Towards Confidence Interval Estimation in Truth Discovery

Houping Xiao, *Student Member, IEEE*, Jing Gao, *Member, IEEE*, Qi Li, Fenglong Ma, Lu Su, *Member, IEEE*, Yunlong Feng, and Aidong Zhang, *Fellow, IEEE*

Abstract—The demand for automatic extraction of true information (i.e., truths) from conflicting multi-source data has soared recently. A variety of *truth discovery* methods have witnessed great successes via jointly estimating source reliability and truths. All existing truth discovery methods focus on providing a point estimator for each object's truth, but in many real-world applications, confidence interval estimation of truths is more desirable, since confidence interval contains richer information. To address this challenge, in this paper, we propose a novel truth discovery method (*ETCIBoot*) to construct confidence interval estimates as well as identify truths, where the bootstrapping techniques are nicely integrated into the truth discovery procedure. Due to the properties of bootstrapping, the estimators obtained by *ETCIBoot* are more accurate and robust compared with the state-of-the-art truth discovery approaches. The proposed framework is further adapted to deal with large-scale truth discovery task in distributed paradigm. Theoretically, we prove the asymptotical consistency of the confidence interval obtained by *ETCIBoot*. Experimentally, we demonstrate that *ETCIBoot* is not only effective in constructing confidence intervals but also able to obtain better truth estimates.

Index Terms—Truth discovery, confidence interval estimation, bootstrapping

1 Introduction

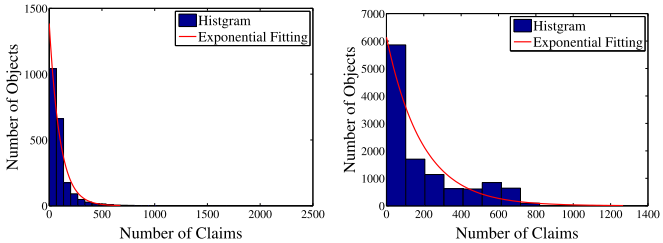
Today, we are living in a data-rich world, and the information on an object (e.g., population/weather/air quality of a particular city) is usually provided by multiple sources. Inevitably, there exist conflicts among the multi-source data due to a variety of reasons, such as background noise, hardware quality or malicious intent to manipulate data. An important question is how to identify the true information (i.e., truths) among the multiple conflicting pieces of information. Because of the volume issue, we cannot expect people to detect truth for each object manually. Thus, the demand for automatic extraction of truths from conflicting multi-source data has soared recently.

A commonly used multi-source aggregation strategy is averaging or voting. The main drawback of these approaches is that they treat the reliability of each source equally. In real-world applications, however, different sources may have different degrees of reliability and more importantly, their reliability degrees are usually unknown *a priori*. To address this problem, a variety of truth discovery methods [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18] have been proposed. Although these methods vary in many aspects, they share a common underlying principle: If a piece of information is provided by a reliable source, it is more likely to be trustworthy, and the source that more often provides trustworthy information is more reliable. Following this principle, existing methods are designed to jointly estimate source reliability and truths by assigning larger weights to the reliable sources.

All existing truth discovery methods [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [15], [16], [17], [19], [20], [21], [22], focus on providing a point estimator for each object's truth, i.e., the estimate is a single value. However, important confidence information is missing in this single-value estimate. For example, two objects A and B receive the same truth estimate, e.g., 25. Even though the estimates are the same, the confidence in these estimates could differ significantly–A may receive 1000 claims around 25 while B only receives one claim of 25. Clearly the confidence in A's truth estimate is much higher. Therefore, instead of a point estimation, an estimated confidence interval of the truth is more desirable. An α-level confidence interval [23] is an interval (a, b) such that $\mathbb{P}(\theta \in (a, b)) = \alpha$ for a given $\alpha \in (0, 1)$, where θ denotes the truth in our scenario. The width of the interval reflects the confidence in the estimate–A smaller interval indicates the higher confidence in the estimate and a larger interval means that the estimate has more possible choices within the interval. In the example we just mentioned, suppose the 95 percent confidence interval of A and B's estimates are (24.9,25.1) and (0, 50), respectively. Although both truth estimates are 25, we are more certain that A is close to 25. With such confidence information, the decision makers can use the truth estimates more wisely. However, such important confidence information cannot be obtained by the traditional point estimation strategy adopted by existing truth discovery methods.

- *H. Xiao, J. Gao, Q. Li, F. Ma, L. Su, and A. Zhang are with the State University of New York at Buffalo, 338 Davis Hall, Buffalo, NY 14260. E-mail: {houpingx, jing, qli22, fenglong, lusu, azhang}@buffalo.edu.*
- *Y. Feng is with the State University of New York at Albany, Albany, NY 12222. E-mail: ylfeng@albany.edu.*

Manuscript received 8 May 2017; revised 26 Apr. 2018; accepted 3 May 2018. Date of publication 15 May 2018; date of current version 4 Feb. 2019. (Corresponding author: Houping Xiao.) Recommended for acceptance by W. Wang. 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/TKDE.2018.2837026



(a) Flight Status Dataset (b) Game Dataset

Fig. 1. Visualization of the long-tail phenomenon.

The estimation of confidence intervals for objects' truths can benefit any truth discovery scenario by providing additional information (i.e., confidence) in the output, but its advantage is more obvious on long-tail data. A multi-source data is said to be long-tail in the sense that most objects receive a few claims from a small number of sources and only a few objects receive many claims from a large number of sources. As discussed in the aforementioned example, the difference in the confidence of the truth estimates is usually caused by the difference in the number of claims received by the objects. When an object receives more claims, a smaller confidence interval is obtained, and thus the estimate of this truth is more certain. It is essential to provide confidence intervals rather than points for the truth estimates on such long-tail data, which are ubiquitous. The Flight Status and Game applications used in our experiments are examples of such long-tail phenomena (The details are deferred to Section 4.3). In Fig. 1, we present the histograms in terms of the number of claims and fit them into an exponential distribution, a typical long-tail distribution, respectively.

To address the problem, in this paper, we propose a novel method, **E**stimating **T**ruth and **C**onfidence **I**nterval via **Boot**strapping (*ETCIBoot*) to construct confidence interval estimates for truth discovery tasks. We adopt the iterative two-step procedure used in traditional truth discovery methods: 1) Update truth estimates based on the current estimates of source weights (source reliability degrees), and 2) update source weights based on the current estimates of truths. At the truth computation step, instead of giving a point estimation, we now adopt the following procedure to obtain confidence interval estimates. *ETCIBoot* obtains multiple estimates of an object's truth, using bootstrapping techniques. Each estimate is obtained by calculating the weighted averaging or voting on a new set of sources which are bootstrapped from available sources. A statistic T that involves the truths is constructed. Its distribution F is usually unknown *a priori*. Based on these multiple estimates obtained via bootstrapping, we derive an estimator $\widehat{T}$ of T and further approximate F by $\widehat{F}$ (i.e., the distribution of $\widehat{T}$). The confidence intervals of the truths are naturally implied in the distribution of $\widehat{T}$ (i.e., $\widehat{F}$). Theoretically, we prove that $\widehat{T}$ is asymptotically consistent to T in distribution, and the end points of the confidence intervals converge to the true ones at $O_p(n^{-\frac{3}{2}})$, where n is the number of claims.

Besides providing confidence intervals, *ETCIBoot* is also able to provide more accurate and robust truth estimates if we use the average of the multiple estimates as the point estimator. Existing truth discovery methods typically compute weighted mean in the truth computation step, and thus the truth estimates can be quite sensitive to some outlying claims. In contrast, *ETCIBoot* adopts bootstrapping procedure to improve the robustness of estimation. The truth estimates are defined as the mean of bootstrap samples. These samples capture the distribution of claims in which the outlying claims' effect can be greatly reduced.

The proposed *ETCIBoot* is further extended to the distributed truth discovery paradigm to handle large-scale data. In many applications, data are distributed on multiple machines at different locations instead of storing at one single machine. The communication is usually expensive or even restricted between these machines, because the data volume is too large to store or process in one single machine, or the data cannot be shared among machines due to privacy concerns. To solve the truth discovery task in this scenario, we propose a two-step D-*ETCIBoot* algorithm: 1) We first bootstrap at every local machine and obtain an initialized truth estimate, and 2) we collect all the truth estimates and construct a new statistics $\widetilde{T}$ for confidence interval construction and truth estimation.

We conduct experiments on both simulated and real-world datasets. Experimental results show that the proposed *ETCIBoot* can effectively construct confidence intervals for all objects and achieve better truth estimates compared with the state-of-the-art truth discovery methods. We further compare the proposed D-*ETCIBoot* with *ETCIBoot* on all datasets in terms of the accuracy as well as efficiency.

To sum up, the paper makes the following contributions:

- To the best of our knowledge, we are the first to illustrate the importance of confidence interval estimation in *truth discovery*, and propose an effective method (*ETCIBoot*) to address the problem. The proposed *ETCIBoot* is further adapted to solve large-scale truth discovery task in the distributed scenario.
- Theoretically, we prove that the confidence interval obtained by *ETCIBoot* is asymptotically consistent.
- The point estimates obtained by *ETCIBoot* are more accurate and robust compared with existing approaches due to the properties of bootstrap sampling, which is nicely integrated into the truth discovery procedure in *ETCIBoot*.
- Experimental results show the effectiveness of *ETCIBoot* in constructing confidence intervals as well as identifying truths. Compared with *ETCIBoot*, D-*ETCIBoot* not only achieves comparable accuracy but also significantly speeds up truth discovery tasks.

2 Problem Setting

In this section, we first introduce terminologies and notations which will be used thoughout the paper. Then, the problem is formally defined.

Definition 1. *An object is an item of interest. Its true information is defined as a truth.*

Definition 2. *The reliability of a source measures the quality of its information. A source weight is proportional to its reliability, i.e., the higher the quality of a source's information, the larger its reliability, and the larger its weight. Typically, the source reliability or weight is unknown a priori.*

Problem Setting. Suppose that there are $\mathcal{S} := \{s\}_1^S$ sources, providing claims on objects $\mathcal{N} := \{n\}_1^N$, where an object

TABLE 1
Notations

Notation	Definition
$\mathcal{S}$	the set of sources
$\mathcal{N}$	the set of objects
x_n^s	the claim on object n made by source s
x_n^*	the true claim of the nth object
$\hat{x}_n$	the estimator of the claim for object n
ϵ_s	the s source's error
σ_s^2	the sth source's variance of claims
ω_s	the weight of source s
$\mathcal{S}_n$	the subset of sources available for object n
$\mathcal{N}_s$	the subset of objects claimed by source s
$\mathcal{X}_n$	the data set available for object n
$\mathcal{X}$	the whole data set for all objects

may receive claims from only a subset of $\mathcal{S}$. The truths of objects $\mathcal{N}$ are denoted as $\{x_n^*\}_{n\in\mathcal{N}}$, which are unknown *a priori*. For the object n, $\mathcal{S}_n$ is the set of sources which provide claims on it. The multi-source data for the nth object is denoted as $\mathcal{X}_n := \{x_n^s\}_{s\in\mathcal{S}_n}$, where x_n^s represents the claim provided by the sth source for the object n. The whole data collection on objects $\mathcal{N}$ is further denoted as $\mathcal{X} := \cup_{n=1}^{N}\mathcal{X}_n$.

