Dynamic Source Weight Computation for Truth Inference over Data Streams

Yi Yang
Auckland University of Technology
Auckland, New Zealand
yi.yang@aut.ac.nz

Quan Bai
University of Tasmania
Hobart, Australia
quan.bai@utas.edu.au

Qing Liu
Data61, CSIRO
Hobart, Australia
q.liu@data61.csiro.au

ABSTRACT

Truth inference, a method that resolves conflicts among multi-agent data, has been widely studied in the field of AI. Most existing truth inference methods use iterative approaches to achieve high accuracy, but are inefficient to infer object truths over data streams. The methods developed for streaming data can achieve high efficiency but suffer from low accuracy. In this paper, we propose a novel truth inference method, Dynamic Source Weight Computation truth inference (DSWC), that can work with a wide range of iterative-based truth inference methods to dynamically compute source weights over data streams. Specifically, we use Taylor expansion to analyze the unit error of object truths inferred by source weights computed at a previous timestamp. If the source weight at present is predicted to be able to limit the error under a threshold, we use the source weights computed previously to approximate object truths at present to avoid the expensive source weight computation step. Compared with the existing work, the proposed method is more effective in predicting source weights and can be applied to a wider range of applications. Experimental results based on four real-world datasets demonstrate that DSWC is both accurate and efficient for truth inference over data streams.

KEYWORDS

Truth Inference; Multi-agent Reliabilities

ACM Reference Format:

Yi Yang, Quan Bai, and Qing Liu. 2019. Dynamic Source Weight Computation for Truth Inference over Data Streams. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 9 pages.

1 INTRODUCTION

In the era of big data, it is easy to obtain data for the same objects from multiple agents, e.g., websites, crowd workers and social sensors. Each agent, performed as a data source[1], can provide her observations, views or answers to the objects that we are interested in. Unfortunately, the multi-source data on the same object is usually conflicting. Generally, the multi-source data cannot be used as inputs by machine learning or analytic tasks because poor quality data could have a negative impact on the performance of the learning algorithms. Thus, truth inference has emerged and it aims at inferring the truth of each object from multi-source data.

[1]We will use agent and source interchangeably in this paper.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

Most studies on truth inference infer object truths by modeling source reliabilities. The principle is that sources which frequently provide trustworthy information are reliable and has high weights, while the information supported by reliable sources is identified as the truth for each object. Motivated by this principle, significant progress has been made to address truth inference in the applications of crowdsourcing [30], world wide web [14], social sensing [1, 22], healthcare [17], information extraction [9, 27] and knowledge base construction [4, 6, 7]. Most existing researches on truth inference focus on static data where no temporal dimension is involved [8, 10, 11, 18, 22–24, 28, 29]. It is assumed that all the data is available in a batch before the truth inference algorithm is conducted. Iterative approaches are usually adopted by these methods to learn accurate agent quality and object truths simultaneously.

In recent years, significant advances have been made in mobile and web technologies. It has led to the proliferation of many streaming data intensive applications, in which data in streaming format is being collected sequentially in large volume and high speed from multiple agents. Effective and efficient truth inference methods for such high-speed data streams are important to a wide range of applications. Although the iterative-based methods can be applied to the streaming data directly, they are computationally expensive to process streaming data. Therefore, it is demanded to have efficient methods which process the streaming data in short response time. In light of this challenge, some incremental truth inference methods are designed specifically for streaming data [15, 16]. One disadvantage of the incremental methods is that they assume the source weight converges to a certain value over time. However, the source weights are changing constantly [21]. Therefore, they cannot compute the source weights accurately at each timestamp. As a result, the incremental methods are not accurate at inferring object truths over data streams. Recently, the authors in [12] developed ASRA that adopts iterative-based methods to infer object truths over data streams. ASRA neglects the source weight computation step at certain timestamps if the error of truth inference caused by not changing source weights is predicted to be small. However, ASRA is limited that it can only be applied to few applications and is not able to incorporate prior beliefs to compute source weights and object truths. Furthermore, its estimation model cannot estimate source weights correctly over data streams, which results in large errors and more iterative processes conducted.

To address the accuracy and efficiency issues of streaming data truth inference and the limitation of ASRA, we develop a novel truth inference method, ***D**ynamic **S**ource **W**eight **C**omputation* (DSWC) truth inference, which is able to work with a wide range of iterative truth inference methods for both high accuracy and efficiency. Specifically, we define *unit error*, which is the truth inference error caused by not changing source weights at certain timestamps. We analyze the relationship between unit error and source weight evolution. We prove that the unit error is limited if the source weight evolution satisfies a certain condition. If the unit error is under a threshold, we skip the expensive iterative process and compute object truths directly. As the source weight evolution is unknown before the source weights are computed, we develop a model to predict source weight evolution over time. Finally, we integrate the error analysis and the prediction model to present the DSWC algorithm. In summary, we make the following contributions:

- We theoretically analyze the relationship between unit error and source weight evolution when each object is only observed or reported by a subset of sources/agents. We prove that the unit error can be limited within a threshold if the source weight evolution satisfies a certain condition.
- We propose a model which accurately predicts source weight evolution and reports the posterior distribution of source weight and source weight evolution.
- By integrating the error analysis and the prediction model, we develop DSWC algorithm that can work with a wide range of iterative-based methods for truth inference over data streams to achieve better accuracy as well as efficiency for streaming data.
- The experiments on four datasets demonstrate high performance of the proposed method.

2 TRUTH INFERENCE & RELATED WORK

In this section, we define the truth inference problem and review existing truth inference methods. The problem definition of truth inference on data streams is given below.

Problem Definition. Suppose there are a set of objects O and a set of agents/sources S. Each object o at a timestamp i can be observed by a set of sources $S_{o,i}$ where $S_{o,i} \subseteq S$. The observation of an object o by a source s at timestamp i is denoted as $v^s_{o,i}$. The goal of truth inference is to infer the truth for each object at each timestamp, i.e. $\{v^*_{o,i}\}$ where $v^*_{o,i}$ is defined as the truth of object o at timestamp i.

Early studies of truth inference mostly focus on static scenarios where no temporal dimension is involved, i.e., all the observations are reported at the same timestamp. In order to infer object truths in a static dataset, Demartini et al. proposed ZenCrowd [3] which uses a probabilistic graphical model to jointly estimate categorical object truths and source weights. DS [2] uses a confusion matrix to model the source weights, in which the confusion matrix represents the probability distribution of the source's possible categorical observation for an object given the truth of the object. Based on DS, LFC [20] was developed to infer both categorical and numerical object truths from crowdsourced observations. In [26], the authors proposed TruthFinder which adopts Bayesian analysis to compute source weights and object truths. In [8], 3-estimates algorithm was proposed to estimate object truths and source weights along with object difficulties. Li et al. [10] proposed a confidence aware truth inference algorithm to deal with datasets with long-tail distribution. [28] developed a probabilistic graphical model (PGM) based method (GTM) for inferring numerical object truths. In [29], the authors studied the truth inference problem with multi-valued truths and used false positive and false negative rates to model the reliability degrees of sources. In [11], an optimization based method (CRH) was developed for inferring heterogeneous object truths. By analyzing streaming data, OTD [25] was developed to regularize inferred truths by ARIMA-mined patterns.

