Conflicts to Harmony: A Framework for Resolving Conflicts in Heterogeneous Data by Truth Discovery

Yaliang Li, Qi Li, Jing Gao, *Member, IEEE*, Lu Su, *Member, IEEE*, Bo Zhao, Wei Fan, and Jiawei Han, *Fellow, IEEE*

Abstract—In many applications, one can obtain descriptions about the same objects or events from a variety of sources. As a result, this will inevitably lead to data or information conflicts. One important problem is to identify the true information (i.e., the *truths*) among conflicting sources of data. It is intuitive to trust reliable sources more when deriving the truths, but it is usually unknown which one is more reliable *a priori*. Moreover, each source possesses a variety of properties with different data types. An accurate estimation of source reliability has to be made by modeling multiple properties in a unified model. Existing conflict resolution work either does not conduct source reliability estimation, or models multiple properties separately. In this paper, we propose to resolve conflicts among multiple sources of heterogeneous data types. We model the problem using an optimization framework where truths and source reliability are defined as two sets of unknown variables. The objective is to minimize the overall weighted deviation between the truths and the multi-source observations where each source is weighted by its reliability. Different loss functions can be incorporated into this framework to recognize the characteristics of various data types, and efficient computation approaches are developed. The proposed framework is further adapted to deal with streaming data in an incremental fashion and large-scale data in MapReduce model. Experiments on real-world weather, stock, and flight data as well as simulated multi-source data demonstrate the advantage of jointly modeling different data types in the proposed framework.

Index Terms—Data fusion, truth discovery, heterogeneous data

1 Introduction

Recently, the Big Data challenge is motivated by a dramatic increase in our ability to extract and collect data from the physical world. One important property of Big Data is its wide *variety*, i.e., data about the same object can be obtained from various sources. For example, customer information can be found from multiple databases in a company, a patient's medical records may be scattered in different hospitals, and a natural event may be observed and recorded by multiple laboratories.

Due to recording or transmission errors, device malfunction, or malicious intent to manipulate the data, data sources usually contain noisy, outdated, missing or erroneous records, and thus multiple sources may provide conflicting information. In almost every industry, decisions based on untrustworthy information can cause serious damage. For example, erroneous account information in a company database may cause financial losses; wrong diagnosis based on incorrect measurements of a patient may lead to serious consequences; and scientific discoveries may be guided to the wrong direction if they are derived from incorrect data. Therefore, it is critical to *identify the most trustworthy answers from multiple sources of conflicting information*. This is a non-trivial problem due to the following two major challenges.

1.1 Source Reliability

Resolving conflicts from multiple sources have been studied in the database community for years [1], [2], [3] resulting in multiple ways to handle conflicts in data integration. Among them, one commonly used approach to eliminate conflicts for categorical data is to conduct majority voting so that information with the highest number of occurrences is regarded as the correct answer; and for continuous values, we can simply take the mean or median as the answer. The issue of such Voting/Averaging approaches is that they assume all the sources are equally reliable, and thus the votes from different sources are uniformly weighted. In the complicated world that we have today, it is crucial to *estimate source reliability* to find out the correct information from conflicting data, especially when there exist sources providing low quality information, such as faulty sensors that keep emanating wrong data, and spam users who propagate false information on the Internet. However, there is no oracle telling us which source is more reliable and which piece of information is correct.

- *Y. Li, Q. Li, J. Gao, and L. Su are with the State University of New York at Buffalo, 338 Davis Hall, Buffalo, NY 14260. E-mail: {yaliangl, qli22, jing, lusu}@buffalo.edu.*
- *B. Zhao is with LinkedIn, 2029 Stierlin Ct., Mountain View, CA 94043. E-mail: bo.zhao.uiuc@gmail.com.*
- *W. Fan is with Baidu Research Big Data Lab, 1195 Bordeaux Drive, Sunnyvale, CA 94089. E-mail: fanwei03@baidu.com.*
- *J. Han is with the University of Illinois at Urbana-Champaign, 201 N Goodwin Ave, Urbana, IL 61801. E-mail: hanj@illinois.edu.*

Manuscript received 9 Oct. 2014; revised 16 Dec. 2015; accepted 16 Apr. 2016. Date of publication 27 Apr. 2016; date of current version 5 July 2016. Recommended for acceptance by N. Ramakrishnan.
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.2016.2559481

1.2 Heterogeneous Data

Motivated by the importance but lack of knowledge in source reliability, many truth discovery approaches have been proposed to estimate it and infer true facts [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23]. However, these approaches are mainly designed for single-type data and they do not take advantage of a joint inference on *data with heterogeneous types*.

In real data integration tasks, heterogeneous data is ubiquitous. An object usually possesses multiple types of data. For example, in the integration of multiple health record databases, a patient's record includes age, height, weight, address, measurements, etc; we may want to infer correct information for a city's population, area, mayor, and founding year among conflicting information presented on the Internet; and when we combine the predictions from multiple weather forecast tools, we need to resolve conflicts in weather condition, temperature, humidity, wind speed, wind direction, etc. In all these cases, the data to be integrated involve categorical, continuous or even more complicated data types.

Due to the wide existence of missing values, we usually do not have sufficient amount of data to estimate source reliability correctly purely from one type of data. When source reliability is consistent on the entire data set, which is often valid in reality, a model that infers from various data types together will generate accurate estimates of source reliability, which will in turn help infer accurate information. Therefore, instead of separately inferring trustworthy information for individual data types, we should develop a unified model that conducts a joint estimation on all types of data simultaneously.

However, it is non-trivial to unify different types of data in one model. During source reliability estimation, we need to estimate how close a source input is to the correct answer, but different data types should be treated differently in this process because the concept of closeness varies among different data types. For categorical data, each observation will be either correct or wrong (i.e., whether the observation is the same as or different from the true fact). It is very different when a property has continuous values. For example, if the true temperature is 80 F, then an observation of 79 F is closer to the true value than 70 F. If such differences are not taken into account and we regard each continuous input as a fact, we will inevitably make wrong estimates of source reliability and derive incorrect results. Therefore, we need a framework that can estimate information trustworthiness and take each data type's characteristics into account to seamlessly integrate data of heterogeneous data types.

1.3 Summary of Proposed CRH Framework

These observations motivate us to develop a **Conflict Resolution on Heterogeneous Data** (CRH) framework to infer the truths (also referred to as the true information or correct answers) from multiple conflicting sources each of which involves a variety of data types. We formulate the problem as an optimization problem to minimize the overall weighted deviation between the identified truths and the input. The weights in the objective function correspond to source reliability degrees. We propose to leverage heterogeneous data types by allowing any loss function for any type of data, and find out both truths and source reliability by solving the joint optimization problem. In the experiments (Section 3), we show that the proposed CRH framework outperforms existing conflict resolution approaches applied separately or jointly on heterogeneous data because each baseline approach either does not conduct source reliability estimation, or takes incomplete single-type data, or ignores the unique characteristics of each data type. The proposed CRH framework is also extended to work incrementally in streaming data scenario, and further a parallel version of CRH is presented to handle large-scale data sets.

In summary, we make the following contributions:

- We design a general optimization framework to model the conflict resolution problem on heterogeneous data by incorporating source reliability estimation. The proposed objective function characterizes the overall difference between unknown truths and input data while modeling source reliability as unknown source weights in the framework.
- Under this framework, weight assignment schemes are introduced to capture source reliability distributions. Various loss functions can be plugged into the framework to characterize different types of data. In particular, we discuss several common choices and illustrate their effectiveness in modeling conflict resolution on heterogeneous data.
- We propose an algorithm to solve the optimization problem by iteratively updating truths and source weights. We derive effective solutions for commonly used loss functions and weight assignment schemes, show the convergence of the algorithm, and demonstrate that the running time is linear in the number of observations. We propose an incremental version of CRH to fit streaming scenarios, and develop a parallel version of CRH under MapReduce model.
- We validate the proposed algorithm on both real-world and simulated data sets, and the results demonstrate the advantages of the proposed approach in resolving conflicts from multi-source heterogeneous data. The CRH framework can improve the performance of existing approaches due to its ability of tightly coupling various data types in the conflict resolution and source reliability estimation. Incremental CRH demonstrates similar accuracy to CRH but can run much faster and can be applied in real time to deal with never-ending streaming data. Parallel CRH's efficiency and ability to handle large-scale data are demonstrated by experimental results on Hadoop cluster.

2 Methodology

In this section, we describe our design of the CRH model, which computes truths and source weights from multi-source heterogeneous data. We formulate the conflict resolution problem as an optimization problem which models the truths as the weighted combination of the observations from multiple sources and incorporates a variety of loss functions for heterogeneous data types. An iterative weight and truth computation procedure is introduced to solve this optimization problem. Under this general framework, we further develop

both incremental and parallel versions of CRH for streaming and large-scale data sets.

2.1 Problem Formulation

We start by introducing important terms and defining the conflict resolution problem.

