by the proposed sublinear algorithm is expressed as $2 \times m \times m_2 \times 24 \times log(\frac{1}{\delta_2}) \times 8$, while the direct computation of entire data needs to process $N \times 365 \times 24 \times 8$. In the objective 3), $N = 100,000$, $\delta_2 = 0.005$, $m_2 = 10, 15, 20, 25, 30, 35$, and m varies as $m = 2{,}000, 3{,}000, 10{,}000, 15{,}000$. We render the numerical computation load in Table 2, where the second column corresponds to the direct computation and the 3rd to 6th columns correspond to the proposed algorithm with some specific parameter settings of (m, m_2). As can be seen from the table, the proposed sublinear algorithm greatly reduces the computation load at the price of acceptable estimation error on the percentage α as indicated in the objective 3). We also investigate the numerical computation load by considering the minimum amount of data required in the computation of proposed sublinear methods with varying parameters and given fixed input data. Suppose we have $N = 100,000$ users and therefore 7.008 GB data as input, we vary the parameters ϵ_1, δ_1, δ_2 and fix $\epsilon_2 = 0.5$ (ϵ_2 is problem-oriented and data-driven. In our application, it is set according to the distance between the two benchmark distributions) to see what is the minimum amount of data involved in the computation of proposed sublinear methods in order to guarantee performance. We render the numerical computation load in Table 3 as a function of parameters ϵ_1, δ_1 and δ_2. As can be seen in Table 3, the smaller error bounds and larger confidence impose more computation load for our algorithms. Moreover, the error bound parameter ϵ_1, which is highly related to the subsampled number of users m, is the major factor that influences the computation load, compared with the rest two factors. Another interesting observation can be found in the first row of Table 3 that even if the parameters change to have higher confidence, i.e., smaller δ_2, the required data to

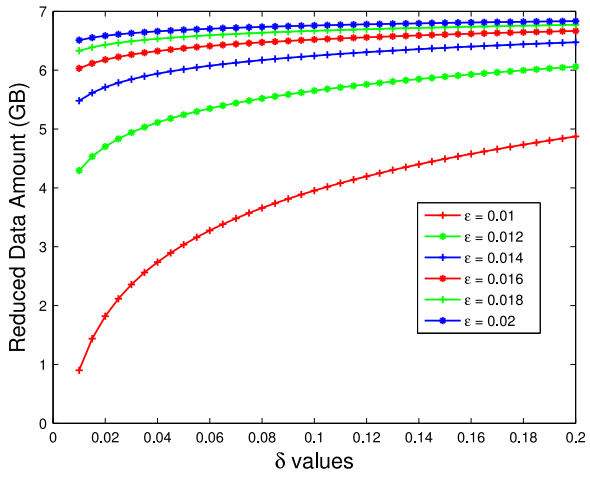

Fig. 9. Reduced data amount (DB) versus overall confidence parameter δ.

be processed is less. This is so because the smaller δ_2 indicates higher confidence of the success of the classification algorithm AlgoDist(). As a consequence, we no longer need to sample as many users as before to guarantee the overall algorithm, as indicated in Theorem 2. To further demonstrate the computational efficiency of the proposed sublinear methods, we also investigate the amount of data (GB) that is reduced by employing our algorithms instead of direct computation on the original 7.008 GB input data. First, we fix the parameters $\epsilon_2 = 0.5$ and $\delta_2 = 0.001$, and plot the reduced data in GB unit by varying the confidence parameter δ of overall algorithm (note that $\delta = \delta_1$ as discussed before). The results are shown as in Fig. 9 with different overall error bound ϵ. We then fix the internal parameters $\epsilon_2 = 0.5$ and $\delta_2 = 0.001$, and plot the reduced data in GB unit by varying the error bound parameter ϵ of overall algorithm (note that $\epsilon = \epsilon_1$ as discussed before). The results are shown as in Fig. 10 with different overall confidence ϵ. As can be seen from both Figs. 9 and 10, if the confidence or the error bound is relaxed, our proposed method can reduce larger amount of data involved in computation. Another discover is that the error bound will influence the reduced data more gently while the impact of the confidence parameter saturates fast as δ increases. This provides a guidance in practice that if we aim to reduce the computation load, there is no need to relax the confidence too much.