As both source weights and object truths are unknown, the methods discussed above adopt an iterative approach to compute object truths and source weights until convergence. A well-known truth inference strategy is **weighted aggregation** [10, 11, 15, 16, 25, 28]. The weighted aggregation can be derived by using coordinate descent if the truth inference problem is modeled as an optimization task [11], or expectation maximization if the truth inference problem is modeled by a probabilistic graphical model [28]. The high-level view of weighted aggregation is given below.

$$v^*_{o,i} = \frac{\sum_{s \in S_{o,i}} w^s_i \times v^s_{o,i} + a}{\sum_{s \in S_{o,i}} w^s_i + b} \tag{1}$$

In Equation (1), w^s_i is the weight of source s at timestamp i. It is modeled as a positive number which reflects the reliability of source s. A source's weight is higher if its observations are closer to the truths. a and b can be set differently to capture different characteristics when inferring object truths. For example, a and b are set to 0 for basic weighted aggregation [11]. If PGM is used to infer object truths where the object truths are generated from Gaussian distributions [28], then $a = \frac{\mu_{o,i}}{\sigma^2_{o,i}}$ and $b = \frac{1}{\sigma^2_{o,i}}$, where $\mu_{o,i}$ and $\sigma^2_{o,i}$ are the mean and variance parameters of the Gaussian distribution that generates the truth of object o at timestamp i. In a data stream, the object truths usually evolve smoothly over time, i.e., the truths of an object in adjacent timestamps are very close. To capture this temporal smoothness [15], a and b can be set to $\lambda v^*_{o,i-1}$ and λ, respectively, where λ is a smooth factor (hyperparameter), and $v^*_{o,i-1}$ is the inferred truth of object o at the previous timestamp $i-1$. A larger λ enforces the truth at current timestamp to be very close to the inferred truth at the previous timestamp. If an observation at timestamp i is significantly different from $v^*_{o,i-1}$, this observation can be treated as an outlier and discarded. Other more advanced outlier detection techniques can also be applied here, but the focus of this paper is not a novel outlier detection method, we will not discuss outlier detection further.

The **source weights computation strategy** can be derived differently by different methods. For example, CRH [11] and DyOP [15] use the following equations to compute source weights.

$$\text{CRH: } w^s_i = -\log \frac{\sum_{o \in O^s_i} (v^s_{o,i} - v^*_{o,i})^2}{\sum_{s \in S} \sum_{o \in O^s_i} (v^s_{o,i} - v^*_{o,i})^2} \tag{2}$$

$$\text{DyOP: } w^s_i = \frac{|O^s_i|}{\sum_{o \in O^s_i} (v^s_{o,i} - v^*_{o,i})^2} \tag{3}$$

In the above two equations, O^s_i denotes the objects that are observed by source s at timestamp i. $\sum_{o \in O^s_i} (v^s_{o,i} - v^*_{o,i})^2$ represents the error that s makes on observing the objects at timestamp i. By incorporating prior beliefs, we can assume the source weight is generated from an Inverse-Gamm distribution, and GTM can be applied to compute source weights [28]:

$$\text{GTM: } w_i^s = \frac{2(\beta_1 + 1) + |O_i^s|}{2\beta_2 + \sum_{o \in O_i^s} (v_{o,i}^s - v_{o,i}^*)^2} \tag{4}$$

In Equation (4), β_1 and β_2 are the hyperparameters of an Inverse-Gamma distribution which encode the prior beliefs of w_i^s. Although the source weights are computed differently by different methods, we can observe that all the methods assign high weights to the reliable sources whose observations are closer to the object truths.

Normally, iterative-based methods can achieve high accuracy. However, the iterative process is computationally expensive. For data arriving from streams, it is inefficient if an iterative process needs to be conducted at each timestamp. To improve the efficiency of truth inference over data streams, [15] proposed an incremental truth inference method which transforms their optimization-based framework (DyOP) to a probabilistic model DynaTD. Thus, data needs to be scanned only once without conducting iterative processes. As information usually evolves smoothly over time, based on DynaTD, DynaTD+s was proposed by adding a smoothness constraint to infer object truths. iCRH [16] was developed to infer truths of heterogeneous data incrementally over data streams. These methods are efficient because they give up using the iterative processes to compute source weights at each timestamp. Instead, they compute each source weight and object truth exactly once at each timestamp without reaching convergence. The consequence of adopting this approach is that the incremental methods cannot compute accurate source weights at each timestamp, which results in large errors when inferring object truths.

In order to leverage accuracy and efficiency of streaming data truth inference, ASRA [12] was developed recently. ASRA uses iterative-based methods to compute source weights only at certain timestamps to reduce the frequency of iterative processes. It analyzes the error of inferred object truths by using source weights computed at a previous timestamp. If the error is predicted to be small, it uses the previously computed source weights to infer object truths at the current timestamp. However, ASRA is limited in the following ways.

- ASRA assumes that every object must be observed by all the agents at every timestamp, i.e., $\forall t \in [1, T]$, $|S_{o,t}| = |S|$. If this condition is not satisfied, its theoretical analysis does not hold. This condition is not realistic for many real-world applications, such as crowdsourcing and social sensing, in which each agent reports only a small set of objects.
- The source weight evolution estimation model of ASRA does not consider the covariance of source weights at each timestamp, which may produce inaccurate estimates.
- ASRA cannot incorporate priors if prior knowledge about the object truths and source weights are available.

The proposed method, DSWC, aims at balancing the accuracy and efficiency and addressing the limitations of ASRA for truth inference over data streams. Specifically, DSWC can work with a wide range of iterative-based methods, including methods that incorporate prior beliefs. Moreover, the error analysis described in Section 4 is based on Taylor expansion, it only requires each source observes a subset of objects, which is more practical for real-world applications. The source weight prediction model developed in Section 5 is able to capture the covariance of source weights over time, which ensures the accuracy and efficiency of DSWC. In the next section, we present the preliminaries of this paper.

3 PRELIMINARY

In this paper, we study numerical truth inference problem over data streams. We adopt weighted aggregation in Equation (1) to infer object truths. From Equation (1) we can see that the truth of an object at timestamp i is determined by the the weights of sources who observe it at timestamp i. We can also write the weighted aggregation as a function of source weights given below

$$f_{o,i}(\{w_i^s\}) = v_{o,i}^* = \frac{\sum_{s \in S_{o,i}} w_i^s \times v_{o,i}^s + a}{\sum_{s \in S_{o,i}} w_i^s + b} \tag{5}$$

where $\{w_i^s\}$ is the weights of sources that observe object o at timestamp i, and $\{v_{o,i}^s\}$, a and b are all constants. By Equation (5), we can observe that the inferred object truth is sensitive to the change of the source weights. If we vary the values of source weights, then the inferred object truth is changed. In real-world applications, source weights usually change smoothly over time [15]. At timestamp j, if we use the source weights computed at a previous timestamp i, where $i < j$, to infer the truth directly without computing the source weights iteratively, it will produce a small error on the inferred truth. Whereas, the efficiency can be improved by skipping the iterative process. Inspired by this idea, we develop a novel method, **D**ynamic **S**ource **W**eight **C**omputation truth inference (DSWC). It can work with a range of iterative-based methods which use weighted aggregation to dynamically compute source weights only at certain timestamps to achieve both high accuracy and efficiency. Specifically, we define *unit error* $\phi_{i/j}^o$ in Equation (6) to measure the deviation of inferred object truth at timestamp j by using source weights computed at timestamp i.

$$\phi_{i/j}^o = \left(\frac{v_{o,j}^* - v_{o,i/j}^*}{v_{o,j}^m}\right)^2 = \left(\frac{f_{o,j}(\{w_j^s\}) - f_{o,j}(\{w_i^s\})}{v_{o,j}^m}\right)^2 \tag{6}$$

In Equation (6), $v_{o,j}^*$ is the truth inferred by the source weights $\{w_j^s\}$ computed at timestamp j, i.e., $f_{o,j}(\{w_j^s\})$, and $v_{o,i/j}^*$ is the approximate truth of object o at timestamp j inferred by the source weights $\{w_i^s\}$ computed at timestamp i, i.e., $f_{o,j}(\{w_i^s\})$. $v_{o,j}^m$ is a scaling factor and defined as the absolute maximum value of observations for o at timestamp j, i.e. $v_{o,j}^m = \max\{v_{o,j}^s\}_{s \in S_{o,j}}$. If the unit error is under a user-defined tolerable threshold ϵ, then we choose to use $\{w_i^s\}$ to approximate the object truths at timestamp j without conducting an expensive iterative process.