Definition 1. *An* object *is a person or thing of interest; a* property *is a feature used to describe the object; and a* source *describes the place where information about objects' properties can be collected. An* observation *is the data describing a property of an object from a source. An* entry *is a property of an object, and the* truth *of an entry is defined as its accurate information, which is unique.*

The mathematical notations are as follows. Suppose there are N objects, each of which has M properties whose data types can be different, and these objects are observed by K sources. The observation of the mth property for the ith object made by the kth source is $v_{im}^{(k)}$. The collection of observations made on all the objects by the kth source is $\mathcal{X}^{(k)}$, and it is a matrix whose imth entry is $v_{im}^{(k)}$. $v_{im}^{(*)}$ denotes the truth of the mth property for the ith object. The truths of all the objects on all the properties are stored in a truth table $\mathcal{X}^{(*)}$ whose imth entry is $v_{im}^{(*)}$.

To simplify the notations, we assume that the observations of all the sources about all the objects are available in the formulation. However, the proposed framework is general enough to cover the cases with missing observations. More discussion can be found in Section 2.5.

Definition 2 (Source Weights). *Source weights are denoted as $\mathcal{W} = \{w_1, w_2, \ldots, w_K\}$ in which w_k is the reliability degree of the kth source. A higher w_k indicates that the kth source is more reliable and observations from this source is more likely to be accurate.*

In real-world applications, ground truths and source reliability are usually unknown *a priori*. In ensemble learning [24] and mixture of experts [25], methods have been proposed to combine different learners (sources) in weighted manners, but these methods need supervision to derive the weights for sources. In contrast, existing truth discovery approaches [4], [5], [6], [7], [8], [9], [10], [11], [12], [13], [14], [15], [16] and the proposed method estimate source reliability in an unsupervised manner.

However, existing truth discovery approaches assume that the data has only one type. For heterogeneous data, if the source reliability estimation is conducted on individual properties separately, the estimated reliability result is not accurate enough due to insufficient observations. In the light of this challenge, the proposed framework unifies heterogeneous properties in the source reliability estimation. It will output both source weights and a truth table which are computed simultaneously by estimating source reliability from all the properties.

2.2 CRH Framework

The basic idea behind the proposed framework is that reliable sources provide trustworthy observations, so the truths should be close to the observations from reliable sources, and thus we should minimize the weighted deviation from the truths to the multi-source input where the weight reflects the reliability degree of sources. Based on this principle, we propose the following optimization framework that can unify heterogeneous properties in this process:

$$\begin{aligned} \min_{\mathcal{X}^{(*)}, \mathcal{W}} \quad & f(\mathcal{X}^{(*)}, \mathcal{W}) = \sum_{k=1}^{K} w_k \sum_{i=1}^{N} \sum_{m=1}^{M} d_m(v_{im}^{(*)}, v_{im}^{(k)}) \\ \text{s.t.} \quad & \delta(\mathcal{W}) = 1, \quad \mathcal{W} \in \mathcal{S}. \end{aligned} \tag{1}$$

We are searching for the values for two sets of unknown variables $\mathcal{X}^{(*)}$ and $\mathcal{W}$, which correspond to the collection of truths and source weights respectively, by minimizing the objective function $f(\mathcal{X}^{(*)}, \mathcal{W})$. There are two types of functions that need to be plugged into this framework:

- *Loss function*. d_m refers to a loss function defined based on the data type of the mth property. This function measures the distance between the truth $v_{im}^{(*)}$ and the observation $v_{im}^{(k)}$. It should output a high value when the observation deviates from the truth and a low value when the observation is close to the truth.
- *Regularization function*. $\delta(\mathcal{W})$ reflects the distributions of source weights. It is also required mathematically. If each source weight w_k is unconstrained, then the optimization problem is unbounded because we can simply take w_k to be $-\infty$. To constrain the source weights $\mathcal{W}$ into a certain range, we need to specify the regularization function $\delta(\mathcal{W})$ and the domain $\mathcal{S}$. Note that we set the value of $\delta(\mathcal{W})$ to be 1 for the sake of simplicity. Different constants for $\delta(\mathcal{W})$ do not affect the results, as we can divide $\delta(\mathcal{W})$ by the constant.

These two types of functions should be chosen based on our knowledge on the characteristics of heterogeneous data and the source reliability distributions, and more details about these functions will be discussed later. Intuitively, if a source is more reliable (i.e., w_k is high), high penalty will be received if this source's observation is quite different from the truth (i.e., difference between $v_{im}^{(*)}$ and $v_{im}^{(k)}$ is big). In contrast, the observation made by an unreliable source with a low w_k is allowed to be different from the truth. In order to minimize the objective function, the truths $\mathcal{X}^{(*)}$ will rely more on the sources with high weights.

The truths $\mathcal{X}^{(*)}$ and source weights $\mathcal{W}$ should be learned together by optimizing the objective function through a joint procedure. In an optimization problem that involves two sets of variables, it is natural to iteratively update the values of one set to minimize the objective function while maintaining the values of another set until convergence. This iterative procedure, referred to as block coordinate descent approach [26], will keep reducing the value of the objective function. To minimize the objective function in Eq(1), we iteratively conduct the following two steps.

Step I: Source Weights Update. With an estimation of the truths $\mathcal{X}^{(*)}$, we weight each source based on the difference between the truths and the observations made by the source:

$$\mathcal{W} \leftarrow \arg\min_{\mathcal{W}} f(\mathcal{X}^{(*)}, \mathcal{W}) \quad \text{s.t.} \quad \delta(\mathcal{W}) = 1, \quad \mathcal{W} \in \mathcal{S}. \tag{2}$$

At this step, we fix the values for the truths and compute the source weights that jointly minimize the objective function subject to the regularization constraints.

Step II: Truths Update. At this step, the weight of each source w_k is fixed, and we update the truth for each entry to minimize the difference between the truth and the sources' observations where sources are weighted by their weights:

$$v_{im}^{(*)} \leftarrow \arg\min_{v} \sum_{k=1}^{K} w_k \cdot d_m(v, v_{im}^{(k)}). \tag{3}$$

By deriving the truth using this equation for every entry, we can obtain the collection of truths $\mathcal{X}^{(*)}$ which minimizes $f(\mathcal{X}^{(*)}, \mathcal{W})$ with fixed $\mathcal{W}$.

Algorithm 1. CRH **Framework**

Input: Data from K sources: $\{\mathcal{X}^{(1)}, \ldots, \mathcal{X}^{(K)}\}$.
Output: Truths $\mathcal{X}^{(*)} = \{v_{im}^{(*)}\}_{i=1,m=1}^{N,M}$, source weights $\mathcal{W} = \{w_1, \ldots, w_K\}$.

```
Initialize the truths X^(*);
repeat
    Update source weights W according to Eq(2) to reflect sources' reliability based on the estimated truths;
    for i ← 1 to N do
        for m ← 1 to M do
            Update the truth of the ith object on the mth property v_im^(*) according to Eq(3) based on the current estimation of source weights;
        end for
    end for
until Convergence criterion is satisfied;
return X^(*) and W.
```

The pseudo code of CRH framework is summarized in Algorithm 1. We start with an initial estimate of truths and then iteratively conduct the source weight update and truth update steps until convergence. In the following, we explain the two steps in detail using example functions, and discuss the convergence and other practical issues.

2.3 Source Weight Assignment

First, we discuss the following regularization function:

$$\delta(\mathcal{W}) = \sum_{k=1}^{K} \exp(-w_k). \tag{4}$$

This function regularizes the value of w_k by constraining the sum of $\exp(-w_k)$. Suppose that the truths are fixed, the optimization problem Eq(1) with constraint Eq(4) is convex, and the global optimal solution is given by

$$w_k = -\log\left(\frac{\sum_{i=1}^{N}\sum_{m=1}^{M} d_m(v_{im}^{(*)}, v_{im}^{(k)})}{\sum_{k'=1}^{K}\sum_{i=1}^{N}\sum_{m=1}^{M} d_m(v_{im}^{(*)}, v_{im}^{(k')})}\right). \tag{5}$$

This weight computation equation indicates that a source's weight is inversely proportional to the difference between its observations and the truths at the log scale. The negative log function maps a number in the range of 0 and 1 to a range of 0 and ∞, so it helps to enlarge the difference in the source weights. A source whose observations are more often close to the truths will have a higher weight. Therefore, Eq(4) is a reasonable constraint function which leads to meaningful and intuitive weight update formula.

In order to distinguish source weights even better so that reliable sources can play a more important role in deriving the truths, we use the maximum rather than the sum of the deviations as the normalization factor when computing the weights. It still ensures that a source's weight is inversely proportional to the difference between its observations and the truths at the log scale.

The aforementioned weight assignment scheme considers a combination of sources. By setting different regularization functions, we can conduct source selection under the framework. For example, the following function defined based on L^p-norm can be used to select sources:

$$\delta(\mathcal{W}) = \sqrt[p]{w_1^p + w_2^p + \ldots + w_K^p} = 1, \quad w_k \in \mathbb{R}^+ \ (k = 1, \ldots, K), \tag{6}$$

where p is a positive integer. When p equals to 1 or 2, it corresponds to the most widely used L^1-norm or L^2-norm. If L^p-norm regularization is employed, the optimal value of the problem in Eq(1) will be 0, which is achieved when we select one of the sources and set its weight to be 1, set all the other source weights to be 0, and simply regard the chosen source's observations as the truths. Different from the regularization function shown in Eq(4), this regularization function does not combine multiple sources but rather assumes that there only exists one reliable source.

