2017 IEEE International Conference on Data Mining

Discovering Truths from Distributed Data

Yaqing Wang, Fenglong Ma, Lu Su, and Jing Gao
SUNY Buffalo, Buffalo, USA
{*yaqingwa, fenglong, lusu, jing*}@*buffalo.edu*

***Abstract*—In the big data era, the information about the same object collected from multiple sources is inevitably conflicting. The task of identifying true information (i.e., the truths) among conflicting data is referred to as truth discovery, which incorporates the estimation of source reliability degrees into the aggregation of multi-source data. However, in many real-world applications, large-scale data are distributed across multiple servers. Traditional truth discovery approaches cannot handle this scenario due to the constraints of communication overhead and privacy concern. Another limitation of most existing work is that they ignore the differences among objects, i.e., they treat all the objects equally. This limitation would be exacerbated in distributed environments where significant differences exist among the objects. To tackle the aforementioned issues, in this paper, we propose a novel distributed truth discovery framework (DTD), which can effectively and efficiently aggregate conflicting data stored across distributed servers, with the differences among the objects as well as the importance level of each server being considered. The proposed framework consists of two steps: the local truth computation step conducted by each local server and the central truth estimation step taking place in the central server. Specifically, we introduce the uncertainty values to model the differences among objects, and propose a new uncertainty-based truth discovery method (UbTD) for calculating the true information of objects in each local server. The outputs of the local truth computation step include the estimated local truths and the variances of objects, which are the input information of the central truth estimation step. To infer the final true information in the central server, we propose a new algorithm to aggregate the outputs of all the local servers with the quality of different local servers taken into account. The proposed distributed truth discovery framework can infer object truths without delivering any raw data to the central server, and thus can reduce communication overhead as well as preserve data privacy. Experimental results on three real world datasets show that the proposed DTD framework can efficiently estimate object truths with accuracy guarantee, and the proposed UbTD algorithm significantly outperforms the state-of-the-art batch truth discovery approaches.**

***Keywords*-Truth discovery; distributed system; uncertainty estimation**

I. Introduction

We are living in the era of big data, and there are usually multiple sources where we can collect information about the same object (e.g., the air quality of a city, the traffic condition of a road segment, or even a question to be answered). Inevitably, conflicts exist among the information provided by different sources. Thus, how to automatically obtain the true information (i.e., the truths) of the objects from the conflicting multi-source information is a challenging research topic. To address this challenge, truth discovery techniques, which take the estimation of source reliability into account, have been proposed to infer the true information of objects from multi-source data [1]–[15]. Although these approaches are different, the same principle applies: If the information is provided by a reliable source, then it is more likely to be trustworthy; and if a source provides lots of trustworthy information, the source is more likely to be reliable.

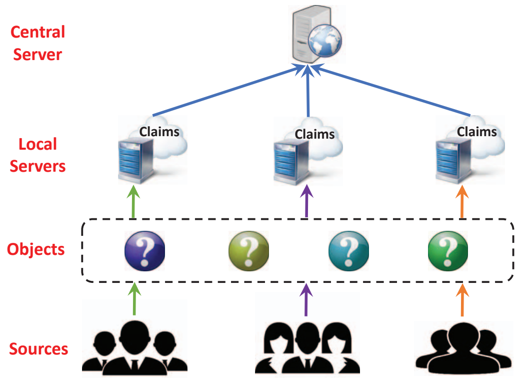


Figure 1: The Scenario of Distributed Truth Discovery.

In this paper, we consider a new scenario of truth discovery, namely, *distributed truth discovery*. In practice, the information (referred to as *claims*) about the observed objects provided by different sources are usually distributed across a bunch of local servers, as shown in Figure 1. Due to the concerns of communication overhead as well as privacy, in many cases it is not allowed to upload all the raw data stored in different local servers to the central sever.

Under this distributed setting, traditional truth discovery methods cannot be directly applied since they usually require all the information being gathered in a central server. To address this challenge, a straghtforward strategy can be adopted: We first separately run an existing truth discovery approach on each local server, and then apply majority voting or averaging on local estimated truths to obtain the final estimated truths in the central server. Unfortunately, this naive approach may not work well when the quality of most local servers is low. Clearly, the truths estimated by different servers would have different accuracy, due to the difference in the quantity and quality of their data. To achieve accurate estimation of the final truths, it is important

2374-8486/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDM.2017.60

to take the quality of local servers into consideration when aggregating data across distributed servers.

As aforementioned, to obtain accurate local truths, we can directly apply the state-of-the-art truth discovery approach in each local server. However, a drawback of most existing truth discovery approaches is that they do not distinguish the differences among objects, in other words, they treat all the objects equally. The object difference stems from two main factors: (1) The inner factors, i.e., the characteristics of the objects themselves, such as the difficulty level of questions. (2) The outer factors, such as the number of claims on each object and the quality of sources that make claims on this object. In practice, the differences among objects are unavoidable, *especially in the distributed environments*. They may significantly influence the local truth estimation. Thus, another challenge is how to properly model the difference among objects during the local truth estimation process.

To tackle the above challenges, in this paper, we propose a novel distributed truth discovery framework, named DTD, which incorporates the estimation of the uncertainty of each object's claims. This model can automatically infer true information of objects from distributed conflicting information. The proposed DTD consists of two components:

- *Local truth computation*, i.e., estmiating the local truths and variances of objects in each local server;
- *Central truth estimation*, i.e., inferring the final truths in the central server with the outputs from all the local servers.

In the local truth computation step, we propose an **u**ncertainty-**b**ased batch **t**ruth **d**iscovery approach (called UbTD), which models the differences among objects as the uncertainty values used for estimating the truths in local servers. Specifically, we assume that the sample variances can represent the uncertainty of objects and furthermore provide theoretical proof for this assumption. Based on the learned uncertainty values of the objects, for each source in the local server, we select a subset of objects with low uncertainty. These selected objects and their claims are used to calculate the reliability degrees of sources. Finally, for each server, the local estimated truths can be inferred according to the learned reliability degrees of sources.

In the central truth estimation step, we propose a new approach to infer the final truths of the objects, which simultaneously considers the quality of local servers, and the estimated truths as well as object variances uploaded by the local servers. Experimental results on three real world datasets show that the proposed DTD can efficiently and effectively estimate the true information of the objects, and the proposed UbTD significantly outperforms the state-of-the-art batch truth discovery approaches.

In summary, we make the following contributions:

- To the best of our knowledge, we are the first to propose a novel distributed truth discovery framework, named DTD, which can infer true information from noisy and conflicting distributed data.
- The proposed DTD framework neither uploads the raw data in local servers to the central server nor needs communications among local severs, which not only preserves data privacy but also reduces communication overhead.
- We empirically show that the proposed DTD framework can efficiently estimate object truths with accuracy guarantee in distributed environments. Moreover, the proposed UbTD outperforms existing truth discovery approaches on three real world datasets.

II. Problem Settings

To clearly define our problem, we introduce some terminologies and notations used in this paper. Since we aim to learn the true information of objects, the definition of an object is given as follows:

Definition 2.1: An object $n \in \mathcal{N}$ is an item of interest, such as a question, the departure or arrival time of a flight and the temperature of a city at a certain time, where $\mathcal{N}$ is the set of objects.