For the sth source, we assume that the difference ϵ_s between its claims and truths follows a normal distribution with mean 0 and variance σ_s^2, i.e., $\epsilon_s \sim \text{Normal}(0, \sigma_s^2)$. This assumption is commonly used in existing truth discovery works [3], [4], [5]. ϵ_s captures the error of source s, and a small ϵ_s implies that the claims are close to the truths. σ_s^2 measures the quality of the claims provided by the sth source. We further denote the weight of source s as ω_s. Definition 2 implies that the larger σ_s^2, the smaller ω_s.

We summarize the notations in Table 2.

Truth Discovery Task. Truth discovery task is formally defined as follows: Given the multi-source data $\mathcal{X}$, the goal of a *truth discovery* approach is to obtain estimates $\hat{x}_n$ which are as close to x_n^* as possible ($\forall n \in \mathcal{N}$). Besides, for any $\alpha \in (0, 1)$, we can also provide an α-level two-sided confidence interval for the truth of each object.

Example 1. Table 2 shows a sample census dataset. In this particular example, an object is a state and a claim is a tuple in the table. Also, $\mathcal{N}$ = {NY,CA} and $\mathcal{S}$ = {Source i}$_{i=1}^{8}$. For instance, Source 1 claims that New York Sate has a population of 19.889 million in 2016, so it corresponds to $x_n^1 = 19.889$. It can be easily seen that the claims from different sources are conflicting. As there are no ground truths available in real applications, truth discovery methods have been proposed to extract an accurate answer from such conflicting information. Moreover, in this paper, we will also provide a confidence interval for each object. Namely, for the population of New York Sate, we will provide a 95 percent confidence interval, i.e., $\mathbb{P}(x_n^* \in (x_{lower}, x_{upper})) = 95$ percent. Such confidence intervals contain much more information than a single point estimation. For instance, we can provide a minimum or maximum population for a particular sate for decision makers.

TABLE 2
A Sample Census Data

Object	Source ID	Population(Million)
NY	Source 1	19.889
NY	Source 2	19.378
CA	Source 1	39.497
CA	Source 2	39.250
CA	Source 3	39.309
CA	Source 4	39.350
CA	Source 5	39.145
CA	Source 6	39.200
CA	Source 7	39.250
CA	Source 8	39.100

3 Methodology

In this section, we first review some preliminaries about *truth discovery* and *confidence interval* in Section 3.1. We then introduce two main components of *ETCIBoot*: a novel strategy for data aggregation (*ETBoot*) and a method for confidence interval construction (*CIC*) in Sections 3.2 and 3.3, respectively. The proposed *ETCIBoot* is further summarized in Section 3.4. Finally, we present the theoretical analysis of the confidence interval estimates obtained by *ETCIBoot* in Section 3.5.

3.1 Preliminary

3.1.1 Truth Discovery

The goal of a truth discovery task is to identify objects' truths (i.e., true information) from conflicting multi-source data. Many truth discovery methods have been proposed to estimate truths and weights iteratively. Details can be found in Section 5. We briefly review each step as follows.

Weight Update. Source weights play important roles in truth discovery. The underlying principle is that: If a source more often provides reliable information, it has a larger weight, and consequently this source contributes more in the truth estimation step discussed below. Based on this principle, various weight update strategies have been proposed. In this paper, we adopt the weight estimation introduced in [4]. A source weight is inversely proportional to its total difference from the estimated truth, that is,

$$\omega_s \propto \frac{\chi^2_{(\frac{\alpha}{2},|\mathcal{N}_s|)}}{\sum_{n\in\mathcal{N}_s}\left(x_n^s - \hat{x}_n\right)^2}, \tag{1}$$

where $\chi^2_{(\frac{\alpha}{2},|\mathcal{N}_s|)}$ is the $\frac{\alpha}{2}$th percentile of a χ^2-distribution with $|\mathcal{N}_s|$ degree. It is to capture the effect of the number of claims so that small sources get their weights reduced.

Truth Estimation. A commonly used strategy is weighted averaging for continuous data or weighted voting for categorical data, namely,

$$\hat{x}_n = \frac{\sum_{s\in\mathcal{S}_n}\omega_s x_n^s}{\sum_{s\in\mathcal{S}_n}\omega_s}, \text{or}, \hat{x}_n = \arg\max_x \frac{\sum_{s\in\mathcal{S}_n}\omega_s \mathbb{1}(x_n^s, x)}{\sum_{s\in\mathcal{S}_n}\omega_s}, \tag{2}$$

where $\mathbb{1}(x_n^s, x) = 1$ if $x_n^s = x$; otherwise it is 0. The weights are obtained at the *Weight Update* step; the truth estimated at this step will be used to update weights based on Eq. (1).

Providing proper initializations, *Weight Update* and *Truth Estimation* are iteratively executed until the convergence condition is satisfied.

3.1.2 Confidence Interval

Assume that an experiment has a sample set $X = \{x_i\}_{i=1}^{n}$ from $F_\mu(x)$, where F_μ is an accumulative density function

(c.d.f.) with a parameter μ. An α-level two-sided confidence interval for the parameter μ is defined as follows:

Definition 3. *For any $\alpha \in (0,1)$, $(\mu_{\boldsymbol{X},L}, \mu_{\boldsymbol{X},R})$ is called an α-level two-sided confidence interval of a parameter μ if it satisfies the following condition:*

$$\mathbb{P}\big(\mu \in (\mu_{\boldsymbol{X},L}, \mu_{\boldsymbol{X},R})\big) = \alpha. \tag{3}$$

The immediately preceding probability statement Eq. (3) can be read: Prior to the repeated independent trails of the random experiment, α is the probability that the random interval $(\mu_{\boldsymbol{X},L}, \mu_{\boldsymbol{X},R})$ includes the unknown parameter μ.

Given the distribution of the experiment sample set $\boldsymbol{X}$, the exact end points of a confidence interval is defined as:

Definition 4. *The exact end points of an α-level two-sided confidence interval of μ with a known c.d.f. F are:*

$$\begin{cases} \mu_{L,Exact} = \mu - \frac{\mathbb{V}\mathrm{ar}(\mu)}{\sqrt{n}} F^{-1}(1-\alpha), \\ \mu_{R,Exact} = \mu + \frac{\mathbb{V}\mathrm{ar}(\mu)}{\sqrt{n}} F^{-1}(\alpha); \end{cases}, \tag{4}$$

where $F^{-1}(\cdot)$ is the inverse function of c.d.f. F, $\mathbb{V}\mathrm{ar}(\mu)$ is the variance of μ, and n is the number of observed samples.

However, as F is unknown, Eq. (4) is always unknown *a prior*. The major task in this paper is to construct a confidence interval estimate for the truth, as well as identifying it.

3.2 *ETBoot* Strategy

In this part, we introduce a novel bootstrapping-based strategy for identifying truths in truth discovery. We term it as **E**stimating **T**ruth via **Boot**strapping (*ETBoot*). All existing *truth discovery* methods apply weighted averaging or voting using all sources' information. In contrast, *ETBoot* first bootstraps multiple sets of sources and then on each set of the bootstrapped sources it obtains a truth estimate based on Eq. (2). The final truth estimator is defined as the mean of these estimates. Due to the properties of bootstrapping, which are nicely integrated into the truth discovery procedure, *ETBoot* is more robust to the outlying claims and can achieve a better estimate of the truth. Moreover, given any $\alpha \in (0,1)$ *ETBoot* can also construct an α-level two-sided confidence interval of the estimated truth (i.e., Section 3.3).

The detailed procedure of *ETBoot* is as follows: For the nth object, it obtains B estimates of its truth, i.e., $\{\hat{x}_n^b\}_{b=1}^B$, where $\hat{x}_n^b$ is obtained by the following two-step procedure:

- Step 1: *Source Bootstrap.* At the bth iteration, we randomly sample a set of sources $\mathcal{S}_n^b$ from $\mathcal{S}_n$ with replacement such that $|\mathcal{S}_n^b| = |\mathcal{S}_n|$ in this step. The sampled data is denoted as $\boldsymbol{X}_n^b = \{x_s^n\}_{s \in \mathcal{S}_n^b}$.
- Step 2: *Truth Computation.* Based on the sampled data $\boldsymbol{X}_n^b = \{x_s^n\}_{s \in \mathcal{S}_n^b}$, $\hat{x}_n^b$ is calculated based on Eq. (2).

The final estimator $(\hat{x}_n^{Boot})$[1] for the nth object's truth is further defined as:

$$\hat{x}_n^{Boot} = \frac{1}{B}\sum_{b=1}^{B} \hat{x}_n^b. \tag{5}$$

1. We use $\cdot^{Boot}$ to represent the estimator obtained by Bootstrapping throughout the paper.

Compared with existing truth discovery methods which use Eq. (2), the proposed *ETBoot* combines results from multiple bootstrap samples instead of using all the sources at once. This enables *ETBoot* to obtain more robust estimates and confidence interval estimates that will be introduced in Section 3.3. The pseudo code of *ETBoot* for the nth object is summarized in Algorithm 1.

Algorithm 1. *ETBoot* on the nth Object

Input: $\mathcal{S}_n$, $\mathcal{X}_n$, $\{\omega_s\}_{s \in \mathcal{S}_n}$, and a parameter B
Output: Truth $\hat{x}_n^{Boot}$.
1: **for** the bth iteration ($b = 1, \cdots, B$) **do**
2: Bootstrap $\mathcal{S}_n^b$ from $\mathcal{S}_n$; extract $\boldsymbol{X}_n^b$ from $\mathcal{X}_n$ based on $\mathcal{S}_n^b$; calculate $\hat{x}_n^b$ according to Eq. (2);
3: **end for**
4: Calculate $\hat{x}_n^{Boot}$ according to Eq. (5).

3.3 Confidence Interval Construction

Next, we introduce the procedure of constructing an α-level two-sided confidence interval of an object's truth. We illustrate it for the nth object, and the remaining objects follow this procedure.