We can also incorporate integer constraints to conduct source selection with more than one source, i.e., choose j sources out of all K sources:

$$\delta(\mathcal{W}) = \frac{1}{j}(w_1 + w_2 + \ldots + w_K) = 1, \quad w_k \in \{0, 1\} \ (k = 1, \ldots, K). \tag{7}$$

If $w_k = 1$, the kth source is selected in truth computation, otherwise its observations will be ignored when updating the truths in the next step. Recent work [27] shows that both economical and computational costs should be taken into account when conducting source selection, which can be formulated as extra constraints in our framework. Due to the integer constraints defined in Eq(7), Eq(1) becomes an integer programming problem. The details of the solution are omitted here.

In many problems, we will benefit from integrating the observations from multiple sources, but there is a variation in the overall reliability degrees. Therefore, in this paper, we focus on the weight assignment scheme with max normalization factor where sources are integrated and variation is emphasized.

2.4 Truth Computation

The truth computation step (Eq(3)) depends on the data type and loss function. We respect the characteristics of each data type and utilize different loss functions to describe the deviation from the truths for different data

types. Accordingly, truth computation will differ among various data types. Below we discuss truth computation in detail based on several loss functions for categorical and continuous data, the two most common data types.

2.4.1 Categorical Data Type

On categorical data, the most commonly used loss function is 0-1 loss in which an error is incurred if the observation is different from the truth. Formally, if the mth property is categorical, the deviation from the truth $v_{im}^{(*)}$ to the observation $v_{im}^{(k)}$ is:

$$d_m(v_{im}^{(*)}, v_{im}^{(k)}) = \begin{cases} 1 & \text{if } v_{im}^{(k)} \neq v_{im}^{(*)}, \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Suppose that the weights are fixed, based on 0-1 loss function, to minimize the objective function at this step (Eq (3)), the truth on the mth property of the ith object should be the value that receives the highest weighted votes among all possible values:

$$v_{im}^{(*)} \leftarrow \arg\max_{v} \sum_{k=1}^{K} w_k \cdot \mathbb{1}(v, v_{im}^{(k)}), \tag{9}$$

where $\mathbb{1}(x, y) = 1$ if $x = y$, and 0 otherwise. This computation follows the principle that an observation stated by reliable sources will be regarded as the truth.

For the scenarios where multiple values of $v_{im}^{(*)}$ are probable, we introduce a strategy to incorporate probability into truth computation. This strategy is probabilistic-based and we assume that observations from reliable sources should have higher probability to be true. We represent categorical data by binary index vectors, which characterize the probability distributions of observations over all possible values. Formally, if the mth property has L_m possible values and $v_{im}^{(k)}$ is the lth value, then the index vector $I_{im}^{(k)}$ for $v_{im}^{(k)}$ is defined as:

$$I_{im}^{(k)} = (0, \ldots, \underset{l}{1}, 0 \ldots, 0)^T. \tag{10}$$

We can use squared loss function to describe the distance between the index vector $I_{im}^{(k)}$ and the truth vector $I_{im}^{(*)}$:

$$\begin{aligned} d_m(v_{im}^{(*)}, v_{im}^{(k)}) &= d_m(I_{im}^{(*)}, I_{im}^{(k)}) \\ &= (I_{im}^{(*)} - I_{im}^{(k)})^T (I_{im}^{(*)} - I_{im}^{(k)}), \end{aligned} \tag{11}$$

where $I_{im}^{(*)}$ denotes the probability distribution of the truths, in which $v_{im}^{(*)}$ is the corresponding value with the largest probability in $I_{im}^{(*)}$, i.e., the most possible value.

As the weights are fixed, the optimization problem Eq(1) with Eq(11) is convex. The optimal $I_{im}^{(*)}$ is the weighted mean of the probability vectors of all the sources:

$$I_{im}^{(*)} \leftarrow \frac{\sum_{k=1}^{K} w_k \cdot I_{im}^{(k)}}{\sum_{k=1}^{K} w_k}. \tag{12}$$

Comparing with the 0-1 loss strategy, this strategy gives a soft decision instead of a hard decision. However, this method has relatively high space complexity due to the representation of categories for input data.

2.4.2 Continuous Data Type

As for the continuous data, the loss function should characterize the distance from the input to the truth with respect to the variance of entries across sources. One common loss function is the normalized squared loss:

$$d_m(v_{im}^{(*)}, v_{im}^{(k)}) = \frac{(v_{im}^{(*)} - v_{im}^{(k)})^2}{std(v_{im}^{(1)}, \ldots, v_{im}^{(K)})}. \tag{13}$$

Suppose that the weights are fixed, the optimization problem Eq(1) with Eq(13) is convex. The truth that minimizes the overall weighted distance should be the weighted average of the observations:

$$v_{im}^{(*)} \leftarrow \frac{\sum_{k=1}^{K} w_k \cdot v_{im}^{(k)}}{\sum_{k=1}^{K} w_k}. \tag{14}$$

This truth computation strategy simulates the idea that observations from a reliable source should contribute more to the computation of the truth. However, this method is sensitive to the existence of outliers, and thus can only work well in the data set in which outliers are removed.

To mitigate the effect of outliers, we can use the normalized absolute deviation as the loss function:

$$d_m(v_{im}^{(*)}, v_{im}^{(k)}) = \frac{|v_{im}^{(*)} - v_{im}^{(k)}|}{std(v_{im}^{(1)}, \ldots, v_{im}^{(K)})}. \tag{15}$$

Based on this, the truth that minimizes the overall weighted absolute deviation should be the weighted median. Specifically, we use the following definition of weighted median [28, Chapter 9]. Given a set of numbers $\{v^1, ..., v^K\}$ with weights $\{w_1, ...w_K\}$, the weighted median of this set is the number v^j, such that

$$\sum_{k:v^k < v^j} w_k < \frac{1}{2}\sum_{k=1}^{K} w_k \quad \& \quad \sum_{k:v^k > v^j} w_k \leqslant \frac{1}{2}\sum_{k=1}^{K} w_k. \tag{16}$$

The sum of weights on the numbers that are smaller than the weighted median, and the sum of weights on the numbers that are greater than the weighted median should both be roughly half of the total weights on the whole set. To find the weighted median, we compare the cumulative sum computed on numbers smaller than v^j or greater than v^j. Note that conventional median can be regarded as a special case where we give the same weight to all the numbers so that median becomes the number separating the higher half from the lower half. It is known that median is less sensitive to the existence of outliers, and thus the weighted median approach for truth computation is more desirable in noisy environments.

Besides the aforementioned loss functions, the proposed general framework can take any loss function that is selected based on data types and distributions. Some other examples include Mahalanobis distance for continuous data, edit distance or KL divergence for text data, etc. To deal with complex data types, we can either use loss functions defined on raw data or on abstraction of raw data, such as motifs in time series, frequent sub-graphs in graphs, and segments in images. The framework can even be adapted to take the

ensemble of multiple loss functions for a more robust loss computation. We can also convert a similarity function into a loss function, which allows the usage of numerous techniques in similarity computation developed in the data integration community.

2.5 Discussions & Practical Issues

Here we discuss several important issues to make the framework practical and analyze the time complexity of the proposed CRH framework.

Initialization. The initialization of the truths can be obtained using existing conflict resolution methods. In our experiments, we find that the results from Voting/Averaging approaches is typically a good start.

Convexity and convergence. The convexity depends on the loss functions and regularization function. An example of a family of convex loss functions is *Bregman divergence* [29], which includes a variety of loss functions such as squared loss, logistic loss, Itakura-Saito distance, squared Euclidean distance, Mahalanobis distance, KL-divergence and generalized I-divergence. Using several loss functions discussed in this paper, we prove the convergence of the CRH framework as follows. When Eq(4) is used as constraint, Eq(11) or/and Eq(13) is/are used as loss functions, the convergence of CRH framework is guaranteed, and detailed proof can be found at [30]. Although the analysis on non-convex or non-differentiable functions need to be conducted differently [31], [32], we find that some of these approaches work well in practice, such as the absolute deviation for continuous data.

In Algorithm 1, the convergence criterion is that the decrease in the objective function is small enough compared with the previous iteration. In the experiments, we find that the convergence of CRH is easy to judge because the first several iterations incur a huge decrease in the objective function, and once it converges, the results become stable.

Normalization. Another important issue is the normalization of deviations on each property. As illustrated in the weight computation equation (Eq(5)), we need to sum up the deviations to the truths across different properties. If various loss functions applied on different properties have significantly different scales, the weight computation will be biased towards the property that has bigger range in the deviation. To solve this issue, we normalize the output of each loss function on each property so that the deviation computed on all the properties fall into the same range.

Missing values. Note that for the sake of simplicity, we assume that all the sources observe all the objects on all the properties in the proposed optimization framework (Eq(1)), but it can be easily modified to handle missing values when different sources observe different subsets of the objects on different subsets of properties. When the number of observations made by different sources is quite different, we can normalize the overall distance of each source by the number of observations.