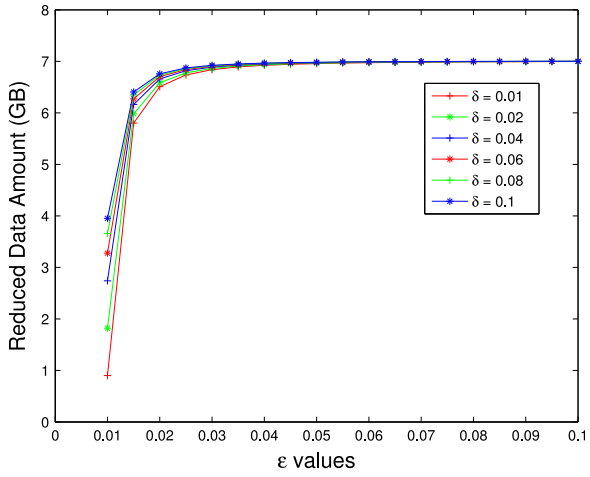

Fig. 10. Reduced data amount (DB) versus overall error bound parameter ϵ.

7 Conclusion

In this paper, we investigated a differentiating user service model for electricity usage. The model is based on an analysis of a real smart metering data trace where we observed that there exists various usage patterns among the power energy customers. One key problem of a differentiating user service model is that the model computation faces a huge amount of data. There is a large number of customers, and for each customer, his/her electricity usage pattern is represented by long period and multi-dimensional data. As a result, the complexity for a differentiate service model is not in the sense of computation, but in big data. We developed a novel sublinear algorithm where we use a sublinear amount of data and we guarantee a small error bounds and a given confidence. We demonstrated by both theoretical proofs and trace-driven evaluations that our algorithm can effectively reduce the amount of data to be processed to a range that is reasonable for the state-of-the-art computing capability.

Appendix A: Proof of Lemma 1

Proof. Assume there are N_1 users who are truly label 1 users, i.e., whose power usage distribution satisfies $||\mathbf{p}^i - \mathbf{q}|| \leq \epsilon_2$ and N_2 users who are truly label 2 users with $||\mathbf{p}^i - \mathbf{q}|| > \epsilon_2$. The total $N = N_1 + N_2$ users' power usage distributions $\{\mathbf{p}^i\}$ are then input to $AlgoDist()$ for classification. Denote $Pr[x \in \omega_1, x \in \psi_1]$ as the joint probability that user x is truly label 1 user (i.e., $x \in \psi_1$) and has been correctly classified as label 1 (i.e., $x \in \omega_1$) by AlgoDist(). Denote its power usage distribution as $\mathbf{p}$. According to Step 1 of Algorithm 2 and Theorem 1, we have

$$\begin{aligned} Pr[x \in \omega_1, x \in \psi_1] &= Pr[x \in \omega_1, ||\mathbf{p} - \mathbf{q}|| \leq \epsilon_2/2] \\ &+ Pr[x \in \omega_1, \epsilon_2/2 < ||\mathbf{p} - \mathbf{q}|| \leq \epsilon_2], \end{aligned} \tag{12}$$

$$\begin{aligned} Pr[x \in \omega_1, x \in \psi_1] &\geq 1 - \delta_2 \\ &+ Pr[x \in \omega_1, \epsilon_2/2 < ||\mathbf{p} - \mathbf{q}|| \leq \epsilon_2], \end{aligned} \tag{13}$$

$$Pr[x \in \omega_1, x \in \psi_1] \geq 1 - \delta_2. \tag{14}$$

Meanwhile, we have

$$Pr[x \in \omega_1, x \in \psi_2] \leq \delta_2. \tag{15}$$

Therefore, in the $\{LabelSet1\}$ produced by Step 3 of Algorithm 2, there are at least $N_1(1 - \delta_2)$ correctly labeled users and at most $N_2\delta_2$ falsely labeled users. Likewise, considering Step 2 of Algorithm 2 and Theorem 1, we have

$$Pr[x \in \omega_1, ||\mathbf{p} - \mathbf{q}|| \geq 2\epsilon_2] \leq \delta_2, \tag{16}$$

$$Pr[x \in \omega_2, ||\mathbf{p} - \mathbf{q}|| \geq 2\epsilon_2] \geq 1 - \delta_2, \tag{17}$$