Different from existing truth discovery problem, we focus on inferring true information under the distributed environment. We assume that there are $\mathcal{L}$ local servers and only one central server. On each local server l, there are sources claiming on the $\mathcal{N}$ objects. Note that sources are *exclusive* among local servers. Then, we have the following definitions.

Definition 2.2: A source $s_l \in \mathcal{S}_l$ is a user or website on the l-th local server that can provide information for objects, where $\mathcal{S}_l$ is the set of sources on l.

Definition 2.3: A claim $x_n^{s_l}$ is a piece of information provided by the source s_l about the given object n on the l-th local sever.

In reality, the claims on the same object n contributed by different sources $\mathcal{S}_l$ on each local sever may be conflicting, which leads to a fact that the reliability of sources is different. Moreover, the data quality of different servers (i.e., the weights of local servers) may be different. Thus, the goal of this paper is to infer the true information (i.e., the truth) for each object n, estimate the reliability degree of source s_l on the local sever l, and learn the quality of each local server. Next, we give the definitions of the estimated truths, reliability degrees of sources on each server, and the weights of servers, which are the outputs of our problem.

Definition 2.4: The local estimated truth $\hat{\mu}_n^l$ is the estimated value for the given object n in the local sever l. The final estimated truth $\hat{\mu}_n$ is the estimated value of the object n in the central server.

Definition 2.5: The reliability degree r_{s_l} of a source s_l on the l-th local server measures the overall quality of claims provided by s_l. A larger r_{s_l} means that the claims provided by the source s_l are more trustworthy.

Definition 2.6: The weight of a local server w_l measures the overall data quality of the local server l. The greater w_l, the higher the data quality on the local sever l.

III. Methodology

In this section, we first introduce the proposed uncertainty-based batch truth discovery approach (UbTD), which runs on the local servers and outputs both the local estimated truth and variance for each object. Based on the estimated truths and variances collected from all the local servers, we can estimate the true information of objects with the proposed distributed truth estimation model DTD. In the following subsections, we will introduce UbTD and DTD in detail.

A. Uncertainty-based Truth Discovery Model

On each local server $l \in \mathcal{L}$, we aim to estimate the true information of each object and learn the reliability degree for each source. To achieve this goal, a simple way is that we directly apply existing truth discovery approaches. However, a drawback of existing models is that they do not consider the characteristics of objects, i.e, they treat all the objects equally. For some objects, such as questions, they have the different difficulty levels. Easy questions can easily obtain consistent answers provided by sources, but for hard questions, the answers may be multifarious. Thus, it is important to model the characteristics of objects (or the differences among objects) in the truth estimation process. Here, we use *uncertainty* to model the objects' characteristics or differences.

- ***Uncertainty***. An object's uncertainty, denoted by u_n^l, can be seen as how difficult or confident to infer the real true information of the object from the claims provided by sources on each local sever l. Note that even for the same object n, on different local servers, the uncertainty u_n^l may be different. Intuitively, the uncertainty of an object is mainly determined by the following two aspects: the difficulty of the object (i.e., the inner factor) and the claims on this object (i.e., the outer factor). On the one hand, if the object is extremely hard, it is difficult to infer its correct information for all the models. Thus, this object will be assigned a higher uncertainty value. On the other hand, if there are only a few sources providing claims on the object, then with insufficient data, the estimated truths do not have higher trustworthiness. It also can lead to a larger uncertainty value.

To mathematically define the uncertainty, we assume that for each object n, it has an underlying continuous distribution with mean μ_n and variance σ_n^2. Here, we treat the mean μ_n as the truth of n and the variance σ_n^2 as the *inner* factor, i.e., the difficulty. For each local server l, $\hat{\mu}_n^l$ is the estimated truth of the object n. Thus, the *outer* factor can be modeled as the square error between the estimated truth and the real true value, i.e., $(\hat{\mu}_n^l - \mu_n)^2$. To theoretically model the uncertainty, we give the following theorem:

Theorem 3.1: The variance $\hat{\sigma}_n^{l2}$ of claims provided by sources on the l-th local server is equal to $\sigma_n^2 + (\hat{\mu}_n^l - \mu_n)^2$.

Proof: See Appendix A for a detailed proof. ■

Actually, we can use $\frac{1}{|\mathcal{S}_n^l|}\sum_{s_l \in \mathcal{S}_n^l}(x_n^{s_l} - \hat{\mu}_n^l)^2$ to estimate $\hat{\sigma}_n^{l2}$, where $\mathcal{S}_n^l$ is the set of sources on the l-th server providing claims for the object n, and $|\mathcal{S}_n^l|$ is the size of $\mathcal{S}_n^l$. Based on the variance $\hat{\sigma}_n^{l2}$, we can formally define the uncertainty u_n^l as follows:

$$u_n^l = 1 - \exp(-\hat{\sigma}_n^{l2}). \tag{1}$$

From Theorem 3.1, we know that $\hat{\sigma}_n^{l2} = \sigma_n^2 + (\hat{\mu}_n^l - \mu_n)^2$. For the uncertainty in Eq. (1), if the object n is extremely hard, i.e $\sigma_n^2 \to \infty$, or the estimated truth $\hat{\mu}_n^l$ is far from the real truth μ_n, i.e $(\hat{\mu}_n^l - \mu_n)^2 \to \infty$, the uncertainty u_n^l will be 1; and if the object is very easy ,i.e $\sigma_n^2 \to 0$, and the estimated truth $\hat{\mu}_n^l$ is close to the real truth μ_n, i.e., $(\hat{\mu}_n^l - \mu_n)^2 \to 0$, then the uncertainty will be 0. Thus, the uncertainty is a real value from 0 to 1.

- ***Uncertainty-based Truth Discovery on the Local Server***. Taking the uncertainty of each object into consideration, we propose a novel uncertainty-based truth discovery approach UbTD. Generally, to solve the truth discovery framework, there are two steps: reliability degree estimation and truth computation. Next, we provide these two steps in detail.

Reliability Degree Estimation. Intuitively, sources $\mathcal{S}_l$ may not have the same confidence when claiming on the same object n. In order to correctly characterize the reliability of each source, we need to remove the objects with high uncertainty. Thus, we propose an uncertainty-based algorithm to sample claims. Based on the sampled claims, we estimate the reliability degree for each source.

Algorithm 1 shows the uncertainty-based sampling method. For each source s_l, we bootstrap a subset of objects which are claimed by the source s_l, based on objects' uncertainty. Then, we can obtain a new sampled dataset for calculating the source reliability.