Source weight consistency. Similar to existing truth discovery methods, CRH framework makes the source weight consistency assumption, which assumes that a source provides truths for all the objects and properties with the same probability. If the assumption does not hold[15], [33], [34], we can adapt the proposed CRH framework to handle such cases by dividing w_k into fine-grained weights, each of which corresponds to a local reliability degree of the source on a subset of properties or objects.

Time complexity. The running time of CRH framework might vary with respect to different loss functions and regularization functions. If we utilize 0-1 loss function Eq(8) and log regularization function Eq(5), the running time is linear with respect to the total number of observations, i.e., $O(KNM)$, where K is the number of sources, N is the number of objects, and M is the number of properties.

2.6 Incremental CRH

In many real truth discovery scenarios, data from multiple sources are collected in a "streaming" manner, i.e., data arrive in sequential chunks. For example, when we crawl the weather prediction information from multiple websites, the data are collected day by day. It is impractical to wait until all the data are collected to estimate source reliability and find the truths. To fit such scenarios, we modify the proposed CRH framework so that the truths and source weights can be learned incrementally. This incremental framework can also be applied to huge data sets that can only tolerate one sequential scan of the data sets.

The basic idea is to obtain the truths for current chunk of data based on the source weights learned from historical data and update the source weights accordingly without revisiting the past data. Applying this in CRH framework, we modify the source weights and truths update steps to conduct incremental CRH (I-CRH). Specifically, for each new chunk of data, we first use the source weights learned from previous data to update the truths, and then update source weights based on the difference between the latest truths and the observations. Note that the time window for data collection decides the size of each data chunk.

To control the effect of past data on truth discovery, we introduce a parameter in the I-CRH method: *Decay rate* α determines the impact of the historical data on current source weights estimation. Intuitively, the recent data should play a more important role in source weight estimation than the past data. Therefore, we introduce the decay parameter $\alpha \in [0, 1]$. The smaller α, the less impact from past data in current source weights estimation.

The pseudo code of the I-CRH framework is summarized in Algorithm 2. We start with an initialization of source weights, and then conduct the truth computation and source weight update steps for each data chunk.

Comparing with the CRH framework, I-CRH method is more efficient. Although the proposed CRH framework runs linearly with respect to the number of observations, it requires several iterations to update truths and source weights. The I-CRH, however, runs only one iteration for each data chunk. The accuracy of the result obtained by I-CRH method is expected to be a bit lower than CRH method because I-CRH only scans data once and have less computation steps, but this leads to better efficiency. I-CRH is ideal in the scenarios when high-volume data arrives at fast speed and requires fast processing.

2.7 Parallel CRH

Nowadays, with the explosion of data and the development of computation devices, it becomes necessary to take the advantage of distributed and parallel computing systems to

process large-scale data sets. In this section, we adapt the proposed CRH framework to parallel paradigm.

Algorithm 2. Incremental CRH Method

Input: Stream data: $\{\mathcal{D}_1, \mathcal{D}_2, \ldots\}$ where $\mathcal{D}_l = \{\mathcal{X}_l^{(1)}, \ldots, \mathcal{X}_l^{(K)}\}$, and decay rate α

Output: Truths for the stream data $\{\mathcal{X}_1^{(*)}, \mathcal{X}_2^{(*)}, \ldots\}$ where $\mathcal{X}_l^{(*)} = \{v_{iml}^{(*)}\}_{i=1,m=1}^{N_l,M_l}$.

1: Initialize source weights $w_k = 1$, and set the accumulated distance for each source $a_k = 0$;
2: **for** each timestamp l **do**
3: Compute the truth for each entry in the current data: $v_{iml}^{(*)} \leftarrow \arg\min_v \sum_{k=1}^{K} w_k \cdot d_m(v, v_{iml}^{(k)})$;
4: Update accumulated distance for each source: $a_k \leftarrow a_k * \alpha + \sum_{i=1}^{N}\sum_{m=1}^{M} d_m(v_{iml}^{(*)}, v_{iml}^{(k)})$;
5: Update source weight w_k according to the accumulated distance;
6: **end for**
7: **return** $\mathcal{X}^{(*)}$.

Many parallel programming models are feasible to parallelize CRH method. Here we particularly discuss the adaptation of CRH method to fit MapReduce framework [35], which has been widely used for large-scale data analysis on the cloud. In MapReduce, the following steps are executed in parallel: 1) the map stage scans the input and output (key, value) pairs; and 2) the reduce stage processes the (key, value) pairs and outputs the final results.

It is obvious that the truth computation step can be executed independently for each object and thus this step is easy to parallelize. The source weight assignment step can be expressed using summation form [36], and thus it can be parallelized by aggregating partial sums. Next we will illustrate the details of the parallel CRH method. Note that the following procedure can work with various loss functions and regularization functions.

2.7.1 Data Format

We first describe the format of data sets as input to parallel CRH method. Since there is information from different sources for a particular entry, to be general, we assume that the input is a tuple of three elements: the ID of the entry (denote as eID), the information from a particular source about this entry (denote as v), and the ID of this particular source (denote as sID). Then each tuple of the input is denoted as (eID, v, sID).

The parallel CRH framework still works in an iterative procedure, and each iteration consists of truth computation and weight assignment steps. Next, we describe the details of these two steps for parallel CRH.

2.7.2 Truth Computation

For truth computation step, we need one MapReduce procedure. First, in Map function, all the input tuples are reorganized into a key/value pair, where the key is the ID of an entry, and the value includes the rest information. Before the key/value pairs are fed into Reducers, they are grouped according to the key values. The pairs with the same key, i.e., the same entry ID, will go to the same Reducer. Then we can easily calculate the truth for each entry based on Eq (3). Note that in the truth computation formula, we also need the source weight information. It is kept in an external file and all Reducer nodes can read it. Initially, source weights are set uniformly ($\frac{1}{K}$ for all sources). This file will then be updated by source weight assignment step as described below. Based on the source weights and key/value pairs with the same entry ID, the Reducer computes the truth for each entry, and outputs key/value pairs, where the key is entry ID and the value is the corresponding truth.

2.7.3 Source Weight Assignment

For the source weight update step, we also need one MapReduce procedure. The general idea is that in Map phase, all the partial errors are computed; then in Reduce phase, the partial errors are summed up. The input format for Mappers is the same as the one in truth computation, i.e., the tuple (eID, v, sID). As shown in Eq(5), the source weights are a function of their errors, which is computed based on the estimated truths and their claimed values: the claimed values can be directly read from the input tuples; the estimated truths, similar to the shared source weights, can be kept in an external file and all Mapper nodes can access it. For each input tuple, the Mapper simply compares the claimed value with the estimated truths and emits the partial errors according to the adopted distance function. Before the $(sID, error)$ pairs are fed into Reducers, they will be sorted by Hadoop. Thus the pairs with the same key, i.e., the same source ID, will feed to the same Reducer. Then we can aggregate all partial errors for each source. As sources may not have claims on all entries, the aggregated errors should be normalized by the number of sources' observations.

As the number of observations can be quite large, the overhead caused by the sorting operation and communication will dominate the running time. In order to reduce the overhead, we further implement a *Combiner* function for source weight assignment. This Combiner function is quite similar to the Reducer, and the only difference is that it does not sum up all the errors, instead, just part of the partial error pairs within each Mapper.

2.7.4 Wrapper Function

To make all the above functions work together, we need a wrapper function to control the iterative procedure. The source weights are initialized by equal values and these weights will be read by the Mappers in the truth computation step. After the truth computation step, we have the estimated truths for all the entries, and this information is written into an external file. Then by comparing the estimated truths and input tuples, the source errors (and source weights) are re-calculated, and the external source weight file will be updated. We repeat the whole procedure until the estimated truths converge or the iteration number meets the threshold we set up.

2.8 Summary

Our major contribution is that we unify data of various types in truth discovery to resolve conflicts on heterogeneous data. The proposed optimization framework (Eq(1)), which targets at minimizing overall weighted difference

between truths and input data, provides a nice way to combine data of various types when deriving source weights and truths. Under this general framework, we discussed several common data types and loss functions, derived effective solutions, and analyzed its convergence. Different from existing truth discovery approaches that focus on facts [4], [5], [9], [10] or continuous data [14], the proposed CRH model learns source reliability degrees jointly from various properties with different data types. Unique characteristics of each data type are considered, and all types contribute to source reliability estimation together. This joint inference improves source reliability estimation and leads to better truth discovery on heterogeneous data.

Other contributions include the development of incremental and parallel versions of CRH method. They are designed for streaming or offline large-scale data sets. They can run by scanning data only once or access the data in parallel, which greatly reduce running time.

3 Experiments

In this section, we report the experimental results on both real-world and simulated data sets, which show that the proposed CRH method is efficient and outperforms state-of-the-art conflict resolution methods when integrating multiple sources of heterogeneous data. We first discuss the experiment setup in Section 3.1, and then present experimental results for CRH method in Section 3.2. Further, the experiments for incremental and parallel CRH methods are reported in Sections 3.3 and 3.4 respectively.