We denote the estimator we are interested in as $\hat{\theta}(\boldsymbol{X}_n)$ corresponding to the dataset $\boldsymbol{X}_n = \{x_n^s\}_{s \in \mathcal{S}_n}$. In our scenario, $\hat{\theta}(\boldsymbol{X}_n)$ denotes the truth estimate. For simplicity, we ignore the subscript $\cdot_n$ for $\boldsymbol{X}_n$. In a *truth discovery* task, the truth estimate is calculated as

$$\hat{\theta}(\boldsymbol{X}) = \frac{\sum_{s \in \mathcal{S}_n} \omega_s x^s}{\sum_{s \in \mathcal{S}_n} \omega_s}. \tag{6}$$

Note that $x_n^s \sim \mathrm{Normal}(x_n^*, \sigma_s^2)$ as $\epsilon_s \sim \mathrm{Normal}(0, \sigma_s^2)$ and $\epsilon_s = x_n^s - x_n^*$, which yields,

$$\mathbb{E}(\hat{\theta}(\boldsymbol{X})) = x_n^*, \quad \text{and,} \quad \mathbb{V}\mathrm{ar}(\hat{\theta}(\boldsymbol{X})) = \frac{\sum_{s \in \mathcal{S}_n} \omega_s^2 \sigma_s^2}{\big(\sum_{s \in \mathcal{S}_n} \omega_s\big)^2}. \tag{7}$$

The corresponding estimate of $\mathbb{V}\mathrm{ar}(\hat{\theta}(\boldsymbol{X}))$ is defined as

$$\widehat{\mathbb{V}\mathrm{ar}}(\hat{\theta}(\boldsymbol{X})) \triangleq \frac{\sum_{s \in \mathcal{S}_n} \omega_s^2 \hat{\sigma}_s^2}{\big(\sum_{s \in \mathcal{S}_n} \omega_s\big)^2}, \tag{8}$$

which is formulated by replacing the population variance with the sample variance. Here, $\hat{\sigma}_s^2 = \frac{\sum_{n \in \mathcal{N}_s} (x_n^s - \hat{x}_n^{Boot})^2}{N_s - 1}$, where $\hat{x}_n^{Boot}$ is obtained by *ETBoot* and $N_s = |\mathcal{N}_s|$. The idea to obtain a confidence interval of the truth x_n^* is that: We first construct a statistic T which is related to x_n^*, and then estimate the accumulated density function of $T \sim F(t)$. In our scenario, T is defined as follows:

$$T = \frac{\hat{\theta}(\boldsymbol{X}) - x_n^*}{[\widehat{\mathbb{V}\mathrm{ar}}(\hat{\theta}(\boldsymbol{X}))]^{\frac{1}{2}} / \sqrt{|\mathcal{S}_n|}}, \tag{9}$$

which measures the error between the truth x_n^* and its estimate $\hat{\theta}(\boldsymbol{X})$. The confidence interval of x_n^* is available once the distribution of T is determined. More precisely, let $T^{(\alpha)}$ indicate the $(100 \cdot \alpha)$th percentile of T, i.e., $\alpha = \int_{-\infty}^{T^{(\alpha)}} dF(t)$. Thus, we have that

$$\mathbb{P}\left(T^{(\alpha/2)} \leq \frac{\hat{\theta}(\boldsymbol{X}) - x_n^*}{[\widehat{\mathbb{V}\mathrm{ar}}(\hat{\theta}(\boldsymbol{X}))]^{\frac{1}{2}}/\sqrt{|\mathcal{S}_n|}} \leq T^{(1-\alpha/2)}\right) = \alpha. \quad (10)$$

Moreover, an α-level two-sided confidence interval of x_n^* is naturally implied in Eq. (10), that is,

$$\left(\hat{\theta}(\boldsymbol{X}) - \frac{T^{(1-\alpha/2)}\left[\widehat{\mathbb{V}\mathrm{ar}}(\hat{\theta}(\boldsymbol{X}))\right]^{\frac{1}{2}}}{\sqrt{|\mathcal{S}_n|}}, \hat{\theta}(\boldsymbol{X}) - \frac{T^{(\alpha/2)}\left[\widehat{\mathbb{V}\mathrm{ar}}(\hat{\theta}(\boldsymbol{X}))\right]^{\frac{1}{2}}}{\sqrt{|\mathcal{S}_n|}}\right). \quad (11)$$

Thus, the width of the confidence interval is proportional to $\frac{1}{\sqrt{|\mathcal{S}_n|}}$. It implies that if an object is claimed by more sources, then the width of its truth's confidence level is smaller, and vice versa. Especially, when the long-tail multi-source data is involved, this phenomenon is clearer.

However, as the T-percentile is usually unknown *a priori*, estimation of $T^{(\alpha)}$ is required. One commonly used strategy is bootstrap sampling [23], [24], [25], [26], [27]. Note that at the bth iteration of *ETBoot* (Algorithm 1), we have bootstrapped $\boldsymbol{X}_n^b$. Based on $\boldsymbol{X}_n^b$, we are able to calculate both $\hat{\theta}(\boldsymbol{X}_n^b)$ and $\widehat{\mathbb{V}\mathrm{ar}}(\boldsymbol{X}_n^b)$, yielding an estimator $\hat{T}_b$ for the statistic T, that is,

$$\widehat{T}_b = \frac{\hat{\theta}(\boldsymbol{X}_n^b) - \hat{\theta}(\boldsymbol{X}_n)}{[\widehat{\mathbb{V}\mathrm{ar}}(\hat{\theta}(\boldsymbol{X}_n^b))]^{\frac{1}{2}}/\sqrt{|\mathcal{S}_n|}}. \quad (12)$$

Moreover, the estimate of $T^{(\alpha)}$ is defined as follows:

$$\widehat{T}^{(\alpha)} = \sup\left\{t \in \{\widehat{T}_1, \cdots, \widehat{T}_B\} : \frac{\#(\widehat{T}_b \leq t)}{B} \leq \alpha\right\}. \quad (13)$$

Eq. (13) provides estimates of Eq. (11). Thus, the estimate of an α-level two-sided confidence interval is defined as follows:

$$\left(\hat{\theta}(\boldsymbol{X}) - \frac{\widehat{T}^{(1-\alpha/2)}\left[\widehat{\mathbb{V}\mathrm{ar}}(\hat{\theta}(\boldsymbol{X}))\right]^{\frac{1}{2}}}{\sqrt{|\mathcal{S}_n|}}, \hat{\theta}(\boldsymbol{X}) - \frac{\widehat{T}^{(\alpha/2)}\left[\widehat{\mathbb{V}\mathrm{ar}}(\hat{\theta}(\boldsymbol{X}))\right]^{\frac{1}{2}}}{\sqrt{|\mathcal{S}_n|}}\right). \quad (14)$$

We summarize the procedure of constructing confidence intervals as *CIC*, i.e., **C**onfidence **I**nterval **C**onstruction. Its pseudo is presented in Algorithm 2 for the nth object.

Algorithm 2. *CIC*

Input: $\{\boldsymbol{X}_n^b\}_{b=1}^B$, $\hat{x}_n^{Boot}$, and a confidence level α.
Output: Endpoints of the α-level two-sided CI
1: Calculate $\hat{\sigma}_s^2$ for $s \in \mathcal{S}_n$;
2: **for** the iteration b ($b = 1, \cdots, B$) **do**
3: Calculate $\widehat{\mathbb{V}\mathrm{ar}}(\hat{\theta}(\boldsymbol{X}_n^b))$ and $\widehat{T}_b$ according to Eq. (12);
4: **end for**
5: Choose $\widehat{T}(1-\alpha/2)$ and $\widehat{T}(\alpha/2)$ according to Eq. (13);
6: Calculate endpoints based on Eq. (14).

3.4 *ETCIBoot* Algorithm

So far, we introduce the update for source weights (i.e., Eq. (1)), a new truth estimation strategy, *ETBoot*, and the construction of confidence intervals for truths via *CIC*. Combining them together, we propose a novel truth discovery approach, **E**stimating **T**ruth and **C**onfidence **I**nterval via **Boot**strapping (*ETCIBoot*), to automatically construct confidence intervals as well as identify objects' truths. The main component of the proposed *ETCIBoot* consists of the following three steps:

- (i) *Weight Update*. Given initialization of truth $\{x_n^0\}_n^N$, source weights are updated based on Eq. (1).
- (ii) *Truth Estimation*. With source weights computed from (i), for each object n, we obtain truth estimators via *ETBoot* to obtain $\hat{x}_n^{Boot}$ associated with $\{\boldsymbol{X}_n^b\}_{b=1}^B$.
- (iii) *Confidence Interval Construction* (CIC). We estimate confidence intervals for all objects' truths.

The above steps are executed iteratively until no truth estimates change anymore. The pseudo code of the proposed *ETCIBoot* algorithm is shown in Algorithm 3. We conduct the proposed algorithm on the toy example, i.e., Table 2. The results are shown in the following table.

TABLE 3
Example on Calculating Confidence Interval

Object ID	# of Claims	$\hat{x}_n$	Confidence Interval (95%)
NY	2	19.480	(19.278, 19.480)
CA	8	39.297	(39.186, 39.297)

The value of a city's population is in millions.

Algorithm 3. *ETCIBoot*

Input: $\mathcal{X}$, and hyperparameters α and B.
Output: Truths $\{\hat{x}_n^{Boot}\}_1^N$ and CIs $\{CI_n(\alpha)\}_1^N$.
1: Initialize truths $x_1^{*,0}, \cdots, x_N^{*,0}$ as average;
2: **while** the convergence condition is not satisfied **do**
3: Compute ω_s for each source s according to Eq. (1);
4: **for** each object n ($n = 1, \cdots, N$) **do**
5: Conduct *ETBoot* to obtain $\hat{x}_n^{Boot}$;
6: Calculate the confidence interval $CI_n(\alpha)$ via *CIC*;
7: **end for**
8: **end while**

Example 2. Based on Example 1, we compute truth estimates and 95 percent-level two-sided confidence intervals, and show the results in Table 3. All the values in this example are in millions. For the object NY, the width of the confidence interval is .3032 which is two times wider than that of the object CA (i.e., .1368). We also define the Relative Width (i.e., rw) as $rw((a,b)) = \frac{2|a-b|}{|a|+|b|}$ to compare the effectiveness of confidence intervals. Then, rw of the object NY is .0156 which is 4.5 times wider than that of the object CA (i.e., .0035). According to these confidence intervals, we can say that the population of CA and NY are $39.186 \sim 39.297$ and $19.278 \sim 19.480$, respectively. As the width of $39.186 \sim 39.297$ is narrower, we have more confidence to obtain an accurate estimate of the CA's population compared with the object NY. Therefore, the more claims provided for an object (i.e., object CA in Table 2), the narrower the width or the relative width of this object, the more confidence for us to obtain an accurate truth estimate. Especially, the object NY only receives 2 claims. Note that, all claims in Table 2 are not largely different from each other. If these two claims largely deviated from each other, the width of the confidence interval becomes even larger. We do not adopt this trivial scenario in the toy example of confidence interval estimation. Instead, we show that our proposed method can also handle the

case where claims are close to each other. Furthermore, we want to use this example to demonstrate that the number of claims which is provided for objects indeed affects our confidence about the final truth estimates.

3.5 Theoretical Analysis

In this subsection, we present the theoretical analysis on the confidence interval estimates, i.e., Eq. (14), obtained via *ETCIBoot*. We first prove that $\widehat{T}$ converges to T in distribution and present it in Proposition 1.