Algorithm 1 Uncertainty-based Sampling Algorithm.

```
Input: Claims on local server $l$ : $\{x_n^{s_l}\}_{n\in\mathcal{N}}$ and uncertainty value $\{u_n^l\}_{n\in\mathcal{N}}$.
Output: Subset of Claims $\{\tilde{x}_n^{s_l}\}_{n\in\mathcal{N}}$
    Bootstrap subset of claims $\{\tilde{x}_n^{s_l}\}_{n\in\mathcal{N}}$ from $\{x_n^{s_l}\}_{n\in N}$ based on uncertainty value $\{u_n^l\}_{n\in\mathcal{N}}$
    return $\{\tilde{x}_n^{s_l}\}_{n\in\mathcal{N}}$.
```

In the reliability degree estimation step, we need to fix the truths $\{\hat{\mu}_n^l\}$. According to the principle of truth discovery, if the reliability degree of this source is high, then the estimated truths may be close to the claims provided by the source s_l; Otherwise, the claims may be far from the estimated truths. If the uncertainty of the object is large, then the source may not provide correct claim. Based on these two principles, we formulate the following objective function:

$$\begin{aligned} &\min_{\{r_{s_l}\}} \sum_{s_l \in \mathcal{S}_l} r_{s_l}^2 g(u_n^l, \tilde{x}_n^{s_l}, \hat{\mu}_n^l) \\ &\text{s.t.} \sum_{s_l \in \mathcal{S}_l} r_{s_l} = 1, r_{s_l} \geq 0, \forall s_l \in \mathcal{S}_l, \end{aligned} \tag{2}$$

where $g(\cdot) = \frac{\sum_{n \in \mathcal{N}}(1-u_n^l)(\tilde{x}_n^{s_l} - \hat{\mu}_n^l)^2}{\sum_{n \in \mathcal{N}}(1-u_n^l)}$ is the weighted error function, which is related to the uncertainty of objects $\{u_n^l\}$, the sampled claims $\{\tilde{x}_n^{s_l}\}$, and the estimated truths $\{\hat{\mu}_n^l\}$ on the local server.

To solve the above objective function in Eq. (2), we can adopt the method of Lagrange multipliers. The Lagrangian of Eq. (2) is given as follows:

$$L = \sum_{s_l \in \mathcal{S}_l} r_{s_l}^2 g(u_n^l, \tilde{x}_n^{s_l}, \hat{\mu}_n^l) + \lambda(\sum_{s_l \in \mathcal{S}_l} r_{s_l} - 1),$$

where λ is a Lagrange multiplier. Let the partial derivative of Lagrangian L with respect to r_{s_l} be 0, and we can obtain the reliability degree r_{s_l} of the source s_l as follows:

$$r_{s_l} \propto \frac{1}{g(u_n^l, \tilde{x}_n^{s_l}, \hat{\mu}_n^l)}. \tag{3}$$

Truth Computation. The final goal of truth discovery problem is to identify true information of objects from conflicting claims provided by multiple sources. If the source s_l has a larger reliability degree r_{s_l}, then the estimated truth $\hat{\mu}_n^l$ may be closer to the claim $x_n^{s_l}$ provided the source s_l. Based on this intuition, a commonly used strategy to calculate the truth is proposed as follows:

$$\hat{\mu}_n^l = \frac{\sum_{s_l \in \mathcal{S}_l} r_{s_l} x_n^{s_l}}{\sum_{s_l \in \mathcal{S}_l} r_{s_l}}. \tag{4}$$

The estimated truth $\hat{\mu}_n^l$ can be seen as the weighted mean of all the claims provided by sources on the object n. Note that when calculating truths, we use the raw data not the sampled data.

Algorithm 2 shows the flow of UbTD. It calculates the uncertainty value for each object according to Eq. (1), then bootstraps claims according to Algorithm 1 for each source, and computes reliability degrees of sources on the sampled claims. Finally, we obtain the estimated truths and variances of objects.

B. Distributed Truth Discovery

In most real-world applications, data about the same object may be stored or collected from a bunch of servers. In order to infer true information of objects located on multiple severs, we propose a novel truth discovery framework, called DTD, to deal with the distributed data. From Section III-A, for each local server l, we can obtain the estimated truths $\{\hat{\mu}_n^l\}$ and the variances $\{\hat{\sigma}_n^{l2}\}$ of objects. The benefits of uploading these two values from local servers $\mathcal{L}$ to the central sever are two folds: (1) This approach can significantly preserve data privacy. (2) They can characterize the data density about the object n on the local sever $l \in \mathcal{L}$, i.e., maintaining the original data properties on the local servers. Let $f_n^l(\cdot)$ represent the density function of the object n on the local server l with mean $\hat{\mu}_n^l$ and variance $\hat{\sigma}_n^{l2}$. Based on this density function, in the central sever, we can still calculate the distance between the claims on the local server l and the estimated final truth denoted as $\hat{\mu}_n$ of the object n, which can be formulated as follows:

$$d_n^l = \int (x - \hat{\mu}_n)^2 f_n^l(x) dx.$$

Similar to the proof of Theorem 3.1 in Appendix A, we can obtain the solution of $d_n^l = \hat{\sigma}_n^{l2} + (\hat{\mu}_n^l - \hat{\mu}_n)^2$, which illustrates that the distance is related to both the sample variance and the estimated truth of the object n on the local sever l.

Algorithm 2 Uncertainty-based Truth Discovery.

```
Input: Claims on local server $l$: $\{x_n^{s_l}\}_{n \in \mathcal{N}, s_l \in \mathcal{S}_l}$.
Output: Reliability of sources $\{r_{s_l}\}_{s_l \in \mathcal{S}_l}$, estimated truths
$\{\hat{\mu}_n^l\}_{n \in \mathcal{N}}$ and variances $\{\hat{\sigma}_n^{l2}\}_{n \in \mathcal{N}}$.
Initialize the estimated truths $\{\hat{\mu}_n^l\}_{n \in \mathcal{N}}$;
while Convergence criterion is not satisfied do
    for $n \leftarrow 1$ to $\mathcal{N}$ do
        Compute uncertainty value $u_n^l$ according to Eq. (1);
    end for
    for $s_l \leftarrow 1$ to $\mathcal{S}_l$ do
        Bootstrap claims $\{\tilde{x}_n^{s_l}\}_{n \in \mathcal{N}}$ according to Algorithm 1;
        Compute reliability $r_{s_l}$ according to Eq. (3);
    end for
    Compute estimated truths $\{\hat{\mu}_n^l\}_{n \in \mathcal{N}}$ according to Eq. (4);
end while
Compute variance $\{\hat{\sigma}_n^{l2}\}_{n \in \mathcal{N}}$;
return $\{r_{s_l}\}_{s_l \in \mathcal{S}_l}$, $\{\hat{\mu}_n^l\}_{n \in \mathcal{N}}$ and $\{\hat{\sigma}_n^{l2}\}_{n \in \mathcal{N}}$.
```

Actually, for the same object, the quality of claims on different local severs may be different because the sources claiming on the objects have different reliability degrees. If the data on the local server l are close to the estimated truth $\hat{\mu}_n$, then the local server may have high quality; otherwise, the quality of this local server is low. Thus, we need to assign a weight w_l to the local sever l for characterizing the overall quality of this server. To estimate the final true information of objects on the central server, we need to minimize the distances between the data on the local servers and the estimated final truths. Based on the distances $\{d_n^l\}$ and the weights $\{w_l\}$ of local severs, we have the following objective function on the central server:

$$\begin{aligned} &\min_{\{w_l\}, \{\hat{\mu}_n\}} \sum_{n \in \mathcal{N}} \sum_{l \in \mathcal{L}} w_l d_n^l \\ &\text{s.t.} \sum_{l \in \mathcal{L}} \exp(-w_l) = 1. \end{aligned} \tag{5}$$

The intuitions behind this objective function are as follows: (1) The proposed objective function minimizes the weighted distances between the claims on each local server and the estimated true information (i.e, the truths). If the local server l has a higher weight, then the estimated truth $\hat{\mu}_n$ on the central server should be close to the uploaded truth $\hat{\mu}_n^l$ from the local sever l in order to minimize the distances. (2) If the claims located on the local server l are close to the truth $\hat{\mu}_n$, then quality of the local server l may be high.