3.1 Experiment Setup

In this part, we present the performance measures and discuss the baseline methods.

3.1.1 Performance Measures

The problem setting is that we have multi-source input and the ground truths. All the conflict resolution methods are conducted in an unsupervised manner in the sense that the ground truths will only be used in evaluation. In this experiment, we focus on two types of data: categorical and continuous. To evaluate the performance of various conflict resolution methods, we adopt the following measures for these two data types:

- Error rate: For categorical data, we use *Error Rate* as the performance measure of an approach, which is computed as the percentage of the approach's output that are different from the ground truths.
- MNAD: For continuous data, we can measure the overall absolute distance from each method's output to the ground truths, which indicates how close the output are to the ground truths. As different entries may have different scales, we normalize the distance on each entry by its own variance, and then calculate their mean. This leads to the measure *Mean Normalized Absolute Distance* (*MNAD*).

For both measures, the *lower* the value, the closer the method's estimation is to the ground truths and thus the *better* the performance.

3.1.2 Baseline Methods

For the proposed CRH method, we use weighted voting (Eq (9)) for categorical data due to its time and space efficiency. On continuous data, we use weighted median (Eq(16)), which is efficient and robust in noisy environment with outliers. Weight assignment is computed by the inverse logarithm of the ratio between the deviation to the truth and the maximum distance so that the difference in source reliability is emphasized. We compare the proposed approach with the following baseline methods that cover a wide variety of ways to resolve conflicts. These approaches can be partitioned into three categories.

- *Conflict resolution methods applied on continuous data only*. The following approaches can only be applied on continuous data, and thus they will ignore the input from categorical properties. Mean and Median are traditional conflict resolution approaches that simply take the mean or median of all observations on each property of each object as the final output, while Gaussian Truth Model (GTM) [14] is a Bayesian probabilistic model based truth discovery approach especially designed for continuous data.
- *Conflict resolution methods applied on categorical data only*. We apply majority voting approach, which takes the value that has the highest number of occurrences as output, on categorical properties only. This is the traditional way of resolving conflicts in categorical data without source reliability estimation.
- *Conflict resolution methods by truth discovery*. Many of the existing truth discovery approaches are developed to find true "facts" for categorical properties. However, we can enforce them to handle data of heterogeneous types by regarding continuous observations as "facts" too. Among these methods, Investment [9] and PooledInvestment [9] "invest" a source's reliability uniformly on the observations it provides and the confidence of an observation is a non-linear (Investment) or linear (PooledInvestment) function defined on the sum of invested reliability from its providers. 2-Estimates [5] takes the assumption that "there is one and only one true value for each entry", and 3-Estimates [5] improves 2-Estimates by considering the difficulty of getting the truth for each entry. Both TruthFinder [4] and AccuSim [10] adopt Bayesian analysis, and similarity function is used to adjust the vote of a value by considering the influences between facts. Meanwhile, AccuSim considers complement vote which is adopted by 2-Estimates and 3-Estimates. Note that in [10], other algorithms have been proposed to tackle source dependency issues in resolving conflicts, which are not compared here because we do not consider source dependency in this paper but leave it for future work.

The comparison between the proposed framework with these baseline approaches on heterogeneous data can show that 1) using both types of data jointly gives better source reliability estimation than using individual data types separately, but 2) an accurate weight can only be obtained by taking unique characteristics of each data type into consideration.

TABLE 1
Statistics of Real-World Data Sets

	Weather Data	Stock Data	Flight Data
# Observations	16,038	11,748,734	2,790,734
# Entries	1,920	326,423	204,422
# Ground Truths	1,740	29,198	16,572

We implement all the baselines and set the parameters according to their authors' suggestions. All the experimental results in this section except for MapReduce experiments are conducted on a Windows machine with 8G RAM, Intel Core i7 processor.

3.2 Experimental Results of CRH Method

In this section, by comparing the proposed CRH approach with the baseline methods, we show the power of simultaneously modeling various data types in a joint framework on both real-world and simulated data sets. We also show the efficiency of the proposed approach on single machine and Hadoop cluster.

3.2.1 Real-World Data Sets

We use three real-world data sets to demonstrate the effectiveness of the proposed method.

Weather forecast data set. Weather forecast integration task is a good test bed because the data contains heterogeneous types of properties. Specifically, we integrate weather forecasting data collected from three platforms: Wunderground,[1] HAM weather,[2] and World Weather Online.[3] On each of them, we crawl the forecasts of three different days as three different sources, so altogether there are nine sources. For each source, we collected data of three properties: high temperature, low temperature and weather condition, among which the first two are continuous and the last is categorical. To get ground truths, we crawl the true weather information for each day. We collected the data for twenty US cities over a month.

Stock data set. The stock data [11], crawled on every work day in July 2011, consists of 1,000 stock symbols and 16 properties from 55 sources, and the ground truths are also provided. Here, we treat the data set as heterogeneous. More specifically, property *volume*, *shares outstanding* and *market cap* are considered as continuous type, and the rest ones are considered as categorical type.

Flight data set. The flight data [11], crawled over one-month period starting from December 2011, consists of 1,200 flights and 6 properties from 38 sources. We conduct pre-processing on the data to convert the gate information into the same format and the time information into minutes. The ground truths are also available. In this work, we show results on the flight data by treating gate information as categorical type and time information as continuous type. Note that we have a different task setting compared with [11] for Stock and Flight data when we treat them as heterogeneous types.

Table 1 shows the statistics of these three data sets. Note that the number of entries does not equal to the number of ground truths because we only have a subset of entries labeled with ground truths. The ground truths are not used by any of the approaches, but only used in the evaluation.

In Table 2, we summarize the performance of all the methods in terms of *Error Rate* on categorical data and *MNAD* on continuous data for three real-wold data sets. Although our approach outputs truths on both data types simultaneously, we evaluate the performance separately on these two data types due to the different measures for different data types. It can be seen that the proposed CRH approach achieves better performance on both types of data compared with all the baselines. For example, on weather data, the number of mismatches from ground truths drops from 266 (the best baseline) to 218 out of 580 entries by using CRH (on categorical data). On Stock and Flight data sets where baselines have already achieved good performance, we still can see the performance improvement of CRH over the best baseline (1,719 → 1,657 out of 23,677, and 427 → 414 out of 4,971). Similarly, the gain on continuous data can be consistently observed on all three data sets.

By outperforming various conflict resolution approaches applied separately on categorical data and continuous data, the proposed CRH approach demonstrates its advantage in modeling source reliability more accurately by jointly inferring from both types of data. GTM can not estimate source reliability accurately merely by continuous data which may not have sufficient information. This also justifies our assumption that each source's reliability on continuous and categorical data is consistent so the estimation over different data types complements each other.

The reason that the proposed CRH approach beats the other conflict resolution approaches that are applied on both types of data is that these approaches cannot capture the unique characteristics of each data type. This is further supported by the fact that the performance of those approaches is relatively better on categorical data, but deviates more from the truths on the continuous data. In contrast to existing approaches, the proposed CRH framework can take data type into consideration, which will provide a better estimation of source reliability, and thus result in more accurate truth estimation.

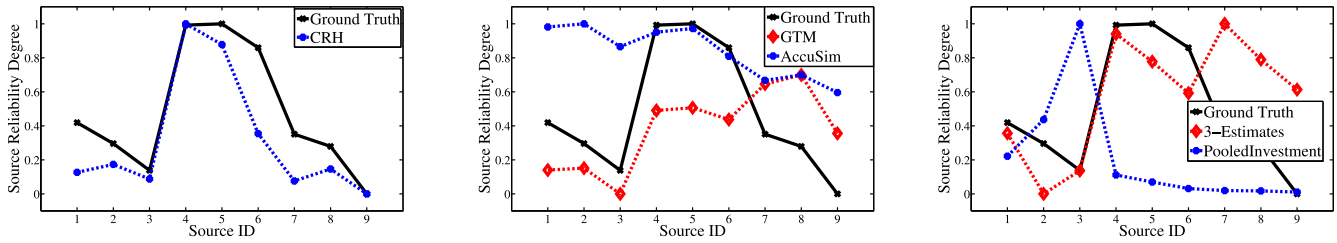
As source reliability is the key to obtain correct truths, we further show the source reliability degrees estimated for the nine sources by various approaches on the weather forecast data set. We choose this data set because it consists of nine sources only, which is more practical to demonstrate. We first compute the true source reliability by comparing the observations made by each source with the ground truths. Reliability of a source is defined as the probability that the source makes correct statements on categorical data, and the chance that the source makes statements close to the truth on continuous data. To simplify the presentation, we combine the reliability scores of continuous and categorical data into one score for each source. To make it clear, we show the source reliability degrees in three plots presented in Fig. 1, each of which shows the comparison between the ground truths and some of the approaches.