$$\begin{aligned} &Pr[x \in \omega_2, ||\mathbf{p} - \mathbf{q}|| \geq 2\epsilon_2] \\ &+ Pr[x \in \omega_2, \epsilon_2 \leq ||\mathbf{p} - \mathbf{q}|| \leq 2\epsilon_2] \geq 1 - \delta_2, \end{aligned} \tag{18}$$

$$Pr[x \in \omega_2, ||\mathbf{p} - \mathbf{q}|| \geq \epsilon_2] \geq 1 - \delta_2, \tag{19}$$

$$Pr[x \in \omega_2, x \in \psi_2] \geq 1 - \delta_2. \tag{20}$$

Meanwhile, we have

$$Pr[x \in \omega_1, x \in \psi_1] \geq 1 - \delta_2, \tag{21}$$

$$Pr[x \in \omega_2, x \in \psi_1] \leq \delta_2. \tag{22}$$

Therefore, in the $\{LabelSet2\}$ produced by Step 4 of Algorithm 2, there are at least $N_2(1-\delta_2)$ correctly labeled users and at most $N_1\delta_2$ falsely labeled users.

Now consider situation of overlapped labeled users in Step 5 of Algorithm 2. Given that the value of δ_2 is usually very small, i.e., high confidence of the classification results, the worst case of the overlapped labeled users are: $N_1(1-\delta_2)$ correctly labeled-as-1 users in the $\{LabelSet1\}$ completely contain $N_1\delta_2$ falsely labeled-as-2 users in the $\{LabelSet2\}$; and $N_2(1-\delta_2)$ correctly labeled-as-2 users in the $\{LabelSet2\}$ completely contain $N_2\delta_2$ falsely labeled-as-1 users in the $\{LabelSet1\}$. To express this using the set notations, we define two sets that are obtained in the $\{LabelSet1\}$ produced by Step 3 of Algorithm 2

$$\Omega_{11} = \{x | x \in \psi_1, x \in \omega_1\}, \tag{23}$$

$$\Omega_{12} = \{x | x \in \psi_2, x \in \omega_1\}. \tag{24}$$

Then we have

$$\{LabelSet1\} = \Omega_{11} + \Omega_{12}, \tag{25}$$

$$Card(\Omega_{11}) \geq N_1(1-\delta_2), \tag{26}$$

$$Card(\Omega_{12}) \leq N_2\delta_2. \tag{27}$$

Define two sets that are obtained in the $\{LabelSet2\}$ produced by Step 4 of Algorithm 2

$$\Omega_{21} = \{x | x \in \psi_1, x \in \omega_2\}, \tag{28}$$

$$\Omega_{22} = \{x | x \in \psi_2, x \in \omega_2\}. \tag{29}$$

Then, we have

$$\{LabelSet2\} = \Omega_{21} + \Omega_{22}, \tag{30}$$

$$Card(\Omega_{21}) \leq N_1\delta_2, \tag{31}$$

$$Card(\Omega_{22}) \geq N_2(1-\delta_2). \tag{32}$$

Define

$$\Delta\Omega_1 = \{x | x \in \Omega_{21}, x \in \Omega_{11}\}, \tag{33}$$

$$\Delta\Omega_2 = \{x | x \in \Omega_{22}, x \in \Omega_{12}\}. \tag{34}$$

Define two sets that are obtained in the $\{LabelSet3\}$ produced by Step 5 of Algorithm 2

$$\Omega_1 = \{x | x \in \psi_1, x \in \omega_1\}, \tag{35}$$

$$\Omega_2 = \{x | x \in \psi_2, x \in \omega_2\}. \tag{36}$$

Then, according to the Algorithm 2, we have

$$\Omega_1 = \Omega_{11} - \Omega_{11} \bigcap \Omega_{21}, \tag{37}$$

$$\Omega_2 = \Omega_{22} - \Omega_{22} \bigcap \Omega_{12}. \tag{38}$$

The worst case happens when $\Omega_{21} \subseteq \Omega_{11}$ and $\Omega_{12} \subseteq \Omega_{22}$, and thus,

$$\Omega_{11} \bigcap \Omega_{21} = \Omega_{21}, \tag{39}$$

$$\Omega_{22} \bigcap \Omega_{12} = \Omega_{12}. \tag{40}$$

Therefore, we have

$$\begin{aligned} &\min Card(\Omega_1) = \min Card(\Omega_{11} - \Omega_{21}) \\ &= \min Card(\Omega_{11}) - \max(\Omega_{21}) \\ &= N_1(1-\delta_2) - N_1\delta_2 = N_1(1-2\delta_2). \end{aligned} \tag{41}$$