In the above objective function, i.e, Eq. (5), we have two sets of variables: the weights of local servers $\{w_l\}$ and the estimated truths $\{\hat{\mu}_n\}$ (in $\{d_n^l\}$). To solve this optimization problem, we adopt the commonly used block coordinate descent approach [16], which leads to an iterative solution consisting of the following two steps.

Local Server Weight Updates. In this step, we fix the estimated truth $\{\hat{\mu}_n\}$. We apply Lagrange multipliers to solve this optimization problem as follows:

$$L' = \sum_{n\in\mathcal{N}}\sum_{l\in\mathcal{L}} w_l d_n^l + \lambda'\big(\sum_{l\in\mathcal{L}}\exp(-w_l) - 1\big), \tag{6}$$

where λ' is a Lagrange multiplier. Let the partial derivative of L' with respect to w_l be 0, and then we can obtain the solution of w_l as follows:

$$w_l = -\log\Big(\frac{\sum_{n\in\mathcal{N}}(\hat{\sigma}_n^{l2} + (\hat{\mu}_n^l - \hat{\mu}_n)^2)}{\sum_{n\in\mathcal{N}}\sum_{l\in\mathcal{L}}(\hat{\sigma}_n^{l2} + (\hat{\mu}_n^l - \hat{\mu}_n)^2)}\Big). \tag{7}$$

Central Server Truth Updates. During this step, we fix the weights of local servers $\{w_l\}$. The estimated truth of the object n can be easily obtained, which is the weighted average of the uploaded estimated truth $\hat{\mu}_n^l$ from each local server l, i.e.,

$$\hat{\mu}_n = \frac{\sum_{l\in\mathcal{L}} w_l \hat{\mu}_n^l}{\sum_{l\in\mathcal{L}} w_l}. \tag{8}$$

Algorithm 3 shows the flow of the proposed DTD. The inputs are the estimated truths and variances from local servers. We first initialize the final estimated truths in the central server, and then iteratively update them and server weights until convergence.

Algorithm 3 **Distributed Truth Discovery.**

```
Input: Local estimated truths $\{\hat{\mu}_n^l\}_{n\in\mathcal{N},l\in\mathcal{L}}$ and variances $\{\hat{\sigma}_n^{l2}\}_{n\in\mathcal{N},l\in\mathcal{L}}$.
Output: Estimated truths $\{\hat{\mu}_n\}_{n\in\mathcal{N}}$ and server weights $\{w_l\}_{l\in\mathcal{L}}$ .
Initialize the estimated truths $\{\hat{\mu}_n\}_{n\in\mathcal{N}}$;
while Convergence criterion is not satisfied do
    Compute the server weights $\{w_l\}_{l\in\mathcal{L}}$ according to Eq. (7);
    for $n \leftarrow 1$ to $\mathcal{N}$ do
        Update the estimated truths $\hat{\mu}_n$ according to Eq. (8);
    end for
end while
return $\{\hat{\mu}_n\}_{n\in\mathcal{N}}$ and $\{w_l\}_{l\in\mathcal{L}}$.
```

IV. Experiments

In this section, we first introduce the three real-world datasets used in the experiments. Then, we introduce the baselines and the evaluation metrics. Finally, we conduct experiments on these datasets for validating the performance of the proposed UbTD and DTD, respectively. The experimental results show that the proposed UbTD outperforms the state-of-the-art batch truth discovery approaches. Moreover, we provide analysis on convergence of UbTD and validate the assumption of UbTD. The experimental results of DTD on the three datasets indicate that the proposed method is better than baselines on both accuracy and efficiency.

A. The Real-World Datasets

In order to evaluate the performance of the proposed methods, UbTD and DTD, we use three datasets in the experiments: one categorical dataset and two continuous datasets. In the following, we introduce these three datasets in detail.

The categorical dataset is named the Game dataset [7], [9], [17], which is collected from multiple online users who play an Android App based on a TV show called "Who Wants to Be a Millionaire". For each question shown on the app, there is only one correct answer provided by the TV game show, and a difficulty level is predefined.

The continuous datasets we used include the Weather dataset and the Flight dataset[1]. For the Weather dataset, high temperature forecast information for 88 big US cities are collected from HAM weather (HAM), Weather Underground (Wunderground), and World Weather Online (WWO). Besides the forecast information, real high temperature observations of each day are also collected as the ground truths for evaluation purpose. The Flight dataset is to extract departure and arrival information for 11,512 flights over 36 sources during one-month starting from December 2011. All the time information is translated into minutes. For example, "9:30 am" will be translated into 570 mins, and "9:30 pm" will be translated into 1,290 mins. The ground truth information is also available for evaluation. The statistics of these three real-world datasets are shown in Table I.

Table I: The Statistics of the Real-World Datasets.

	Game	Weather	Flight
# Sources	37,029	152	36
# Objects	2,103	7,568	138,586
# Observations	214,849	936,989	2,207,379

B. Baselines & Evaluation Measures

To fairly evaluate the performance of the proposed methods, we first describe the state-of-the-art truth discovery

[1] http://lunadong.com/fusionDataSets.htm

approaches as baselines, and then introduce the evaluation metrics.

Baseline Methods. In the experiments, we use both categorical and continuous datasets. For the continuous datasets, the baseline methods include Median, Mean, CATD [7], CRH [8], KDEm [18], ETCIBoot [17] and GTM [3]. For the categorical dataset, we use the following baseline methods: Voting, ETCIBoot [17], Accusim [4], 3-Estimates [6], CRH [8], Investment [19], CATD [7], Zencrowd [2], Dawid&Skene [1], and TruthFinder [5]. Details of these methods are introduced in the related work section. We also use a variant of the proposed UbTD as a baseline, called UbTD−, which reduces the sampling step (Algorithm 1) when calculating the reliability degrees of sources.

Performance Metrics. We applied the following metrics to compare the aggregated results with the ground truths. For the continuous data, we adopt both the mean absolute error (MAE) and the root of mean squared error (RMSE), and for the categorical data, we adopt the error rate as the performance metric. More details are given as follows:

- *MAE*: MAE uses ℓ_1-norm distance between the aggregated results and the ground truths. It penalizes more on the smaller errors.
- *RMSE*: RMSE uses ℓ_2-norm distance between the estimated truths and the ground truths, which penalizes more on the bigger errors.
- *Error Rate*: Error Rate is defined as the percentage of mismatched values between the aggregated results and the ground truths.

For all the above performance metrics, the lower values, the better performance.

C. *Experimental Results of UbTD*

We first validate the performance of the proposed UbTD on the categorical dataset, i.e., the Game dataset, and then show the experimental results on the two continuous datasets.

