TRUTH DISCOVERY IN STREAMING DATA AND CROWDSOURCING APPLICATIONS

A THESIS SUBMITTED TO AUCKLAND UNIVERSITY OF TECHNOLOGY
IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Supervisors

Assoc. Prof. Quan Bai

Dr. Qing Liu

August 2019

By

Yi Yang

School of Engineering, Computer and Mathematical Sciences

Abstract

With the development of Internet and cellular network, it becomes much easier for people to receive information from multiple data sources. However, the data from different sources describing the same entity or object is usually conflicting and erroneous. Therefore, it is important to assess the data veracity, resolve the conflicts and extract the trustworthy information among the multi-source data for the downstream applications.

In this thesis, I focus on the truth discovery models to assess data veracity. Truth discovery is an emerging technique that estimates the most trustworthy information (also known as truth) of each object from the multi-source data. Specifically, a truth discovery model is usually an unsupervised learning model that learns the unknown source reliability from the observed multi-source data to better estimate object truth. This thesis advances truth discovery in applications where data is collected from data streams and crowdsourcing applications, specifically studies how to use object correlation in streaming data truth discovery and how to improve the accuracy and efficiency of streaming data truth discovery. For crowdsourcing applications, the thesis presents two truth discovery models that can better model human behaviors in the truth discovery steps. As most truth discovery methods are unsupervised learning models in which the ground truths of objects are unknown, the thesis also discusses how to use a small set of ground truths to guide the source reliability estimation and develops a semi-supervised truth discovery model to better discover object truths.

Contents

List of Tables

List of Figures

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the qualification of any other degree or diploma of a university or other institution of higher learning.

Signature of candidate

Publications

Yang, Y., Bai, Q., & Liu, Q. (2019a). Dynamic Source Weight Computation for Truth Inference over Data Streams. In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (pp. 277-285).

Yang, Y., Bai, Q., & Liu, Q. (2019b). Modeling Random Guessing and Task Difficulty for Truth Inference in Crowdsourcing. In Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems (pp. 2288-2290).

Yang, Y., Bai, Q., & Liu, Q. (2019c). A probabilistic model for truth discovery with object correlations. Knowledge-Based Systems, 165, 360-373.

Yang, Y., Bai, Q., & Liu, Q. (2018). On the Discovery of Continuous Truth: A Semi-supervised Approach with Partial Ground Truths. In International Conference on Web Information Systems Engineering (pp. 424-438).

Acknowledgements

Undertaking this PhD study is a challenging experience for me, it would not haven been possible for me to complete this PhD study and thesis without the guidance and support from many people.

Firstly, I would like to express my sincere gratitude to my primary advisor Assoc. Prof. Quan Bai for the continuous support of my PhD study and related research. His guidance helped me in all the time of research and writing of this thesis. Without his guidance and constant feedback this PhD would not have been achievable. I am also extremely grateful to my secondary advisor, Dr. Qing Liu, for her constructive advices and guidance to my research works.

I gratefully acknowledge the funding received towards my PhD from Auckland University of Technology.

Last but not the least, I would like to thank my family members who constantly support me during my PhD study.

Intellectual Property Rights

Chapter 1

Introduction

1.1 Background

In the era of big data, data resides in every aspect of our lives. With the development of computing power, Internet and wireless network, it becomes much faster and flexible to generate and publish data and information. Similarly, it becomes easier for people to obtain information from various channels and systems such as World-Wide-Web, social networks, mobile apps, physical sensors, and crowdsourcing platforms. People may use these data for different purposes. For example, a user may use a search engine to ask a question she is interested in, a data scientist may collect data from some meteorological monitoring sensors to forecast the future weather, and an AI expert may obtain some labeled data from a crowdsourcing platform to train a machine learning model. In all these scenarios, one phenomenon that may appear frequently is that data describing the same object could be from multiple sources. However, the multi-source data describing the same object is usually conflicting and noisy. For example, it is well-known that Mariana Trench has the deepest natural trench in the world, if you search the query "what is the depth of the Mariana Trench" on a search engine, such as Google, it gives

	New Zealand	Australia	France	Italy	China
Source 1	Auckland	Sydney	Marseille	Milan	Shanghai
Source 2	Wellington	Canberra	Paris	Rome	Beijing
Source 3	Auckland	Melbourne	Pairs	Venice	Beijing
MV	Auckland	Sydney	Pairs	Milan	Beijing
Weighted Voting	Wellington	Canberra	Pairs	Rome	Beijing

Table 1.1: A Motivative Example

you different results. From National Geographic[1], it reports the depth of Mariana Trench is 11,034 meters. However, Geology[2] reports the depth of Mariana Trench is 10,994 meters. Given the noisy data, it is hard for the users to decide which one to trust. Thus, one important task is to assess the veracity of the data and identify the most trustworthy information as the truth of each object. From this process, it turns the conflicting and noisy data into valuable information for the users to use in the downstream applications.

1.2 From Data Veracity Assessment to Truth Discovery

A straightforward approach of assessing data veracity is using majority voting (MV) or mean. MV works on the categorical data. For each object, MV chooses the data that receives the most votes as the truth of that object. Mean works on continuous data. For each object, mean simply takes the average of the data as the truth of that object. The assumption made underlying MV and mean is that they treat all the sources equally reliable. However, this assumption is usually not held in many scenarios because the reliabilities of sources vary in real-world applications. If we know the reliabilities of each source, it can help us better assess the data veracity and we can identify the truth for each object more accurately. For example, consider the example in Table 1.1. In this example, there are three sources providing information of capitals of five countries. From the information provided by the three sources, we want to find the true capitals

[1]https://www.nationalgeographic.org/activity/mariana-trench-deepest-place-earth/
[2]https://geology.com/records/deepest-part-of-the-ocean.shtml

of the five countries. From this table we can see that the three sources cannot reach an agreement on their claims. It is essential to decide which ones are veracious. MV identifies the truth by checking which candidate receives the most votes. However, it wrongly selects the capital cities of New Zealand, Australia and Italy. If we know that Source 2 is more reliable than the other two sources, we can weigh the votes from Source 2 more than the votes from the other two sources, then we can get better results as shown by the method Weighted Voting in Table 1.1.

The example discussed above demonstrates the effectiveness of data veracity assessment by considering the source reliability. If we know the source reliabilities, we can easily identify object truths accurately. However, the source reliabilities are usually unknown *a priori*. The only data we have on hand is usually the observed data claimed by the sources on each object. Thus, it is a challenging task to assess the source reliabilities from the data.

In the light of this challenge, truth discovery emerges as an effective technique for data veracity assessment by finding the true value of each object claimed by multiple data sources. Different from the naive MV and mean, truth discovery identifies object truths by estimating source reliabilities. The generalized task definition of truth discovery can be described as follows.

Definition 1.1. ***Truth Discovery (Y. Li, Gao et al., 2016)****: Given a set of objects and a set of sources, each source can make its claims on a (sub)set of objects. The goal of truth discovery is to aggregate the truths of the objects from the observed source claims by estimating the source reliabilities.*

As the object (Dong et al., 2015) truths and source reliabilities are usually both unknown *a priori*, truth discovery exploits an unsupervised approach which learns the source reliabilities and object truths simultaneously from the data. For this reason, truth discovery receives a lot of attentions and becomes a hot topic in the community.

1.3 Applications

Truth discovery has been successfully applied in many application domains. In terms of healthcare, an early research work (Dawid & Skene, 1979) that is related to truth discovery was proposed to evaluate the trustworthiness of clinicians' opinions. In the online healthcare community, the users can post health-related questions online and seek advices from the Internet users. However, the online users are not medical experts and some of their answers could be incorrect. The authors in (Mukherjee, Weikum & Danescu-Niculescu-Mizil, 2014; Y. Li et al., 2017) adopt truth discovery technique to analyze and aggregate the users' responses in order to find the most trustworthy answer for each question.

For data fusion and information extraction, the data describing the same entity could be extracted from different data sources, such as database, corpora and web pages. The data extracted from different sources could be conflicting, truth discovery (F. Li, Lee & Hsu, 2014; Yu et al., 2014; Dong et al., 2015) can be used to resolve these conflicts and output the accurate information for each entity.

With the development of social networks and mobile technologies, it becomes much easier for people to update real-time information about some events we are interested in. People can engage directly with the mobile Internet and share real-time experiences at an unprecedented scale in social sensing applications (D. Wang, Abdelzaher & Kaplan, 2015). For example, Waze[3] is a social sensing navigation mobile app that allows the drivers to upload the real-time road conditions, and the other drivers can use this app to view the places having congestions and avoid some traffic problems. However, the data uploaded by the users are not always accurate. Thus, truth discovery (Le et al., 2011; Su et al., 2014; D. Wang, Kaplan & Abdelzaher, 2014; S. Wang et al., 2015; S. Wang, Wang, Su, Kaplan & Abdelzaher, 2014; Gupta, Lamba, Kumaraguru & Joshi, 2013)

[3]https://www.waze.com/

can be used to aggregate the noisy user data and discover accurate information.

Crowdsourcing platforms, such as Amazon Mechanical Turk[4] and Figure Eight[5], provide a cost-effective and efficient way to collect labeled data from the crowd workers. The requesters can post the unlabeled data as tasks on a crowdsourcing platform, and the crowd workers will label the tasks and earn some rewards. However, most crowd workers are not experts and their labeling abilities are different. As a result, the labeled data collected from the crowd workers are usually conflicting and noisy. To tackle this problem, truth discovery techniques can be applied to process the noisy data and output the most trustworthy label for each task.

1.4 Research Motivation

Although truth discovery techniques have been applied in many applications and with different merits, there are still some areas that are left unexplored. The primary research gaps are identified as follows.

- **Streaming Data.** 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. Most of the traditional truth discovery methods are designed for static data, in which time dimension is not involved. These methods adopt an iterative approach that updates source reliabilities and object truths iteratively. Although these iterative-based methods can accurately discover object truths, the iterative process is too computationally expensive to process streaming data in which data arrives sequentially from data streams. To tackle this problem, some incremental truth

[4]https://www.mturk.com/
[5]https://www.figure-eight.com/

discovery methods are developed for steaming data. These incremental truth discovery methods are efficient but they cannot estimate source reliabilities, which results in large errors when estimating object truths. Therefore, it is desired to have a truth discovery method that is both accurate and efficient for estimating object truths when data arrives from data streams.

- **Object Relation in Dynamic Environment.** Usually, the objects are correlated and there exists some relationships among the objects. In the truth discovery literature, there are a few research works that consider object relations in the truth discovery process. It is also claimed that using object relation can improve truth discovery accuracy. However, the existing works that consider object relations are proposed to work in a static environment where all the data is assumed to be available all at once. In a dynamic environment where data arrives sequentially from data streams, the existing methods accounting for object relations cannot efficiently aggregate object truths from the streaming data. Therefore, it is demanded to have an efficient truth discovery method accounting for object relation that can work in both static and dynamic environment.

- **Partially Observed Ground Truths.** In a traditional truth discovery setting, the source reliabilities and object truths are both unknown *a priori*. Most of the truth discovery method exploits an unsupervised approach that learns the source reliabilities and object truths at the same time from the data. Although it is unpractical and expensive to obtain the ground truths of all the objects, sometimes it is possible to get the ground truths of a very small set of objects. Thus, it is important to have a semi-supervised truth discovery method that can use the small amount of valuable ground truths. Yin & Tan et al. (2011) tackled this problem by using graph propagation. However, the proposed method was designed for categorical data, but it does not perform well on continuous data.

- **Human Sources.** In many applications the sources that provide information are human. Different from the non-human sources, human sources have different patterns when providing data. For example, in crowdsourcing applications, a crowd worker (as human source) may spam the tasks by submitting randomly labels without even knowing the details of the tasks. A worker may also have different probabilities of choosing the choices of tasks. Thus, it is crucial to consider the human sources' unique behaviors in the truth discovery process.

1.5 Research Question

According to the research motivations mentioned above, the research objectives are described by four research questions given below.

Research Question 1: How to use object relations in truth discovery in a dynamic environment?

- **Research Sub-question 1.1:** How to model object relations in a dynamic environment?
- **Research Sub-question 1.2:** How to efficiently discover object truths in a dynamic environment when object relation is considered?

Research Question 2: How to achieve both high accuracy and high efficiency for streaming data truth discovery?

- **Research sub-question 2.1:** How to discover object truths efficiently if data arrives from data streams?
- **Research sub-question 2.3:** Given that achieving high accuracy is an objective of streaming data truth discovery, how to further improve stream data truth discovery efficiency?

Research Question 3: If the sources are human, how to model the humans' own characteristics in the truth discovery model?

- **Research sub-question 3.1:** How to model humans' guessing behaviors in a truth discovery model?
- **Research sub-question 3.2:** How to better model humans' labeling process for inferring truths in crowdsourcing applications?

Research Question 4 How to effectively use a small amount of ground truths to better aggregate continuous object truths?

- **Research sub-question 4.1:** How to discover object truths in an un-supervised manner?
- **Research sub-question 4.2:** How to adjust the importance of the ground truths in an un-supervised truth discovery model?

1.6 Research Method

My PhD research method is an iterative process which is summarized in Figure 1.1 on the next page. The first step is to conduct literature review of truth discovery. Next, I identify the research gaps from the existing research works and sub-sequentially propose a research question attempting to address the identified research gap. Then two processes run in parallel. On one hand, I develop a solution that can be used to solve the proposed research question. On the other hand, I collect datasets that can be used to later evaluate the proposed solution. I use three approaches to collect data.