Proposition 1. *Assume that* $x_n^s \sim \mathcal{N}(x_n^*, \sigma_s^2)$, $\forall s \in \mathcal{S}_n$. *Let* T *and* T^* *be defined as Eqs. (9) and (12), respectively. Then, we have*

$$\lim_{|\mathcal{S}_n|\to\infty} \|\mathbb{P}^*(\widehat{T} \le t) - \mathbb{P}(T \le t)\| = 0, \quad \text{a.s.}, \tag{15}$$

where $\mathbb{P}^*$ *is the probability calculated based on the bootstrapping sample distribution,* $|\mathcal{S}_n|$ *is the Cardinality of* $\mathcal{S}_n$, t *is any real number, and a.s. means 'almost surely'.*

Proof. See Appendix A, which can be found on the Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/TKDE.2018.2837026, for a detailed proof. □

Proposition 1 is a straightforward result from Theorem 1 in [25], where the author provides sufficient conditions to guarantee the convergence of the bootstrapping samples. Thus, the proof of Proposition 1 is to testify whether the *ETCIBoot* satisfies these sufficient conditions, as shown in Appendix A in the online supplement. Proposition 1 shows that the bootstrapping estimator $\widehat{T}$ converges to T in distribution. It enables us to use the bootstrapping distribution to approximate the unknown distribution F for confidence interval construction.

Next, in Proposition 2, we show that the upper end point of an α-level one-sided confidence interval obtained via *ETCIBoot* is close to that from the theoretical distribution.

Proposition 2. *Given* $T \sim F(x)$, $\widehat{T} \sim \widehat{F}(x)$ *and a dataset* $\boldsymbol{X}$, *we have that*

$$\hat{\theta}_{\widehat{T},\boldsymbol{X}}(\alpha) = \hat{\theta}_{T,\boldsymbol{X}}(\alpha) + O_p(n^{-3/2}), \tag{16}$$

where $\mathbb{P}^*(\theta(\boldsymbol{X}) \le \hat{\theta}_{\widehat{T},\boldsymbol{X}}(\alpha)) = \alpha$, $\mathbb{P}(\theta(\boldsymbol{X}) \le \hat{\theta}_{T,\boldsymbol{X}}(\alpha)) = \alpha$, $n = |\boldsymbol{X}|$, *and* O_p *means the order holds in probability.*

Proof. See Appendix B in the online supplement for a detailed proof. □

Proposition 2 shows that the endpoint of an α-level one-sided confidence interval obtained by bootstrapping $\widehat{T}$ is close to that obtained by T, provided that there are enough samples. As any α-level two-sided confidence interval can be obtained by two one-sided confidence intervals, the results (Eq. (16)) also hold for Eq. (14). In truth discovery tasks, *ETCIBoot* is able to provide more accurate confidence intervals for the objects' truths, if they receive more claims. This result is more obvious especially on long-tail data.

3.6 Distributed *ETCIBoot* Method

Modern truth discovery applications increasingly involve massive datasets. More specifically, in many real applications, the data is distributed into multiple machines at different locations, between which communication is expensive or restricted. The possible reasons can be either because the data volume is too large to store or process on one single machine, or the data cannot be shared among machines due to privacy concerns, such as healthcare and mobile sensor-sensing applications. We adapt the proposed *ETCIBoot* framework to a distributed paradigm to handle the large-scale truth discovery task. We name the distributed truth discovery algorithm as D-*ETCIBoot*, that is, Distributed Estimation of Truth and Confidence Interval via Bootstrapping. Next, we first illustrate the distributed truth discovery framework and then introduce the details of the proposed D-*ETCIBoot* algorithm.

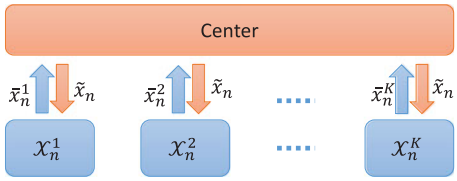


Fig. 2. Illustration of the distributed bootstrapping strategy.

Distributed Truth Discovery Framework. In our scenario, we assume that there are K local machines, over which S sources are either evenly or unevenly distributed. Besides, there is a *Center* (central server) which can be used to calculate the final truth estimates and construct confidence intervals. Take the truth computation of the object n for example. Every local machine has some sources which provide claims on it. We denote the index set of sources within the kth local machine as $\mathcal{S}_{kn}$, i.e., $\mathcal{S}_n = \bigcup_{k=1}^{K} \mathcal{S}_{kn}$. We further denote the claims made by these available sources from the kth local machine as $\mathcal{X}_n^k$, that is, $\mathcal{X}_n^k = \{x_n^s\}_{s\in\mathcal{S}_{kn}}$. In the centralized algorithm, we bootstrap samples from the whole data at once. In contrast, D-*ETCIBoot* adopts a two-step procedure: 1) bootstrap samples from each local machine for initialized truth estimation which will be sent to the center, and 2) calculate the final truth estimates as well as their confidence intervals at the center. More specifically, the main components of the proposed D-*ETCIBoot* consist of the following two steps:

- (i) *Bootstrapping at Local Machines.* At this step, every local machine sends an initialized truth estimate and its variance, calculated via bootstrapping technique.
- (ii) *Truth Estimation and Confidence Interval Construction.* The center collects all the initialized truth estimates and their variances, and calculates the final truth estimators and their corresponding α-level two-sided confidence intervals.

The above steps are executed iteratively until no truth estimates change any more. An illustration obtaining a truth estimate of the object n at each iteration is shown in Fig. 2. The detail of each step at every iteration is further explicated in Sections 3.6.1 and 3.6.2, respectively.

Note that, at the first iteration, the initialized truth estimate at every local machine k is calculated by averaging for continuous data or majority voting for categorical data over the available sources. Namely, for the object n, $\bar{x}_n^{k,0} = \frac{1}{|\mathcal{S}_n^k|}\sum_{s\in\mathcal{S}_n^k} x_n^s$ or $\bar{x}_n^{k,0} = \arg\max_x \sum_{s\in\mathcal{S}_n^k} \mathbb{1}(x_n^s, x)$,[2] which will be sent to the center for further computation. When the

2. $\cdot^0$ represents the first iteration.

center collects $\{\bar{x}_n^{k,0}\}_{k=1}^K$, it will return the truth estimate $\tilde{x}_n^0 = \frac{1}{K}\sum_{k=1}^K \bar{x}_n^{k,0}$ to local machines.

3.6.1 Bootstrapping at Local Machines

After receiving the truth estimate from the center, every local machine (a) calculates the ω_s via Eq. (1) for available $s \in \mathcal{S}_n^k$, and (b) updates the initialized truth $\bar{x}_n^k$ via bootstrapping techniques. Similar to the *ETBoot* strategy, every local machine first samples a source index set $\tilde{\mathcal{S}}_n^k$ with replacement from $\mathcal{S}_n^k$. The claim samples are represented as $\widetilde{\mathcal{X}}_n^k = \{x_n^s\}_{s\in\tilde{\mathcal{S}}_n^k}$. The source reliability is calculated once the center sends back the truth estimator. Based on the source reliability $\{\omega_s\}_{s\in\mathcal{S}_n^k}$, the initialized truth estimate at the kth local machine is thus updated by

$$\bar{x}_n^k = \frac{\sum_{s\in\tilde{\mathcal{S}}_n^k}\omega_s x_n^s}{\sum_{s\in\tilde{\mathcal{S}}_n^k}\omega_s}. \tag{17}$$

Moreover, local machine k also calculates the variance of the initialized truth estimate, that is,

$$\widehat{\mathbb{V}\mathrm{ar}}(\bar{x}_n^k) = \frac{\sum_{s\in\mathcal{S}_n^k}\omega_s^2\hat{\sigma}_{s,k}^2}{(\sum_{s\in\mathcal{S}_n^k}\omega_s)^2}, \tag{18}$$

where $\hat{\sigma}_{s,k}^2 = \frac{\sum_{n\in\mathcal{N}_s}(x_n^s - \tilde{x}_n)^2}{N_s - 1}$. Then, local machine k sends the initialized truth estimate (i.e. $\bar{x}_n^k$) associated with its variance (i.e., $\widehat{\mathbb{V}\mathrm{ar}}(\bar{x}_n^k)$) to the center for further computation.

3.6.2 Truth Estimation & Confidence Interval Construction

When the center collects $\{\bar{x}_n^k\}_{k=1}^K$ from all local machines, it will calculate the final truth estimator as well as an α-level two-sided confidence interval. The final estimate of the nth object's truth $\tilde{x}_n$ is defined as the average of the truth estimates over all local machines. Namely,

$$\tilde{x}_n = \frac{1}{K}\sum_{k=1}^K \bar{x}_n^k. \tag{19}$$

Note that the underlying idea to obtain a confidence interval of the truth x_n^* is as follows. We first need to construct an x_n^*-related-statistic T, and further estimate its accumulated density function $F(t)$. Inheriting from Section 3.3, $T \triangleq \frac{\hat{\theta}(\boldsymbol{X}) - x_n^*}{[\widehat{\mathbb{V}\mathrm{ar}}(\hat{\theta}(\boldsymbol{X}))]^{\frac{1}{2}}/\sqrt{|\mathcal{S}_n|}}$, where $\boldsymbol{X}$ is a sample set. The endpoints of an α-level two-sided confidence interval are the same as shown in Eq. (11). However, one issue is that $T^{(\alpha)}$ is always unknown a priori. To obtain such confidence intervals, we need to estimate $T^{(\alpha)}$. As introduced in the previous step, we have bootstrapped $\mathcal{X}_n^k$ at local machine k. Based on $\mathcal{X}_n^k$, we are able to calculate both $\hat{\theta}(\mathcal{X}_n^k)$ and $\widehat{\mathbb{V}\mathrm{ar}}(\hat{\theta}(\mathcal{X}_n^k))$, where $\hat{\theta}(\mathcal{X}_n^k) = \bar{x}_n^k$. Different from $\widehat{T}$, another estimator $\widetilde{T}_k$ in the distributed truth discovery paradigm for the statistic T is defined as follows:

$$\widetilde{T}_k = \frac{\bar{x}_n^k - \tilde{x}_n}{[\widehat{\mathbb{V}\mathrm{ar}}(\bar{x}_n^k)]^{\frac{1}{2}}\sqrt{|\mathcal{S}_n^k|}}. \tag{20}$$

Based on Eq. (20), the estimate of $T^{(\alpha)}$ is further defined as follows:

$$\widetilde{T}^{(\alpha)} = \sup\Big\{t \in \{\widetilde{T}_1, \cdots, \widetilde{T}_K\} : \frac{\#(\widetilde{T}_k \le t)}{K} \le \alpha\Big\}. \tag{21}$$

Moreover, the estimate of the variance of the final truth estimator is defined as the average of the variances, that is, $\hat{\sigma}_n^2 = \frac{1}{K}\sum_{k=1}^K \widehat{\mathbb{V}\mathrm{ar}}(\hat{\theta}(\mathcal{X}_n^k))$. Combining Eqs. (11) and (21), the estimate of an α-level two-sided confidence interval via the distributed bootstrapping strategy is

$$\Big(\tilde{x}_n - \frac{\widetilde{T}^{(1-\alpha/2)}\hat{\sigma}_n}{\sqrt{K}}, \tilde{x}_n - \frac{\widetilde{T}^{(\alpha/2)}\hat{\sigma}_n}{\sqrt{K}}\Big). \tag{22}$$

The pseudo code of the proposed distributed *ETCIBoot* algorithm is shown in Algorithm 4.