(1) Results on the Categorical Dataset

We analyze the performance of the proposed UbTD from three aspects: the accuracy, convergence and model assumption validation.

• ***Accuracy Analysis.*** For the Game dataset, each claim is a discrete value. In order to fit the input of the proposed UbTD, we follow the data preprocessing approach of [7], [17], i.e., transferring the discrete claims into probability vectors. Table II shows the error rate of all the methods on the Game dataset. From Table II, we can observe that the overall performance of the proposed UbTD is better than of all the baseline approaches in terms of *Error Rate*. Comparison between UbTD and UbTD− shows that it is important to sample a subset based on the uncertainty to calculate the reliability degrees of sources. For the baselines, ETCIBoot achieves the best results. However, in all the ten difficult levels, the proposed UbTD wins ETCIBoot on five levels and draws with ETCIBoot on two levels. For other baselines, the error rate of TruthFinder is larger than other methods' because this method is dramatically affected by the large number of lower quality claims. There are many users providing low quality answers and the number of conflicts is very large on the Game dataset. Thus, it leads to the poor performance of TruthFinder. The error rate of Investment is greater than that of Voting, as Investment estimates the probability of each claim being correct given each user's reliability without considering complement votes. Other baseline methods are all better than Voting but worse than UbTD.

• ***Convergence.*** Figure 2 shows the proposed UbTD can converge to a small error rate. The X-axis represents the number of iterations, and the Y-axis is the error rate. From Figure 2, we can observe that the error rate of UbTD is 0.0590 after only the first iteration, which is better than the performance of most baselines except ETCIBoot and CATD as shown in Table II. With the increase of the number of iterations, the error rate drops dramatically. After the fifth iteration, the error rate of UbTD is smaller than that of CATD. The performance of the proposed UbTD is better than that of the best baseline ETCIBoot after the thirteenth iteration. Finally, the overall error rate of UbTD steadily converges.

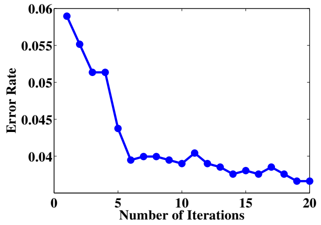


Figure 2: The Convergence on the Game Dataset.

• ***Model Validation***. In the proposed model UbTD, we assume that with the uncertainty of objects increases, the error rate also raises. To validate this assumption, we conduct the following experiment. For each object, we can obtain its corresponding uncertainty value. Then, for each uncertainty value, we round to the nearest tenths. Finally, we calculate the average error rate of all the objects with the same uncertainty value. Figure 3 shows the relationship between uncertainty and error rate. From Figure 3, we can observe that when the uncertainty value increases, the error rate increases. This observation is in accord with our assumption.

On the Game dataset, each question or object has a difficulty level. For difficult questions, users usually cannot answer correctly, i.e., higher error rate. Thus, using difficulty level information, we also can validate our assumption. We first obtain the uncertainty values for questions. Then, for

Table II: Performance Comparison on the Game Dataset.

Method	Error Rate										
	L1 (303)	L2 (295)	L3 (290)	L4 (276)	L5 (253)	L6 (218)	L7 (187)	L8 (138)	L9 (99)	L10 (44)	Overall (2103)
UbTD	**0.0132**	0.0305	**0.0241**	**0.0145**	**0.0395**	0.0550	**0.0374**	**0.0725**	0.1010	**0.0909**	**0.0366**
UbTD−	**0.0132**	**0.0271**	0.0276	0.0254	0.0474	0.0596	**0.0374**	0.1231	0.1111	0.2045	0.0456
ETCIBoot	0.0165	**0.0271**	**0.0241**	0.0217	**0.0395**	**0.0505**	0.0481	0.0870	**0.0707**	0.1364	0.0385
CATD	**0.0132**	**0.0271**	0.0276	0.0290	0.0435	0.0596	0.0481	0.1304	0.1414	0.2045	0.0485
CRH	0.0264	0.0271	0.0345	0.0435	0.0593	0.0872	0.0856	0.2609	0.3535	0.4545	0.0866
ZenCrowd	0.0330	0.0305	0.0345	0.0471	0.0593	0.0872	0.0856	0.2754	0.3636	0.5227	0.0899
AccuSim	0.0264	0.0305	0.0345	0.0507	0.0632	0.0963	0.0909	0.2826	0.3636	0.5000	0.0913
3-Estimates	0.0264	0.0305	0.0310	0.0507	0.0672	0.1055	0.0963	0.2971	0.3737	0.5000	0.0942
Dawid&Skene	0.0297	0.0305	0.0483	0.0507	0.0672	0.1101	0.0963	0.2971	0.3636	0.5227	0.0975
Voting	0.0297	0.0305	0.0414	0.0507	0.0672	0.1101	0.1016	0.3043	0.3737	0.5227	0.0980
Investment	0.0330	0.0407	0.0586	0.0761	0.0870	0.1239	0.1283	0.3406	0.3838	0.5455	0.1151
TruthFinder	0.0693	0.0915	0.1241	0.0942	0.1581	0.2294	0.2674	0.3913	0.5455	0.5455	0.1816

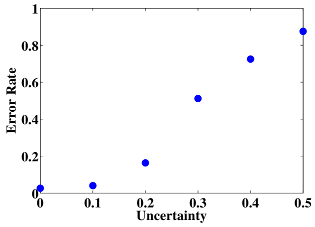


Figure 3: Error Rate w.r.t. Uncertainty on the Game Dataset.

the questions with the same difficulty level, we calculate the average uncertainty values of them. Figure 4 shows the validation results. We can observe that with the difficulty level increases, the average uncertainty raises. It can also validate the initial assumption.

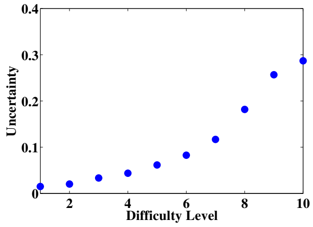


Figure 4: Uncertainty w.r.t. Difficulty Level on the Game Dataset.

(2) Results on the Continuous Datasets

We use two continuous datasets (the Weather and Flight dataset) to validate the performance of UbTD.

- ***Accuracy Analysis***. Table III shows the performance of the proposed UbTD and baselines. On the Weather dataset, the UbTD method achieves the best performance on both MAE and RMSE compared with baseline methods. Since the proposed UbTD incorporates the uncertainty of objects, it makes the proposed approach learn correct source reliability degrees, and in turn, estimate the accurate true information. Compared with the best baseline ETCIBoot, the proposed UbTD reduces MAE and RMSE 16.85% and 16.88%, respectively.

Table III: Performance Comparison on the Continuous Datasets.