1. Search for the publicly available datasets used in existing research works.
2. Crawl data from websites.

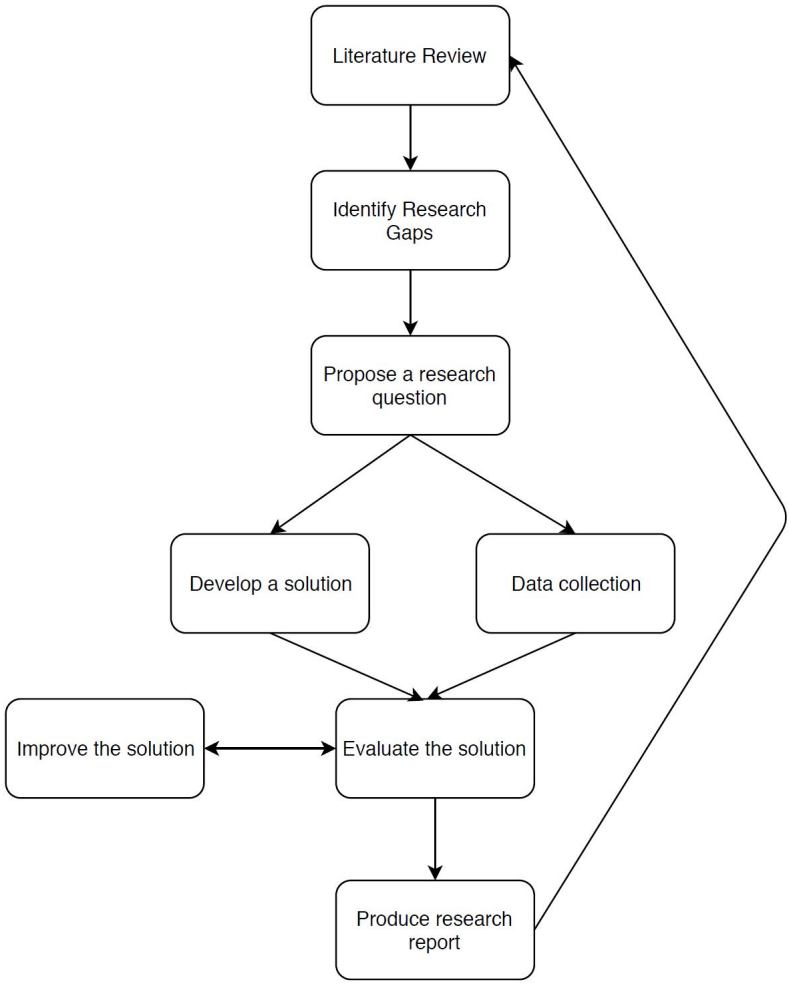


Figure 1.1: Research Method

3. Post the wanted data on a crowdsourcing platform, and ask the crowd workers to label the data.

After a solution is developed and the ideal datasets are collected, I evaluate the developed solution. This is achieved by comparing the proposed solution against state-of-the-art truth discovery methods and see if the performance is improved according to the proposed research question. By analyzing the evaluation results, I would improve the proposed solution if required. If no improvement room can be identified, I produce the research report and then conduct the literature review again to identify new research

gaps.

1.7 Contributions of the Thesis

In this thesis, I propose five truth discovery models which solve different aspects raised in the research questions. Based on the five discovery models, the contributions of the thesis are summarized as follows:

1. I propose the **P**robabilistic **T**ruth **D**iscovery with Object **Corr**elations (PTDCorr) approach that can incorporate object correlations to better estimate source reliabilities and object truths. The PTDCorr model aims at answering Research Question 1.1. Based on PTDCorr, I develop an incremental iPTDCorr algorithm which works efficiently in a dynamic environment. iPTCorr is able to efficiently aggregates truths of objects from a data stream without re-processing the historical data. The iPTCorr model aims at answering Research Question 1.2. The model and results are published in (Yang, Bai & Liu, 2019c).

2. I propose the **D**ynamic **S**ource **W**eight **C**omputation Truth Discovery schema (DSWC) that can efficiently apply iterative-based truth discovery methods to streaming data truth discovery applications. The improvement of efficiency is achieved by running the iterative processes at only certain time-stamps instead of all the time-stamps. The proposed DSWC model aims at answering Research Question 2. The model and results are published in (Yang, Bai & Liu, 2019a).

3. I propose a model, called **C**rowdsourced **T**ruth **D**iscovery modeling **G**uessing and task **D**ifficulty (CTDGD), that aggregates crowdsourcing single-choice tasks' truths by jointly modeling tasks' difficulties and the crowd workers' abilities (reliabilities) and guessing behavior. The proposed CTDGD model aims at

answering Research Question 3.1. The model and results are published in (Yang, Bai & Liu, 2019b).

4. I propose a model, called Confusion-aware Truth Inference (CTI), that aggregates crowdsourcing single-choice tasks' truths by considering choices' confusion degrees brought to the crowd workers. The proposed CTI model aims at answering Research Question 3.2.

5. I propose the **Op**timization-based **S**emi-supervised **T**ruth **D**iscovery (OpSTD) that can use a small set of ground truths to improve the accuracy of truth discovery. The weights of the ground truths can be tuned freely by a hyperparameter. The proposed OpSTD model aims at answering Research Question 4. The model and results are published in (Yang, Bai & Liu, 2018).

In summary, The proposed models, PTDCorr, iPTDCorr and DSWC are developed for general truth discovery tasks in which the data is fed into the applications from data streams. These three methods aim at improving both accuracy and efficiency for streaming data truth discovery. CTDGD and CTI are developed for truth discovery for crowdsourcing applications in which the sources are human. The novelty of CTDGD and CTTI is that they both consider the characteristic human behaviors in the truth discovery steps. Finally, the OpSTD model is a general truth discovery method that can be applied to all kinds of applications if a small set of ground truths are available.

1.8 Thesis Structure

The rest of the thesis is organized as follows:

- **Chapter 2** reviews the state-of-the-art truth discovery models.

- **Chapter 3** presents the models of PTDCorr and iPTDCorr which discover object truths with object correlations in both static and dynamic environment.
- **Chapter 4** presents the model of DSWC that aims at improving both accuracy and efficiency for streaming data truth discovery.
- **Chapter 5** introduces a crowdsourcing truth discovery method CTDGD that estimates object truths by estimating human source reliabilities and object difficulties and modeling human sources' guessing behavior.
- **Chapter 6** presents a crowdsourcing truth discovery method CTI that considers tasks' choice confusion degrees in crowdsourcing applications.
- **Chapter 7** presents the model of OpSTD, a semi-supervised truth discovery method for estimating continuous object truths.
- **Chapter 8** concludes the thesis. It also outlines the future work.

Chapter 2

Literature Review

Truth discovery has received a lot of attentions in recent years, and many truth discovery methods have been proposed with different merits. This chapter aims at providing a preliminarily of truth discovery, and reviews related research works of truth discovery in order to identify research gaps. In Section 2.1, it reviews three common frameworks that are adopted by many truth discovery methods. In Section 2.2, it reviews the aspects of sources that are considered in truth discovery methods. Section 2.3 reviews the aspects of objects that are studied in the existing truth discovery algorithms. Section 2.4 presents the commonly used metrics for evaluating the truth discovery algorithms. In Section 2.5, it lists some representative truth discovery methods. Finally, Section 2.6 summarizes this chapter.

2.1 General Truth Discovery Frameworks

This section starts by introducing the notations and symbols. The generalized truth discovery definition was given in Definition 1.1. Next, the restatement of truth discovery with symbols is given below.

Truth Discovery: There is a set of objects J, for each of the object $j \in J$, the

Symbol	Description
J	The set of all the objects
I	The set of all the sources
j	A single object
i	A single source
a_i	The real weight (a.k.a. reliability, quality) of source i
$\hat{a}_i$	The estimated weight (a.k.a. reliability, quality) of source i
A	The set of all source weights
$\hat{A}$	The set of all estimated source weights
x_{ij}	The claim (observation) of source s on object j
X	The set of all claims
X_j	The set of all claims on object j
X_i	The set of all claims provided by source i
z_j	The real truth of object j
$\hat{z}_j$	The estimated truth of object j
Z	The set of all real object truths
$\hat{Z}$	The set of all estimated object truths
J_i	The set of objects that are claimed by source i
I_j	the set of sources that claim (observe) object j

Table 2.1: Notations in Literature Review

information of the object j can be claimed by a set of sources I. Each object j has an unknown object truth z_j and each source i has an unknown source reliability a_i. Let x_{ij} denote the information of object j claimed by source i where $i \in I$, and $X = \{x_{ij} | i \in I, j \in J\}$ denotes all the claims provided by all the sources. Given all the claims X, the goal of truth discovery is to estimate the truth $\hat{z}_j$ for each object such that the estimated object truth $\hat{z}_j$ shall be close to the unknown real object truth z_j.

The important notations and symbols that will be used in this chapter are summarized in Table 2.1. Note that in the subsequent chapters, the notations are re-defined to better suit the context of the problem solved in that chapter.

As discussed in Chapter 1, truth discovery estimates the object truths by considering source reliabilities. In the truth discovery literature, the source reliability is also known as source weight and source quality. In the rest of this chapter, it uses the terms *source reliability*, *source weight* and *source quality* interchangeably. As the object truths and

source weights are both unknown *a priori*, the following principle is widely adopted (Y. Li, Gao et al., 2016).

Principle of Truth Discovery: A source is assigned with high weight if it frequently provides trustworthy information, and the information supported by the sources with high weights are more likely to be selected as the truth for the objects.

Three general truth discovery frameworks are presented which incorporate the principle of truth discovery (Y. Li, Gao et al., 2016).

2.1.1 Iterative Framework

In truth discovery, the source weights and object truths are inter-dependent. To incorporate the inter-dependencies between source weights and object truths, some truth discovery methods (Dong, Berti-Equille & Srivastava, 2009a; Galland, Abiteboul, Marian & Senellart, 2010; Pasternack & Roth, 2010; Yin, Han & Philip, 2008) are developed as iterative algorithms which treat source weights and object truths as unknown variables. These methods update source weights and object truths alternatively and iteratively until the algorithm converges.

In the object truth update step, the source weights are fixed, and the object truths are updated by weighted voting. For example, in Investment (Pasternack & Roth, 2010), the sources uniformly vote their claims by using their source weights, the object truths are updated by the weighted voting. Specifically, for a possible truth candidate x of object j where $x \in X_j$[1], the trustworthiness $T(x)$ of the candidate x is computed as:

$$T(x) = \sum_{i \in I_x} \frac{a_i}{|X_i|} \tag{2.1}$$

In Equation (2.1), a_i is the weight of source i, a larger weight indicates that the source

[1] X_j is the set of claims on object j. For example, assume there are 7 sources $[i_1, \ldots, i_7]$, the claims of these sources on object j are $[A, A, B, B, C, C, D]$. Then the truth candidates of object j are $\{A, B, C, D\}$ and $x \in \{A, B, C, D\}$.

is more reliable. I_x is the sources whose claims are x, X_i is the set of claims made by source i and $|\cdot|$ denotes its cardinality. Then the truth z_j of object j is updated by choosing the candidate with the highest trustworthiness. From Equation (2.1), we can see that the trustworthiness of a candidate is determined by the weights of the sources which claim it. A candidate is more trustworthy if the sources S_x are more reliable.

In the source weight update step, the object truths computed in the previous step are fixed. The source weights harvest the reliabilities back from the updated object truths, and can be computed as

$$a_i = \sum_{x \in X_i} (T(x) \times \frac{a_i/|X_i|}{\sum_{i' \in I_x} a_{i'}/|X_{i'}|}) \tag{2.2}$$

From Equation (2.2) we can see that the source weight is determined by the trustworthiness of the claims the source provides. A source is assigned with a higher weight value if its claims are more trustworthy.

After the iterative algorithm converges, for each object, it selects the truth candidate with the highest trustworthiness as the estimated truth of the object.

2.1.2 Optimization Framework

There are some truth discovery methods incorporate the principle of truth discovery is incorporated by modeling the truth discovery as an optimization problem (Y. Li et al., 2015; Q. Li, Li, Gao, Zhao et al., 2014; Q. Li, Li, Gao, Su et al., 2014; Aydin et al., 2014; Y. Li, Li et al., 2016, 2016), which aims at optimizing an objective function f defined in Equation (2.3)

$$f = \sum_{j \in J} \sum_{i \in I} a_i \times d(x_{ij}, z_j) \tag{2.3}$$

In Equation 2.3, $d(\cdot)$ is a distance function which measures the distance between a source's claim and the object truth. In other words, the distance between a source's claim and object truth is the error that the source makes when it claims an object. Different distance functions can be plugged in depending on the data types in the truth discovery application. For example, $0-1$ loss function can be used on categorical data and L^2 norm distance can be used on continuous data.

The objective function f represents the overall errors between the claims and the object truths. Truth discovery methods formulated as an optimization problem aim at finding the set of estimated object truths and source weights that minimize the objective function f. One one hand, if $d(x_{ij}, z_j)$ is big, in order to minimize the objective function, it needs to reduce the weight a_i of source i. On the other hand, if a_i is high, in order to minimize the objective function, it needs to adjust z_j closer to x_{ij}.

To find the optimal solutions defined by Equation (2.3), coordinate descent (Bertsekas, 1999) can be adopted. Coordinate descent is an iterative non-linear optimization algorithm that updates the unknown variables alternatively. In the truth discovery problem, there are two sets of unknown variables $Z = \{z_j | j \in J\}$ and source weights $A = \{a_i | i \in I\}$. In each iteration, coordinate descent fixes one set of variables, e.g. Z, and updates the other set of variables, e.g. A. Then it fixes the other set of variables and updates the variables that are fixed previously. Coordinate descent terminates until the algorithm converges, and uses the updated object truths and source weights in the last iteration as the estimated (optimal) object truths and source weights.

2.1.3 Probabilistic Graphical Model Framework

There are some truth discovery methods (Pasternack & Roth, 2013; B. Zhao & Han, 2012; B. Zhao, Rubinstein, Gemmell & Han, 2012; Z. Li, Han, Yu et al., 2016; X. Wang et al., 2016a; Dong et al., 2015; Zhi et al., 2015; X. Wang et al., 2016b) tackle the

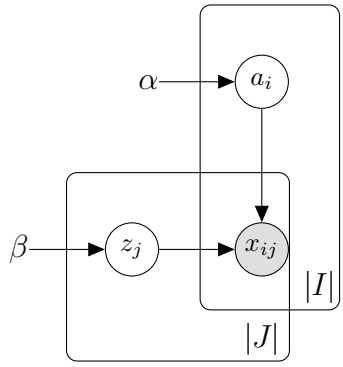


Figure 2.1: Plate Model of PGM

truth discovery problem by using **p**robabilistic **g**raphical **m**odel (PGM), in which the object truth, claims and source are modeled as random variables. Specifically, the unknown random variables in the truth discovery problem are the set of object truths $Z = \{z_j | j \in J\}$ and the set of source weights $A = \{a_i | i \in I\}$, and the observed (known) random variables are the set of claims $X = \{x_{ij} | j \in J, i \in I\}$. In a PGM, the relationships among the random variables are usually depicted by a graph. For example, Zhao et al. (2012) propose GTM that models the truth discovery problem by Bayesian network (a sub-class of GTM), and the graph that depicts GTM can be seen in Figure 2.1.