At timestamp j, the unit error is determined by the change of source weights from timestamp i to j. We use source weight evolution $\delta_{i/j}^s$, given in Equation (7), to capture the absolute difference of source weights from timestamp i to j. Without loss of generality, we assume the source weights at each timestamp are scaled and summed up to 1, i.e., $\forall t \in \{1, \ldots, T\}$, $\sum_{s \in S} w_t^s = 1$.

$$\delta_{i/j}^s = |w_j^s - w_i^s| \tag{7}$$

Next, we will analyze the relationship between unit error and source weight evolution, and present the source weight evolution upper bound for limiting unit error.

4 ERROR ANALYSIS

In this section, we theoretically analyze the upper bound of source weight evolution that limits the unit error for each object.

The approximate truth $v^*_{o,i/j}$, or $f_{o,j}(\{w_i^s\})$, is sensitive to the the source weights $\{w_i^s\}$. The change rate of $f_{o,j}(\{w_i^s\})$ can be captured by its derivative:

$$\frac{\partial f(\{w_i^s\})}{\partial w_i^s} = \frac{\frac{v^s_{o,j}\times(\sum_{s'\in S_{o,j}} w_i^{s'}+b)-(\sum_{s\in S_{o,j}} w_i^{s'}\times v^{s'}_{o,j}+a)}{\sum_{s'\in S_{o,j}} w_i^{s'}+b}}{\frac{(\sum_{s'\in S_{o,j}} w_i^{s'}+b)^2}{\sum_{s'\in S_{o,j}} w_i^{s'}+b}} = \frac{v^s_{o,i}-f(\{w_i^s\})}{\sum_{s'\in S_{o,j}} w_i^{s'}+b} \tag{8}$$

where, to keep the notation uncluttered, we use f to denote $f_{o,j}$. Next, we propose a theorem to show the high order derivative of weight aggregation in Equation (5).

Theorem 4.1. *The n^{th} order partial derivative of $f(\{w_i^s\})$ w.r.t. n source weights (i.e. $w_i^{s1},\ldots,w_i^{sn}$) is:*

$$\frac{\partial^n f(\{w_i^s\})}{\partial w_i^{s1}\ldots\partial w_i^{sn}} = (-1)^{n-1}(n-1)!\frac{\sum_{k=1}^n v^{s_k}_{o,j}-f(\{w_j^s\})}{(\sum_{s'\in S_{o,j}} w_i^{s'}+b)^n} \tag{9}$$

Proof. For any integers $n\geq 1$, let $P(n)$ denotes the statement $\frac{\partial^n f(\{w_i^s\})}{\partial w_i^{s1}\ldots\partial w_i^{sn}} = (-1)^{n-1}(n-1)!\frac{e^{s_1}+e^{s_2}+\cdots+e^{sn}}{(\sum_{s'\in S_{o,j}} w_i^{s'}+b)^n}$ where $e^s = v^s_{o,j} - f(\{w_i^s\})$. We will prove Theorem 4.1 by induction.

Base step ($n=1$): $P(1)$ is true as shown by Equation(8).

Inductive step $P(k)\to P(k+1)$: Fix some integer $k\geq 2$. Assume that $P(k)$ holds. We need to show that $P(k+1)$:

$$\frac{\partial^{k+1} f(\{w_i^s\})}{\partial w_i^{s_1}\ldots\partial w_i^{s_k}\partial w_i^{s_{k+1}}} = (-1)^k(k)!\frac{e^{s_1}+\cdots+e^{s_k}+e^{s_{k+1}}}{(\sum_{s'\in S_{o,j}} w_i^{s'}+b)^{k+1}}$$

Let $\Omega^o_{i/j} = \sum_{s\in S_{o,j}} w_i^s + b$, by the assumption we have

$$\frac{\partial^k f(\{w_i^s\})}{\partial w_i^{s_1}\ldots\partial w_i^{s_k}} = (-1)^{k-1}(k-1)!\frac{(\sum_{z=1}^k v^{s_z}_{o,j})-k\times f(\{w_i^s\})}{(\Omega^o_{i/j})^k}$$

because we defined $e^{s_z} = v^{s_z}_{o,j} - f(\{w_i^s\})$. Rearrange the above equation, we can get:

$$kf(\{w_i^s\}) = \sum_{z=1}^k v^{s_z}_{o,j} - \frac{1}{(-1)^{k-1}(k-1)!}\times\frac{\partial^k f(\{w_i^s\})}{\partial w_i^{s_1}\ldots\partial w_i^{s_k}}(\Omega^o_{i/j})^k$$

Taking the derivative w.r.t. $w_i^{s_{k+1}}$ on both sides, we have:

$$k\frac{\partial f(\{w_i^s\})}{\partial w_i^{s_{k+1}}} = \frac{\partial}{\partial w_i^{s_{k+1}}}\Big(\sum_{z=1}^k v^{s_z}_{o,j} - \frac{1}{(-1)^{k-1}(k-1)!}\times\frac{\partial^k f(\{w_i^s\})}{\partial w_i^{s_1}\ldots\partial w_i^{s_k}}\times(\Omega^o_{i/j})^k\Big)$$
$$= -\frac{1}{(-1)^{k-1}(k-1)!}\Big(\frac{\partial^{k+1} f(\{w_i^s\})}{\partial w_i^{s_1}\ldots\partial w_i^{s_{k+1}}}\times(\Omega^o_{i/j})^k + k(\Omega^o_{i/j})^{k-1}\times\frac{\partial^k f(\{w_i^s\})}{\partial w_i^{s_1}\ldots\partial w_i^{s_k}}\Big)$$

Rearrange the above equation, we can show that:

$$\frac{\partial^{k+1} f(\{w_i^s\})}{\partial w_i^{s_1}\ldots\partial w_i^{s_{k+1}}} = \frac{-1}{(\Omega^o_{i/j})^k}\Big((-1)^{k-1}(k-1)!k\frac{\partial f(\{w_i^s\})}{\partial w_i^{s_{k+1}}} + k(\Omega^o_{i/j})^{k-1}\frac{\partial^k f(\{w_i^s\})}{\partial w_i^{s_1}\ldots\partial w_i^{s_k}}\Big)$$
$$= (-1)^k k!\frac{(v^{s_1}_{o,j}-f(\{w_i^s\}))+\cdots+(v^{s_k}_{o,j}-f(\{w_i^s\}))+(v^{s_{k+1}}_{o,j}-f(\{w_i^s\}))}{(\Omega^o_{i/j})^{k+1}}$$
$$= (-1)^k(k)!\frac{e^{s_1}+\cdots+e^{s_{k+1}}}{(\sum_{s'\in S_{o,j}} w_i^{s'}+b)^{k+1}}$$