Method	Weather		Flight	
	MAE	RMSE	MAE	RMSE
UbTD	**3.6725**	**4.7573**	**4.7149**	58.4209
UbTD−	3.6731	4.7583	4.7355	58.5230
Mean	4.7523	6.1378	8.2100	**51.5379**
Median	4.5791	5.9982	7.6030	58.0486
KDEm	4.4283	5.9153	11.0362	69.9530
GTM	4.4409	5.7567	7.9760	51.7872
ETCIBoot	4.4169	5.7237	8.9288	55.4703
CATD	4.6375	6.0453	6.7832	60.7814
CRH	4.5139	5.9088	7.7923	58.2416

On the Flight dataset, the proposed UbTD obtains the best performance on MAE. For the metric RMSE, the performance of the naive baseline Mean is the best. That is because the Flight dataset contains some special objects, arrival or departure time at “0:00 am” or “0:00 pm”. They are easily mixed, and the difference is around 720 minutes. Since most reliable sources also mistake these two values, it is hard to infer the correct value of this object for all the approaches. To fairly evaluate the performance for all the methods, we adopt another commonly used approach [12], [17] , Tolerance(ϵ), which needs to convert the continuous data to categorical data. Tolerance(ϵ) means that the estimated answer within an ϵ-minute difference of the ground truth can be seen as the correct one. Table IV shows the error rate with different ϵ's on the Flight dataset.

Table IV: Error Rate on the Flight Dataset.

Method	$\epsilon = 1$	$\epsilon = 5$	$\epsilon = 10$
UbTD	**0.0714**	**0.0267**	**0.0159**
UbTD−	0.0731	**0.0267**	0.0160
Mean	0.5880	0.2802	0.1165
Median	0.2840	0.2279	0.1541
GTM	0.4054	0.2665	0.1749
KDEm	0.3544	0.2994	0.2482
ETCIBoot	0.4403	0.2751	0.1565
CATD	0.1582	0.1405	0.1175
CRH	0.2846	0.2318	0.1649

From Table IV, we can see that the proposed UbTD decreases the error rate significantly on all ϵ's compared with baseline methods. When $\epsilon = 1$ min, UbTD can correctly estimate more than 90% of the flight arrival or departure time. With 10 minutes tolerance, UbTD can decrease to 1.59% error rate on all the objects, which reduces more than 86% of errors compared with best baseline method Mean. This big improvement shows the importance of incorporating uncertainty into truth discovery.

- ***Convergence***. To show the convergence of UbTD on the continuous data, we take the Flight dataset as an example shown in Figure 5. The X-axis denotes the number of iterations, and the Y-axis is the MAE value. We can observe that the UbTD converges within four iterations. The MAE reduces significantly in the first three iterations. It is because the proposed UbTD can learn accurate source reliability degrees within a small number of iterations, which helps UbTD to estimate correct answers.

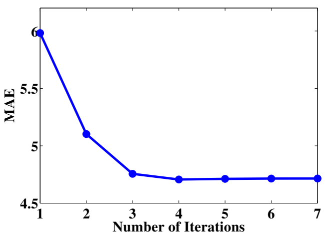


Figure 5: The Convergence on the Flight Dataset.

- ***Assumption Validation***. Similar to the validation on the categorical dataset, we validate our assumption on the continuous dataset. Here, we use the Flight dataset as an example, and the Weather dataset has the similar property. We first obtain the uncertainty levels or groups by rounding up the uncertainty for each object, and then calculate the average MAE of objects in the same group. Figure 6 shows the relationship between uncertainty and MAE value on the Flight dataset. The X-axis is the uncertainty, and the Y-axis denotes the MAE. In Figure 6, there is one special point whose coordinate is (1.0, 13.8705). As we analyze in the *Accuracy Analysis*, the objects "0:00 am" and "0:00 pm" are mixed, and their uncertainty values are all 1.0. In order to clearly show the trends of other uncertainty levels, we do not show this point in Figure 6. From Figure 6, we can observe that the uncertainty and MAE are highly related. With the increase of the uncertainty level, the MAE increases, which satisfies our assumption.

D. Experimental Results of DTD

In Section IV-C, we have shown that the proposed UbTD outperforms all the state-of-the-art truth discovery approaches in a central server. In this subsection, we aim to validate the performance of the proposed approach DTD under the distributed environment. We first introduce the experiment settings, and then validate the performance on both categorical on continuous datasets with accuracy and efficiency.

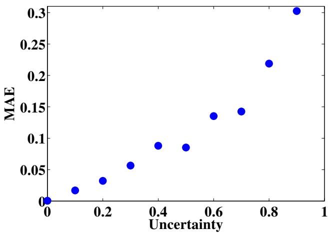


Figure 6: MAE w.r.t. Uncertainty on the Flight Dataset.

- ***Experiment Setups.*** To deploy the distributed environment, we use $\mathcal{L} > 1$ local servers and one central sever. On each local server l, we run the proposed UbTD with a subset of data[2]. Then, each local server only uploads its estimated truth set and variance of each object to the central server. Finally, the central server infers the final estimated truths for objects based on inputs from local servers. Note that there is no communications among local servers. We run DTD 10 times from the data partition to the truth estimation. Then we report the average results in the following experiments.

To validate the performance of the proposed DTD, we select three representative approaches as the baselines: the first two are CRH and CATD, and the last one is the proposed UbTD. For each baseline, we first run it on the local server, and then estimate the true information by simply averaging or voting.

- ***Accuracy Analysis on the Categorical Dataset.*** Table V lists the error rate with different $\mathcal{L}$'s on the Game dataset. We can observe that the proposed DTD can achieve the best performance compared with all the baselines. UbTD$_v$ means that we first run UbTD on each local server and then estimate the truths by *voting* all the outputs of local servers. Compared with UbTD$_v$, the performance of DTD improves significantly. It is because after partitioning the whole data into $\mathcal{L}$ groups by sources, there may be a few groups containing high quality sources, which leads to a fact that the truth estimated by different local servers may have different quality. The proposed DTD which models the quality of outputs from local servers can achieve better performance compared with UbTD$_v$. Compared with the other two baselines CRH$_v$ and CATD$_v$, UbTD$_v$ has lower error rates on different $\mathcal{L}$'s. This also illustrates that the proposed UbTD is better than the existing state-of-the-art batch truth discovery approaches. We use Figure 7 to clearly show the relationship between the number of local servers and the error rate. From Figure 7, we can observe that with the increase of the value of $\mathcal{L}$, the error rate also increases. It

[2] For each dataset, we randomly partition all the sources into $\mathcal{L}$ groups. All the claims provided by sources in group $\mathcal{L}$ are the input data of the l-th local server.

is reasonable because the larger $\mathcal{L}$ means that less data on the local servers. With insufficient data, the algorithms cannot correctly estimate the truths. Thus, the error rate increases.

Table V: Error Rate with Different $\mathcal{L}$'s on the Game Dataset.

Method	$\mathcal{L}=5$	$\mathcal{L}=10$	$\mathcal{L}=15$	$\mathcal{L}=20$
DTD	**0.0565**	**0.0615**	**0.0651**	**0.0658**
UbTD_v	0.0666	0.0713	0.0752	0.0757
CRH_v	0.0897	0.0894	0.0904	0.0933
CATD_v	0.0755	0.0795	0.0817	0.0837

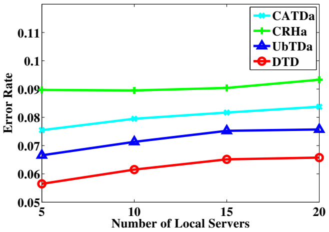


Figure 7: Error Rate w.r.t. Different $\mathcal{L}$'s on the Game Dataset.