In Figure 2.1, the random variables (a_i, x_{ij} and z_j) are drawn in the circles, and the circle of the observed variable (x_{ij}) is shaded. The hyperparameters are drawn with no borders (α and β). There are two plates with labels at the bottom right, which represents the occurrence of random variables in the model. We can read from this figure that there are $|I|$ sources and $|J|$ objects. Here I make the assumption that each source observes all the objects. Thus, there are $|I| \times |J|$ observations in total. However, the generative model also supports sparsity in the claim set, i.e., each source claims a subset of objects.

In the context of GTM, both object truths and claims are continuous. Thus, it assumes that object truth z_j is generated from a Normal distribution with hyperparameter β. It assumes that the source weight is generated from an Inverse-Gamma distribution with hyperparameter α. Intuitively, the claim x_{ij} is dependent on both the object truth

and the source weight, thus, x_{ij} is assumed to be generated from a Normal distribution $\mathcal{N}(z_j, 1/a_i)$ with mean z_j and precision (inverse of variance) a_i. Therefore, taking a random draw $x_{ij} \sim \mathcal{N}(z_j, 1/a_i)$, x_{ij} is closer to the object truth if a_i is high.

Given the formulation of GTM, the joint likelihood function can be formulated in Equation (2.4)

$$p(A, Z, X) = \prod_{i \in I} p(a_i|\alpha) \prod_{j \in J} \Big(p(z_j|\beta) \prod_{i \in I} p(x_{ij}|z_j, a_i) \Big) \tag{2.4}$$

To find the optimal object truths and source weights, probabilistic inference techniques, such as Expectation-Maximization (EM) and Maximum A Posterior (MAP), can be used. In the case of GTM, as the Normal distribution that generates the object truths and the Inverse-Gamma distribution that generates the source weight are conjugate priors of the Normal distribution that generates the claim, the object truths and source weights can be inferred efficiently by EM algorithm.

2.1.4 Summary

Although the three models discussed above solve truth discovery problem differently, the algorithm steps that estimate the object truths and source weights are similar. In the iterative based methods, the update equations of object truths and source weights are derived directly by capturing the inter-dependencies between the object truths and source weights. They perform an iterative process to update object truths and source weights alternatively. In the optimization based methods, an objective function aiming at minimizing the overall error is formulated first, then it can use optimization algorithms, such as coordinate descent, to minimize the objective function. The coordinate descent is an iterative algorithm that updates the object truths and source weights alternatively. In the PGM based methods, the dependencies between object truths, source weights and claims are modeled as random variables and the relationships among them are

established by conditional probabilities. In order to find the optimal solutions for the object truths and source weights, probabilistic inference algorithm, such as EM algorithm, can be used. The EM algorithm also leads to an iterative solution. Hence, we summarize the general steps of truth discovery in Algorithm 2.1.

Algorithm 2.1: General Steps of Truth Discovery

Input : Claims X
Output : Estimated object truths Z and estimated source weights A

```
1  Initialize sources weights A
2  repeat
3      for j ∈ J do
4          compute object truth z_j based on source weights A and X
5      end
6      for i ∈ I do
7          compute source weight a_i based on object truths Z and X
8      end
9  until Convergence condition is met
10 return Ẑ and Â
```

In Algorithm 2.1, truth discovery starts with the initialization of source weights (Line 1). If prior knowledge is available, the source weights can be initialized differently according to the prior knowledge. Otherwise, the sources are treated equally initially and they are assigned the same source weights. Then it conducts an iterative process to update the object truths and source weights based on the derived update equations (Lines 2 - 8). The iterative process stops until it meets some convergence condition (Line 9). The convergence condition is determined by the used algorithm. Finally, it returns the updated object truths and source weights in the last iteration as the estimated object truths and source weights (Line 10).

2.2 Aspects of Source

In this section, the aspects of sources that impact truth discovery are reviewed.

2.2.1 Source Reliability Modeling

In this subsection, two source reliability modeling techniques are reviewed.

Single Number and One-coin Model

A majority of truth discovery research works model the source reliability as a single number. In (Aydin et al., 2014; Demartini, Difallah & Cudré-Mauroux, 2012; Karger, Oh & Shah, 2011), the source reliability is modeled as a single number between 0 and 1: $a_i \in [0, 1]$, it represents the probability that the claims provided by source i are the truths. Some works (Q. Li, Li, Gao, Zhao et al., 2014; Y. Li, Li et al., 2016; Whitehill, Wu, Bergsma, Movellan & Ruvolo, 2009) extend this idea by modeling the source reliability on the real line $a_i \in (-\infty, +\infty)$. A source with high w^s is more reliable and its claims are more trustworthy. If the object truths are categorical (see Section 2.3.2 on page 39) such that the object truth can only be chosen from K choices, modeling source reliability is also known as the one-coin model. In one-coin model, it assumes that a source has the same probability that its claim is wrong. For example, given the source reliability a_i, the probability that x_{ij} is the truth of object j is

$$p(x_{ij} = k | z_j = k, a_i) = f(a_i)$$

where $k \in [1, K]$ f is a function that maps a_i to probability if $a_i \in (-\infty, +\infty)$. Then, for any of the rest wrong choices $k' \neq k \cap k' \in [1, K]$, the probability that source s claims k' is

$$p(x_{ij} = k' | z_j = k, a_i) = \frac{1}{K-1}(1 - f(a_i))$$

Two-coin and Confusion Matrix Model

If the object truths are binary, i.e., the object truth can only be chosen from two choices. The source reliability can be modeled by two-coin model (Raykar et al., 2010). In

two-coin model, the source reliability is modeled as two numbers $a_i = \{\alpha_i, \beta_i\}$. For example, assume the object truths can be taken from $\{0, 1\}$. If the object truth $z_j = 1$, the sensitivity (true positive rate) for a source i is defined as the probability that the source' claim on object j is 1

$$\alpha_i := p(x_{ij} = 1 | z_j = 1)$$

If the object truth is 0, the specificity (1 - false positive rate) is defined the probability that the source's claim on object o is 0

$$\beta_i := p(x_{ij} = 0 | z_j = 0)$$

Based on the two-coin model, the source reliability can be modeled by a confusion matrix if the object truths can be taken from K choices $[1, K]$ (Raykar et al., 2010; Dawid & Skene, 1979; Kim & Ghahramani, 2012; Venanzi, Guiver, Kazai, Kohli & Shokouhi, 2014). In this case, the source reliability a_i is modeled as a $K \times K$ confusion matrix in which all the entries are non-negative and each row sums up to 1. The (m, n) entry of a source's confusion matrix represents the conditional probability that the source's claim is n given the object truth is m. The confusion matrix model is a generalization of the two-coin model. When $K = 2$, the confusion matrix model is equivalent to the two-coin model.

Confidence

In some applications, the number of observations provided by each source is usually different. Some datasets present a long-tail phenomenon in which the majority of the objects are claimed by only few sources, and many sources only claim a few of objects. If a source only claims few objects, the estimated source weight could be inaccurate

because the estimated source weight is not statistically confident. To tackle this problem, confidence interval estimation is used when estimating the source weights. The authors develop CATD (Q. Li, Li, Gao, Su et al., 2014) that uses Chi-square distribution with 95% confidence interval when estimating the source weights. Specifically, the estimated source weights are scaled down by $\mathcal{X}_{(0.975,|J_i|)}$ where $|J_i|$ is the number of objects claimed by source i. For example, if there are two sources i_1 and i_2 who claim 100 and 10 objects respectively. i_1 may claim 80 objects correctly, while i_2 may claim 8 objects correctly. If the confidence interval is not considered, the estimated source weights of i_1 and i_2 could be the same. By using the confidence interval, the source weight of i_2 is scaled down because it is not statically confident in estimating the source weight of i_2.

Diverse Source Reliabilities

In most truth discovery methods, it assumes that a source has the same reliability when claiming all the objects. In some cases, the objects can be divided into clusters based on some features of the objects. For example, the objects can be politics-related or sports-related. Given a sports website (source), the sports-related information provided from this website is meant to be more trustworthy than the politics-related information. Inspired by this idea, existing works (Ma et al., 2015; Welinder, Branson, Perona & Belongie, 2010; Z. Zhao, Wei, Zhou, Chen & Ng, 2015; Zheng, Li & Cheng, 2016) model source's diverse reliabilities. The general approach to diverse source reliabilities modeling is that the source weight a_i is modeled as a vector with K values and K is the number of clusters that the objects can have. Thus, it uses the k^{th} element in a_i to estimate the truth of an object if the object belongs to cluster k.

2.2.2 Source Relationship

Most truth discovery methods have the source independence assumption such that the sources claim objects independently. In other words, the sources are uncorrelated with each other, and each source provides their claims without the knowledge of other sources. In some cases, the sources may not be independent and there exists some relationships between the sources.

One type of relationship that exists among sources is copying, i.e., a source copies information from other sources. In (Dong, Berti-Equille, Hu & Srivastava, 2010; Dong et al., 2009a; Dong, Berti-Equille & Srivastava, 2009b; Pochampally, Das Sarma, Dong, Meliou & Srivastava, 2014), the authors propose truth discovery models that consider information copying between sources. The general idea is that it is common if sources share many true claims. However, if two sources share many false claims, then the two sources may be highly correlated and one may copy information from the other one. Taking ACCU (Dong et al., 2009a) as an example, given two sources A and B, the authors consider two copying factors. (1) The copying direction, i.e., does A copy B or B copy A or there is no copying relationship between them. (2) If A copies information from B, A may not copy all the information from B, then what is the likelihood that A copies from B and how much information is copied. The copying detection is tightly combined with the truth discovery steps when estimating source weights and object truths, if the copying relationship is detected, then the weight of the copying source is deceased accordingly. Based on this idea, (Dong et al., 2009b) uses Hidden Markov Model to detect copying relationship in a dynamic environment in which the claims are made at different time-stamps. In (Pochampally et al., 2014), the authors study more complex copying relationship such as co-copying, positive and negative correlation among sources.

The other type of source relationship is clustering. If some sources belong to the

same cluster or community, they may share some common attributes, e.g., reliabilities. In (Venanzi et al., 2014), the authors propose a community based truth discovery algorithm *CommunityBCC*. In CommunityBCC, it assumes that the sources comform to a few different community, where each community represents a cluster of sources with similar source weights.

2.2.3 Human Sources

There usually exists two types of sources involved in the truth discovery studies: human sources and non-human sources. Non-human sources involves websites, databases, physical sensors, etc. The errors made by a non-human source is usually caused by missing records, typos, out-of-date data, measurement precision, etc. The non-human sources usually do not provide erroneous information on purpose. The errors made by human sources are more complex to analyze because the way a human source may have different intentions when providing information. In crowdsourcing truth discovery literature (Whitehill et al., 2009; Ghosh, Kale & McAfee, 2011; Karger et al., 2011; Raykar & Yu, 2012; Ipeirotis, Provost & Wang, 2010), based on the worker (human source) intention, the workers can be divided into three types: the honest worker, the spammer source and adversarial workers. The trustworthiness of data provided by an honest worker is largely based on the worker's ability, i.e., if the worker knows the truth of an object, she is willing to provide it. Of course, if the information provided by an honest worker is false, it is because the worker's ability is not high enough and she does not know the truth of the object. If a worker is a spammer, she spams the crowdsourcing platform by providing random information to the objects without making an effort. If a worker is adversarial, she would intentionally provide false information given that she knows the truth of the objects. It is found that if a worker's claims are random, her claims are useless in estimating the truths of objects (H. Li, Zhao & Fuxman,

2014). Thus, the claims provided by spammers should be cleansed. It turns out that the claims provided by adversarial workers can potentially benefit truth finding (Raykar & Yu, 2012). If a worker is identified as an adversarial worker, we can decrease the trustworthinesses of the truth candidates the adversarial worker claim, instead, we can relatively increase the trustworthinesses of the truth candidates the adversarial worker does not claim.

2.3 Aspects of Object

In this section, it reviews the aspects of objects studied in the truth discovery literature.

2.3.1 Object Truth Assignment

The truth assignment of most truth discovery models can be categorized into the the *scoring method* and *labeling method*. The scoring method (Yin & Tan, 2011; Yin et al., 2008) first identifies the truth candidates for each object, then for each object, it assigns a trustworthiness score for each truth candidate based on the reliabilities of sources which vote (claim) the candidate. Finally, it performs a post decision-making process to select the truth from the candidates. Usually the candidate with the highest trustworthiness score is selected as the truth. The scoring method is widely adopted in the iterative based truth discovery methods (see Section 2.1.1).

The *labeling method* (B. Zhao & Han, 2012; Q. Li, Li, Gao, Zhao et al., 2014; Y. Li et al., 2015) directly assigns a label or a true value to an object instead of selecting one from the truth candidates. For example, if three sources provide the information of Auckland temperature on a given day as 25.6, 26.1 and 26.5 and the weights of the three sources are 1.0, 1.1 and 1.3 respectively. By weighted voting, the estimated truth is computed as $\frac{25.6*1.0+26.1*1.1+26.5*1.5}{1.0+1.1+1.5} \approx 26.13$. The labeling method is usually used in

optimization based truth discovery methods (see Section 2.1.2) and PGM based truth discovery methods (see Section 2.1.3).

2.3.2 Object Truth Data Type

Generally, based on the data type, the object truth can be divided into two types: categorical and continuous. If an object truth is categorical, then the truth can only take certain values, e.g., K values indexed from 1 to K. If an object truth is continuous, then the truth can take infinite number of possible values within a range, e.g., any number on the real line. Different object truth types are usually modeled differently in different truth discovery frameworks.

The iterative based truth discovery methods are mainly developed for estimating categorical object truths by assigning trustworthiness scores to each truth candidate. As the iterative based truth discovery methods usually adopt scoring method to assign object truths, they need to construct truth candidate set for each object first. Although the continuous truth can take infinite number of possible values within a range, the number of sources that claim an object is always finite in any applications. Therefore, these iterative based truth discovery methods can also be extended to estimate continuous object truths by treating them as discrete values. For example, 5 sources claim the temperature of Auckland on a given day and the claims are $25.1, 25.1, 25.3, 25.3, 25.4$ respectively. A candidate set $\{25.1, 25.3, 25.4\}$ can be constructed first and the truth discovery algorithm assigns trustworthiness score to each truth candidate based on the source reliabilities.