Conclusion: By induction, it is proved that for all integers $n\geq 1$, $P(n)$ is true. Therefore, Equation (9) holds. □

Next, we analyze the unit error by using Taylor Expansion:

$$\sqrt{\phi^o_{i/j}} = \frac{|f(\{w_s^j\})-f(\{w_s^i\})|}{v^m_{o,j}} = \frac{1}{v^m_{o,j}}\Big|\sum_{x\in S_{o,j}}\frac{\partial f}{\partial w_i^x}\Delta w^x_{i/j} + \frac{1}{2!}\sum_{x\in S_{o,j}}\sum_{y\in S_{o,j}}\frac{\partial^2 f}{\partial w_i^x\partial w_i^y}\Delta w^x_{i/j}\Delta w^y_{i/j}+\ldots\Big| \tag{10}$$

where $\Delta w^x_{i/j} = w^x_j - w^x_i$. Based on the Equations (8 - 10), we present the following proposition to show the upper bound of source weight evolution to ensure $\phi^o_{i/j}\leq\epsilon$.

Proposition 4.2. *Given a unit error threshold ϵ and an object o, $\phi^o_{i/j}\leq\epsilon$ if the source weight evolution $\delta^s_{i/j}$ for each source $s\in S_{o,j}$ satisfies the following condition:*

$$\delta^s_{i/j} \leq \frac{\sqrt{\epsilon}\times\Omega^o_{i/j}}{|S_{o,j}|\times(\xi_{o,i/j}+\sqrt{\epsilon})} \tag{11}$$

where $\Omega^o_{i/j} = \sum_{s\in S_{o,j}} w_i^s + b$, *and* $\xi_{o,i/j} = \max\big\{\frac{|v^s_{o,j}-f(\{w_i^s\})|}{v^m_{o,j}}\big\}_{s\in S_{o,j}}$.

Proof. By definition, $\delta^s_{i/j} = |\Delta w^s_{i/j}|$. Substituting the derivatives (Equations (8) and (9)) into Equation (10), we can infer:

$$\sqrt{\phi^o_{i/j}} \leq \sum_{x\in S_{o,j}}\frac{\xi_{o,i/j}}{\Omega^o_{i/j}}\delta^x_{i/j} + \frac{1}{2}\sum_{x\in S_{o,j}}\sum_{y\in S_{o,j}}\frac{2\xi_{o,i/j}}{(\Omega^o_{i/j})^2}\delta^x_{i/j}\delta^y_{i/j}+\ldots$$

Substituting Formula (11) in the above inequation, by the sum of geometric series we can show that:

$$\sqrt{\phi^o_{i/j}} \leq \xi_{o,i/j}\times\Big(\frac{\sqrt{\epsilon}}{\xi_{o,i/j}+\sqrt{\epsilon}} + (\frac{\sqrt{\epsilon}}{\xi_{o,i/j}+\sqrt{\epsilon}})^2+\ldots\Big) = \xi_{o,i/j}\times\frac{\sqrt{\epsilon}}{\xi_{o,i/j}+\sqrt{\epsilon}-\sqrt{\epsilon}} = \sqrt{\epsilon}$$

Hence, $\phi^o_{i/j}\leq\epsilon$. □

Proposition 4.2 states that for an object o, if every source $s\in S_{o,j}$ satisfies the condition given in Formula (11), then we can use $\{w_i^s\}$ to approximate $v^s_{o,j}$ and ensure $\phi^o_{i/j}\leq\epsilon$ at the same time. Based on Proposition 4.2, we propose Proposition 4.3 that defines the upper bound of source weight evolution which guarantees that all the objects' unit errors are under ϵ.

Proposition 4.3. *For each source s, if $\delta^s_{i/j}\leq r^s_{i/j}$ where $r^s_{i/j} = \min(\{\frac{\sqrt{\epsilon}\times\Omega^o_{i/j}}{|S_{o,j}|\times(\xi_{o,i/j}+\sqrt{\epsilon})}\}_{o\in O^s_j})$, then for each object $o\in O$, $\phi^o_{i/j}\leq\epsilon$.*

Proposition 4.3 states that for a source $s\in S$, the upper bound of its source weight evolution should be no more than $r^s_{i/j}$ to ensure the unit errors of its observed objects under ϵ. Hence, for each object o, ensuring $p(\phi^o_{i/j}\leq\epsilon)\geq\alpha$ is equivalent to ensure $p(\delta^s_{i/j}\leq r^s_{i/j})\geq\alpha$ for all the sources.

5 PREDICTION MODEL

In the last section, we present the source weight evolution upper bound that limits the unit error. However, the source weight evolution is unknown unless computing the source weights at the current timestamp j. In order to avoid the iterative process at each timestamp, we propose a source weight prediction model to predict the source weights $\{w_j^s\}$ instead of computing them iteratively. Specifically, we predict the probability of $\phi_{i/j}^o \leq \epsilon$. Given a user-defined confidence threshold α, if $p(\phi_{i/j}^o \leq \epsilon) \geq \alpha$, then we choose to infer object truths at timestamp j by source weight computed at timestamp i. Otherwise, it conducts the iterative process at timestamp j to obtain accurate object truths and source weights. Next, we will describe the prediction model in details.

The source weight w_j^s is computed differently by different methods. Thus, we treat w_j^s as a random function $g^s(j)$. Similarly, given a vector of timestamps $\boldsymbol{t} = [1, \ldots, i]^T$, we use $g^s(\boldsymbol{t})$ to denote the vector of weights of s over $\boldsymbol{t}$, i.e., $g^s(\boldsymbol{t}) = [w_1^s, \ldots, w_i^s]^T$. Then, we model $g^s(\boldsymbol{t})$ as a Gaussian Process (GP) $g^s(\boldsymbol{t}) \sim \mathcal{N}(m(\boldsymbol{t}), \boldsymbol{K^{(i)}})$ where $m(\boldsymbol{t})$ is a prior mean function for $g^s(\boldsymbol{t})$, $\boldsymbol{K^{(i)}}$ is a $i \times i$ covariance matrix at timestamp i. The (x, y) entry in $\boldsymbol{K^{(i)}}$ stores the covariance between $g^s(x)$ and $g^s(y)$, which is measured by a kernel function $k(x, y)$. Then w_j^s can be predicted by $p(g^s(j)|j, \boldsymbol{t}, g^s(\boldsymbol{t})) = \mathcal{N}(\mu_j, \sigma_j^2)$ with mean μ_j and variance σ_j^2 given below [19].

$$
\begin{aligned}
\mu_j &= m(j) + \boldsymbol{k_j}(\boldsymbol{K^{(i)}})^{-1}(g^s(\boldsymbol{t}) - m(\boldsymbol{t})) \\
\sigma_j^2 &= k(j, j) - \boldsymbol{j}(\boldsymbol{K^{(i)}})^{-1}(\boldsymbol{k_j})^T \\
\boldsymbol{k_j} &= [k(j, 1), \ldots, k(j, i)]
\end{aligned}
$$

By definition, $\delta_{i/j}^s = |w_j^s - w_i^s|$. Therefore, we can evaluate the probability $p(\delta_{i/j}^s \leq r_{i/j}^s)$ by evaluating $p(-r_{i/j}^s \leq g^s(j) - w_i^s \leq r_{i/j}^s)$, which can be calculated by using the cumulative probability of Normal distribution $\mathcal{N}(\mu_j - w_i^s, \sigma_j^2)$.