Algorithm 4. D-*ETCIBoot* Algorithm

Input: Data collection $\{\mathcal{X}_n^k\}_{k,n=1}^{K,N}$, a confidence level α.
Output: Truths $\{\tilde{x}_n\}_1^N$ and their CIs $\{CI_n(\alpha)\}_1^N$.

```
Every machine calculates x̄_n^k and sends it to the center;
The center calculates x̃_n^0 and sends it to all machines
while the convergence condition is not satisfied do
    for each local machine k (k = 1, ..., K) do
        Adopts the truth estimate obtained in previous step to
        calculate {ω_s^k}_{s∈S_n^k} according to Eq. (1);
        for each object n (n = 1, ..., N) do
            Bootstraps X̃_n^k, calculates (x̄_n^k, V̂ar(x̄_n^k)), and sends
            (x̄_n^k, V̂ar(x̄_n^k)) to the center;
            The center calculates truth estimator x̃_n and sends it
            back to all local machines;
            The center calculates the confidence interval CI_n(α)
            based on Eq. (22);
        end for
    end for
end while
```

4 Experiments

In this part, we introduce the experimental setup, test the *ETCIBoot* and baselines on simulated datasets generated in different scenarios and real-world datasets, and compare the proposed *ETCIBoot* and D-*ETCIBoot* on both simulated and real world data in terms of accuracy as well as efficiency. Experiments show that: (1) *ETCIBoot* outperforms the state-of-the-art truth discovery methods in most cases, (2) *ETCIBoot* can provide accurate confidence interval estimates, and (3) D-*ETCIBoot* can achieve comparable accuracy compared with *ETCIBoot* with a significant speed-up.

4.1 Experimental Setup

In this part, we introduce the baseline methods and discuss the measurements for evaluation.

Baselines. For all truth discovery methods, we conduct them on the same input data in an unsupervised manner. Although ground truths are available, we only use them for evaluation. For different data types, different baselines are adopted, including both the naive conflict resolution methods and the state-of-the-art truth discovery methods. More precisely, for continuous data we use Median, Mean, CATD [4], CRH [3] and GTM [5]. Baselines used for categorical data include: Voting, Accusim [6], 3-estimate [9], CRH [3], Investment [8], CATD [4], ZenCrowd [10],

TABLE 4
Comparison on Simulated Data: All Scenarios

Method	Scenario 1 (Uniform$(0,1)$)		Scenario 2 (Gamma$(1,3)$)		Scenario 3 (FoldedNormal$(1,2)$)		Scenario 4 (Beta$(1,\frac{1}{2})$)	
	MAE	*RMSE*	*MAE*	*RMSE*	*MAE*	*RMSE*	*MAE*	*RMSE*
ETCIBoot	**.0226**	**.0290**	**.0231**	**.0291**	.0223	**.0286**	.0233	**.0298**
CATD	.0228	.0297	.0237	.0307	**.0222**	.0291	**.0216**	.0283
CRH	.0378	.0518	.0379	.0509	.0367	.0494	.0398	.0550
Median	.0708	.0944	.0724	.0975	.0717	.0964	.0766	.1030
Mean	.1953	.2423	.1922	.2455	.1960	.2437	.1975	.2455
GTM	.0815	.1018	.0838	.1044	.0808	.1010	.0830	.1032

Dawid&Skene [19], and TruthFinder [7]. Details of baselines are discussed in the related work (i.e., Section 5).

Measurements. As the experiments involve both continuous and categorical data, we introduce different measurements. For data of continuous type, we adopt both the mean of absolute error (*MAE*) and the root of mean square error (*RMSE*); *Error Rate* is used for date of categorical type. The details of the measurements are:

- *MAE*: *MAE* measures the L^1-norm between the methods' output and the ground truths. It tends to penalize more on small errors.
- *RMSE*: *RMSE* measures the L^2-norm between the methods' output and the ground truths. It tends to penalize more on the large distance and less on the small distance comparing with *MAE*.
- *Error Rate*: *Error Rate* is defined as the percentage of mismatched values between the output of each method and the ground truths.

For all measurements, the smaller the value, the better the method.

4.2 Simulated Datasets

In this subsection, we test the proposed *ETCIBoot* on several simulated datasets, which capture different scenarios involving various distributions of source reliability. We first introduce the procedure of generating simulated datasets, and then test the effectiveness of *ETCIBoot* in identifying truths comparing with baselines on these datasets. Last but not least, we compare the confidence intervals obtained by *ETCIBoot* with that by theoretical distribution and show the advantage of bootstrapping.

Data Generation. The procedure of generating simulated data is shown as follows:

- (i) We first generate a vector of the number of claims $\boldsymbol{C}$, e.g., $\boldsymbol{C} = (5, 10, 15, \cdots, 50)$.
- (ii) For each $c_i \in \boldsymbol{C}$, there are $o_i = e^7 \cdot c_i^{-1.5}$ objects which will receive c_i claims. This power law function is used to create the long-tail multi-source data. Thus, there are totally $O = \sum_i o_i$ objects and $S = \max_i\{c_i\}$ sources.
- (iii) For each source, we randomly generate its reliability $\sigma_s^2 \sim F$, where F is a pre-defined distribution. Thus, for each source, its claims are generated from $\text{Normal}(0, \sigma_s^2)$. Here, σ_s^2 captures reliability degree of the sth source's information. The larger value the σ_s^2, the lower reliability degree of the sth source.

Experiments. In the following experiments, we simulate different scenarios via changing source reliability distributions F. We set $\boldsymbol{C} = 70 : 100$; thus, there are 31 objects and 100 sources. Note that the number of objects is not large. This is used to better display the experimental results on the confidence interval estimates. To reduce the randomness, we repeat the experiment 100 times and report the average results. As the simulated data is continuous, *MAE* and *RMSE* are used for evaluation. We simulate 4 scenarios and the detail of each scenario is discussed as follows. Note that σ_s^2 represents the source reliability degree. The larger value the σ_s^2, the lower reliability degree the source.

Scenario 1: $\sigma_s^2 \sim \text{Uniform}(0, 1)$. In this scenario, all source reliability degrees are uniformly distributed in $(0, 1)$.

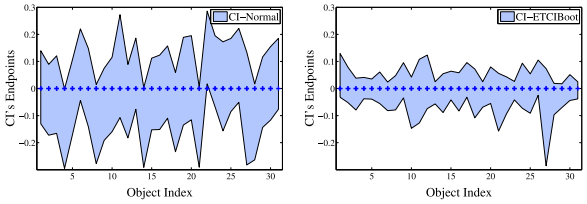
Scenario 2: $\sigma_s^2 \sim \text{Gamma}(1, 3)$. In this scenario, most of the sources are reliable with high reliability degrees. However, there are a few unreliable sources with very small reliability degrees.

Scenario 3: $\sigma_s^2 \sim \text{FoldedNormal}(1, 2)$. As Folded Normal is a long-tail distribution, in this scenarios, it generates a few unreliable sources. Compared with Scenarios 1 and 2, the reliable sources have higher reliability degrees.

Scenario 4: $\sigma_s^2 \sim \text{Beta}(1, \frac{1}{2})$. In this scenario, source reliability degrees are within $0 \sim 1$. Compared with other scenarios, there are much more reliable sources.

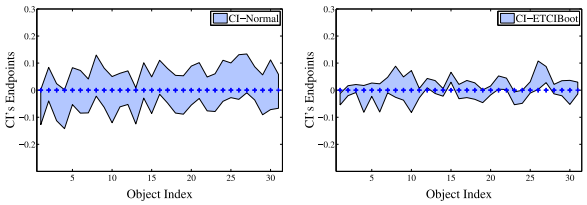
Comparison with Baselines. Table 4 shows that the proposed *ETCIBoot* outperforms all baselines in all scenarios in terms of both *MAE* and *RMSE*. When estimating the truth for each object n, *ETCIBoot* obtains multiple truth estimates which are calculated according to Eq. (2) based on the bootstrapped claims. Then, the final truth estimator is defined as the average of these estimates. Experimentally, we generate $10 * |\mathcal{S}_n|$ bootstrapping samples. Due to the properties of bootstrapping, *ETCIBoot* is robust to the outlying claims provided by some sources. However, as existing truth discovery methods typically compute weighted mean to obtain one single point estimate, they are more sensitive to the outlying claims. So, the *ETCIBoot* performs better than baselines as confirmed in the experimental results. Also, as there are more reliable sources in Scenarios 3 and 4, the results are better compared with those in Scenarios 1 and 2. It confirms the underlying intuition of truth discovery: the more the reliable sources, the better the results.

Confidence Interval Comparison. For confidence interval comparison, we compare the results of *ETCIBoot* with that obtained by theoretical distribution, i.e., normal distribution. Note that $\hat{x}_n \sim \text{Normal}(x_n^*, \frac{\sum_{s\in\mathcal{S}_n} \omega_s^2\sigma_s^2}{(\sum_{s\in\mathcal{S}_n} \omega_s)^2})$ (based on



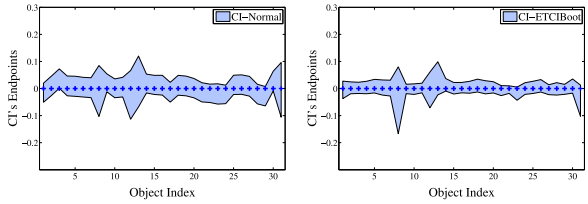
(a) *CI-Normal* (b) *CI-ETCIBoot*

Fig. 3. Scenario 1: $\mathrm{Uniform}(0, 5)$.



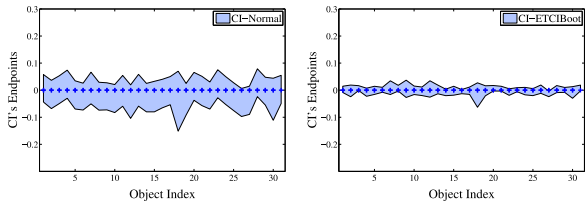
(a) *CI-Normal* (b) *CI-ETCIBoot*

Fig. 4. Scenario 2: $\mathrm{Gamma}(1, 3)$.



(a) *CI-Normal* (b) *CI-ETCIBoot*

Fig. 5. Scenario 3: $\mathrm{FoldedNormal}(1, 2)$.



(a) *CI-Normal* (b) *CI-ETCIBoot*

Fig. 6. Scenario 4: $\mathrm{Beta}(1, .5)$.

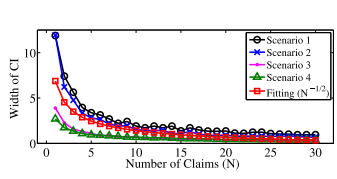
Eq. (2)). As the true σ_s^2 is known for each source, we know the theoretical distribution for $\hat{x}_n$, based on which we can further obtain the 95 percent-level confidence interval. We term the confidence interval obtained in this way as *CI-Normal*. The confidence interval (i.e., Eq. (14)) for the truths' estimators, which is obtained by the *ETCIBoot* using the bootstrapping technique, is referred to as *CI-ETCIBoot*.