In contrast, the PGM based truth discovery methods can only be used to estimate one type of object truths. In a PGM, the basic building blocks are random variables, and the random variables are expressed by probability distributions. If the object truth is

categorical, the truth discovery methods usually assume the random variable representing an object truth has a Categorical distribution. If the object truth is continuous, the truth discovery methods usually assume an object truth random variable has a Normal distribution.

The optimization based truth discovery methods can estimate heterogeneous object truth data type. The optimization based truth discovery methods aim at minimizing the overall error between sources' claims and object truths, it uses a distance function (see Section 2.1.2) to measure the error between a single claim and the corresponding object truth. If the object truth is categorical, the 0-1 loss function can be used as the distance function. If the object truth is continuous, the L^2 norm between the claim and object truth can be used as the distance function.

2.3.3 Multiple Object Truths

Many truth discovery methods have the single truth assumption such that each object has only one truth and the truth candidates are mutually exclusive. With this assumption, truth discovery outputs only one most trustworthy information for each object. While this assumption is valid in many applications, but it is not always true. For example, each book may have multiple authors, and each movie has more than one stars. In this case, the object truth is a set of values instead of a single value. The existing truth discovery methods, that has the single truth assumption, can be extended to estimate truths in these applications by treating each truth set as a single inseparable value, and output the most trustworthy set as the truth for each object. This extension works but the performance is usually not good because the information provided by many sources may be partially correct. In this case, a better strategy is to estimate the trustworthiness of each value in the set claimed by each source, and output the multiple trustworthy values as the truths for each object. To tackle this problem, B. Zhao et al. (2012) propose a

PGM based model that discover multiple truths for each object by considering source's false positive and false negative claims. Thus, it can discover multiple truths for each object simultaneously.

2.3.4 Object Difficulty

In (Ma et al., 2015; Whitehill et al., 2009; Galland et al., 2010), the truth discovery methods estimate object difficulty along with source reliabilities and object truths. Estimating object difficulty is useful when the sources react differently to objects with different difficulties, and this is widely used in crowdsourcing truth discovery literatures.

A common way of modeling object difficulty is to treat it as a random variable in a PGM. In (Whitehill et al., 2009), the authors propose a PGM based method that discovers categorical object truths. The difficulty d_o of object o is modeled as a random variable generated from a known prior distribution with positive support (e.g., Gamma distribution), the weight w^s of source s is modeled as a random variable generated from a known prior distribution with $(-\infty, +\infty)$ support (e.g., Normal Distribution). Given the object truth and source weight, the probability that the source's claim is equal to the object truth is

$$p(v_o^s = k | v_o^\star = k, w^s, d_o) = \frac{1}{1 + \exp(-w^s \times d_o)}$$

By the one-coin model presented in Section 2.2.1, the probability that the source's claim is not the object truth is

$$p(v_o^s = k' | v_o^\star = k, w^s, d_o) = \frac{1}{K-1}(1 - \frac{1}{1 + \exp(-w^s \times d_o)})$$

Then the object truths, source weights and object difficulties can be inferred by an inference algorithm.

2.3.5 Object Relation

In many cases, the objects are related to each other. Some truth discovery methods ((Pasternack & Roth, 2010; Yu et al., 2014; S. Wang et al., 2014; Y. Li et al., 2015; Meng et al., 2015; S. Wang et al., 2015) use the object relations to improve truth discovery performance. For example, if we know that A is the father of B and B is the father of C, then we can reason that A is the grandfather or ancestor of C. If such information is available beforehand, it would be more accurate to verify the trustworthiness of some observations. In (Pasternack & Roth, 2010), the authors translates such prior knowledge into propositional logic and use it in the truth discovery process. Ouyang et al. (2015) propose a method that can incorporate spatial correlation among objects in the truth discovery step. Object temporal correlation is studied in (S. Wang et al., 2014; Y. Li et al., 2015). It is found that (Meng et al., 2015) using object correlation can greatly improve the effectiveness if some objects are claim by few sources, which is common in many real-world applications.

2.3.6 Streaming Data

Most of the existing truth discovery methods are batch algorithms designed for static data in which time dimension is not involved. The batch algorithms are usually conducted in an iterative manner which update source weights and object truths alternatively and iteratively. In many real-world applications which data arrives sequentially from data streams, it is infeasible to use the batch algorithms to infer the object truths as the batch algorithms are computationally expensive. To tackle this challenge, some truth discovery methods are developed for truth finding over data streams. D. Wang et al. (2013) propose an EM truth discovery algorithm for processing streaming data in which the algorithm only scans the data once. To avoid the iterative updates of source weights and object truths, Y. Li et al. (2015) turns the optimization based truth discovery

framework to a probabilistic model, and the inference algorithm can update the source weights and object truths efficiently over data streams. Based on the solution derived from the optimization based truth discovery methods Y.Li, Li et al. (2016) propose an incremental algorithm that updates source weights and object truths exactly only once at each time-stamp.

2.3.7 Partially Observed Object Truth

Almost all the existing truth discovery methods are developed as unsupervised learning algorithms in which both source weights and object truths are not known as inputs. Although it is very expensive to obtain the ground truths for all the objects, it is sometimes practical to get a very small set of ground truths, i.e., the object truths are partially observed. In (Yin & Tan, 2011), the authors propose a semi-supervised truth discovery method called SSTF. SSTF can use a small set of ground truths to guide the estimation of source weights. It treats the truth candidates of each object and source weights as nodes in a graph, and uses graph propagation technique to propagate the trustworthiness of the ground truths to the truth candidates of the objects whose ground truths are unknown. Finally, SSTF selects the single most trustworthy truth candidate as the truth for each object.

2.4 Performance Metrics

In this section, the commonly used truth discovery performance metrics are reviewed. The performance of truth discovery is usually evaluated from two perspectives: *effectiveness* (also known as accuracy) and *efficiency*.

2.4.1 Effectiveness Metrics

The effectiveness of truth discovery is evaluated differently according to the object truth data type.

Categorical Object Truth

- **Error Rate:** The percentage of the object truths that are incorrectly estimated.
- **Accuracy:** The percentage of the object truths that are correctly estimated (1 – *error rate*).
- **Precision:** Precision measures the percentage of estimated object truths are indeed true. This is used in multiple truths truth discovery.
- **Recall:** Recall measures the percentage of object truths that are correctly estimated. This is used in multiple truths truth discovery.

Continuous Object Truth

- **Mean Absolute Error (MAE):** MAE measures L^1 norm distance between the estimated object truths and the ground truths. It penalizes more on the smaller errors.
- **Root of Mean Squared Error (RMSE):** RMSE measures L^2 norm distance between the estimated object truths and the ground truths. It penalizes more on the big errors.

2.4.2 Efficiency Metrics

The efficiency of truth discovery is usually evaluated by the *run time*, *convergence steps* and *time complexity*.

- **Run time:** Run time measures the actual elapsed time that a truth discovery method needs to use to conduct all the truth discovery steps.
- **Convergence Steps:** As many truth discovery algorithms have the iterative solution. The convergence step can be used the measure how many iterations that an algorithm needs to take to converge.
- **Time Complexity:** It measures the truth discovery algorithm time complexity in terms of Big-O notion. If the truth discovery algorithm has an iterative solution, the time complexity is usually measured within a single iteration.

2.5 Representative Truth Discovery Methods

In this section, it will present some representative truth discovery methods and briefly describes the specific problem they attempt to solve. Some of these truth discovery methods are also used as baselines to evaluate the truth discovery models proposed in the subsequent chapters.

2.5.1 Truth Discovery Methods for Static Data

The following methods are developed for static data. The algorithm flows of these methods are iterative based. They need to update source weights and object truths iteratively and iteratively until convergence. These methods are generally accurate, but computationally expensive if applied to data streams directly.

- **TruthFinder (Yin et al., 2008):** TruthFinder is an iterative based truth discovery method. It adopts Bayesian analysis to iteratively update source weights and object truths.

- **Accu (Dong et al., 2009a):** Accu also adopts Bayesian analysis to update source weights and object truths. Furthermore, it considers copying relations among sources.
- **3Estimates (Galland et al., 2010):** An iterative based truth discovery method that considers the difficulty of getting the truth for each object.
- **SSTF (Yin & Tan, 2011):** A semi-supervised truth discovery method that can use a small set of ground truths to guide the source weights estimation.
- **LTM (Zhao et al.,, 2011):** LTM is a PGM based truth discovery method aiming at discovering multiple truths for each object.
- **GTM (Zhao & Han, 2012):** GTM is a PGM based truth discovery method designed for estimating continuous object truths.
- **LCA (Pasternack & Roth, 2013):** LCA is a PGM based truth discovery method that considers various latent factors that may impact the truth discovery process.
- **CRH (Li et al., 2014):** CRH is an optimization based truth discovery method that can estimate heterogeneous object truths.
- **CATD (Li et al., 2014):** CATD is an optimization based method that considers the long-tail phenomenon. It uses confidence interval to penalize the weights of sources which claim few objects.
- **DyOP (Y. Li et al., 2015):** DyOP is an optimization based truth discovery method that aims at estimating continuous object truths at different timestamps.
- **OTD (Yao et al., 2018):** OTD is an optimization based truth discovery method develop for estimating truths on time series. It uses AutoRegressive Integrated Moving Average (ARIMA) to learn the trends and patterns from the observed time series, and then uses the trends and patterns to assist truth estimation.

2.5.2 Truth Discovery Methods for Streaming Data

The following methods are developed for streaming data. Different from the above truth discovery methods having iterative based algorithm flows, the truth discovery methods listed below updates source weights and object truths exactly once at each time-stamp over the data stream. These methods can estimate object truths efficiently when data arrives sequentially from data streams but the accuracy is usually lower than the truth discovery methods designed for static data.

- **iCRH (Y. Li, Li et al., 2016):** iCRH is the incremental version of CRH. It is developed for estimating heterogeneous object truths efficiently in a dynamic environment.

- **DynaTD (Y. Li et al., 2015):** An incremental truth discovery method that estimates continuous object truths over data streams without re-visiting the historical data.

- **DynaTD+s (Y. Li et al., 2015):** An extension of DynaTD that considers smoothing factor. The smoothing factor is used to enforce the object truths at adjacent timestamps to have similar values.

- **DynaTD+d (Y. Li et al., 2015):** An extension of DynaTD that considers decay factor. DynaTD uses all the errors accumulated from past to now to estimate the source weights at the current timestamp. The decay factor is used to penalize contributions of historical errors (errors accumulated in the past) to the source weight estimation at the current timestamp.

- **DynaTD+all (Y. Li et al., 2015):** An extension of DynaTD that considers both smoothing factor and decay factor.

- **ASRA (T. Li, Gu, Zhou, Ma & Yu, 2017):** A truth discovery that balances effectiveness and efficiency for estimating object truths over time.

2.5.3 Truth Discovery Methods for Crowdsourcing Applications

The following methods are developed for crowdsourcing applications. In the context of crowdsourcing, the source is known as worker and the object is known as task. Hence, the sources in crowdsourcing applications are human.

- **DS (Dawid & Skene, 1979):** A truth discovery method models worker reliability as confusion matrix.
- **LFC (Raykar et al., 2010):** Based on DS, it further incorporates task features in the truth discovery steps.
- **ZC (Demartini et al., 2012):** A truth discovery method models worker reliability as worker probability (a single number).
- **GLAD (Whitehill et al., 2009):** A truth discovery method accounting for task difficulty.
- **GEM (Kurve, Miller & Kesidis, 2014):** A truth discovery method accounting for both task difficulty and worker intention.

2.5.4 Naive Methods

The following two methods estimate object truths without considering source reliabilities.

- **Majority Voting (MV):** MV selects the truth candidate with the most votes as the object truth. If two truth candidates tie, it selects a random one. MV can be applied when object truths are categorical.

- **Mean:** Mean outputs the mean of truth candidates as the object truth. Mean can be applied when object truths are continuous.

2.6 Summary

In this chapter, I have given a detailed review of truth discovery methods from four perspectives: general frameworks, aspects of sources, aspects of objects and the performance metrics used to evaluate truth discovery methods. I have also selected and presented a list of truth discovery methods that are representative in the truth discovery literature. Some of these methods will also be used as baselines to evaluate the truth discovery models developed in the subsequent chapters.

To conclude this chapter, I will list the identified research gaps by analyzing the pros and cons of the existing works.

- Most existing truth discovery methods cannot efficiently combine object correlations into the truth discovery steps in a dynamic environment where data arrives sequentially from data streams.

- Existing truth discovery methods designed for static data can achieve high accuracy but are computationally expensive to be applied to stream data. Existing truth discovery methods designed for streaming data can achieve high efficiency but sacrifice the accuracy. A method is needed to balance the accuracy of truth discovery over data streams and still guarantees the efficiency of truth finding.

- It lacks a semi-supervised truth discovery method that is specifically designed for continuous data.

- The human factors in crowdsourcing applications are not well exploited in most existing truth discovery methods.

In the subsequent chapters, I will discuss the identified research gaps in details and presents the solutions to fill the research gaps in truth discovery.

Chapter 3

A Probabilistic Model for Truth Discovery with Object Correlations

3.1 Overview

There are some existing truths discovery methods that consider object correlations in the truth discovery steps (reviewed in Section 2.3.5). However, they are batch methods in natural, and are very expensive to run in dynamic environments where data arrives sequentially and accumulates over time. Some efficient truth discovery methods (reviewed in Section 2.3.6) are developed for processing data in a dynamic environment, but the object correlations information is not considered. Object correlation is existed in many real-world applications and it has important research value to truth discovery problems. If object correlation can be incorporated in the truth discovery steps, the accuracy of truth discovery can be improved. This chapter presents a novel truth discovery model that uses object correlation in the truth discovery process. I formulate the truth discovery task as a probabilistic inference problem. The developed probabilistic model considers not only source reliability but also object correlations to infer object truths. Furthermore, I extend the probabilistic model and develop an incremental truth

discovery method to process data efficiently in a dynamic environment and use temporal correlation to infer object truths. The contributions of this chapter are summarized as following:

In this chapter, two truth discovery models, PTDCorr and iPTDCorr, are developed. PTDCorr is a PGM based truth discovery model that capturing object correlations in the truth discovery steps. PTDCorr is a batch algorithm that can be used to estimate object truths from static data. In order to efficiently process streaming data in a dynamic environment, an incremental truth discovery model, iPTDCorr, is developed based on PTDCorr.