The proposed GP-based prediction model has the following benefits to predict source weights and source weight evolution over data streams. (1) It reports the probability distribution of w_j^s, which is suitable for evaluating $p(\delta_{i/j}^s \leq r_{i/j}^s)$. (2) It is nonparametric. The prediction model treats the source weight as a random function, which can be used to predict source weights computed by different methods. (3) It uses kernels to measure the covariance and the similarity of source weights at different timestamps. Different kernel functions can be applied for different applications. (4) It considers the covariance of source weights over data streams, which makes the prediction more robust.

Update Prediction Model. At each timestamp, $\boldsymbol{K^{(i)}}$ needs to be updated for future prediction. We categorize the update procedure into the following two cases:

Case 1: $p(\delta_{i/j}^s \leq r_{i/j}^s) \geq \alpha$: In this case, we use $\{w_i^s\}$ to approximate $v_{o,j}^s$, and we do not update $\boldsymbol{K^{(i)}}$.

Case 2: $p(\delta_{i/j}^s \leq r_{i/j}^s) < \alpha$: In this case, we need to compute $\{w_j^s\}$. The procedure of updating $\boldsymbol{K^{(i)}}$ for this case is summarized in Algorithm 1. In Algorithm 1, it will augment $\boldsymbol{K^{(i)}}$ $(j - i)$ times. In each augment, it first computes the covariance between the source weights at timestamp t and the previous ones (Line 2). If the

Algorithm 1: Update Covariance Matrix $\boldsymbol{K}$

Input : $\boldsymbol{K^{(i)}}$ at timestamp i
Output: $\boldsymbol{K^{(j)}}$ at timestamp j

1 **for** $t = i + 1 \ldots j$ **do**
2 $\quad \boldsymbol{K_t} = [k(t, 1), \ldots, k(t, t-1)]$
3 $\quad$ **if** $t = j$ **then** $\rho = 0$
4 $\quad$ **else** $\rho = \sigma_t^2$
5 $\quad \boldsymbol{K^{(t)}} = \begin{bmatrix} \boldsymbol{K^{(t-1)}} & (\boldsymbol{K_t})^T \\ \boldsymbol{K_t} & k(t, t) + \rho \end{bmatrix}$
6 **end**
7 **return** $\boldsymbol{K^{(j)}}$

source weight w_t^s is predicted, there will be an error ρ involved in the predicted source weight, and this error can be captured by the variance σ_t^2 of this distribution (Lines 3-7). Then, $\boldsymbol{K^{(i)}}$ is augmented by adding new covariances and error of w_t^s in it.

Gaussian Process needs to retain all the historical information which measures the covariance between each source weight at different timestamps in $\boldsymbol{K^{(i)}}$. From Algorithm 1, we can see that the size of $\boldsymbol{K^{(i)}}$ is increased by $2i+1$ for each augment. As $\boldsymbol{K^{(i)}}$ becomes larger, the matrix inversion becomes computationally expensive, which will make the prediction inefficient. In real-world applications, the present weight of a source may not be correlated with its weights computed or predicted long time ago. Hence, we can use a sliding window to maintain the covariances of L most recently source weights in $\boldsymbol{K^{(i)}}$. By using the sliding window technique to update $\boldsymbol{K^{(i)}}$, the size of $\boldsymbol{K^{(i)}}$ will be at most L^2 and the inversing $\boldsymbol{K^{(i)}}$ is not an issue.

6 DSWC ALGORITHM FLOW

By integrating the error analysis and prediction model, we present DSWC algorithm in Algorithm 2.

In Algorithm 2, i is the last timestamp at which the source weights are computed by an iterative process, j is the current timestamp, and L is the size of the sliding window. In the beginning of the truth inference process, it computes source weights by an iterative process (*iterative_process()*) in the first L timestamps to obtain accurate source weights to initialize the prediction model (Lines 3-4). In *iterative_process()*, it computes the source weights and truths (Lines 15 - 19). An existing iterative approach is adopted here, e.g. DyOP or CRH. It ensures the source weights and truths are accurately computed at this timestamp. Line 20 scales the source weights to make them sum up to 1. Then it updates the covariance matrix of the prediction model (Line 21), and marks the current timestamp as the last timestamp to compute source weights (Line 22). After the first L timestamps, the prediction model is initialized and ready to use. At each timestamp, if $p(\delta_{i/j}^s \leq r_{i/j}^s) \geq \alpha$ is satisfied (Line 6), it uses $\{w_i^s\}$ to approximate object truths at timestamp j (Line 7). Otherwise, iterative_process() will be conducted at the current timestamp to compute source weights and object truths.

7 EXPERIMENTS

In this section, we present the experimental results conducted by using four real-world datasets to evaluate the performance of DSWC

Algorithm 2: DSWC Truth Inference

Input : Observations $\{V_j\}_{j=[1,T]}$, ϵ and α
Output: Truths at each timestamp $\{V_j^*\}_{j=[1,T]}$

$i \leftarrow 1$;
for $j = 1 \rightarrow T$ **do**
  if $j \leq L$ **then**
    iterative_process();
  else
    if *all sources satisfy* $p(\delta^s_{i/j} \leq r^s_{i/j}) \geq \alpha$ **then**
      $V_j^* \leftarrow \{v^*_{o,i/j}\}$
    else
      iterative_process();
    end
  end
end
return $\{V_j^*\}$
Procedure `iterative_process()`
  Initialize the truths V_j^*;
  repeat
    Compute source weights;
    Compute truths;
  until *Convergence condition satisfied*
  Scale source weights;
  Update $\boldsymbol{K}$ for sources;
  $i = j$;

algorithm. We implement all the methods in Java. Experiments are conducted on a Windows PC with Intel i7 CPU and 16 GBs RAM.

7.1 Experiment Setup

7.1.1 Datasets. the dataset descriptions are given below.

- **Weather** [5]: This dataset contains 18 sources that record daily weather information for 30 cities over 6 months. 17 sources[2] are selected from the dataset. The daily temperature property is used in the experiments.
- **Stock** [13]: This dataset records data for 1000 stocks collected from 55 sources over 21 working days in 2011. The open price property is used in the experiments.
- **Forecast**: We crawled hourly weather forecast data from five sources (Aeris[3], Apixu[4], Darksky[5], World Weather Online[6] and Wunderground[7]) for 42 different locations (objects) in New York city over 180 hours. We also collected the ground truths for evaluation.
- **Rates**: This dataset[8] contains 756 pairs of exchange rates over 439 days and use them as ground truths for the objects. 20 sources are generated with smoothly evolved source weights over 439 days. Observations are generated by adding different levels of Gaussian noises based on source weights upon the ground truth for each day. Different from the other three datasets, the likelihood of conflicting observations is high.

7.1.2 Performance Metrics. Efficiency is evaluated by runtime. Accuracy is evaluated by Mean Absolute Error (MAE).

7.1.3 Methods. We apply the source weight computation strategies of CRH, DyOP and GTM, as shown in Equations (2), (3) and (4), in DSWC, and denote them as DSWC(CRH), DSWC(DyOP) and DSWC(GTM). By applying the temporal smoothing constraint when computing object truths on DSWC(CRH), DSWC(DyOP) and DSWC(GTM), we denote them as DSWC(CRH+s), DSWC(DyOP+s) and DSWC(GTM+s). For all the experiments, $m(\boldsymbol{t})$ in the prediction model returns the mean of the most recent source weights in sliding window L. We use squared exponential kernel to measure the covariance between source weights at different timestamps.