Fig. 1a shows that the source reliability degree estimated by CRH method is in general consistent with that obtained from the ground truths. By characterizing different data types in a joint model, CRH can successfully distinguish

1. http://www.wunderground.com
2. http://www.hamweather.com
3. http://www.worldweatheronline.com

TABLE 2
Performance Comparison on Real-World Data Sets

Method	Weather Data		Stock Data		Flight Data	
	Error Rate	MNAD	Error Rate	MNAD	Error Rate	MNAD
CRH	**0.3759**	**4.6947**	**0.0700**	**2.6445**	**0.0823**	**4.8613**
Mean	NA	4.7840	NA	7.1858	NA	8.2894
Median	NA	4.9878	NA	3.9334	NA	7.8471
GTM	NA	4.7914	NA	3.4253	NA	7.6703
Voting	0.4844	NA	0.0817	NA	0.0859	NA
Investment	0.4913	5.2361	0.0983	2.8081	0.0919	6.4153
PooledInvestment	0.4948	5.5788	0.0990	2.7940	0.0925	5.8562
2-Estimates	0.5327	5.5258	0.0726	2.8509	0.0885	7.4347
3-Estimates	0.4810	5.1943	0.0818	2.7749	0.0881	7.1983
TruthFinder	0.4586	5.1293	0.1194	2.7140	0.0950	8.1351
AccuSim	0.4672	5.0862	0.0726	2.8503	0.0881	7.3204



(a) CRH (b) Baselines (c) Baselines

Fig. 1. Comparison of source reliability degrees with ground truths.

good sources from bad ones, and accordingly derive the truth based on good sources. In Figs. 1b and 1c, we show the reliability degrees of ninr sources estimated by GTM, AccuSim, 3-Estimates and PooledInvestment compared with the ground truth reliability. We particularly show the results on these approaches because 3-Estimates and PooledInvestment are improved solutions compared with 2-Estimates and Investment respectively claimed in the corresponding papers, and TruthFinder has similar performance with AccuSim on this data set. As different methods adopt various functions to estimate the source reliability scores, to make them comparable, we normalize all the scores into the range $[0, 1]$. Among these approaches, 3-Estimates and GTM calculate the unreliability degrees, so we convert their scores to reliability degrees to show the comparison. The plots show that the baseline methods can capture the difference among sources in making accurate claims to a certain extent, but the patterns in source reliability detected by them are not very consistent with the ground truths, which can thus explain the increased error in truth detection in Table 2.

3.2.2 Noisy Multi-Source Simulations

To further demonstrate the advantages of the proposed framework in the environment involving various reliability degrees and different loss functions, we conduct experiments on simulated data sets generated from UCI machine learning data sets. We choose two data sets: UCI Adult[4] and Bank[5] data sets, each of which has both continuous and categorical properties. The original data sets are regarded as the ground truths, and based on each of them, we generate a data set consisting of multiple conflicting sources by injecting different levels of noise into the ground truths as the input to our approach and baseline methods. Gaussian noise is added to each continuous property, and values in categorical properties are randomly flipped to generate facts that deviate from the ground truths. To better simulate the real-world data, we round the continuous type data based on their physical meaning. A parameter γ is used to control the reliability degree of each source (a lower γ indicates a lower chance that the ground truths are altered to generate observations). For continuous data, γ is proportional to the variance of the Gaussian noise. For categorical data, a threshold θ ($\theta \in [0, 1]$) is set according to γ. For each object, we first draw a random number x from $Uniform(0, 1)$. If $x < \theta$, the corresponding claimed value from will be perturbed by randomly choosing one of the other possible values. Otherwise, the original value is kept. In this way, we generate data sets which contain 8 sources with various degrees of reliability ($\gamma = \{0.1, 0.4, 0.7, 1, 1.3, 1.6, 1.9, 2\}$). Table 3 shows the statistics of these two data sets.

Table 4 summarizes the results of all the approaches on these two data sets. It can be seen that CRH can fully recover

TABLE 3
Statistics of Simulated Data Sets

	Adult Data	Bank Data
# Observations	3,646,832	5,787,008
# Entries	455,854	723,376
# Ground Truths	455,854	723,376

4. http://archive.ics.uci.edu/ml/datasets/Adult
5. http://archive.ics.uci.edu/ml/datasets/Bank+Marketing

TABLE 4
Performance Comparison on Simulated Data Sets

Method	Adult Data		Bank Data	
	Error Rate	MNAD	Error Rate	MNAD
CRH	**0.0000**	**0.0637**	**0.0000**	**0.0789**
Mean	NA	0.3673	NA	0.3671
Median	NA	0.2470	NA	0.2491
GTM	NA	0.0810	NA	0.0948
Voting	0.1029	NA	0.2314	NA
Investment	0.0530	0.1391	0.1197	0.1588
PooledInvestment	0.0215	0.1008	0.0241	0.0866
2-Estimates	0.0497	0.1355	0.1152	0.1583
3-Estimates	0.0497	0.1355	0.1152	0.1583
TruthFinder	0.0346	0.1272	0.1097	0.1589
AccuSim	0.0288	0.1145	0.0681	0.1571

all the truths on categorical data, and find the true value for continuous data with very small distance by inferring accurate source reliability degrees. Similar to the experiments on the weather data set, we can still observe the great improvement in truth detection performance compared with baseline approaches due to the proposed method's advantage in source reliability estimation. Existing approaches cannot provide accurate estimate of source reliability because they either take incomplete data (only categorical or continuous), or do not model the characteristics of both data types jointly.

On these simulated data sets, we also investigate how the performance of the proposed CRH approach varies with respect to different distributions of source reliability degrees. To illustrate the effect more clearly, we choose two reliability degrees: $\gamma = 0.1$ and $\gamma = 2$, which correspond to reliable and unreliable sources respectively. We now fix the total number of sources as 8, and change the number of reliable sources to conduct a series of experiments. Figs. 2 and 3 show the performance of the proposed approach and baseline methods on Adult and Bank data sets respectively. In each of them, we show the performance on categorical and continuous data respectively when we vary the number of reliable sources from 0 to 8 (out of 8 sources in total).

The following observations can be made from the results: 1) The plots support our previous findings that the CRH framework outperforms existing conflict resolution techniques, which ignore the unique characteristics of each data type. When sources are equally reliable or unreliable (number of reliable sources equals to 0 or 8), the CRH model achieves similar performance as that of voting/averaging approaches. However, when the reliability degree varies across sources, CRH performs much better. 2) In general, it is easier to detect truths when we have a bigger number of reliable sources. However, on categorical data, even when only 1 out of 8 sources is reliable, CRH can still discover most of the truths. Clearly, the proposed approach can successfully infer source reliability and thus detect the truths that are stated by the minority. 3) On continuous data, we can see that the convergence rate is slower than that on categorical data. Conflict resolution on continuous data is in general more difficult due to the higher complexity of the truth space and more complicated definition of closeness to the truths.

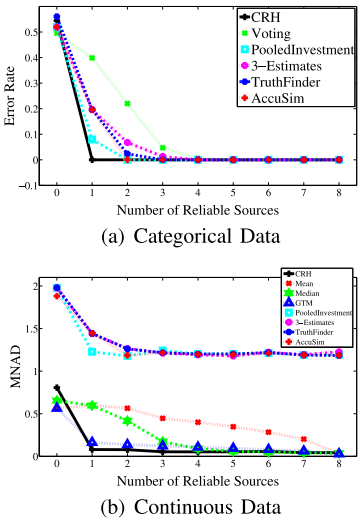


Fig. 2. Performance w.r.t. # reliable sources on adult data set.

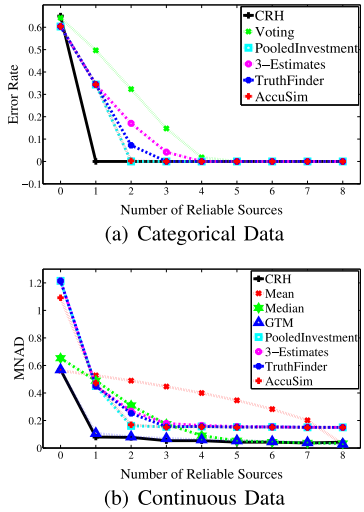


Fig. 3. Performance w.r.t. # reliable sources on bank data set.

3.3 Experimental Results of I-CRH Method

We apply the incremental CRH (I-CRH) method on three real world data sets: weather data set, stock data set, and flight data set. Table 5 summarizes the performance of CRH and I-CRH in terms of Error Rate on categorical data, MNAD on continuous data and running time (in seconds). We can observe that though I-CRH method performs slightly worse than CRH framework on Error Rate and MNAD, its running time is significantly shorter than CRH.

To demonstrate the performance of I-CRH method, we plot the source reliability degrees it estimates on weather data set. We first show the source reliability degrees at each timestamp in Fig. 4a. It is clear that all source reliability degrees reach a stable stage after few timestamps. To further validate whether the stabilized source weights estimated by I-CRH method are consistent with the estimation by CRH framework, we shows its estimated source reliability degrees at the first timestamp and the sixth timestamp (when they become stable) comparing with the source reliability degrees estimated by the CRH framework in Fig. 4b. Although at the first timestamp, I-CRH and CRH methods have slightly different estimation on source weights, I-CRH converges to CRH after few timestamps. It implies that the I-CRH method can provide similar result as CRH method

TABLE 5
Performance Comparison of CRH and I-CRH

Method	Weather Data			Stock Data			Flight Data		
	Error Rate	MNAD	Time (s)	Error Rate	MNAD	Time (s)	Error Rate	MNAD	Time (s)
CRH	0.3759	4.6947	13.182	0.0700	2.6445	162.24	0.0823	4.8613	138.64
I-CRH	0.4	4.7996	3.8064	0.0749	2.6494	70.091	0.0837	5.2295	79.794

after several timestamps when source weight consistency assumption holds.