- I develop a chain graph based framework, Probabilistic Truth Discovery with Object Correlations (PTDCorr), in which source reliabilities, sources' claims and object truths are modeled as random variables.
- An optimization-based inference solution is developed to infer object truths in the chain graph model.
- Based on PTDCorr, I develop an incremental truth discovery algorithm, iPTDCorr, which works efficiently in dynamic environments. iPTDCorr is able to incorporate time-invariant correlations between different objects as well as temporal correlations for the same object to effectively infer object truths. Furthermore, iPTDCorr infers object truths by processing all data only once without re-processing the historical data.
- I conduct experiments on three datasets to evaluate the performance of the developed methods. Experimental results show that the developed methods outperform the existing truth discovery methods in inferring object truths for correlated objects.

The rest of the chapter is organized as follows. In Section 3.2, the review of related

works is presented. Section 3.3 introduces the key definitions and formally defines the problem of truth discovery with object correlations. In Section 3.4, I describe the PTDCorr model and an optimization based solution for truth inference. In Section 3.5, theoretical analysis of PTDCorr is presented. Section 3.6 describes the incremental iPTDCorr algorithm that is able to infer object truths by using object correlations in a dynamic environment. Experimental results are demonstrated in Section 3.7. Finally, this chapter is concluded in Section 3.8.

3.2 Related Work

There are some works that study the object correlations in truth discovery. The authors (D. Wang, Abdelzaher, Kaplan, Ganti et al., 2013; S. Wang et al., 2015) developed probabilistic models to solve the truth discovery problem in the context of social sensing. However, the models are limited to binary data and cannot be generalized to estimate real valued object truths. Meng et al. (2015) developed an optimization-based truth framework, and models object correlations as a regularization term in an objective function. The method is able to aggregate continuous object truths with correlation. However, the method is a batch algorithm and it cannot update object truths dynamically. In a recent work, (Liu, Liu, Duan, Hu & Wei, 2017), the authors developed a source-object network model to resolve conflicts among linked data. The linked data is described by RDF, and the relationship of the linked data is expressed as RDF triples. Correlations exist among RDF triples, and each RDF triple is treated as an object in this work. If one object has a high probability of being trustworthy, then its correlated objects also have high probabilities of being trustworthy. However, in this chapter's setting, the object correlations are used to reinforce the correlated object truths. This is able to correct the inferred object truths if these objects are claimed by few or unreliable sources.

Truth discovery in a dynamic environment has also received a lot of attention over the years. In (Pal, Rastogi, Machanavajjhala & Bohannon, 2012), the problem is modeled by a hidden semi-Markovian process that aims to solve truth discovery with missing updates and lagged claims. In (D. Wang, Abdelzaher, Kaplan & Aggarwal, 2013), the authors used Fisher information to recursively update the estimation parameters of truth discovery in a social sensing environment where data changes dynamically over time. In (Y. Li et al., 2015), an incremental truth discovery method was proposed to find truths over time. It also considered the smoothing data change and source weight evolution. Based on the work in (Y. Li et al., 2015), authors in (T. Li et al., 2017) developed a method which is able to trade off the accuracy and efficiency of the truth discovery model flexibly over time by tuning the parameters. Li et al. (2016) extend the CRH framework in (Li et al., 2015) to make it able to process data incrementally. Both (Y. Li et al., 2015) and (Q. Li, Li, Gao, Zhao et al., 2014) explored an incremental approach to estimate object truths over data streams. However, these methods assume that the source weights converge over time instead of converging at each timestamp. Thus, these methods can achieve optimal efficiency but sacrifice much accuracy. Among all the above truth discovery methods that are capable of working in a data stream, none of them captures object correlation and uses it to improve the performance.

3.3 Problem Definition

This section introduces the key definitions in the proposed truth discovery models.

Definition 3.1. ***Object, source and claim****: An object, j, is a thing or an event which is associated with a numerical value that is interested in. The set of all objects is denoted as J where $j \in J$. A source, i, is an information provider which can provide information on an object. The set of all sources is denoted as I where $i \in I$. A claim, x_{ij}, is a value of object j provided by source i. X is used to denote the set of all claims, X_j denotes*

the set of claims about object j and X_i denotes the set of claims provided by source i.

Definition 3.2. ***Object truth and inferred object truth****: An object truth, denoted as z_j, is the factual or ground truth of j, and it is unknown a priori in truth discovery models. An inferred object truth, denoted as $\hat{z}_j$, is regarded as an estimated truth of object o computed by truth discovery models. Z denotes the set of all object truths and $\hat{Z}$ denotes the set of all the inferred object truths.*

Definition 3.3. ***Source weight****: A source weight, a_i, is a source's true reliability degree and it is modeled as a positive real number. The larger the source weight, the more reliable the source is. The estimated source weight, $\hat{a}i$, is the source weight estimated by truth discovery models. A is used to denote the set of all source weights and $\hat{A}$ denotes the set of all estimated source weights.*

Definition 3.4. ***Object correlation****: Object correlation is a relationship between two objects, j and j'. It can be measured by a correlation coefficient $c(j, j')$ where $c(j, j') \in [0, 1]$. If $c(j, j') = 1$, that means that j and j' have a strong correlation and the truths of j and j' could be very close. Conversely, j and j' have no correlation relationship if $c(j, j')$ is 0. $\mathcal{C}$ is used to denote the set of all objects' positive correlation coefficients where $\mathcal{C} = \{c(j, j') | j \in J, j' \in J, j \neq j', c(j, j') > 0\}$.*

Object correlation, as a priori knowledge, can be observed and pre-defined using correlation coefficient in many applications. For example, it is common to use Gaussian kernel (Equation (3.1)) to define two objects' spatial correlation coefficient. In Equation (3.1), two objects are considered correlated if their spatial distance $d(j, j')$ is within a threshold δ, where σ determines how fast $c(j, j')$ is approaching 0 as the distance becomes larger.

$$c(j, j') = \begin{cases} \exp(-\frac{d^2(j,j')}{\sigma^2}), & \text{if } d(j, j') \leq \delta; \\ 0, & \text{otherwise}. \end{cases} \tag{3.1}$$

Notation	Description
J	set of all the objects
I	set of the sources
X_i	set of all claims from source i
z_j	truth of object j
Z	set of all truths of objects
$\hat{z}_j$	inferred truth of object j
$\hat{Z}$	set of all the inferred truths
a_i	source weight of source i
A	set of all sources weights
$\hat{a}_i$	inferred source weight of i
$\hat{W}$	set of all inferred source weights
$\mu_{j,x}$	mean of claims of object j
$\sigma^2_{j,x}$	variance of claims of object j
x_{ij}	claims of object j from source i
X_j	set of all claims of object j
X_i	set of all claims of objects from source i
X	set of all claims of all the objects
$c(j, j')$	correlation coefficient between j and j'
c^t	temporal correlation coefficient
$\mathcal{C}$	set of all objects' positive correlation coefficients
λ_j	balancing factors of object j
θ	weighting factor
E	set of undirected edges connecting correlated objects
E_j	a subset of E that involves j in the edges

Table 3.1: Notations in Chapter 3

In situations where data arrives sequentially, a superscript t is used to denote a timestamp of claims, object truths and source weights. For example, z^t_{ij} denotes the claim of object j from source i at time t, a^t_i denotes the source weight of source i at time t and A^t denotes the set of all source weights at time t.

The notations used in this paper are summarized in Table 3.1. The superscript timestamp t is omitted in the notations. Given the notations, the truth discovery problem that is studied in this chapter is defined as follows:

Truth Discovery with Object Correlations: Given a set of objects J and their corresponding object coefficients $\mathcal{C}$, the claims X of J provided by a set of sources I

can be observed. The truth discovery with object correlation aims at computing a set of inferred object truths $\hat{Z}$ for each object. In a dynamic environment where data arrives sequentially at different timestamps, the inferred object truths at timestamp t need to be computed without re-visiting the claims X arrived before t.

3.4 Probabilistic Truth Discovery with Object Correlations Model

In this section, I introduce the framework of the Probabilistic Truth Discovery with Object Correlations model (PTDCorr). Secondly, an optimization-based approach derived from the probabilistic model is proposed to infer object truths. This is followed by some theoretical analysis in the next section.

3.4.1 The PTDCorr Framework

As discussed, there are three major elements that may contribute to the computation of inferred truths, namely, object truths, source weights and claims. Assuming there are n objects of J and m sources of, there are at most mn claims in X where $X = \{x_{ij} | i \in I, j \in J\}$. Figure 3.1 shows the chain graph model that can characterize the generative process of the proposed methods. Source weight a_i, object truth z_j and claim of an object x_{ij} are modeled as random variables, and represented as the nodes in the chain graph. Shaded nodes are observed variables and unshaded nodes are hidden variables. As the dependencies of random variables are modeled in the chain graph, the terms *random variable* and *node* interchangeably in this chapter.

The chain graph in Figure 3.1 is a *partially directed acyclic graph* (PDAG). The nodes can be disjointly partitioned into several *chain components* where each chain component is an induced sub-graph of the chain graph in Figure 3.1. The nodes that

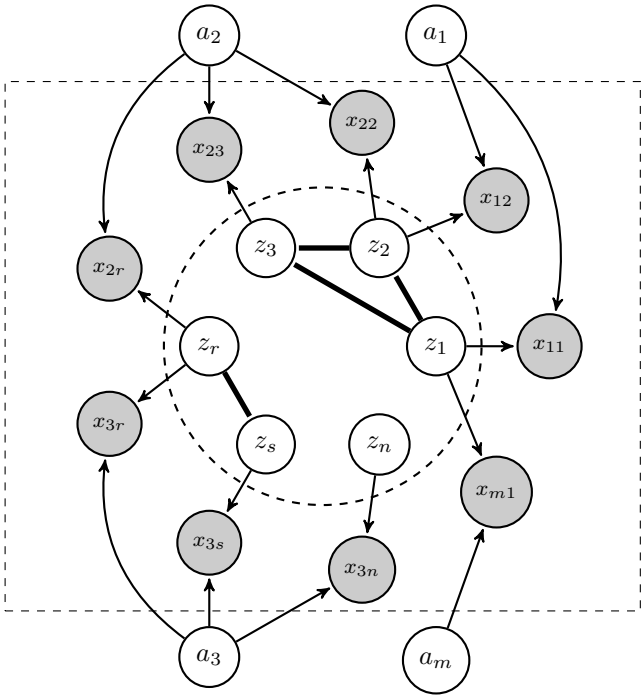


Figure 3.1: Chain Graph Model

are connected by undirected edges belong to the same chain component. The nodes in different chain components are connected by directed edges (Koller & Friedman, 2009). Based on the types of the random variables that the nodes represent, the chain components in Figure 3.1 can be classified into three layers: the inner layer Z enclosed in dotted circle, the outer layer A outside the dotted rectangle, and the middle layer X enclosed in between dotted circle and rectangle.

The inner layer Z is an undirected graph $\mathcal{G}_Z = \{Z, E\}$ representing the relationship of object truths where E is a set of undirected edges that connect the correlated nodes *s.t.* $E = \{(j, j')|c(j, j') \in \mathcal{C}\}$, two nodes are connected by an undirected edge if they are correlated. Assume Z can be partitioned into K chain components Z_k where $\bigcup_{k \in K} Z_k = Z$. The induced sub-graph $H_{Z_k} = \{Z_k, E_k\}$ is undirected and connected where $\bigcup_{k \in K} E_k = E$, the nodes within a sub-graph H_{Z_k} are correlated with at least one nodes and can reach other nodes within the chain component by at least one path. There are some chain components which have only one node (*e.g.* z_n in Figure 3.1). This indicates that the

node is uncorrelated with the rest of the nodes in Z.

Each chain component Z_k in Z is associated with a Markov Random Field and can be factorized as:

$$p(Z_k) = \frac{1}{P_{Z_k}} \prod_{z_j \in Z_k} \psi_j(z_j) \prod_{(j,j') \in E_k} \psi_{j,j'}(z_j, z_{j'}),$$

where P_{Z_k} is a partition function, $\psi_j(z_j)$ is a node potential which provides the prior probability for z_j; $\psi_{j,j'}(z_j, z_{j'})$ is an edge potential which reinforces the distributions of two correlated object truths z_j and $z_{j'}$. Therefore, the joint probability over all chain components in the inner layer Z can be factorized as:

$$\begin{aligned} p(Z) &= \prod_{k \in K} p(Z_k) \\ &= \prod_{k \in K} \Big(\frac{1}{P_{Z_k}} \prod_{z_j \in Z} \psi_j(z_j) \prod_{(j,j') \in E_k} \psi_{j,j'}(z_j, z_{j'}) \Big) \\ &= \frac{1}{\prod_{k \in K} P_{Z_k}} \prod_{j \in J} \psi_j(z_j) \prod_{(j,j') \in E} \psi_{j,j'}(z_j, z_{j'}) \end{aligned} \tag{3.2}$$

In the outer layer A, sources are assumed to be independent, *i.e.* sources make claims independently. Each random variable a_i forms a chain component, which is associated with a Markov Random Field with only one node and can be factorized as:

$$p(a_i) = \frac{1}{P_{a_i}} \psi_i(a_i)$$

where P_{a_i} is a partition function, $\psi_i(a_i)$ is a node potential which provides the prior probability for a_i. Therefore, the joint probability over all chain components in the

outer layer A can be factorized as:

$$\begin{aligned} p(A) &= \prod_{a_i \in A} \frac{1}{P_{a_i}} \psi_i(a_i) \\ &= \frac{1}{\prod_{a_i \in A} P_{a_i}} \prod_{i \in I} \psi_i(a_i) \end{aligned} \tag{3.3}$$

In the middle layer X, each chain component contains only one node x_{ij}. As a claim is dependent on both its object truth and the source which claims it. Node x_{ij} is a child of the corresponding object truth z_j and source weight a_i in the chain graph model. Each chain component in X is associated with a conditional probability and can be factorized as:

$$p(x_{ij}|z_j, a_i) = \frac{1}{P_{x_{ij}}} \psi_x(x_{ij}, z_j, a_i)$$

where $P_{x_{ij}}$ is a partition function, $\psi_x(x_{ij}, z_j, a_i)$ is a node potential which provides the conditional probability for x_{ij}. The joint probability over all chain components over X can be factorized as:

$$\begin{aligned} p(X|A, Z) &= \prod_{x_{ij} \in X} p(x_{ij}|z_j, a_i) \\ &= \prod_{j \in J} \prod_{x_{ij} \in X_j} \frac{1}{Z_{x_{ij}}} \psi_x(x_{ij}, z_j, a_i) \end{aligned} \tag{3.4}$$

where X_j are the claims about object j.