The baseline truth inference methods include the iterative-based methods: DyOP, GTM, CRH, LFC and OTD. As GTM can incorporate prior beliefs, we incorporate the information of object truths and source weights computed at the previous timestamp into the Bayesian prior distributions at current timestamp for inferring object truths and computing source weights. The incremental methods include: DynaTD, DynaTD+s and iCRH. The ASRA methods include ASRA(DyOP), ASRA(CRH), ASRA(DyOP+s) and ASRA(CRH+s). ASRA cannot work with GTM because ASRA cannot work with weighted aggregation that encodes prior beliefs. The descriptions of the baselines can be found in Section 2.

In the stock dataset, each source averagely observes 897 objects, two sources observe less than 200 objects at each day, and no source observe all the 1000 objects at any day. It does not meet the condition required by ASRA unless removing some objects from the dataset, which is not practical. Therefore, ASRA cannot be performed on this dataset.

7.2 Prediction Model Evaluation

In this section, we evaluate the effectiveness of the proposed prediction model against the one proposed in ASRA. In order to approximate object truths over the data streams by using the previously computed source weights, the source weight evolution must satisfy the condition, $\delta^s_{i/j} \leq r^s_{i/j}$, to ensure $\phi^o_{i/j} \leq \epsilon$ with probability at least α. Therefore, the prediction results at any timestamp can be categorized into the following cases.

- True Positive (TP): The actual source weight evolution condition is satisfied, and the truth inference method does not compute source weights.
- True Negative (TN): The actual source weight evolution condition is not satisfied, and the truth inference method computes source weights.
- False Positive (FP): The actual source weight evolution condition is not satisfied, but the truth inference method does not compute source weights.
- False Negative (FN): The actual source weight evolution condition is satisfied, but the truth inference method computes source weights.

Higher TP and TN indicate that the prediction model predicts source weight evolution correctly. Thus, we use *accuracy* (*accuracy* = $\frac{\text{TP + TN}}{\text{TP + TN + FP + FN}}$) to measure the effectiveness of the prediction

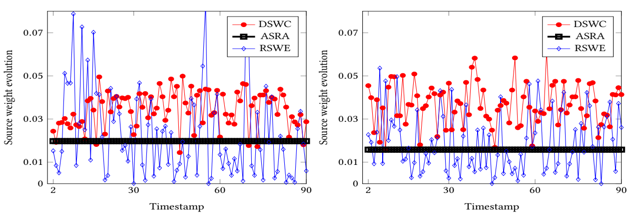
[2]The source we do not use does not contain temperature data.
[3]www.aerisweather.com
[4]www.apixu.com
[5]darksky.net/about/
[6]www.worldweatheronline.com
[7]www.wunderground.com
[8]Data collected from https://fixer.io/.

Parameter Settings		Method	TP	TN	FP	FN	Accuracy
α=0.7	ε=0.01	DSWC	0.19	0.6	0.01	0.2	**0.79**
		ASRA	0.13	0.55	0.06	0.26	0.69
	ε=0.1	DSWC	0.56	0.15	0.12	0.17	**0.71**
		ASRA	0.51	0.13	0.14	0.22	0.64
	ε=0.5	DSWC	0.8	0.16	0.01	0.03	**0.96**
		ASRA	0.68	0.13	0.04	0.15	0.81
ε=0.1	α=0.5	DSWC	0.61	0.17	0.1	0.12	**0.78**
		ASRA	0.56	0.14	0.13	0.17	0.7
	α=0.9	DSWC	0.25	0.27	0	0.48	**0.52**
		ASRA	0.18	0.22	0.05	0.55	0.4

Table 1: Prediction Model Evaluation for Weather Dataset

Parameter Settings		Method	TP	TN	FP	FN	Accuracy
α=0.7	ε=0.01	DSWC	0.27	0.58	0.09	0.06	**0.85**
		ASRA	0.25	0.49	0.18	0.08	0.74
	ε=0.1	DSWC	0.68	0.2	0.07	0.07	**0.88**
		ASRA	0.53	0.15	0.12	0.2	0.68
	ε=0.5	DSWC	0.75	0.13	0.01	0.11	**0.88**
		ASRA	0.73	0.11	0.03	0.13	0.84
ε=0.1	α=0.5	DSWC	0.68	0.19	0.07	0.06	**0.87**
		ASRA	0.55	0.14	0.12	0.19	0.69
	α=0.9	DSWC	0.23	0.25	0.02	0.5	**0.48**
		ASRA	0.15	0.22	0.05	0.58	0.37

Table 2: Prediction Model Evaluation for Rates Dataset



(a) Weather **(b) Rates**

Figure 1: Source Weight Evolution Condition Comparison

model. Two parameters, ϵ and α are varied to evaluate the performance with different settings.

Due to space limitation, we only show the experimental results conducted on weather and rates datasets in Tables 1 and 2. We can observe that the prediction model in DSWC outperforms the probabilistic model in ASRA under all parameter settings. The reason is that the prediction model of DSWC evaluates the covariances between the source weights over time. However, ASRA models the satisfaction of source weight evolution as a Bernoulli random variable. It overlooks the correlation of the source weights over the data streams, which results in less accurate prediction results.

Furthermore, we can observe that TN of ASRA is usually smaller than that in DSWC, this causes ASRA to compute source weights more frequently, which makes the truth inference process inefficient. Note that the accuracy of the prediction model is not high when $\alpha = 0.9$. The reason is that the variance of the posterior distribution is not small enough to assert $p(\delta^s_{i/j} \le r^s_{i/j}) \ge \alpha$, which results in a higher FN. However, if α is set to a relatively smaller number, the prediction model performs much better. In the experiments, when $\alpha = 0.7$, the accuracies of the prediction model in DSWC for the weather dataset are all above 0.7, and the accuracies are all above 0.85 in the rates dataset.

In summary, the prediction model in DSWC is effective. It predicts the source weight evolution correctly most of the times. This guarantees DSWC algorithm is both accurate and efficient for computing object truths over data streams.

7.3 Source Weight Evolution Condition

We use the weather and rates datasets to test the source weight evolution condition that satisfies $\phi^o_{j-1/j} \le \epsilon$ between consecutive timestamps for DSWC and ASRA. ϵ is set to 0.1, DyOP is run on the datasets with ground truths to obtain real source weights at each timestamp. As the source weight evolution condition is different for each source computed by $r^s_{i/j}$ in Proposition 4.3 for DSWC, we choose a random source from each dataset and compute the upper bound by the real source weights. The source weight evolution conditions for DSWC (red), ASRA (black) and real source weight evolutions (RSWE, blue) over the first 90 timestamps are plotted in Figure 1. We can observe that most of the blue dots are under red ones. It means the source weight evolution condition computed by DSWC can capture most of the real source weight evolution that ensures $\phi_{j-1/j} \le \epsilon$. There are many blue dots above the black dots but under the red dots. This indicates that DSWC is capable of capturing most of the true source weight evolution ensuring $\phi^o_{j-1/j}$ but ASRA cannot. For the moments when the blue dot is above the red one, it means the source weight evolution $\delta^s_{j-1/j}$ cannot guarantee $\phi^o_{j-1/j} \le \epsilon$. In summary, DSWC allows source weights to change more between adjacent timestamps, but still guarantees that the unit error is less than the user-defined threshold.