We also check the effect of time window size and decay rate in I-CRH method. Time window controls the size of each chunk of data and α determines the impact of past data on source weight estimation. Fig. 5 shows the performance with respect to the time window size. Note that the time window determines how often we apply the I-CRH method to the data. When the window size is too small, there are not sufficient data to estimate accurate source weights so the error rate is high. However, once there are enough data, the performance improves. Although performance may drop slightly if window size is too large, the performance of I-CRH is mostly steady with various time window sizes. Fig. 6 shows the effect of the decay rate α. It can be seen that the performance of I-CRH is not sensitive to different values of α.

3.4 Experimental Results of Parallel CRH Method

In this section, we evaluate the running time of CRH on Hadoop cluster using simulated data sets. Based on the Adult data set, we generate large-scale data sets by adding different noise levels on the original data set as we discussed before. By controlling the number of instances and properties, we can vary the number of entries. On the other side, we can easily change the number of sources. For each simulated source, we generate the claimed value for all the entries, so the number of observation is the product of the number of entries and the number of sources. The proposed CRH framework is implemented using MapReduce model. The experiments are conducted on a Dell Hadoop cluster with Intel Xeon E5-2403 processor (4x 1.80 GHz, 48 GB RAM).

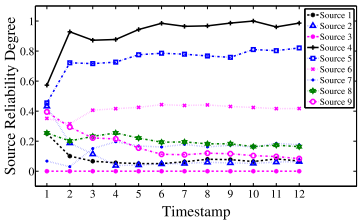
As shown in Table 6, the number of observations vary from 10^4 to 10^8. The fusion process using the MapReduce version of the proposed approach can finish in a short time. The running time mainly comes from the setup overhead when the number of observations is not very large, but the speed-up in the execution time is more obvious when the number of observations increases. For example, on a data set with size 10^8, the whole process only takes 669 s.

To make the result more interpretable, we plot the running time with respect to the number of entries or the number of sources. From Fig. 7, we can observe that when we keep the number of sources unchanged, the running time linearly grows with the number of entries. We also shows the result from another perspective: When we keep the number of entries unchanged, the running time linearly grows with the number of sources.

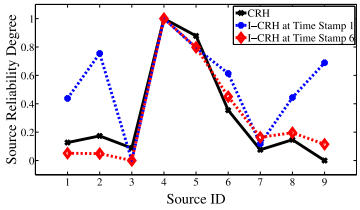
It is also important to check the effect of the number of nodes in Hadoop system. Here we only show the experiment with different number of Reducer nodes. For the number of Mapper nodes, results are similar. We keep the number of observation unchanged (4×10^8), start with only two

TABLE 6
Running Time on Hadoop Cluster

# Observations	Time (s)
1×10^4	94
1×10^5	96
1×10^6	100
1×10^7	193
1×10^8	669
4×10^8	1,384
Pearson Correlation	0.9811



(a) w.r.t. Timestamp



(b) w.r.t. CRH Methods

Fig. 4. Source reliability degree comparison.

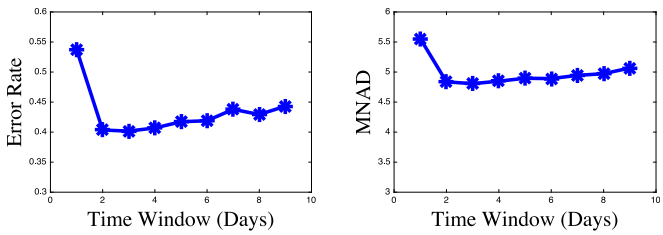


Fig. 5. Error rate and MNAD w.r.t. time window.

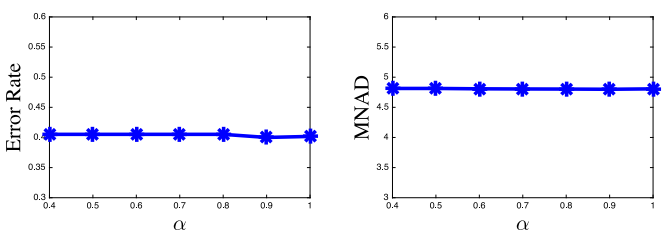


Fig. 6. Error rate and MNAD w.r.t. decay rate α.

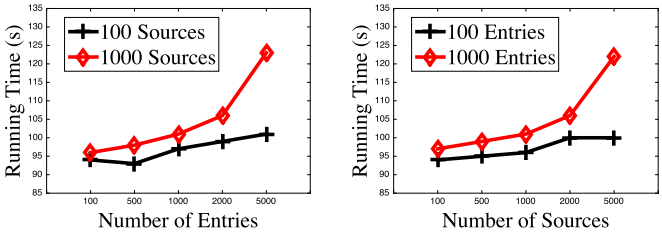


Fig. 7. Running time w.r.t. number of observations.

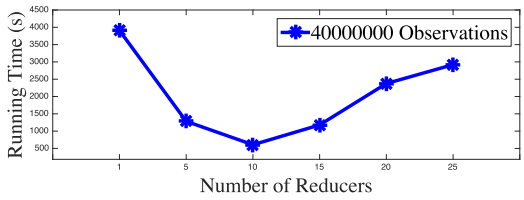


Fig. 8. Running time w.r.t. number of reducers.

Reducer nodes, and then include more nodes to do the Reducer jobs. Fig. 8 shows the running time with respect to the number of Reducer nodes. For MapReduce, it is not necessary that more nodes lead to faster speed, because the overhead such as communication cost has to be considered. For a given input, its size along with other factors determine the optimal number of reducers. For example, from Fig. 8 we can see that with 10 Reducer nodes, the system achieves the best performance. When we set the number of reducers to be 25, it takes even longer time to complete the task.

4 Conclusions

To extract insightful knowledge from an overwhelming amount of information generated by numerous industries, it is crucial to automatically identify trustworthy information and sources from multiple conflicting data sources. As heterogeneous data is ubiquitous, a joint estimation on various data types can lead to better estimation of truths and source reliability. However, existing conflict resolution work either regards all the sources equally reliable, or models different data types individually. Therefore, we propose to model the conflict resolution problem on data of heterogeneous types using a general optimization framework called CRH that integrates the truth finding process on various data types seamlessly. In this model, truth is defined as the value that incurs the smallest weighted deviation from multi-source input in which weights represent source reliability degrees. We derive a two-step iterative procedure including the computation of truths and source weights as a solution to the optimization problem. The advantage of this framework is its ability of taking various loss and regularization functions to characterize different data types and weight distributions effectively. We also extend the CRH method to streaming and parallel processing scenarios by developing effective incremental and MapReduce based CRH methods. We conduct experiments on weather, stock and flight data sets collected from multiple platforms as well as simulated multi-source data generated from UCI machine learning data sets. Results demonstrate the advantage of the proposed CRH approach over existing conflict resolution approaches in finding truths from heterogeneous data. Further, the incremental and parallel CRH methods demonstrate high efficiency and scalability on streaming and large-scale data sets.

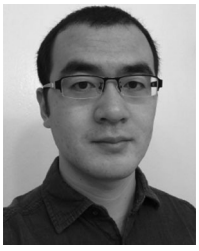
Acknowledgments

Yaliang Li and Qi Li contributed equally to this work and should be considered as joint first authors. The work was supported in part by the US National Science Foundation under Grant US National Science Foundation IIS-1319973 and CNS-1566374, the US Army Research Laboratory under Cooperative Agreement No. W911NF-09-2-0053 (NS-CTA), and the US Army Research Office under Cooperative Agreement No. W911NF-13-1-0193.