The partition functions in Equations (3.2), (3.3) and (3.4) are used to renormalize the potential functions. In the proposed method, they are all set to 1 because the probability density functions are used to install potential functions. These functions will be introduced in Section 3.4.3.

Given the structure of the chain graph and the chain components, the posterior

distribution $p(Z, A|X)$ is given in Equation (3.5):

$$\begin{aligned} p(Z, A|X) &\propto p(Z, A, X) \\ &\propto \prod_{j \in J} \psi_j(z_j) \times \prod_{(j,j') \in E} \psi_{j,j'}(z_j, z_{j'}) \times \\ &\quad \prod_{i \in I} \psi_i(a_i) \times \prod_{j \in J} \prod_{x_{ij} \in X_j} \psi_x(x_{ij}, z_j, a_i) \end{aligned} \tag{3.5}$$

The objective of the proposed truth discovery framework is to find the optimal inferred truth $\hat{z}_j$ for each object and source weight $\hat{a}_i$ for each source to maximize the posterior distribution shown in Equation (3.5).

3.4.2 Design Philosophy

The chain graph model shown in Figure 3.1 reflects the design philosophy of solving the truth discovery problem with object correlations from the following aspects. Reliable sources are able to **propagate** their influences over some objects even if they do not provide information on those objects. Assuming that there are no undirected edges in G_Z, sources would only influence the objects which are claimed by the sources. Thus, the object truths are only dependent on their claims and sources which provide the claims. As the random variables of correlated object truths are connected, an object truth is further directly influenced by its correlated object truths and the sources which provide claims on its correlated objects. For example, assume that i_1 and i_n as represented by a_1 and a_m, are reliable sources. i_2, represented by a_2, is not reliable in Figure 3.1. In this case, the inferred object truth $\hat{z}_1$ is trustworthy because it is claimed by two reliable sources i_1 and i_m. In contrast, it is hard to correctly infer the object truth of j_3 because j_3 is only claimed by one unreliable source i_2, the inferred truth $\hat{z}_3$ solely based on the claim x_{23} may not be accurate. However, j_1 and j_3 are correlated, and $\hat{z_1}$ and $\hat{z}_3$ could be similar. In the proposed probabilistic model, $\hat{z}_1$ is able to reinforce $\hat{z}_3$

and adjust $\hat{z}_3$ to a value closer to $\hat{z}_1$. it can be seen that a_1 and a_m indirectly flow their high reliability degrees to influence $\hat{z}_3$ by the correlation relationship encoded in the chain graph model. Thus, the inferred truth $\hat{z}_3$ is influenced directly by the correlation relationship between $\hat{z}_3$ and $\hat{z}_1$.

3.4.3 Potential Functions

In this section, the potential functions defined in the PTDCorr model are introduced.

Node Potentials

Object truth node potential $\psi_j(z_j)$. For each object, its truth is among the claims it received. It is more likely to be at the interval where the majority of claims are distributed. Hence, the object truth z_j is modeled as a real number and it is generated from a Normal distribution. The node potential function $\psi_j(z_j)$ can be represented as:

$$\begin{aligned} \psi_j(z_j) &\sim \mathcal{N}(\mu_{j,x}, \sigma^2_{j,x}) \\ &\propto (\sigma^2_{j,x})^{-\frac{1}{2}} \exp(-\frac{(z_j - \mu_{j,x})^2}{2\sigma^2_{j,x}}) \end{aligned} \tag{3.6}$$

where $\mu_{j,x}$ and $\sigma^2_{j,x}$ are the mean and variance of object j's claims.

Source weight node potential ψ_i. The source weight a_i is a positive real number and it can be modeled by a Gamma distribution. The node potential $\psi_i(a_i)$ function can be represented as:

$$\begin{aligned} \psi_i(a_i) &\sim Gamma(\alpha, \beta) \\ &\propto (a_i)^{\alpha-1} \exp(-\beta a_i) \end{aligned} \tag{3.7}$$

where $\alpha \geq 1$ and $\beta > 0$ and they are hyper parameters [1] that control the prior belief of a

[1] To make sure that the proposed inference algorithm in Section 3.4.5 converges, the shape parameter

source's reliability degree. If prior knowledge is available, α and β can be adjusted to change the source weight distribution; otherwise, source weights are initially treated equally and the same α and β are used for all source weight node potentials.

Claim node potential $\psi_x(x_{ij}, z_j, a_i)$. A claim is dependent on its object truth and the source which makes the claim. The claim from a reliable source is trustworthy and it is more likely to be the truth or closer to the truth. Guided by this intuition, the claim x_{ij} is modeled as a real number that is generated from a Normal distribution. The node potential $\psi_{x_{ij}}(x_{ij}, z_j, a_i)$ can be represented as:

$$\begin{aligned} \psi_x(x_{ij}, z_j, a_i) &\sim \mathcal{N}(z_j, \frac{1}{a_i}) \\ &\propto (a_i)^{\frac{1}{2}} \exp(-\frac{(x_{ij} - z_j)^2}{2\frac{1}{a_i}}) \end{aligned} \tag{3.8}$$

where z_j is the object truth of the claim and a_i is the weight of the source which provides the claim. Since the Normal distribution, which generates the object truth, and the Gamma distribution that generates the source weight, are both conjugate priors of the Normal distribution that generates the claim, this makes the proposed model a conjugate model and the inference performed (as described in Section 3.4.5) is traceable.

Edge Potentials

An undirected edge between two nodes z_j and $z_{j'}$ is used to reinforce the generated truths by utilizing the knowledge of object correlations. This imposes a soft smoothness constraint over the chain graph, indicating that neighboring nodes connected by undirected edges should take similar values. For truth discovery in real-world applications, object truths are unknown *a priori*. However, the truths of two objects can be approximated if these two objects are correlated. Specifically, given two correlated objects j and

α of Gamma distribution is constrained to be greater or equal to 1 to make the p.d.f. of Gamma distribution finite. See Section 3.5 for further explanation.

j' where $c(j, j') > 0$, their object truths could be numerically similar, and the absolute difference between two correlated object truths $|z_{j'} - z_j|$ should be close to 0. Guided by this intuition, a Normal distribution is applied to represent the edge potential function given Equation (3.9):

$$\begin{aligned} \psi_{j,j'}(z_j, z_{j'}) = p(|z_{j'} - z_j|) &\sim \mathcal{N}(0, \frac{1}{\theta\lambda_j c(j,j')}) \\ &\propto \big(\theta\lambda_j c(j,j')\big)^{\frac{1}{2}} \exp(-\frac{\theta\lambda_j c(j,j')(z_{j'} - z_j)^2}{2}) \end{aligned} \tag{3.9}$$

where θ and λ_j are two positive hyper parameters that adjust the effect of correlated object truths to the inferred truth for inference. They will be further discussed in Section 3.4.5.

The distribution in Equation (3.9) models the probability of $|z_{j'} - z_j|$, and the variance $\theta\lambda_o c(j, j')$ controls its probability. If $\theta\lambda_j c(j, j')$ is big, the variance of the distribution is small. $\psi_{j,j'}(z_j, z_{j'})$ attains a high value only if $z_{j'}$ is very close to z_j. Thus, it puts a strong constraint on the correlated object truths and it reinforces them to be very similar. Conversely, if $\theta\lambda_j c(j, j')$ is small, the variance of the distribution is big. It relaxes the constraint and the correlated object truths do not have to be very close to each other.

3.4.4 Outlier Removal

The PTDCorr model is a probabilistic generative model, the object truth is modeled as a Gaussian random variable using the mean and variance of the object's claims as its parameters. However, the mean parameter of a Gaussian random variable could be shifted infinitely by outliers, which could have bad influences on the truth inference and source weight estimation. In this subsection, a two-stage outlier removal algorithm is developed.

The outlier removal algorithm is described in Algorithm 3.1. In the first stage (Lines

Algorithm 3.1: Outlier Removal

Input : Set of claims X, thresholds δ_1, δ_2 and δ_3
Output : Claims X in which outliers are removed

for $x_{ij} \in X$ **do**
  if $x_{ij} \geq \delta_1$ *or* $x_{ij} \leq \delta_2$ **then**
    $X_j = X_j - x_{ij}$
  end
end
for $j \in J$ **do**
  $has\_outlier = False$
  repeat
    $\tilde{x}_j$ = median(X_j)
    $\tilde{\sigma}_j$ = standard_deviation(X_j)
    for $x_{ij} \in X_j$ **do**
      if $\frac{|x_{ij} - \tilde{x}_j|}{\tilde{\sigma}_j} > \delta_3$ **then**
        $X_j = X_j - x_{ij}$
        $has\_outlier = True$
      end
    end
  until $has\_outlier = False$
end
return X

1 - 8), it removes the extreme claims based on the two given thresholds δ_1 and δ_2. δ_1 and δ_2 are application-specific. For example, in the application which reports gas prices in the US, the gas price cannot be over \$6 per gallon and the gas price must be positive. Thus, δ_1 = 6 and δ_2 = 0. The second stage (Lines 6 - 18) iteratively removes outliers until no new outlier is detected. For each object, it first computes the median and standard deviation of the object's claims (Lines 9 -10). Then it uses the median and standard deviation to compute the z-value for each claim and compares it with the threshold δ_3 (Line 12). The threshold δ_3 can be understood as the number of standard deviations. If the z-value of the claim is greater than the threshold δ_3, it is treated as an outlier and is removed from the set of claims. Note that the median is used to measure the central tendency of the claims of an object as the median is less sensitive to outliers than the mean.

Other more advanced outlier detection and removal techniques (Han, Pei & Kamber, 2011) can also be deployed here, but the z-value based outlier removal algorithm described in Algorithm 3.1 is simple and effective. The primary goal of the outlier removal step is to remove the extreme values that are bad for the inference. Thus, the truth discovery algorithm will estimate source weights and infer object truths.

3.4.5 Truth Inference

As discussed in Subsection 3.4.1, the objective of the proposed truth discovery framework is to find the optimal inferred truth $\hat{z}_j$ for each object and estimated source weight $\hat{a}_i$ for each source. *Maximum A Posteriori* (MAP) estimation can be performed to infer the optimal set of object truths $\hat{Z}$ where $\hat{z}_j \in \hat{Z}$ and optimal set of source weights $\hat{A}$ where $\hat{a}_i \in \hat{A}$. The MAP estimation aims at maximizing $p(Z, A|X)$ in Equation (3.5), it can be formulated as an energy minimization problem (Koller & Friedman, 2009) where the energy, denoted as a function f, corresponds to the negative log likelihood of

the posterior probability $p(Z, A|X)$ and is defined as:

$$f(Z, A) \propto -\ln p(Z, A, X)$$

By plugging in the potential functions:

$$\begin{aligned} f(Z, A) \propto & \sum_{j \in J} \Big(\frac{(z_j - \mu_{j,x})^2}{\sigma_{j,x}^2} \Big) + \theta \sum_{(j,j') \in E} \Big(\lambda_j c(j, j') (z_{j'} - z_j)^2 \Big) \\ & + 2 \sum_{i \in I} \Big((1 - \alpha) \ln a_i + \beta a_i \Big) + \sum_{j \in J} \sum_{x_{ij} \in X_j} \Big(a_i (x_{ij} - z_j)^2 - \ln a_i \Big) \end{aligned} \tag{3.10}$$

Minimizing the energy in Equation (3.10) can be viewed as an optimization task that seeks the optimal $\{\hat{a}_i\}$ and $\{\hat{z}_j\}$ that minimize f:

$$\hat{\mathcal{A}} = \arg\min_{A} \int f(Z, A) dZ$$

$$\hat{Z} = \arg\min_{Z} \int f(Z, A) dA$$

Block coordinate descent (Bertsekas, 1999), in which one set of variables is updated while fixing the other, can be adopted to solve the above optimization problem. The validation of this approach is proved in Section 3.5. By setting $\frac{df}{dz_j} = 0$ and $\frac{df}{da_i} = 0$, update rules of $\hat{a}_i$ and $\hat{z}_j$ for block coordinate descent can be derived in Equations (3.11) and (3.12).

$$\hat{a}_i = \frac{2(\alpha - 1) + |X_i|}{2\beta + \sum_{x_{ij} \in X_i} (x_{ij} - z_j)^2} \tag{3.11}$$

$$\hat{z}_j = \frac{\frac{\mu_{j,x}}{\sigma^2_{j,x}} + \theta\lambda_j \sum_{j' \in E_j} c(j,j') z_{j'} + \sum_{x_{ij} \in X_j} a_i x_{ij}}{\frac{1}{\sigma^2_{j,x}} + \theta\lambda_j \sum_{j' \in E_j} c(j,j') + \sum_{i \in I_j} a_i} \tag{3.12}$$

In Equation (3.11), X_i is the set of claims provided by source i, $|X_i|$ is the number of claims provided by source i, and $\sum_{x_{ij} \in X_i}(x_{ij} - z_j)^2$ is the total squared error the source makes on its claims. It can be seen that the source weight is inversely proportional to the error, it coincides with the design of the proposed method that sources are assigned high weights if they make less errors on the claims.

In Equation (3.12), X_j is the set of claims for object j, I_j is the set of sources that provide claims on object j, and E_j is the set of edges that connect correlated objects involving object j. It abuses the notation here by using the same symbol E_j to denote the set of objects that are correlated with object j. From this equation it can be seen that the inferred truth is determined by three parts: (1) mean of claims weighted by their variance, (2) correlated object truths weighted by correlation coefficients, and (3) claims weighted by source weights. θ and λ_j are used to adjust the contribution of correlated object truths in the inferred truth. The use of θ and λ_j are analyzed below.

Let $|E_j|$ denotes the number of correlated objects of object j and $|I_j|$ denotes the number of sources which provide information on object j. $|E_j|$ is independent of $|X_j|$. It is known that correlation coefficient c is bounded between 0 and 1, but a_i is an unbounded positive real number. Assume θ and λ_j have no effect on the inferred object truths, i.e. $\theta = 1$ and $\lambda_j = 1$. If any of the source weights $a_i \gg 1$, but $|E_j|$ is numerical close to $|I_j|$, then $\sum_{i \in I_j} a_i \gg \sum_{(j,j') \in E_j} c(j,j')$. This makes correlated object truths insignificant to the inferred truth. Similarly, if $|E_j| \gg |I_j|$ and each a_i is not highly greater than 1 which is the upper bound of c, it makes the weighted claims insignificant to the inferred truth.