Table VI: MAE with Different $\mathcal{L}$'s on the Continuous Datasets.

Dataset	$\mathcal{L}$	Method			
		DTD	UbTD_a	CRH_a	CATD_a
Weather	5	**3.8108**	3.8373	5.1667	4.2323
	10	**4.0456**	4.1117	6.5400	4.3735
	15	**4.1795**	4.2530	6.9867	4.4846
	20	**4.2926**	4.3735	7.1271	4.5326
Flight	4	**6.1313**	7.2108	56.5856	28.9426
	6	**6.4824**	7.6193	55.6556	35.2040
	8	**6.5700**	8.0353	77.6273	39.4831
	10	**6.8645**	7.8709	116.1163	33.4058

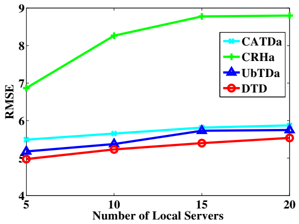


Figure 8: RMSE w.r.t. Different $\mathcal{L}$'s on the Weather Dataset.

- ***Accuracy Analysis on the Continuous Datasets.*** Table VI shows the MAE values of all the methods with different $\mathcal{L}$'s on the Weather and Flight datasets. For all the baselines, we first run the algorithm and then *average* all the outputs from local severs as the final truths. Since the number of sources on the Flight dataset is only 36, we set $\mathcal{L}$ as 4, 6, 8 and 10. From Table VI, we can see that the proposed distributed truth discovery method DTD achieves the best performance compared with baseline methods on the two continuous datasets. Even in the distributed scenario, the MAE values of the proposed DTD are smaller than those of most state-of-the-art batch methods as shown in Table III. The performance of all the approaches slightly decreases as the number of local servers increases, because local servers own less sources, and it is hard to infer true information with insufficient data. Compared with UbTD_a[3], we can see that the combination algorithm helps to improve the local truth discovery model further, and DTD can provide more stable results on the different number of servers.

Figure 8 shows the relationship between $\mathcal{L}$ and RMSE on the Weather dataset. Similar with the Error Rate on the Game dataset or MAE shown in Table VI, the RMSE increases with the increase of $\mathcal{L}$.

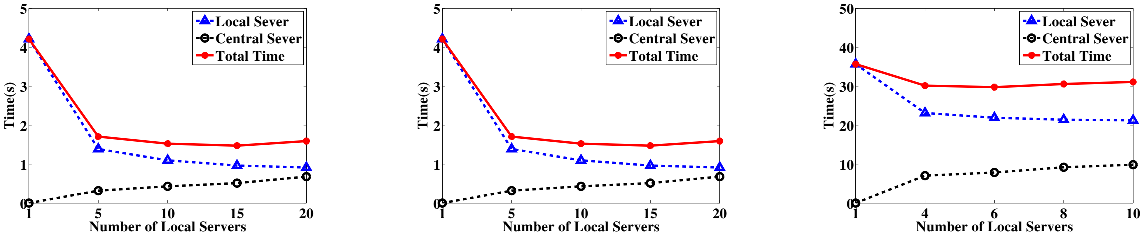
- ***Efficiency Evaluation.*** From the results of accuracy analysis in Table V and VI on both categorical and continuous datasets, we can safely conclude that even the number of servers is big, DTD still obtains high accuracy. In this section, we use running time to evaluate the efficiency of the proposed DTD. The running time of the proposed DTD includes two parts: the one denoted as t_l is to run UbTD on each local server l, and the other one t_c is for the combination on the central server. In order to show the relationship between running time and the number of local servers, we assume that the number of sources on each local server is relatively balanced. We record all the running time of different servers and take the maximum value as the local running time $t_m = \max\{t_1, t_2, \cdots, t_{\mathcal{L}}\}$. We also record the running time t_c on the central server. The final running time t is equal to $t_m + t_c$. Figure 9 shows the running time of DTD on all the three datasets. Note that when the number of servers is 1, it actually denotes the time of running on the whole datasets. Thus, we set the time of combination as 0. From Figure 9, we can observe that as the number of servers increases, the running time of local server significantly decreases, and the aggregation time in the center server increases steadily. However, the overall running time is still lower than that on the whole dataset. It illustrates that the proposed DTD can improve the efficiency of truth discovery with the accuracy guarantee.

V. Related Work

Truth discovery aims to resolve conflicts among the multi-source information, which has attracted significant attentions recently [1]–[9], [12], [13], [15], [20]–[25]. Truth Discovery estimates reliability degrees of sources from the data and infers the truth information simultaneously. The advantage of truth discovery compared with naive methods such as majority voting and averaging is that it incorporates the reliability degrees of sources, instead of treating each source equally.

In [8], the authors formulate the truth discovery problem into a optimization framework (CRH) and plug in different

[3] UbTD_a means that we first run UbTD on each local server and then estimate the truths by *averaging* all the outputs of local servers.



(a) Game Dataset

(b) Weather Dataset

(c) Flight Dataset

Figure 9: Running Time v.s. Number of Local Servers.

types of distance functions to tackle the heterogeneous data. CATD [7] also formulates the problem into a optimization framework and incorporates the long-tail phenomenon into the model. GTM [3] is a probabilistic model which is designed for the continuous data. TruthFinder [5] works on the categorical data and adopts the Bayesian-based heuristic algorithm. AccuSim [4] considers source correlations when sources are not independent and may copy each other. Investment [19] follows the idea that sources invest their reliability degrees into their claims. 3-Estimate [6] adopts the idea of complement votes and considers the difficulty of getting the truth when computing source weights. Dawid&Skene [1] and ZenCrowd [2] apply the Expectation-Maximization technique to update source weights and truths based on a confusion matrix. Unlike most truth discovery methods, ETCIBoot [17] uses the bootstrap procedure to provide confidence intervals instead of point estimators of truths. In [9], the authors proposed a fine-grained truth discovery method to handle the case that users have the various expertises on different levels. However, they only consider the topic differences among objects, and our proposed method considers more general differences brought by the inner and outer factors. In [14], [26], the authors propose the parallel versions of the truth discovery algorithms. However, their algorithms have to communicate among servers in each iteration, which is not efficient and easily causes the privacy concerns. The proposed distributed truth discovery framework (DTD) only needs to upload the estimated truths and variances from local servers to the central server. This can protect data privacy well and largely reduce the communication overhead.

VI. Conclusions

In the big data era, the huge volume of data are usually stored in multiple servers. The information distributed across the severs about the same object may be conflicting. In order to infer the true information of objects, truth discovery approaches can be applied. However, most of the existing truth discovery methods cannot work under distributed environments, and they ignore the differences among objects, which affects the accuracy of discovered truths. In order to tackle the aforementioned challenges, we proposed a distributed truth discovery framework (DTD) which can resolve conflicts among the data stored over distributed servers. It only uploads each local server's estimated truths to the central server, and thus reduces the communication overhead and protects the data privacy. In addition, we also design a new algorithm named UbTD, which models the differences among objects as uncertainty values. With the estimation of uncertainty, UbTD provides robust and accurate results. Experiments conducted on real world datasets show that our DTD framework can efficiently provide accurate estimated truths for distributed data. We also experimentally show that UbTD improves the performance compared with the state-of-the-art batch truth discovery approaches.