We report the results in Scenarios $1 \sim 4$ in Figs. 3, 4, 5, 6, respectively. From Figs. 3, 4, 5, 6, we can draw the following conclusions: (1) The *CI-ETCIBoot* is much smaller than *CI-Normal* in all simulated scenarios. Note that the smaller the confidence interval, the more confident the estimator. For example, in Scenario 1 the shaded area (i.e., the area between the lower and upper bound curves) of *CI-Normal* in Fig. 3a is larger than that of *CI-ETCIBoot* in Fig. 3(b). Similar conclusions can be drawn in other scenarios. Thus, the experimental results show the power of the *ETCIBoot* on constructing effective confidence intervals. (2) As most sources are reliable in Scenarios $2 \sim 4$, comparing with Scenario 1, the width of *CI-ETCIBoot* or *CI-Normal* in other scenarios is smaller, which indicates the higher overall confidence in these scenarios.

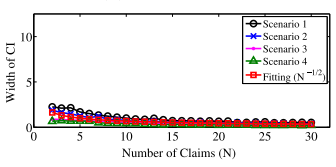
Next we conduct experiments to illustrate the relationship between the width of confidence interval and the number of claims on long-tail data. We follow the same procedure to generate the simulated data, except that we choose the number of claims as 2 to 30. If there is only one claim, it is impossible to construct the confidence interval. We present the width of *CI-Normal* and *CI-ETCIBoot* in all scenarios in Figs. 7a and 7b, respectively. Meanwhile, we also fit them into a polynomial function of N ($N^{-\frac{1}{2}}$), respectively. The red line with square marker represents the fitting line, averaging over all scenarios. From Figs. 7a and 7b, we can see that the width of the 95 percent confidence interval, obtained via either normal distribution or *ETCIBoot*, decreases with respect to the number of claims at an error rate $N^{-\frac{1}{2}}$, where N is the number of claims. It confirms the theoretical analysis that if an object receives more claims then its estimator is more accurate. Moreover, the width of *CI-ETCIBoot* is much smaller than that of *CI-Normal*, which demonstrates that *ETCIBoot* is able to provide a more confident estimator. This advantage is achieved by incorporating bootstrapping techniques into truth discovery procedure in *ETCIBoot*.

4.3 Real-World Datasets

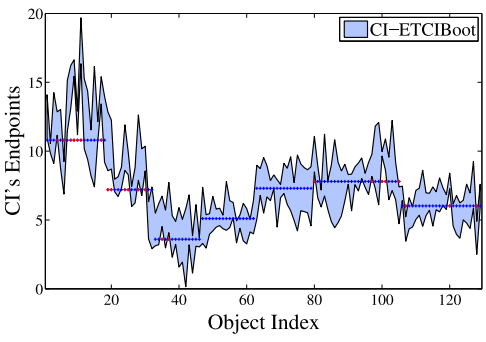
In this subsection, we present the experimental results on two continuous datasets and two categorical datasets.



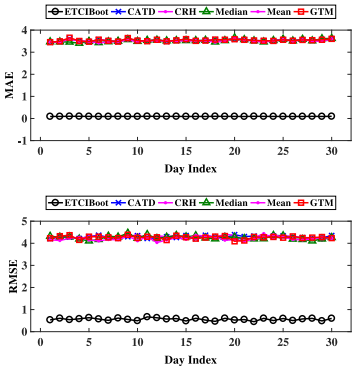
(a) *CI-Normal*



(b) *CI-ETCIBoot*



(c) Indoor Floorplan dataset: CI-**ETCIBoot**



(d) Comparison on Flight data over 30 days

Fig. 7. Experiments: (a) and (b) Width of Confidence Interval w.r.t # of Claims. (c) Visualization of Confidence Interval of *ETCIBoot*. (d) *MAE* and *RMSE* comparison of truth discovery methods on Flight Data.

TABLE 5
Comparison on Game Dataset

Method	*Error Rate*										
	Level 1 (303)	Level 2 (295)	Level 3 (290)	Level 4 (276)	Level 5 (253)	Level 6 (218)	Level 7 (187)	Level 8 (138)	Level 9 (99)	Level 10 (44)	All Levels (2103)
ETCIBoot	.0165	**.0271**	**.0241**	**.0217**	**.0395**	**.0505**	**.0481**	**.0870**	**.0707**	**.1364**	**.0385**
CATD	**.0132**	**.0271**	.0276	.0290	.0435	.0596	**.0481**	.1304	.1414	.2045	.0485
CRH	.0264	**.0271**	.0345	.0435	.0593	.0872	.0856	.2609	.3535	.4545	.0866
ZenCrowd	.0330	.0305	.0345	.0471	.0593	.0872	.0856	.2754	.3636	.5227	.0899
AccuSim	.0264	.0305	.0345	.0507	.0632	.0963	.0909	.2826	.3636	.5000	.0913
3-Estimates	.0264	.0305	.0310	.0507	.0672	.1055	.0963	.2971	.3737	.5000	.0942
Dawid&Skene	.0297	.0305	.0483	.0507	.0672	.1101	.0963	.2971	.3636	.5227	.0975
Voting	.0297	.0305	.0414	.0507	.0672	.1101	.1016	.3043	.3737	.5227	.0980
Investment	.0330	.0407	.0586	.0761	.0870	.1239	.1283	.3406	.3838	.5455	.1151
TruthFinder	.0693	.0915	.1241	.0942	.1581	.2294	.2674	.3913	.5455	.5455	.1816

Experiments show that the proposed *ETCIBoot* is able to obtain more accurate estimates of truths comparing with baselines. We first introduce the description of the datasets and then report the results.

4.3.1 Continuous Data

Dataset Description. The following datasets of continuous data type are used in experiments:

- Indoor Floorplan Dataset: We develop an Android App to estimate the walking distances of smartphone users via multiplying their step sizes by step count inferred using the in-phone accelerometer. There are totally 247 users and 129 objects (i.e., indoor hallways). The ground truth is obtained by manually measuring the indoor hallways. The goal is to estimate the distance of indoor hallways from the data provided by a crowd of users.
- Flight Status Dataset: The flight data [28] is collected by extracting departure/arrival time for 11,512 flights from 38 sources on every day in December 2011. We present the time in terms of the minutes from 00:00. There are 11,146 flights that have departure/arrival ground truths. The goal is to estimate the departure/arrival time for each flight.

Result Analysis. We present the results of *ETCIBoot* and baselines with respect to *MAE* and *RMSE* on the continuous datasets in Table 6. The results show that the proposed *ETCIBoot* can achieve the best performance on both datasets.

On Indoor Floorplan dataset, as the number of objects is small, we also present the confidence intervals obtained by *ETCIBoot* for each object in Fig. 7c. The figure shows that in most cases the confidence intervals provided by *ETCIBoot* contains the corresponding objects' truths. However, there are some confidence intervals which do not contain truths. A possible reason is: These objects are claimed by a few sources and the information provided by these sources is far away from the truth. Take the 9th object for example. There are only 4 sources which provide claims, among which the smallest value is 14.3 that is still very larger than the ground truth 10.8. It is impossible to correctly identify these objects' truths for any truth discovery method. So, the CI estimates obtained by *ETCIBoot* do not contain the truths for these objects.

On Flight Status dataset, the data on each day is treated as a single data collection. As there are many flights only claimed by a few sources, the performance of baselines is not satisfactory. We conduct a case study on Day 1 dataset. We count the statistics on how many claims of an object receives to show the long-tail phenomenon: (1) there are about 61.1 percent of flights which only receive claims from at most 5 out of 38 sources; (2) only 2.3 percent of flights have received claims from more than 25 sources. Similar phenomenon can be found on other days' data.

Consequently, we can see that the proposed *ETCIBoot* outperforms all baselines, as shown in Fig. 7d. We do not present the confidence interval for the flights due to the page limit and the large number of flights.

4.3.2 Categorical Data

Dataset Description. We introduce the details of two categorical datasets and their tasks as follows:

- Game Dataset: Game dataset [4] collects answers from multiple users based on a TV game show "Who Wants to Be a Millionaire" via an Android App. There are 37,029 Android users and 2,103 questions. Ground truths are available for evaluation. The goal is to identify each question's answer from the users' answers.
- SFV Dataset: SFV dataset is built upon the annal Slot Filling Validation (SFV) competition of the NITS Text Analysis Conference Knowledge Base Population track [29]. In this task, given a query (an object), e.g., the birthday of Obama, 18 slot filling systems (sources) extract useful claims independently from a large-scale corpus. The 2011 SFV dataset[3] contains $2,538$ claims from 18 sources for 328 objects. The goal is to extract the true answer for each query from the systems' claims.

Result Analysis. For categorical data, we first encode the claims into probability vectors and then apply the methods proposed for continuous data, such as *ETCIBoot*, CATD, etc. The detailed procedure is: For a question with 4 possible choices, the first choice is encoded into a 4-element vector $(1, 0, 0, 0)$. In Tables 5 and 7, we present the experimental results of the proposed *ETCIBoot* as well as baselines on the SFV and Game datasets, respectively.

On Game dataset, the number of sources (37,029) is sufficient for bootstrapping. Although CATD performs best

3. http://www.nist.gov/tac/2011/

TABLE 6
Comparison on Continuous Data

Method	Indoor Floorplan		Flight Status	
	MAE	*RMSE*	*MAE*	*RMSE*
ETCIBoot	**.9349**	**1.3249**	**.0913**	**.5697**
CATD	.9960	1.385	3.443	4.2318
CRH	1.193	1.596	3.446	4.240
Median	1.380	1.786	3.468	4.261
Mean	1.785	2.285	3.433	4.225
GTM	1.285	1.483	3.450	4.242

among all baselines, the proposed *ETCIBoot* achieves even better performance compared with CATD. Especially, on the Levels 8, 9, and 10, the proposed *ETCIBoot* improves the results by 33.28, 50.00 and 33.30 percent, respectively, compared with the best baseline CATD. As *ETCIBoot* integrates bootstrapping techniques into the truth discovery procedure, it is more robust to the wrong claims compared with baselines. Thus, *ETCIBoot* can obtain the best performance. Note that there are 81 objects on which no sources provide correct answers. Therefore, the lowest error rate for any truth discovery method is .0380. *ETCIBoot* can achieve error rate at .0385, which shows its effectiveness in identifying truths.

On SFV dataset, there are only 18 sources, so we have a limited number of sources to bootstrap at each iteration of *ETCIBoot*. Thus, the result of the proposed *ETCIBoot* (.0945) is not the best, but still comparable with the two best methods: AccuSim (.0701) and TruthFinder (.0793).

4.4 Experimental Results of D-*ETCIBoot* Method

Next, we compare the proposed D-*ETCIBoot* or *ETCIBoot* with the state-of-the-art truth discovery methods CATD and CRH in terms of both accuracy and efficiency.