To tackle this problem, for each object j, **balancing factor**, λ_j, is used to balance

between the weighted correlated object truths and weighted claims in the inferred truth. λ_j can be computed by the following Equation:

$$\lambda_j = \frac{\alpha}{\beta} \times \frac{|I_j|}{|E_j|} \tag{3.13}$$

In Equation (3.13), $\frac{\alpha}{\beta}$ is the average mean[2] of the Gamma distribution that is used to draw source weights, as it balances the correlation coefficients (bounded between 0 and 1) with source weights. $\frac{|I_j|}{|E_j|}$ balances the difference between the number of sources that provide information on object j and the number of j's correlated objects. The balancing factor λ_j is hard to estimate if it was set empirically for different objects. By Equation (3.13), λ_j can be computed for each object without any manual configuration.

By using the balancing factor λ_j, the effect of weighted correlated object truths can be balanced with the weighted claims. In some datasets, if the number of correlated objects per object is large or there are few reliable sources available, it can further increase the contribution of weighted object truths for inferred truth. A **weighting factor**, θ, is used to adjust the effect of correlated object truths in the inferred truth after it is balanced by λ_j. The value of θ can be set empirically in different datasets.

In this model, there are two sets of variables, Z and A, involved in the optimization problem. However, variables in Z are correlated, it is trivial if they are updated in the same block by using block coordinate descent. Instead, the variables in Z can be divided into independent blocks of variables (Meng et al., 2015) $\{Q_1, Q_2, \ldots, Q_l\} \subset Q = Z$ where the variables in one block are uncorrelated from each other. Hence, an object truth $\hat{z}_j$ is updated in one block while its correlated object truths are fixed in other blocks. The algorithm flow for the proposed truth discovery is summarized in Algorithm 3.2.

[2]Average mean of the Gamma distributions is $\frac{\sum_{i \epsilon I_j} \frac{\alpha}{\beta}}{|I_j|}$, since the source weights are initialized equally, it can be simplified into $\frac{\alpha}{\beta}$.

Algorithm 3.2: Truth Inference

Input : Set of claims X and set of objects' correlation coefficients $\mathcal{C}$
Output : $\hat{Z}$ and $\hat{A}$

Initialize the set of estimated source weights $\hat{A}$
Initialize empty set of inferred object truths $\hat{Z}$
Partition $\hat{Z}$ to independent blocks Q by using object correlations $\mathcal{C}$
$\hat{Z}$, $\hat{A} \leftarrow$ inference(Q,X,$\mathcal{C}$,$\hat{Z}$,$\hat{A}$)
return $\hat{Z}$,$\hat{A}$

Procedure `inference`(Q,X,$\mathcal{C}$,$\hat{Z}$,$\hat{A}$)
    repeat
        for $Q_k \in Q$ **do**
            for $z_j \in Q_k$ **do**
                Update $\hat{z}_j$ by Equation (3.12)
            end
        end
        for $a_i \in A$ **do**
            Update $\hat{a}_i$ by Equation (3.11)
        end
    until *Convergence condition is met*
    return $\hat{Z}$,$\hat{A}$

3.5 Theoretical Analysis

In this section, it will show that block coordinate descent is a valid method to minimize the energy defined in Equation (3.10). It will also analyze the time complexity of the proposed method.

Theorem 3.1. *Function f in Equation (3.10) converges to the global minimum when using block coordinate descent to iteratively update blocks of variables in $\{Q_1, Q_2, \ldots, Q_l, A\}$ by Equations (3.11) and (3.12). The solution of $\hat{A}$ and $\hat{Z}$ is a stationary point w.r.t. f.*

Proof. Let $\mathcal{Y}$ denotes the set of blocks of variables where $\mathcal{Y} = \{Q_1, Q_2, \ldots, Q_l, A\}$ and the size of $\mathcal{Y}$ is $l + 1$. Then the optimization problem can be refined as:

$$\text{minimize } f(y), \ \textit{s.t. } y \in \mathcal{Y}$$

According to (Bertsekas, 1999), let $\{y^u\}$ be the sequence generated by the the following rule:

$$y_i^{u+1} = \arg\min_{\xi \in \mathcal{Y}_v} f(y_1^{u+1}, \ldots, y_{v-1}^{u+1}, \xi, y_{v+1}^{u}, \ldots, y_{l+1}^{u}),$$
$$v = 1, \ldots, l + 1.$$

where u is the iterate index, then every limit point of $\{y^u\}$ is a stationary point and $f(\{y^u\})$ is the global minimum of f if f satisfies two conditions:

1. f is continuously differentiable over $\mathcal{Y}$.
2. For each $y_v \in \mathcal{Y}_v$,

$$f(y_1, y_2, \ldots, y_{v-1}, \xi, y_{v+1}, \ldots, y_{l+1})$$

viewed as a function of ξ while the other blocks of variables are fixed, attains a

unique minimum $\bar{\xi}$ over $\mathcal{Y}_v$, and is monotonically non-increasing in the interval from y_v to $\bar{\xi}$.

Then, it will show that f satisfies the above two conditions in the following two cases:

- Case 1: Update block A while fixing Q. In this case, as $\alpha \geq 1$, $f_Q(A)$ involves linear combination negative logarithm functions, linear functions *w.r.t.* a_i and constants. Both negative logarithm functions and linear functions are continuously differentiable over A, negative logarithm functions are strictly convex over A, and linear functions are affine and convex over A. Hence, $f_Q(A)$ is continuously differentiable and strictly convex over A. Thus, f satisfies the above two conditions when updating A.

- Case 2: Update block Q_o while fixing $\{\mathcal{Y} \setminus Q_o\}$ where $Q_o \in Q$. In this case, $f_{\{\mathcal{Y} \setminus Q_o\}}(Q_o)$ is a linear combination of quadratic functions which have the form $d(z_j - e)$ where d and e are constants and $d > 0$, and constants. Since the quadratic functions are continuously differentiable and strictly convex over Q_o, $f_{\{\mathcal{Y} \setminus Q_o\}}(Q_o)$ is continuously differentiable and strictly convex over Q_o. It concludes that f satisfies the above two conditions when updating Q_o.

Therefore, it is valid to use block coordinate descent to minimize energy function f. □

Time Complexity Analysis: For each iteration, in the source weight update step, each source needs to compute the error between its claims and the corresponding object truth. Since each source can provide up to n claims and there are m sources, the cost of source weight update is $O(mn)$. In the object truth update step, for each object, it receives up to m claims and has at most $n - 1$ correlated objects. The costs of computing the mean and variance of object's claims are both $O(mn)$. The cost of computing sum of correlated objects' truths is $O(mn)$ for all the objects, and the cost of computing weighted object's claims is $O(mn)$. Overall, one iteration requires $O(mn)$ time. As

there are at most mn claims, the time complexity of one iteration is linear *w.r.t.* the number of claims.

The existing work on the convergence rate of block coordinate descent assumes the objective function is Lipschitz continuous (Beck & Tetruashvili, 2013), but the objective function in Equation (3.10) does not satisfy Lipschitz continuity[3]. However, the performance of the proposed methods for many real-world applications is reasonable and practical as demonstrated in Section 3.7.2, and the algorithm converges very quickly as shown in Section 3.7.5.

3.6 Incremental Truth Inference

As presented in Section 3.4, the PTDCorr model runs in a batch mode. It needs to process the whole dataset all together to infer object truths and source weights. However, this method is infeasible in some applications where data arrives sequentially. In these scenarios, object truths, claims and source weights change over time, therefore, the inferred object truths and source weights could be different at different timestamps. Furthermore, data is accumulated over time. It would be very expensive to re-process all the historical data, as even a small amount of data is added at each timestamp. Instead, the truth discovery algorithm should compute the inferred truth on real time with a short response time. To tackle this problem, I extend the PTDCorr model and develop an incremental truth discovery model (iPTDCorr). In this section, it will first introduce the incremental source weight estimation method. Then it will describe how to infer object truths by capturing temporal correlation.

[3]This is because $a_i \in (0, +\infty)$, $\lim_{a_i \to 0} \frac{\partial f}{\partial a_i} = \infty$.

3.6.1 Incremental Source Weight Estimation

The data is assumed to come into an application in a sequential order. The idea of iPTDCorr is to retain the source weights at timestamp $t-1$ where t is the current timestamp, and uses the source weights computed at timestamp $t-1$ as the prior information to estimate the source weights at t. Thus, the posterior distribution of each source weight after timestamp $t-1$ can be computed by Equation (3.14).

$$\begin{aligned}
p(a_i^{t-1}|X^{1:t-1}, Z^{1:t-1}) &\propto p(a_i^{t-1})p(X^{1:t-1}|a_i^{t-1}, Z^{1:t-1}) \\
&\propto p(a_i^{t-1}) \prod_{k=1}^{t-1} p(X_i^k|a_i^{t-1}, Z_{1:t-1}) \\
&\propto (a_i^{t-1})^{\alpha-1} \times \exp(-\beta a_i^{t-1}) \times \\
&\quad \prod_{k=1}^{t-1} \prod_{x_{ij}^k \in X_i^k} (a_i^{t-1})^{\frac{1}{2}} \times \exp(-\frac{a_i^{t-1}(z_j^k - x_{ij}^k)^2}{2}) \\
&\propto (a_i^{t-1})^{\alpha-1+\frac{\sum_{k=1}^{t-1}|X_i^k|}{2}} \times \\
&\quad \exp(-a_i^{t-1}\frac{2\beta + \sum_{k=1}^{t-1}\sum_{x_{ij}^k \in X_i^k}(z_j^k - x_{ij}^k)^2}{2})
\end{aligned} \tag{3.14}$$

Equation (3.14) indicates that $p(a_i^{t-1}|X_i^{1:t-1}, Z^{1:t-1})$ follows a Gamma distribution:

$$Gamma(\alpha + \frac{\sum_{k=1}^{t-1}|X_i^k|}{2}, \beta + \frac{\sum_{k=1}^{t-1}\sum_{x_{ij}^k \in X_i^k}(z_j^k - x_{ij}^k)^2}{2})$$

Let b_i denotes $\sum_{k=1}^{t-1}|X_i^k|$, which is the historical total number of claims that a source i has claimed from timestamps 1 to $t-1$. Let d_i denotes $\sum_{k=1}^{t-1}\sum_{x_{ij}^k \in X_i^k}(z_j^k - x_{ij}^k)^2$, which represents the historical total errors that a source i has made through timestamps 1 to $t-1$. By using $p(a_i^{t-1}|X_i^{1:t-1}, Z^{1:t-1})$ as the prior distribution, the source weight of i at timestamp t can be computed as :

$$\hat{a}_i^t = \frac{2(\alpha-1) + b_i + |X_i^t|}{2\beta + d_i + \sum_{x_{ij}^t \in X_i^t}(z_j^t - x_{ij}^t)^2} \tag{3.15}$$

As time goes on, b_i and d_i may become very large and dominate the source weight computation at timestamp t, which makes $|X_i^t|$ and $\sum_{x_{ij}^t \in X_i^t}(z_j^t - x_{ij}^t)^2$ for the current timestamp t insignificant in Equation (3.15). In order to solve this problem, a decay factor ω can be used to exponentially shrink the effect of historical claim counts and errors to infer object truths at the current timestamp. Specifically,

$$b_i = \sum_{k=1}^{t-1} \omega^{t-k} |X_i^k| \tag{3.16}$$

$$d_i = \sum_{k=1}^{t-1} \omega^{t-k} \sum_{x_{ij}^k \in X_i^k} (z_j^k - x_{ij}^k)^2 \tag{3.17}$$

From Equations (3.16) and (3.17), it can seen that the more recent historical claim counts and errors weigh more when estimating the source weights. Once the truths at timestamp $t-1$ are inferred, b_i and d_i are updated and they are treated as constants when estimating source weights at timestamp t.

The incremental source weight update rule ensures that the historical claims do not need to be re-processed and stored and makes truth inference more efficient if data arrives sequentially.

3.6.2 Temporal Correlation

In the applications where data arrives sequentially, there usually exists temporal correlation among the truths of an object. For example, the hourly temperatures of a place usually evolve smoothly over time, so the temperatures in a short period of time are correlated and close. The temporal correlation can also be incorporated into the proposed model to infer object truths. Specifically, given an object j and two timestamps t and t', the states of object j at timestamps t and t' can be viewed as two pseudo objects j^t and $j^{t'}$. The temporal correlation between j^t and $j^{t'}$ can be measured by a temporal

correlation coefficient $c^t(j^t, j^{t'})$. If $c^t(j^t, j^{t'}) > 0$, j^t and $j^{t'}$ are temporally correlated, and the truths of object j at timestamps t and t' could be close.

By incorporating the temporal correlation, the truth of object j at timestamp t can be computed by Equation (3.18).

$$\hat{z}_j^t = \frac{\frac{\mu_{j,x}^t}{{\sigma_{j,x}^t}^2} + \theta\lambda_j^t\Big[\sum_{j'\in E_j} c(j,j')z_{j'}^t + \sum_{j^{t'}\in E_j^t} c^t(j^t,j^{t'})\hat{z}_j^{t'}\Big] + \sum_{x_{ij}^t\in X_j^t} a_i^t x_{ij}^t}{\frac{1}{{\sigma_{j,x}^t}^2} + \theta\lambda_j^t\Big[\sum_{j'\in E_j} c(j,j') + \sum_{j^{t'}\in E_j^t} c^t(j^t,j^{t'})\Big] + \sum_{i\in I_j^t} a_i^t} \tag{3.18}$$

In Equation (3.18), E_j is the set of objects that have a time-invariant correlation with object j, and E_j^t is the set of pseudo objects that have a temporal correlation with object j at timestamp t. $\hat{z}_j^{t'}$ is the inferred object truth of the pseudo object $j^{t'}$, it is also the truth of object j at timestamp t'. $\hat{z}_j^{t'}$ is treated as a constant instead of a random variable at timestamp t, because it was inferred at the previous timestamp t'. λ_j^t is the balancing factor for object j at timestamp t. It uses the same idea discussed in Section 3.4.5 to compute λ_j^t:

$$\lambda_j^t = \frac{\sum_{i\in I_j^t}(\frac{\alpha+b_i/2}{\beta+d_i/2})}{|I_j^t|} \times \frac{|I_j^t|}{|E_j| + |E_j^t|} \tag{3.19}$$

In Equation (3.19), $\frac{\alpha+b_i/2}{\beta+d_i/2}$ is the mean of the prior distribution for each source weight at timestamp t, $\frac{\sum_{i\in I_j^t}(\frac{\alpha+a_i/2}{\beta+d_i/2})}{|I_j^t|}$ is the average mean of the source weight's distributions for the sources that claim object j at timestamp t, and $|E_j| + |E_j^t|$ is the size of the correlated objects.