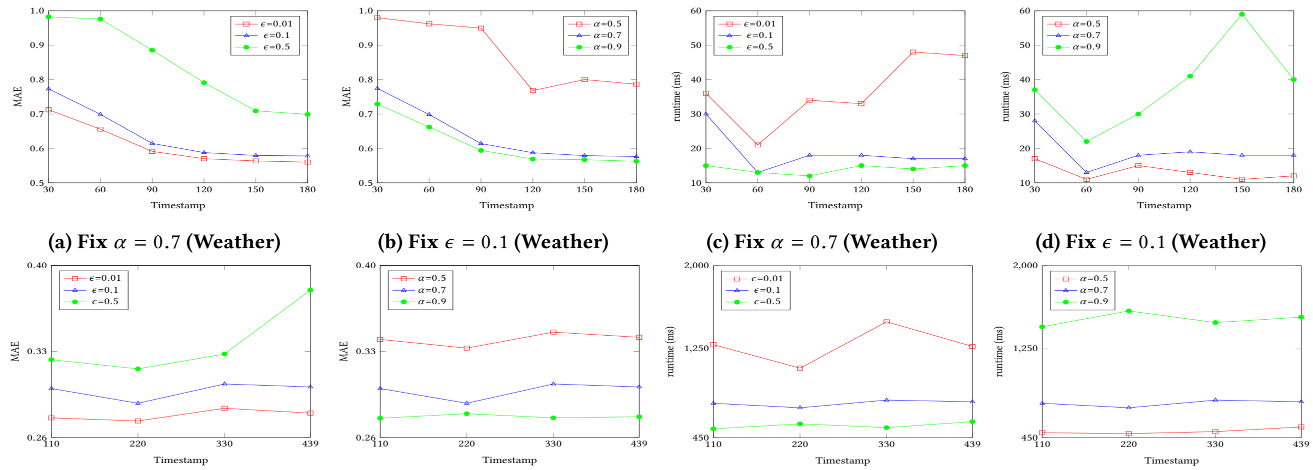
7.4 Parameters Analysis

We test the effect of parameters ϵ and α on the performance of DSWC. We conduct experiments by fixing one parameter and varying the other. Due to space limitation, only the results of weather and rates datasets run by DSWC(DyOP) are illustrated in Figure 2. In the figure, on one hand, we can see that as ϵ increases, MAE increases but the runtime decreases for both datasets. The reason is that larger ϵ increases the probability $p(\delta^s_{i/j} \le r^s_{i/j})$, which results in less iterative processes conducted over time. In this case, the truth inference process is configured to tolerate a large error, which runs more efficiently but less accurately. On the other hand, as α increases, MAE decreases but the runtime increases. This is because a larger α makes $p(\delta^s_{i/j} \le r^s_{i/j}) \ge \alpha$ less likely to hold. In this case, the truth inference is configured to tolerate a tiny error, which requires more iterative processes conducted. The experimental results on the other two datasets present a similar pattern to that of the weather and rates datasets shown in Figure 2.

7.5 Performance Comparison

We evaluate the performance of DSWC against the baselines with the following parameter settings. Weather dataset: $\epsilon = 0.1$, $\alpha = 0.7$ and $L = 5$. Stock dataset: $\epsilon = 10^{-3}$, $\alpha = 0.7$ and $L = 5$. Rates dataset: $\epsilon = 0.1$, $\alpha = 0.7$ and $L = 8$.

Table 3 summarizes the experimental results for all the methods conducted on the four datasets. For weather dataset, in terms of accuracy, DSWC is more accurate than the incremental methods. The reason is that the incremental methods cannot compute accurate source weights at each timestamp, which results in large



(a) Fix $\alpha = 0.7$ (Weather) **(b) Fix $\epsilon = 0.1$ (Weather)** **(c) Fix $\alpha = 0.7$ (Weather)** **(d) Fix $\epsilon = 0.1$ (Weather)**

(e) Fix $\alpha = 0.7$ (Rates) **(f) Fix $\epsilon = 0.1$ (Rates)** **(g) Fix $\alpha = 0.7$ (Rates)** **(h) Fix $\epsilon = 0.1$ (Rates)**

Figure 2: Parameters Analysis

Method	Weather		Stock		Forecast		Rates	
	MAE	Time(ms)	MAE	Time(ms)	MAE	Time(ms)	MAE	Time(ms)
DynaTD	0.9082	**53**	0.0278	**246**	1.356	**45**	0.5879	**1178**
DynaTD+s	0.8587	54	0.0243	265	1.2951	**45**	0.5633	1197
iCRH	0.9578	55	0.0291	308	1.3568	46	0.6125	1237
DyOP	0.6138	253	0.0196	1882	0.9929	124	0.2758	10439
CRH	0.6461	254	0.0194	2000	0.9953	125	0.2633	11253
LFC	0.6466	255	0.0194	2013	0.9983	125	0.2619	11274
GTM	0.6103	231	0.0193	1931	0.9547	121	0.2599	11887
OTD	0.5935	301	0.0192	1927	0.9654	144	0.2587	10889
ASRA(DyOP)	0.6678	117	N/A	N/A	1.0523	85	0.3233	3587
ASRA(DyOP+s)	0.6124	121	N/A	N/A	1.0395	87	0.3057	3653
ASRA(CRH)	0.6813	145	N/A	N/A	1.0531	86	0.3357	3777
ASRA(CRH+s)	0.65	147	N/A	N/A	1.0377	87	0.3291	3971
DSWC(DyOP)	0.6254	114	0.0209	805	1.0439	82	0.2997	2876
DSWC(DyOP+s)	**0.5905**	116	**0.019**	832	1.0215	82	0.2533	2805
DSWC(CRH)	0.6513	138	0.0215	877	1.0439	83	0.291	3011
DSWC(CRH+s)	0.6231	140	0.0193	883	1.0201	84	**0.2498**	3319
DSWC(GTM)	0.6215	139	0.0208	855	0.9765	82	0.2915	2885
DSWC(GTM+s)	0.5911	140	0.0195	861	**0.9455**	83	0.2531	2896

Table 3: Accuracy and Efficiency Comparison

errors when inferring truths. Compared with iterative-based methods, DSWC(DyOP) and DSWC(CRH) are less accurate since they approximate object truths at certain timestamps without updating source weights. However, DSWC(DyOP+s) and DSWC(CRH+s) are more accurate than the iterative-based methods because they infer object truths with smoothness constraint, but the iterative-based methods do not consider this when inferring truths. Note that although OTD uses a point estimate produced by ARIMA to assist its truth aggregation, it does not perform better than DSWC because ARIMA may not predict the truths correctly if the time series does not present a significant trend. DSWC methods are also more accurate than ASRA. The reason is that the source weight prediction model of DSWC is more accurate to predict source weight evolution, which results in less unsuccessful predictions that fail to assert $p(\delta^s_{i/j} \leq r^s_{i/j}) \geq \alpha$.

In terms of efficiency, the incremental methods have the best performance because they scan data only once. DSWC and ASRA only compute source weights at certain timestamps. Therefore, they are more efficient than the iterative-based methods which compute source weights at each timestamp. Compared with ASRA, DSWC is more efficient. The reason is that DSWC's prediction model can predict source weight evolution more accurately and DSWC has a more flexible source weight evolution condition to limit the unit error, which results in less number of iterative processes conducted to compute source weights over the data streams.