Similarly, we have

$$\min Card(\Omega_2) = N_2(1-2\delta_2). \tag{42}$$

Hence,

$$Pr[\text{labeled as } 1, \text{true } 1] \geq (1-2\delta_2), \tag{43}$$

$$Pr[\text{labeled as } 2, \text{true } 2] \geq (1-2\delta_2). \tag{44}$$

The worst case of correctly labeled users in the final result $\{LabelSet3\}$ are

$$\begin{aligned} &\min Card(\Omega_1) + \min Card(\Omega_2) \\ &\qquad = N_1(1-2\delta_2) + N_2(1-2\delta_2). \end{aligned} \tag{45}$$

In the worst case, the accuracy is

$$\begin{aligned} &\frac{\min Card(\Omega_1) + \min Card(\Omega_2)}{Card(\{LabelSet3\})} = \frac{(N_1+N_2)(1-2\delta_2)}{N} \\ &= 1 - 2\delta_2. \end{aligned} \tag{46}$$

So the classification accuracy is at least $1-2\delta_2$. □

ACKNOWLEDGMENTS

This work is supported through University of Houston Electric Power Analysis Consortium.

REFERENCES

[1] V. Gungor, "Smart grid technologies: Communication technologies and standards," *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 529–539, Nov. 2011.
[2] S. Chen, K. Xu, Z. Li, F. Yin, and H. Wang, "A privacy-aware communication scheme in advanced metering infrastructure (AMI) systems," in *Proc. IEEE Wireless Commun. Netw. Conf.*, Apr. 2013, pp. 1860–1863.
[3] S. Amin and B. Wollenberg, "Toward a smart grid: Power delivery for the 21st century," *IEEE Power Energy Mag.*, vol. 3, no. 5, pp. 34–41, Sep. 2005.
[4] H. Farhangi, "The path of the smart grid," *IEEE Power Energy Mag.*, vol. 8, no. 1, pp. 18–28, Jan. 2010.
[5] W. Shepherd and D. Shepherd, *Energy Studies*. River Edge, NJ, USA: World Scientific, 1998.