4.4.1 Experiments on Simulated Datasets

The data generation procedure is similar to that described in Section 4.2. Recall that the claims from source s are generated from a Gaussian distribution, i.e., $\mathrm{Normal}(0, \sigma_s^2)$. σ^2 here plays an important role in the data generation procedure, as it represents the source reliability degree. The larger value the σ^2 is, the lower reliability degree the source is, and the less claims the source correctly makes. We simulate the data on four different scenarios: $\sigma^2 \sim \mathrm{Uniform}(0, 1)$, $\mathrm{Gamma}(1, 3)$, $\mathrm{FoldedNormal}(1, 2)$, and $\mathrm{Beta}(1, \frac{1}{2})$. We randomly partition the data into K local parts and then evaluate D-*ETCIBoot* on these partitions distributed on K machines. In distributed scenarios, we run both CATD and CRH on each local node to obtain a local truth estimate and then average or vote all local truth estimates for continuous or categorical data type, respectively. In the following experiments, K is set to be 5, 10, and 15. To reduce the randomness, we run each experiment 100 times and report the averages of *MAE*, *RMSE*, and running time on each local machine in Table 8. We measure the running time on a machine with a 2.8 GHz Intel Core i7 processor and 16 GB memory.

Result Analysis. From Table 8, we can see that the results of the proposed D-*ETCIBoot* are slightly worse than those of *ETCIBoot*, but D-*ETCIBoot* takes much less running time. As the number of local machines K increases, the accuracy usually drops while the running time dramatically decreases. For instance, when there are 15 local machines, D-*ETCIBoot* only needs about .0705 seconds on each local machine while it takes 1.188 seconds (about 17 times) for *ETCIBoot* to process the whole dataset, but the best *MAE* and *RMSE* are obtained when the whole dataset is processed on one single machine. As mentioned in [17], the more sources at one machine, the better the estimate of both the truth and the source reliability. In distributed truth discovery scenario, claims are distributed into multiple local machines. Thus, each machine has less information to estimate the source reliability. As a result, the accuracy of the D-*ETCIBoot* is worse than that of *ETCIBoot*. However, each machine bootstraps less number of samples comparing with *ETCIBoot*, so D-*ETCIBoot* takes less time which makes it more efficient in handling large-scale data. Moreover, compared with CATD and CRH, the proposed D-*ETCIBoot* can achieve higher accuracy but with less time.

TABLE 7
Comparison on *SFV* Dataset

Method	*Error Rate*
ETCIBoot	.0945
CATD	.1037
CRH	.0854
ZenCrowd	.1010
AccuSim	**.0701**
3-Estimates	.1128
Voting	.1128
Dawid&Skene	.0985
Investment	.2896
TruthFinder	.0793

4.4.2 Experiments on Real World Datasets

In this part, we present experimental results on real world datasets. Details of the datasets can be found in Section 4.3. Due to the page limit, we report experiments on the Indoor Floorplan application for continuous data type and Game data for categorical one. But experiments on the remaining datasets can be obtained when required. More detailed experiment setting is as follows: For Indoor Floorplan, K is set to be 5, 10, and 15. For Game data with 37,029 sources/users, K is 50, 100, and 150. For each dataset, we run the proposed D-*ETCIBoot* and baselines (i.e. CATD and CRH) 20 times. We report the averages of *MAE*, *RMSE* and running time of each local machine for the continuous data in Table 9. For the categorical data, the averages of *Error Rate* and running time of each local machine are reported in Table 10.

Result Analysis. Table 9 shows the results of both *ETCIBoot* and D-*ETCIBoot* in terms of accuracy and efficiency for the Indoor Floorplan data. From Table 9, we can see that the accuracy of D-*ETCIBoot* is lower with less running time when compared with *ETCIBoot*. Similar results can be founded for both CATD and CRH. As the number of local machines K increases, the performance of the proposed D-*ETCIBoot* (or the distributed version of CATD or CRH) is less accurate while its running time is lower. More specifically, we can see that the proposed D-*ETCIBoot* can achieve comparable accuracy in terms of both *MAE* and *RMSE* but

TABLE 8
Comparison on Simulated Data: All Scenarios

Nodes	Methods	Scenario 1 (Uniform$(0,1)$) *MAE*	Scenario 1 *RMSE*	Scenario 1 Time	Scenario 2 (Gamma$(1,3)$) *MAE*	Scenario 2 *RMSE*	Scenario 2 Time	Scenario 3 (FoldedNormal$(1,2)$) *MAE*	Scenario 3 *RMSE*	Scenario 3 Time	Scenario 4 (Beta$(1,\frac{1}{5})$) *MAE*	Scenario 4 *RMSE*	Scenario 4 Time
	ETCIBoot	**.0226**	**.0290**	**1.188**	**.0231**	**.0291**	**1.272**	.0223	**.0285**	**1.283**	.0233	**.0298**	**1.461**
$K=1$	CATD	.0228	.0297	2.003	.0237	.0307	2.166	**.0222**	.0291	2.207	**.0216**	.0283	2.440
	CRH	.0378	.0518	2.003	.0379	.0509	2.166	.0367	.0494	2.207	.0398	.0550	2.440
	D-*ETCIBoot*	**.0554**	**.0692**	**.2213**	**.0550**	**.0692**	**.2292**	**.0583**	**.0737**	**.2121**	**.0560**	**.0703**	**.2082**
$K=5$	CATD	.0600	.0746	.5091	.0574	.0725	.5360	.0638	.0805	.4901	.0639	.0795	.4805
	CRH	0678	.0866	.5077	.0667	.0852	.5353	.0718	.0915	.4908	.0698	.0881	.4803
	D-*ETCIBoot*	**.0835**	**.1056**	**.1064**	**.0810**	**.1019**	**.1184**	**.0824**	**.1033**	**.1093**	**.0838**	**.1054**	**.1109**
$K=10$	CATD	.0945	.1183	.3001	.0902	.1136	.3393	.0888	.1111	.3104	.0927	.1161	.3137
	CRH	.0961	.1215	.3001	.0929	.1174	.3393	.0923	1164	.3124	.0919	.1167	.3144
	D-*ETCIBoot*	**.1164**	**.1477**	**.0705**	**.1143**	**.1452**	**.0743**	**.1138**	**.1436**	**.0707**	**.1095**	**.1393**	**.0738**
$K=15$	CATD	.1249	.1585	.2310	.1263	.1589	.2428	.1245	.1558	.2300	.1213	.1530	.2424
	CRH	.1321	.1684	.4109	.1333	.1704	.4359	.1328	.1689	.4139	.1261	.1600	.4470

with less running time when compared with CATD or CRH. Similar results can be found on the categorical dataset, Game data, as shown in Table 10. Overall, the proposed D-*ETCIBoot* can still achieve comparable accuracy using less running time when compared with the proposed *ETCIBoot*. The efficiency of the proposed D-*ETCIBoot* is more obvious on large-scale datasets. For instance, on the Game dataset, there are 37,029 sources and 2,103 objects. *ETCIBoot* takes about 300 seconds to obtain the final results, while D-*ETCIBoot* only needs about 4 seconds on each machine when $K=50$. Moreover, D-*ETCIBoot* only takes about 1.5 seconds on each machine when $K=150$, which shows a significant speed-up compared with the running time of *ETCIBoot* (i.e., 1,300 seconds). On the other hand, the distributed versions of truth discovery methods can achieve comparable accuracy in less running time. Moreover, the proposed D-*ETCIBoot* is more efficient than baselines but can also achieve comparable accuracy.

5 Related Work

Truth discovery has become an eye-catching term recently and many methods have been proposed to identify true information (i.e., truths) from the conflicting multi-source data. The advantage of truth discovery over the naive aggregation methods such as averaging or voting is that it can capture the variance in sources' reliability degrees. So, truth discovery methods can estimate source reliability automatically from the data, which is integrated into truth estimation as source weight. Consequently, the more reliable sources contribute more in the final aggregation.

A large variety of truth discovery methods have been designed to jointly estimate truths and source reliability. In [3], the authors formulate the truth discovery task into an optimization framework (CRH). They propose to minimize the overall weighted distance between claims from sources and aggregated results. CATD [4] is a statistical method that has been proposed to deal with long-tail phenomenon in truth discovery tasks, where confidence interval is incorporated in source weight estimation. However, CATD does not consider the long-tail phenomenon on objects, which can be solved by *ETCIBoot*. In [5], the authors propose a probabilistic model based truth discovery framework (GTM). Both AccuSim [6] and TruthFinder [7] adopt Bayesian analysis to estimate source reliability and update truths iteratively. In [8], the authors take the prior knowledge on truth and background information into consideration and propose Investment method. 3-Estimate [9] considers the difficulty of getting the

TABLE 9
Comparison on Continuous Data

Nodes	Methods	Indoor Floorplan *MAE*	Indoor Floorplan *RMSE*	Indoor Floorplan Time
	ETCIBoot	.9399	1.309	**1.413**
$K=1$	CATD	.9960	1.385	2.818
	CRH	1.193	1.596	2.918
	D-*ETCIBoot*	1.420	2.026	**.5706**
$K=5$	CATD	1.329	1.943	1.246
	CRH	1.527	2.122	1.336
	D-*ETCIBoot*	1.634	2.116	**.2663**
$K=10$	CATD	1.589	2.055	.6430
	CRH	1.595	2.055	.8039
	D-*ETCIBoot*	1.687	2.246	**.1571**
$K=15$	CATD	1.619	2.201	.4349
	CRH	1.579	2.159	.5696

TABLE 10
Comparison on Categorical Data

Nodes	Methods	Game *Error Rate*	Game Time
	ETCIBoot	.0385	300.0
$K=1$	CATD	.0485	156.4
	CRH	.0866	**108.3**
	D-*ETCIBoot*	.0889	**3.957**
$K=50$	CATD	.0875	4.408
	CRH	.0927	3.954
	D-*ETCIBoot*	.0932	**1.853**
$K=100$	CATD	.0903	2.294
	CRH	.0932	2.383
	D-*ETCIBoot*	.0980	**1.227**
$K=150$	CATD	.0970	1.511
	CRH	.0999	1.650

truth for each object when calculating source weights as well as complement vote. A topic related to truth discovery is crowdsourcing aggregation [10], [19], [30], [31], [32]. Dawid&-Skene [19] and ZenCrowd [10] use Expectation Maximization technique to update source weights and truths simultaneously, based on a confusion matrix. [30] conductes a comprehensive survey on crowdsourcing data management from the perspective of fundamental techniques. In [31], the authors survey many existing algorithm of inferring truth from crowdsourced data in both database and data mining areas. [32] proposes a domain-aware crowdsourcing system using Knowledge Base to interpret the domain knowledge of each questions to adaptively assign tasks to the crowds. However, the setting of truth discovery involves open-domain answer space, i.e., each object may have different candidate answers in terms of size and content, so most crowdsourcing models are not suitable, since they need to estimate the confusion matrix.