For the stock, forecast and rates datasets, the experimental results present a similar pattern to that of the weather dataset, except that ASRA cannot be performed on the stock dataset because its condition is not satisfied. Furthermore, we can observe that the accuracy of DSWC(GTM+s) performs significantly better than the other methods in terms of accuracy on forecast dataset. For example, MAE of DSWC(GTM+s) is 7.5% lower than DSWC(CRH+s), and 8.9% lower than ASRA(CRH+s). The reason of this is that each object is observed by few sources (only 5) in this dataset. Incorporating priors can help to reduce the "overfit" phenomenon in this dataset for better truth inference.

8 CONCLUSION

In this paper, we propose a novel method, DSWC, that can work with a wide range of truth inference methods to improve accuracy and efficiency for truth inference over data streams. DSWC dynamically computes agent/source weights over data streams. The error analysis and the source wight prediction model guarantee a high accuracy even if the source weights are only computed at certain timestamps. Compared with the existing work ASRA, DSWC can incorporate prior beliefs for computing object truths and DSWC's prediction model is more robust to predict source weights and source weight evolutions. Furthermore, DSWC does not need to satisfy the condition that all sources must observe all objects at each timestamp. Thus, it fits into more application scenarios. Experiments on four datasets demonstrate that the proposed method is both accurate and efficient for truth inference over data streams.

REFERENCES

[1] Charu C Aggarwal. 2013. *Managing and mining sensor data*. Springer Science & Business Media.

[2] Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied statistics* (1979), 20–28.

[3] Gianluca Demartini, Djellel Eddine Difallah, and Philippe Cudré-Mauroux. 2012. ZenCrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on World Wide Web*. ACM, 469–478.

[4] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. 2014. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 601–610.

[5] Xin Luna Dong, Laure Berti-Equille, Yifan Hu, and Divesh Srivastava. 2010. Global detection of complex copying relationships between sources. *Proceedings of the VLDB Endowment* 3, 1-2 (2010), 1358–1369.

[6] Xin Luna Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Kevin Murphy, Shaohua Sun, and Wei Zhang. 2014. From data fusion to knowledge fusion. *Proceedings of the VLDB Endowment* 7, 10 (2014), 881–892.

[7] Xin Luna Dong, Evgeniy Gabrilovich, Kevin Murphy, Van Dang, Wilko Horn, Camillo Lugaresi, Shaohua Sun, and Wei Zhang. 2015. Knowledge-based trust: Estimating the trustworthiness of web sources. *Proceedings of the VLDB Endowment* 8, 9 (2015), 938–949.

[8] Alban Galland, Serge Abiteboul, Amélie Marian, and Pierre Senellart. 2010. Corroborating information from disagreeing views. In *Proceedings of the third ACM international conference on Web search and data mining*. ACM, 131–140.

[9] Furong Li, Mong Li Lee, and Wynne Hsu. 2014. Entity profiling with varying source reliabilities. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1146–1155.

[10] Qi Li, Yaliang Li, Jing Gao, Lu Su, Bo Zhao, Murat Demirbas, Wei Fan, and Jiawei Han. 2014. A confidence-aware approach for truth discovery on long-tail data. *Proceedings of the VLDB Endowment* 8, 4 (2014), 425–436.

[11] Qi Li, Yaliang Li, Jing Gao, Bo Zhao, Wei Fan, and Jiawei Han. 2014. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 1187–1198.

[12] Tianyi Li, Yu Gu, Xiangmin Zhou, Qian Ma, and Ge Yu. 2017. An Effective and Efficient Truth Discovery Framework over Data Streams.. In *EDBT*. 180–191.

[13] Xian Li, Xin Luna Dong, Kenneth Lyons, Weiyi Meng, and Divesh Srivastava. 2012. Truth finding on the deep web: Is the problem solved?. In *Proceedings of the VLDB Endowment*, Vol. 6. VLDB Endowment, 97–108.

[14] Yaliang Li, Jing Gao, Chuishi Meng, Qi Li, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. 2016. A survey on truth discovery. *Acm Sigkdd Explorations Newsletter* 17, 2 (2016), 1–16.

[15] Yaliang Li, Qi Li, Jing Gao, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. 2015. On the discovery of evolving truth. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 675–684.

[16] Yaliang Li, Qi Li, Jing Gao, Lu Su, Bo Zhao, Wei Fan, and Jiawei Han. 2016. Conflicts to harmony: A framework for resolving conflicts in heterogeneous data by truth discovery. *IEEE Transactions on Knowledge and Data Engineering* 28, 8 (2016), 1986–1999.

[17] Subhabrata Mukherjee, Gerhard Weikum, and Cristian Danescu-Niculescu-Mizil. 2014. People on drugs: credibility of user statements in health communities. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 65–74.

[18] Robin Wentao Ouyang, Mani Srivastava, Alice Toniolo, and Timothy J Norman. 2016. Truth discovery in crowdsourced detection of spatial events. *IEEE transactions on knowledge and data engineering* 28, 4 (2016), 1047–1060.

[19] Carl Edward Rasmussen. 2004. Gaussian processes in machine learning. In *Advanced lectures on machine learning*. Springer, 63–71.

[20] Vikas C Raykar, Shipeng Yu, Linda H Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. 2010. Learning from crowds. *Journal of Machine Learning Research* 11, Apr (2010), 1297–1322.

[21] Theodoros Rekatsinas, Xin Luna Dong, and Divesh Srivastava. 2014. Characterizing and selecting fresh data sources. In *Proceedings of the 2014 ACM SIGMOD international conference on Management of data*. ACM, 919–930.

[22] Dong Wang, Lance Kaplan, Hieu Le, and Tarek Abdelzaher. 2012. On truth discovery in social sensing: A maximum likelihood estimation approach. In *Proceedings of the 11th international conference on Information Processing in Sensor Networks*. ACM, 233–244.

[23] Yi Yang, Quan Bai, and Qing Liu. 2018. On the Discovery of Continuous Truth: A Semi-supervised Approach with Partial Ground Truths. In *International Conference on Web Information Systems Engineering*. Springer, 424–438.

[24] Yi Yang, Quan Bai, and Qing Liu. 2019. A probabilistic model for truth discovery with object correlations. *Knowledge-Based Systems* 165 (2019), 360–373.

[25] Liuyi Yao, Lu Su, Qi Li, Yaliang Li, Fenglong Ma, Jing Gao, and Aidong Zhang. 2018. Online Truth Discovery on Time Series Data. In *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 162–170.

[26] Xiaoxin Yin, Jiawei Han, and S Yu Philip. 2008. Truth discovery with multiple conflicting information providers on the web. *IEEE Transactions on Knowledge and Data Engineering* 20, 6 (2008), 796–808.

[27] Dian Yu, Hongzhao Huang, Taylor Cassidy, Heng Ji, Chi Wang, Shi Zhi, Jiawei Han, Clare Voss, and Malik Magdon-Ismail. 2014. The wisdom of minority: Unsupervised slot filling validation based on multi-dimensional truth-finding. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*. 1567–1578.

[28] Bo Zhao and Jiawei Han. 2012. A probabilistic model for estimating real-valued truth from conflicting sources. *Proc. of QDB* (2012).

[29] Bo Zhao, Benjamin IP Rubinstein, Jim Gemmell, and Jiawei Han. 2012. A bayesian approach to discovering truth from conflicting sources for data integration. *Proceedings of the VLDB Endowment* 5, 6 (2012), 550–561.

[30] Yudian Zheng, Guoliang Li, Yuanbing Li, Caihua Shan, and Reynold Cheng. 2017. Truth inference in crowdsourcing: is the problem solved? *Proceedings of the VLDB Endowment* 10, 5 (2017), 541–552.