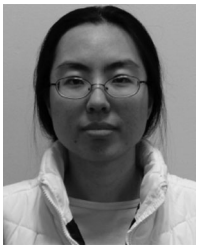
References

[1] J. Bleiholder and F. Naumann, "Data fusion," *ACM Comput. Surveys*, vol. 41, no. 1, pp. 1:1–1:41, 2009.
[2] X. L. Dong and F. Naumann, "Data fusion: Resolving data conflicts for integration," *Proc. VLDB Endowment*, vol. 2, no. 2, pp. 1654–1655, 2009.
[3] Z. Jiang, "A decision-theoretic framework for numerical attribute value reconciliation," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 7, pp. 1153–1169, Jul. 2012.
[4] X. Yin, J. Han, and P. S. Yu, "Truth discovery with multiple conflicting information providers on the web," in *Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2007, pp. 1048–1052.
[5] 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.
[6] B. Zhao, B. I. P. Rubinstein, J. Gemmell, and J. Han, "A Bayesian approach to discovering truth from conflicting sources for data integration," *Proc. VLDB Endowment*, vol. 5, no. 6, pp. 550–561, 2012.
[7] X. L. Dong and D. Srivastava, "Big data integration," in *Proc. Int. Conf. Data Eng.*, 2013, pp. 1245–1248.
[8] V. Vydiswaran, C. Zhai, and D. Roth, "Content-driven trust propagation framework," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2011, pp. 974–982.
[9] J. Pasternack and D. Roth, "Making better informed trust decisions with generalized fact-finding," in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 2324–2329.
[10] X. L. Dong, L. Berti-Equille, and D. Srivastava,, "Integrating conflicting data: The role of source dependence," *Proc. VLDB Endowment*, vol. 2, no. 1, pp. 550–561, 2009.
[11] X. Li, X. L. Dong, K. B. Lyons, W. Meng, and D. Srivastava,, "Truth finding on the deep web: Is the problem solved?" *Proc. VLDB Endowment*, vol. 6, no. 2, pp. 97–108, 2012.
[12] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher, "On truth discovery in social sensing: A maximum likelihood estimation approach," in *Proc. 11th Int. Conf. Inf. Process. Sensor Netw.*, 2012, pp. 233–244.
[13] L. Su, Q. Li, S. Hu, S. Wang, J. Gao, H. Liu, T. Abdelzaher, J. Han, X. Liu, Y. Gao, and L. Kaplan, "Generalized decision aggregation in distributed sensing systems," in *Proc. IEEE Real-Time Syst. Symp.*, 2014, pp. 1–10.
[14] B. Zhao and J. Han, "A probabilistic model for estimating real-valued truth from conflicting sources," in *Proc. 10th Int. Workshop Quality Databases*, 2012.
[15] 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.
[16] A. Marian and M. Wu, "Corroborating information from web sources," *IEEE Data Eng. Bull.*, vol. 34, no. 3, pp. 11–17, Sept. 2011.
[17] J. Pasternack and D. Roth, "Latent credibility analysis," in *Proc. Int. Conf. World Wide Web*, 2013, pp. 1009–1020.
[18] X. L. Dong and D. Srivastava, "Compact explanation of data fusion decisions," in *Proc. Int. Conf. World Wide Web*, 2013, pp. 379–390.
[19] R. Pochampally, A. D. Sarma, X. L. Dong, A. Meliou, and D. Srivastava, "Fusing data with correlations," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2014, pp. 433–444.
[20] T. Rekatsinas, X. L. Dong, and D. Srivastava, "Characterizing and selecting fresh data sources," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2014, pp. 919–930.
[21] D. Yu, H. Huang, T. Cassidy, H. Ji, C. Wang, S. Zhi, J. Han, C. Voss, and M. Magdon-Ismail, "The wisdom of minority: Unsupervised slot filling validation based on multi-dimensional truth-finding," in *Proc. 25th Int. Conf. Comput. Linguistics*, 2014, pp. 1567–1578.

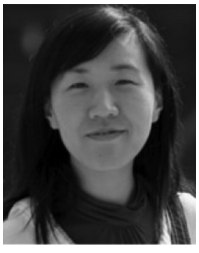
[22] S. Wang, L. Su, S. Li, S. Hu, T. Amin, H. Wang, S. Yao, L. Kaplan, and T. Abdelzaher, "Scalable social sensing of interdependent phenomena," in *Proc. 14th Int. Conf. Inf. Process. Sensor Netw.*, 2015, pp. 202–213.
[23] Q. Li, Y. Li, J. Gao, L. Su, B. Zhao, M. Demirbas, W. Fan, and J. Han, "A confidence-aware approach for truth discovery on long-tail data," *Proc. VLDB Endowment*, vol. 8, no. 4, pp. 425–436, 2014.
[24] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. London, U.K.: Chapman & Hall, 2012.
[25] S. Masoudnia and R. Ebrahimpour, "Mixture of experts: A literature survey," *Artif. Intell. Rev.*, vol. 42, no. 2, pp. 275–293, 2014.
[26] D. P. Bertsekas, *Non-Linear Programming*. Belmont, MA, USA: Athena Scientific, 1999.
[27] X. L. Dong, B. Saha, and D. Srivastava,, "Less is more: Selecting sources wisely for integration," *Proc. VLDB Endowment*, vol. 6, no. 2, pp. 37–48, 2012.
[28] T. H. Cormen, R. L. Rivest, C. E. Leiserson, and C. Stein, *Introduction to Algorithms*. Cambridge, MA, USA: MIT press, 2009.
[29] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, "Clustering with Bregman divergences," *J. Mach. Learn. Res.*, vol. 6, pp. 1705–1749, 2005.
[30] 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.
[31] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *J. Optimization Theory Appl.*, vol. 109, no. 3, pp. 475–494, 2001.
[32] J. Nocedal and S. Wright, *Numerical Optimization*. New York, NY, USA: Springer, 2006.
[33] M. Gupta, Y. Sun, and J. Han, "Trust analysis with clustering," in *Proc. 20th Int. Conf. Companion World Wide Web*, 2011, pp. 53–54.
[34] 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. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2015, pp. 745–754.
[35] J. Dean and S. Ghemawat, "Mapreduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, 2008.
[36] C.-T. Chu, S. K. Kim, Y.-A. Lin, Y. Yu, G. Bradski, A. Y. Ng, and K. Olukotun, "Map-reduce for machine learning on multicore," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 281–288.



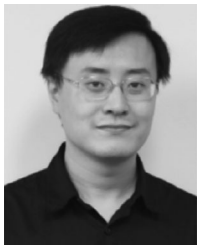
Yaliang Li received the BS degree from the Nanjing University of Posts and Telecommunications in 2010 and is currently working toward the PhD degree in the Department of Computer Science and Engineering at SUNY Buffalo. His research topics include truth discovery, text and web mining, privacy-preserving data mining, and data mining application in healthcare.



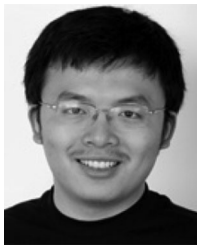
Qi Li received the BS degree in mathematics from Xidian University and the MS degree in statistics from the University of Illinois at Urbana-Champaign, in 2010 and 2012, respectively. She is currently working toward the PhD degree in the Department of Computer Science and Engineering at SUNY Buffalo. Her research interest includes truth discovery, data aggregation, and crowdsourcing.



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 assistant professor in 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 member of the IEEE.



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 2013 and 2012, respectively. He is an assistant professor in the Department of Computer Science and Engineering at SUNY Buffalo. His research focuses on the general areas of cyber-physical systems, wireless and sensor networks, and mobile computing. He was with the IBM T. J. Watson Research Center and National Center for Supercomputing Applications. He is a member of the ACM and the IEEE.



Bo Zhao received the PhD degree from the University of Illinois at Urbana-Champaign. He is a senior engineer at LinkedIn, before that he was a researcher at Microsoft Research Silicon Valley. His research interests include truth discovery, data integration, knowledge bases, crowdsourcing, and more recently recommender systems.

Wei Fan received the PhD degree in computer science from Columbia University in 2001. He is currently the senior director and deputy head of Baidu Big Data Lab in Sunnyvale, California. His main research interests and experiences are in various areas of data mining and database systems, such as, deep learning, stream computing, high-performance computing, extremely skewed distribution, cost-sensitive learning, risk analysis, ensemble methods, easy-to-use nonparametric methods, graph mining, predictive feature discovery, feature selection, sample selection bias, transfer learning, time series analysis, bioinformatics, social network analysis, novel applications, and commercial data mining systems. His coauthored paper received ICDM'06/KDD11/KDD12/KDD13/KDD97 Best Paper & Best Paper Runner-up Awards. He led the team that used his Random Decision Tree (www.dice.com) method to win the 2008 ICDM Data Mining Cup Championship. He received the 2010 IBM Outstanding Technical Achievement Award for his contribution to IBM Infosphere Streams. He is the associate editor of the *ACM Transactions on Knowledge Discovery and Data Mining* (*TKDD*). During his times as the associate director in Huawei Noah's Ark Lab in Hong Kong from August 2012 to December 2014, he has led his colleagues to develop Huawei StreamSMART a streaming platform for online and real-time processing, query, and mining of very fast streaming data. StreamSMART is three to five times faster than STORM and 10 times faster than SparkStreaming, and was used in Beijing Telecom, Saudi Arabia STC, Norway Telenor, and a few other mobile carriers in Asia. Since joining Baidu Big Data Lab, he has been working on medical and healthcare research and applications, such as deep learning-based disease diagnosis based on NLP input as well as medical dialogue robot.

Jiawei Han is Abel Bliss professor, the Department of Computer Science, University of Illinois at Urbana-Champaign. His research interests include data mining, data warehousing, information network analysis, etc., with more than 600 conference and journal publications. He is the director of IPAN, supported by the Network Science Collaborative Technology Alliance program of the US Army Research Lab, and the codirector of KnowEnG: A Knowledge Engine for Genomics, one of the NIH supported Big Data to Knowledge (BD2K) Centers. He is a fellow of the ACM and the IEEE.