However, most existing truth discovery methods have the following limitations: (1) Most of them apply weighted averaging, so they are sensitive to outlying claims, and (2) they focus on point estimation of the truth, where important confidence information is missing. In this paper, we illustrate the importance of confidence interval estimation in *truth discovery*, and propose effective methods (*ETCIBoot* and D-*ETCIBoot*) to address it. By integrating bootstrapping into truth discovery, *ETCIBoot* is robust compared with the state-of-the-art truth discovery methods.

6 Conclusions

In this paper, we first illustrate the importance of confidence interval estimation in truth discovery, which has never been discussed in existing work. To address the problem, we propose a novel truth discovery method (*ETCIBoot*) to construct confidence interval estimates as well as identify truths. The bootstrapping techniques are nicely integrated into the truth discovery procedure in *ETCIBoot*. Due to the properties of bootstrapping, the estimators obtained by *ETCIBoot* are more accurate and robust compared with the state-of-the-art truth discovery approaches. Moreover, we propose D-*ETCIBoot* in the distributed truth discovery paradigm to deal with large-scale data. Theoretically, we prove that the confidence interval obtained by *ETCIBoot* is asymptotically consistent. Experimentally, we demonstrate that *ETCIBoot* is not only effective in constructing confidence intervals but also able to obtain better truth estimates. The efficiency of the D-*ETCIBoot* is also confirmed on both simulated and real-world datasets.

Acknowledgments

This work was sponsored in part by the US National Science Foundation under grant IIS 1319973, IIS 1553411, CNS 1566374, CNS 1742845, and CNS 1652503. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agency.

References

[1] X. Dong, E. Gabrilovich, G. Heitz, W. Horn, N. Lao, K. Murphy, T. Strohmann, S. Sun, and W. Zhang, "Knowledge vault: A web-scale approach to probabilistic knowledge fusion," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 601–610.

[2] A. Marian and M. Wu, "Corroborating information from web sources," *Data Eng. Bull.*, vol. 34, no. 3, pp. 11–17, 2011.

[3] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han, "Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2014, pp. 1187–1198.

[4] Q. Li, Y. Li, J. Gao, L. Su, B. Zhao, M. Demirbas, W. Fan, and J. Han, "A confidence-aware approach for truth discovery on long-tail data," *Proc. VLDB Endowment*, vol. 8, no. 4, pp. 425–436, 2014.

[5] B. Zhao and J. Han, "A probabilistic model for estimating real-valued truth from conflicting sources," in *Proc. QDB*, 2012.

[6] X. L. Dong, L. Berti-Equille, and D. Srivastava,"Integrating conflicting data: the role of source dependence," *Proc. VLDB Endowment*, vol. 550–561, pp. 550–561, 2009.

[7] X. Yin, J. Han, and P. S. Yu, "Truth discovery with multiple conflicting information providers on the web," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 6, pp. 796–808, Jun. 2008.

[8] J. Pasternack and D. Roth, "Knowing what to believe (when you already know something)," in *Proc. 23rd Int. Conf. Comput. Linguistics*, 2010, pp. 877–885.

[9] A. Galland, S. Abiteboul, A. Marian, and P. Senellart, "Corroborating information from disagreeing views," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining*, 2010, pp. 131–140.

[10] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux, "Zencrowd: Leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking," in *Proc. 21st Int. Conf. World Wide Web*, 2012, pp. 469–478.

[11] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher, "On truth discovery in social sensing: A maximum likelihood estimation approach," in *Proc. ACM/IEEE 11th Int. Conf. Inf. Process. Sensor Netw.*, 2012, pp. 233–244.

[12] S. Zhi, B. Zhao, W. Tong, J. Gao, D. Yu, H. Ji, and J. Han, "Modeling truth existence in truth discovery," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 1543–1552.

[13] Z. Zhao, J. Cheng, and W. Ng, "Truth discovery in data streams: A single-pass probabilistic approach," in *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage.*, 2014, pp. 1589–1598.

[14] B. Zhao, B. I. Rubinstein, J. Gemmell, and J. Han, "A bayesian approach to discovering truth from conflicting sources for data integration," *Proc. VLDB Endowment*, vol. 5, pp. 550–561, 2012.

[15] C. Meng, W. Jiang, Y. Li, J. Gao, L. Su, H. Ding, and Y. Cheng, "Truth discovery on crowd sensing of correlated entities," in *Proc. SenSys*, 2015, pp. 169–182.

[16] F. Ma, Y. Li, Q. Li, M. Qiu, J. Gao, S. Zhi, L. Su, B. Zhao, H. Ji, and J. Han, "Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 745–754.

[17] H. Xiao, J. Gao, Z. Wang, S. Wang, L. Su, and H. Liu, "A truth discovery approach with theoretical guarantee," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1925–1934.

[18] H. Xiao, J. Gao, Q. Li, F. Ma, L. Su, Y. Feng, and A. Zhang, "Towards confidence in the truth: A bootstrapping based truth discovery approach," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1935–1944.

[19] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the EM algorithm," *Appl. Statist.*, pp. 20–28, 1979.

[20] G.-J. Qi, C. C. Aggarwal, J. Han, and T. Huang, "Mining collective intelligence in diverse groups," in *Proc. 22nd Int. Conf. World Wide Web*, 2013, pp. 1041–1052.

[21] C. Miao, W. Jiang, L. Su, Y. Li, S. Guo, Z. Qin, H. Xiao, J. Gao, and K. Ren, "Cloud-enabled privacy-preserving truth discovery in crowd sensing systems," in *Proc. SenSys*, 2015, pp. 183–196.

[22] Y. Li, J. Gao, C. Meng, Q. Li, L. Su, B. Zhao, W. Fan, and J. Han, "A survey on truth discovery," *ACM Sigkdd Explorations Newsletter*, vol. 17, no. 2, pp. 1–16, 2016.

[23] R. V. Hogg and A. T. Craig, *Introduction to Mathematical Statistics*. Hoboken, NJ, USA: Wiley, 1978.

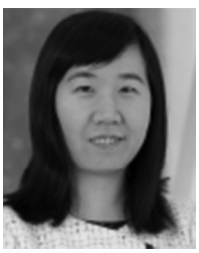
[24] D. Cheng and Y. Liu, "Parallel gibbs sampling for hierarchical dirichlet processes via gamma processes equivalence," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 562–571.

[25] R. Y. Liu, "Bootstrap procedures under some non-i.i.d. models," *Ann. Statist.*, vol. 16, no. 4, pp. 1696–1708, 1988.

[26] A. Q. Li, A. Ahmed, S. Ravi, and A. J. Smola, "Reducing the sampling complexity of topic models," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 891–900.

[27] B. Efron and R. J. Tibshirani, *An Introduction to the Bootstrap*. Boca Raton, FL, USA: CRC Press, 1994.
[28] X. Li, X. L. Dong, K. Lyons, W. Meng, and D. Srivastava, "Truth finding on the deep web: is the problem solved?" *Proc. PVLDB Endowment*, vol. 6, no. 2, pp. 97–108, 2012.
[29] H. Ji, R. Grishman, H. T. Dang, K. Griffitt, and J. Ellis, "Overview of the tac 2010 knowledge base population track," in *Proc. TAC*, 2010, pp. 3–13.
[30] G. Li, Y. Zheng, J. Fan, J. Wang, and R. Cheng, "Crowdsourced data management: Overview and challenges," in *Proc. ACM Int. Conf. Manage. Data*, 2017, pp. 1711–1716.
[31] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng, "Truth inference in crowdsourcing: is the problem solved?" *Proc. VLDB Endowment*, vol. 10, no. 5, pp. 541–552, 2017.
[32] Y. Zheng, G. Li, and R. Cheng, "Docs: A domain-aware crowdsourcing system using knowledge bases," *Proc. VLDB Endowment*, vol. 10, no. 4, pp. 361–372, 2016.

Houping Xiao received the BS degree in statistics from Beijing Normal University. He is working toward the PhD degree in the Department of Computer Science and Engineering at SUNY Buffalo. His research interests include data mining and machine learning, including truth discovery, multi-source information trustworthiness analysis, privacy-preserving data mining, distributed machine learning, etc. He is a student member of the IEEE.

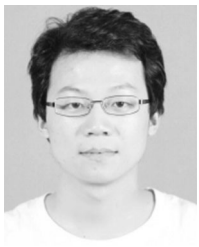


Jing Gao received the PhD degree from the Computer Science Department, University of Illinois at Urbana-Champaign, in 2011, and subsequently joined SUNY Buffalo, in 2012. She is an associate professor with the Department of Computer Science and Engineering at SUNY Buffalo. She is broadly interested in data and information analysis with a focus on truth discovery, information integration, ensemble methods, mining data streams, transfer learning, and anomaly detection. She is a recipient of NSF CAREER Award (2016) and IBM Faculty Award (2013). She is a member of the IEEE.

Qi Li received the BS degree in mathematics from Xidian University and the MS degree in statistics from the University of Illinois at Urbana-Champaign, in 2010 and 2012, respectively. She has defended her PhD dissertation and is expected to graduate with a PhD degree from the Department of Computer Science and Engineering at SUNY Buffalo. Her research interest includes truth discovery, data aggregation, and crowdsourcing.

Fenglong Ma received BE and ME degrees from the Dalian University of Technology and is currently working toward the PhD degree in the Department of Computer Science and Engineering at SUNY Buffalo. His research interests include data mining, and machine learning, including truth discovery, healthcare data mining, and probabilistic graphical model.

Lu Su received the MS degree in statistics and the PhD degree in computer science, both from the University of Illinois at Urbana-Champaign, in 2012 and 2013, respectively. He is an assistant professor with the Department of Computer Science and Engineering at SUNY Buffalo. His research interests include general areas of mobile and crowd sensing systems, Internet of Things, and cyber-physical systems. He has also worked with the IBM T. J. Watson Research Center and National Center for Supercomputing Applications. He is the recipient of NSF CAREER Award, University at Buffalo Young Investigator Award, ICCPS'17 Best Paper Award, and the ICDCS'17 Best Student Paper Award. He is a member of the ACM and IEEE.



Yunlong Feng received the PhD degree in mathematics from the University of Science and Technology of China and the City University of Hong Kong. He is an assistant professor with the Department of Mathematics and Statistics at SUNY Albany. His research interests include machine learning, statistical learning theory, and nonparametric statistics, with recent emphasis on the following topics: kernel methods, robust learning, tensor-based learning, and learning with non-i.i.d observations.

Aidong Zhang is a SUNY distinguished professor with the Department of Computer Science and Engineering, State University of New York at Buffalo, and a program director in the Information & Intelligent Systems Division, National Science Foundation. Her research interests include data mining, bioinformatics, multimedia and database systems, and content-based image retrieval. She is an author of more than 250 research publications in these areas. She has chaired or served on more than 100 program committees of international conferences and workshops, and currently serves on several journal editorial boards. She has published two books *Protein Interaction Networks: Computational Analysis* (Cambridge University Press, 2009) and *Advanced Analysis of Gene Expression Microarray Data* (World Scientific Publishing Co., Inc. 2006). She is a recipient of the National Science Foundation CAREER award and the State University of New York (SUNY) Chancellor's Research Recognition award. She is a fellow of the IEEE.