Acknowledgment

The authors would like to thank the anonymous referees for their valuable comments and helpful suggestions. This work is supported in part by the US National Science Foundation under grants IIS-1319973, CNS-1566374, and CNS-1652503. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

[1] A. P. Dawid and A. M. Skene, "Maximum likelihood estimation of observer error-rates using the em algorithm," *Applied Statistics*, pp. 20–28, 1979.

[2] G. Demartini, D. E. Difallah, and P. Cudré-Mauroux, "Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking," in *WWW*, 2012, pp. 469–478.

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

[4] X. L. Dong, L. Berti-Equille, and D. Srivastava, "Integrating conflicting data: the role of source dependence," *PVLDB*, vol. 2, no. 1, pp. 550–561, 2009.

[5] X. Yin, J. Han, and S. Y. Philip, "Truth discovery with multiple conflicting information providers on the web," *IEEE TKDE*, vol. 20, no. 6, pp. 796–808, 2008.

[6] A. Galland, S. Abiteboul, A. Marian, and P. Senellart, “Corroborating information from disagreeing views,” in *WSDM*. ACM, 2010, pp. 131–140.

[7] 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,” *PVLDB*, vol. 8, no. 4, pp. 425–436, 2014.

[8] 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 *SIGMOD*, 2014, pp. 1187–1198.

[9] 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 *SIGKDD*, 2015, pp. 745–754.

[10] S. Yao, M. T. Amin, L. Su, S. Hu, S. Li, S. Wang, Y. Zhao, T. Abdelzaher, L. Kaplan, C. Aggarwal, and A. Yener, “Recursive ground truth estimator for social data streams,” in *IPSN*, 2016, pp. 1–12.

[11] B. Zhao, B. I. Rubinstein, J. Gemmell, and J. Han, “A bayesian approach to discovering truth from conflicting sources for data integration,” *PVLDB*, vol. 5, no. 6, pp. 550–561, 2012.

[12] Z. Zhao, J. Cheng, and W. Ng, “Truth discovery in data streams: A single-pass probabilistic approach,” in *CIKM*, 2014, pp. 1589–1598.

[13] S. Zhi, B. Zhao, W. Tong, J. Gao, D. Yu, H. Ji, and J. Han, “Modeling truth existence in truth discovery,” in *SIGKDD*, 2015, pp. 1543–1552.

[14] R. W. Ouyang, L. M. Kaplan, A. Toniolo, M. Srivastava, and T. J. Norman, “Parallel and streaming truth discovery in large-scale quantitative crowdsourcing,” *IEEE TPDS*, vol. 27, no. 10, pp. 2984–2997, 2016.

[15] Y. Li, Q. Li, J. Gao, L. Su, B. Zhao, W. Fan, and J. Han, “On the discovery of evolving truth,” in *SIGKDD*, 2015, pp. 675–684.

[16] D. P. Bertsekas, *Nonlinear programming*. Athena scientific Belmont, 1999.

[17] 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 *SIGKDD*, 2016, pp. 1935–1944.

[18] M. Wan, X. Chen, L. Kaplan, J. Han, J. Gao, and B. Zhao, “From truth discovery to trustworthy opinion discovery: An uncertainty-aware quantitative modeling approach,” in *SIGKDD*, 2016, pp. 1885–1894.

[19] J. Pasternack and D. Roth, “Knowing what to believe (when you already know something),” in *Coling*, 2010, pp. 877–885.

[20] X. Li, X. L. Dong, K. Lyons, W. Meng, and D. Srivastava, “Truth finding on the deep web: Is the problem solved?” in *PVLDB*, vol. 6, no. 2, 2012, pp. 97–108.

[21] F. Ma, C. Meng, H. Xiao, Q. Li, J. Gao, L. Su, and A. Zhang, “Unsupervised discovery of drug side-effects from heterogeneous data sources,” in *SIGKDD*, 2017, pp. 967–976.

[22] 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 *SIGKDD*. ACM, 2014, pp. 601–610.

[23] H. Zhang, Q. Li, F. Ma, H. Xiao, Y. Li, J. Gao, and L. Su, “Influence-aware truth discovery,” in *CIKM*, 2016, pp. 851–860.

[24] Y. Li, J. Gao, C. Meng, Q. Li, L. Su, B. Zhao, W. Fan, and J. Han, “A survey on truth discovery,” *SIGKDD Explor. Newsl.*, vol. 17, no. 2, pp. 1–16, 2016.

[25] H. Xiao, J. Gao, Z. Wang, S. Wang, L. Su, and H. Liu, “A truth discovery approach with theoretical guarantee,” in *SIGKDD*, 2016, pp. 1925–1934.

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

Appendix

A. Proof of Theorem 3.1

Proof: For the continuous distribution, the variance $\hat{\sigma}_n^{l2}$ can be defined as

$$\hat{\sigma}_n^{l2} = \int (x - \hat{\mu}_n^l)^2 f_n(x)\mathrm{d}x \tag{9}$$

where $\hat{\mu}_n^l$ is the estimated truth of claims on the n-th object in the local serve l, and $f_n(\cdot)$ is the density function of the underlying distribution on the n-th object. Let μ_n be the mean of the underlying distribution on the object n, and we have

$$\begin{aligned}\hat{\sigma}_n^{l2} =& \int (x - \mu_n + \mu_n - \hat{\mu}_n^l)^2 f_n(x)\mathrm{d}x \\ =& \underbrace{\int (x-\mu_n)^2 f_n(x)\mathrm{d}x}_{T_1} + \underbrace{\int 2(x-\mu_n)(\mu_n - \hat{\mu}_n^l) f_n(x)\mathrm{d}x}_{T_2} \\ &+ \underbrace{\int (\mu_n - \hat{\mu}_n^l)^2 f_n(x)\mathrm{d}x.}_{T_3}\end{aligned} \tag{10}$$

In Eq. (10), for the first term, according to Eq. (9), we have that T_1 is the variance of the underlying distribution, i.e.,

$$T_1 = \int (x - \mu_n)^2 f_n(x)\mathrm{d}x = \sigma_n^2.$$

For the second term, we have

$$\begin{aligned}T_2 =& \int 2(x-\mu_n)(\mu_n - \hat{\mu}_n^l) f_n(x)\mathrm{d}x \\ =&2(\mu_n - \hat{\mu}_n^l) \int (x - \mu_n) f_n(x)\mathrm{d}x \\ =&2(\mu_n - \hat{\mu}_n^l)\Big(\int x f_n(x)\mathrm{d}x - \mu_n\Big),\end{aligned}$$

where $\mu_n = \int x f_n(x)\mathrm{d}x$, and thus, the value of T_2 is 0. For the last term, since $(\mu_n - \hat{\mu}_n^l)^2$ is a constant, we have

$$T_3 = \int (\mu_n - \hat{\mu}_n^l)^2 f_n(x)\mathrm{d}x = (\mu_n - \hat{\mu}_n^l)^2.$$

Thus, we have

$$\hat{\sigma}_n^{l2} = T_1 + T_2 + T_3 = \sigma_n^2 + (\mu_n - \hat{\mu}_n^l)^2.$$

■