[6] V. Ford and A. Siraj, "Clustering of smart meter data for disaggregation," in *Proc. IEEE Global Conf. Signal Inf. Process.*, Dec. 2013, pp. 507–510.
[7] C. Joe-Wong, S. Sen, S. Ha, and M. Chiang, "Optimized day-ahead pricing for smart grids with device-specific scheduling flexibility," *IEEE J. Sel. Areas Commun.*, vol. 30, no. 6, pp. 1075–1085, Jul. 2012.
[8] M. Roozbehani, M. Dahleh, and S. Mitter, "Dynamic pricing and stabilization of supply and demand in modern electric power grids," presented at the 1st IEEE Int. Conf. Smart Grid Commun., Gaithersburg, MD, Oct. 2010.
[9] Q. Wang, M. Liu, and R. Jain, "Dynamic pricing of power in smart-grid networks," in *Proc. IEEE 51st Annu. Conf. Decision Control*, Dec. 2012, pp. 1099–1104.
[10] W. Fan and A. Bifet, "Mining big data: Current status, and forecast to the future," *SIGKDD Explorations Newslett.*, vol. 14, no. 2, pp. 1–5, Apr. 2013.
[11] M. Arlitt, M. Marwah, G. Bellala, A. Shah, J. Healey, and B. Vandiver, "IoTAbench: An internet of things analytics benchmark," in *Proc. 6th ACM/SPEC Int. Conf. Performance Eng.*, Jan. 2015, pp. 133–144.
[12] X. Liu, L. Golab, W. Golab, and I. F. Ilyas, "Benchmarking smart meter data analytics," in *Proc. Int. Conf. Extending Database Technol.*, Mar. 2015, Art. no. 385396.
[13] R. Rubinfeld and A. Shapira, "Sublinear time algorithms," *SIAM J. Discrete Mathematics*, vol. 25, no. 4, pp. 1562–1588, Feb. 2011.
[14] D. Wang, Y. Long, and F. Ergun, "A layered architecture for delay sensitive sensor networks," in *Proc. 2nd Annu. IEEE Commun. Soc. Conf. Sensor Ad Hoc Commun. Netw.*, Sep. 2005, pp. 24–34.
[15] T. Batu, L. Fortnow, R. Rubinfeld, W. Smith, and P. White, "Testing that distributions are close," in *Proc. 41st Annu. Symp. Found. Comput. Sci.*, Nov. 2000, pp. 259–269.
[16] A.-H. Mohsenian-Rad, V. Wong, J. Jatskevich, R. Schober, and A. Leon-Garcia, "Autonomous demand-side management based on game-theoretic energy consumption scheduling for the future smart grid," *IEEE Trans. Smart Grid*, vol. 1, no. 3, pp. 320–331, Dec. 2010.
[17] S. Shao, T. Zhang, M. Pipattanasomporn, and S. Rahman, "Impact of TOU rates on distribution load shapes in a smart grid with PHEV penetration," in *Proc. IEEE PES Transmission Distribution Conf. Exposition*, Apr. 2010, pp. 1–6.
[18] L. P. Qian, Y. Zhang, J. Huang, and Y. Wu, "Demand response management via real-time electricity price control in smart grids," *IEEE J. Sel. Areas. Commun.*, vol. 31, no. 7, pp. 1268–1280, Jul. 2013.
[19] S. Bu, F. Yu, and P. Liu, "Dynamic pricing for demand-side management in the smart grid," in *Proc. IEEE Online Conf. Green Commun.*, Sep. 2011, pp. 47–51.
[20] T. Kim and H. Poor, "Scheduling power consumption with price uncertainty," *IEEE Trans. Smart Grid*, vol. 2, no. 3, pp. 519–527, Sep. 2011.
[21] F. de Carvalho, P. Brito, and H.-H. Bock, "Dynamic clustering for interval data based on L2 distance," *Comput. Statistics*, vol. 21, no. 2, pp. 231–250, 2006.

Erte Pan (S'14) received the BE degree in electrical and computer engineering from Wuhan University, China, in 2010. He has been working toward the PhD degree in the Department of Electrical and Computer Engineering, University of Houston, since September 2011. He has been a research assistant in the Wireless Networking, Signal Processing and Security Lab, since June 2013. His research interests include nonparametric inference, deep learning networks, big data analysis, and sublinear methods on smart grid networks. He served as a peer reviewer of the *IEEE Transactions on Medical Imaging*, the *IEEE Transactions on Wireless Communications*, and the *IEEE Transactions on Smart Grid*. He is a student member of the IEEE.

Dan Wang (S'06-M'07-SM'13) received the BS degree from Peking University, Beijing, the MS degree from Case Western Reserve University, Cleveland, Ohio, and the PhD degree from Simon Fraser University, Vancouver, Canada, all in computer science. He is an associate professor in the Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong. His recent research interests include green computing, smart cities and big data. He is a senior member of the IEEE.

Zhu Han (S'01-M'04-SM'09-F'14) received the BS degree in electronic engineering from Tsinghua University, in 1997, and the MS and PhD degrees in electrical engineering from the University of Maryland, College Park, in 1999 and 2003, respectively. From 2000 to 2002, he was a R&D engineer with JDSU, Germantown, Maryland. From 2003 to 2006, he was a research associate with the University of Maryland. From 2006 to 2008, he was an assistant professor with Boise State University, Idaho. Currently, he is a professor in the Electrical and Computer Engineering Department as well as Computer Science Department, University of Houston, Texas. His research interests include wireless resource allocation and management, wireless communications and networking, game theory, wireless multimedia, security, and smart grid communication. He received an NSF Career Award in 2010, the Fred W. Ellersick Prize of the IEEE Communication Society in 2011, the EURASIP Best Paper Award for the *Journal on Advances in Signal Processing in 2015*, several best paper awards in IEEE conferences, and is currently an IEEE Communications Society distinguished lecturer. He is a fellow of the IEEE.