Dataset	Objects	Sources	Claims in total	timestamps	Avg. number of correlated object per object
Gas Price Dataset	3197	30	95910	N/A	7.2
Weather Dataset	42	5	37800	180	1.1
Synthetic Dataset	4000	30	2400000	20	99

Table 3.2: Datasets Statistics in Chapter 3 Experiments

3.7 Experiments

In this section, it experimentally compares the proposed methods with the state-of-the art truth discovery methods on both real datasets and synthetic datasets. All the experiments are conducted on a PC with an Intel i7 processor and 16 GB RAM.

3.7.1 Experiments Setup

In this section, it describes the datasets, baseline methods and performance metrics used to evaluate the proposed PTDCorr and iPTDCorr methods.

Datasets

- Gas Price Dataset: The regular gas prices of 3197 gas stations in the US from Gasbuddy[4] are collected for one day as the ground truth. The gas stations are in 30 major cities in the US. Gas prices are reported by various users. In this experiment, 30 users with various reliability degrees and their claims are generated. In this dataset, a gas price in a gas station is an object, and a source is a user who reports gas prices. The gas prices is analyzed on the interactive map on Gasbuddy and it is found that gas prices are similar if the distance between is within 5 kms. Hence, 5 kms is used as the threshold to determine if two stations' gas prices are correlated and choose Gaussian kernel to install correlation coefficient function c.

- Weather Dataset: The weather forecast data from five weather forecast providers

[4]www.gasbuddy.com

(Aeris[5], Apixu[6], Darksky[7], World Weather Online[8] and Wunderground[9]) for 42 different locations in New York city with 37800 claims are collected. The ground truths of the weather condition are also collected for evaluation. In this dataset, an object is a forecast temperature at a location which has different true values at different timestamps. A source is a website which provides weather forecast information. Two objects are considered to be correlated if the suburbs they belong to are next to each other. It uses a constant, 0.8, to install correlation coefficient function.

- Synthetic Dataset: This dataset contains 30 synthetically generated sources with different reliability degrees and 4,000 generated objects spanning 20 timestamps, hence, there are in total 2,400,000 claims in this dataset. The objects are divided into 40 clusters where objects are correlated if they are in the same cluster. The ground truths of objects at each timestamp within the same cluster and the ground truths of the same object at adjacent timestamps are numerically similar. For each source, Gaussian noise based on the source's reliability degree is added to each object's ground truth for each timestamp and is used as the source's claims. This dataset will be used to evaluate incremental truth discovery methods.

The statistics of the three datasets are summarized in Table 3.2. The last column shows the average number of correlated objects per object in the datasets.

Baseline Methods

PTDCorr and iPTDCorr models are compared with the state-of-the art truth discovery solutions, including the incremental truth discovery methods: DynaTD (Y. Li et al.,

[5]www.aerisweather.com
[6]www.apixu.com
[7]darksky.net/about/
[8]www.worldweatheronline.com
[9]www.wunderground.com

2015), DynaTD+all (Y. Li et al., 2015), and iCRH (Y. Li, Li et al., 2016); the static truth discovery methods: TDCorr (Meng et al., 2015), GTM (B. Zhao & Han, 2012) and CRH (Q. Li, Li, Gao, Zhao et al., 2014), iCRH (Y. Li, Li et al., 2016); and the naive methods which do not consider source reliability: Mean and Median. The descriptions of these methods can be seen in Section 2.4.

GTM is a batch method and it is not designed to work on datasets which involve temporal relations or in a dynamic environment. Hence the same object at different timestamps are treated as different objects when testing GTM on weather dataset and synthetic dataset.

Performance Metrics

The data in the datasets are continuous, the difference between inferred truth and ground truth can be measured by their numerical distance. Hence, MAE and RMSE (see Section 2.4 for description).

3.7.2 Performance Comparison

In this section, it presents the experiment results conducted on the three datasets. All the baseline methods are implemented and tuned with the parameters that result in the best performances.

Effectiveness

In many real-world scenarios, an object may not be reported by all the sources. Hence the experiments is conducted on different coverages of sources. The coverage is defined as the percentage of sources that make claims on the objects. For example, in the gas price dataset, there are 30 sources in total. If the coverage is 100%, it means that each object is reported by all the 30 sources. If the coverage is 60%, it means that each object

Gas Price Dataset										
Metric	MAE					RMSE				
Method \ Coverage	20%	40%	60%	80%	100%	20%	40%	60%	80%	100%
PTDCorr	**0.198**	**0.182**	**0.176**	**0.172**	**0.167**	**0.06**	**0.047**	**0.042**	**0.04**	**0.037**
TDCorr	0.226	0.221	0.22	0.219	0.195	0.073	0.066	0.065	0.063	0.06
GTM	0.409	0.301	0.26	0.233	0.214	0.263	0.142	0.104	0.086	0.072
CRH	0.43	0.329	0.289	0.267	0.251	0.290	0.169	0.131	0.11	0.096
Mean	0.469	0.377	0.336	0.291	0.273	0.344	0.223	0.175	0.13	0.113
Median	0.491	0.366	0.307	0.276	0.252	0.382	0.213	0.151	0.12	0.101

Table 3.3: Gas Price Dataset Experimental Result

is reported by 18 randomly selected sources. For the gas price and synthetic datasets, the experiments are conducted under the coverage of 20%, 40%, 60%, 80% and 100%. For the weather dataset, experiments are conducted under the coverage of 40%, 60%, 80% and 100%. It is trivial to test for the coverage under 40% for this dataset because each object is only claimed by 1 source if the coverage is below 40%.

The experiment results for the gas price dataset is summarized in Table 3.3. The baseline methods iCRH, DynaTD and DynaTD+all are designed to work on data having temporal relations. Hence, they are not tested with the gas price dataset. From the table it can be seen that the proposed method outperforms all the baseline methods under all the coverages. Among all the baseline methods, only TDCorr uses object correlation and it is a batch truth discovery method. According to the experiment results, PTDCorr improves the accuracy by more than 30% in comparison to GTM and CRH. When the coverage is low at 20%, PTDCorr performs 32% better than GRM and 67% better than CRH. The result is very encouraging because in many real-world applications, it is common that the coverage is normally low. The proposed method can provide a good solution for these scenarios. PTDCorr also outperforms TDCorr by at least 12% for all coverages. This demonstrates that the proposed method can better utilize object correlation in truth discovery.

Table 3.4 shows the experimental results of the weather dataset. The weather dataset contains weather forecasts over time provided by multiple sources but the

Weather Dataset								
Metric	MAE				RMSE			
Method \ Coverage	40%	60%	80%	100%	40%	60%	80%	100%
iPTDCorr	**1.159**	**1.042**	**0.959**	**0.902**	**2.402**	**1.798**	**1.473**	**1.355**
TDCorr	1.23	1.121	1.022	0.95	2.597	2.029	1.627	1.455
DynaTD	1.34	1.2	1.12	1.02	2.988	2.125	1.676	1.573
DynaTD+all	1.265	1.123	1.027	0.985	2.839	1.952	1.679	1.566
iCRH	1.331	1.158	1.035	0.989	2.849	1.998	1.705	1.68
GTM	1.336	1.19	1.056	0.991	2.858	2.029	1.797	1.688
Mean	1.647	1.463	1.359	1.287	3.776	2.597	2.432	2.325
Median	1.517	1.390	1.293	1.226	3.746	3.021	2.592	2.172

Table 3.4: Weather Dataset Experimental Result

Synthetic Dataset										
Metric	MAE					RMSE				
Method \ Coverage	20%	40%	60%	80%	100%	20%	40%	60%	80%	100%
iPTDCorr	**0.196**	**0.179**	**0.156**	**0.142**	**0.131**	**0.081**	**0.05**	**0.039**	**0.032**	**0.027**
TDCorr	0.247	0.223	0.199	0.18	0.164	0.097	0.079	0.062	0.051	0.042
DynaTD	0.379	0.268	0.225	0.192	0.172	0.223	0.115	0.076	0.06	0.049
DynaTD+all	0.345	0.257	0.22	0.19	0.168	0.188	0.105	0.073	0.056	0.045
iCRH	0.38	0.271	0.223	0.191	0.172	0.222	0.117	0.078	0.057	0.048
GTM	0.386	0.273	0.223	0.193	0.172	0.234	0.117	0.078	0.058	0.047
Mean	0.406	0.288	0.235	0.203	0.181	0.26	0.13	0.087	0.065	0.052
Median	0.443	0.328	0.273	0.238	0.213	0.311	0.169	0.117	0.089	0.072

Table 3.5: Synthetic Dataset

average number of correlated objects per object is very low. Therefore, the impact of object correlations on inferred truth may be small. Even in this situation, the proposed incremental model iPTDCorr outperforms all the baseline methods under all the coverages. DynaTD and DynaTD+all perform better than TDCorr when the coverage is high. However, when the coverage is 40%, TDCorr provides better results. This further demonstrates that object correlation has a positive impact on truth discovery. It becomes more critical in low coverage situations which are common for many real-world applications. However, TDCorr cannot update source weights and object truths incrementally for each timestamp, hence its performance is not as good as the proposed iPTDCorr.

The experimental results for the synthetic dataset is summarized in Table 3.5.

iPTDCorr has the lowest errors compared to the baseline methods under all coverages. Being different from the weather dataset, the average number of correlated objects per object is much larger. Hence, the evidence of effectiveness is prominent. When coverage is 100%, iPTDCorr outperforms TDCorr, DynaTD+all and iCRH by 20%, 22% and 23% respectively. Its error is nearly 24% lower than the batch method GTM. When the coverage is 20%, iPTDCorr remains a 21% advantage over TDCorr, and performs 43%, 48% and 49% better than DynaTD+all, iCRH and GTM respectively.

It can be observed from the experimental results that the performances of PTDCorr, iPTDCorr and TDCorr increase as the coverage decreases for all the three datasets. The reason is that when the coverage is low, an object is claimed by less sources. It increases the chances that objects are claimed by unreliable sources. By utilizing object correlation, the inferred object truths claimed by reliable sources can be propagated to those who are claimed by unreliable sources to improve the effectiveness of truth discovery. This also coincides with the design philosophy of the proposed method that reliable sources are able to propagate their influences to the objects even if they do not report information on those objects directly. Compared with TDCorr, the performance of PTDCorr and iPTDCorr are better because (1) the probabilistic model captures the dependencies among object truth, source weight, claim and object correlation in a more systematic way; (2) iPTDCorr is able to estimate the source weights incrementally for each timestamp in a dynamic environment.

Efficiency

In this section, I compare the efficiencies of the proposed methods. The data is preloaded into the memory and the independent blocks of variables are constructed based on the correlations among objects. For the weather dataset and the synthetic dataset, the arrival of data is modeled coming from a stream. For the weather dataset which has 180 timestamps, there are few claims for each timestamp and the running time for processing

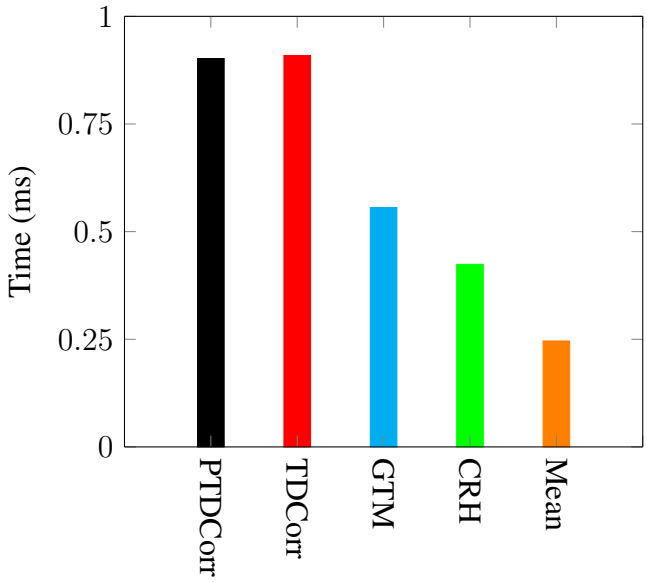


Figure 3.2: PTDCorr Running Time on Gas Price Dataset

each timestamp is very small. Hence, it reports the accumulated running time for every 9 timestamps for the weather dataset. The running times of PTDCorr/iPTDCorr and the state-of-art truth discovery methods are plotted in Figures 3.2, 3.3 and 3.4 respectively. For baseline methods, the running times of DynaTD and DynaTD+all are very close, and the running times of Mean and Median are also very close. Hence, it only shows the result of DynaTD and Mean for clear presentation. Among all the methods, Mean has the optimal efficiency because it does not estimate source reliabilities in comparison to the other truth discovery methods.

The efficiency experiment conducted on the gas price dataset is shown in Figure 3.2. The running times of PTDCorr and TDCorr are similar. However, these two methods are slower than GTM and CRH because PTDCorr and TDCorr use object correlations in their models. This results in extra computation when summing the weighted correlated object truths in the truth update step.

Figure 3.3 shows the results of the weather dataset. It can seen that the running times of TDCorr and GTM grow linearly over time and they have a similar running time.

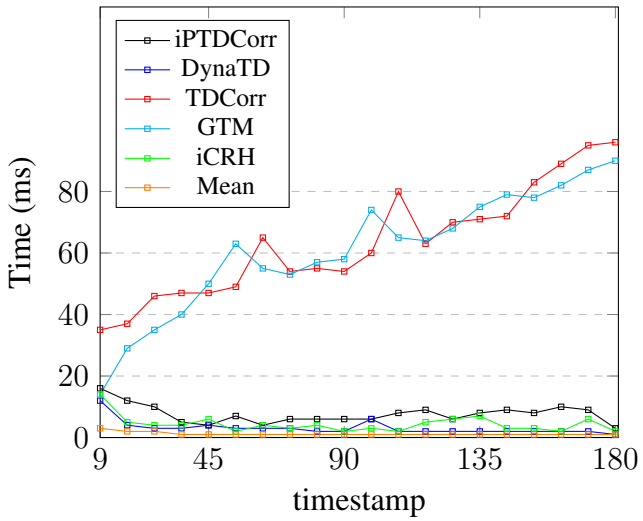


Figure 3.3: iPTDCorr Running Time on Weather Dataset

The reason is that these algorithms have to be re-executed over the whole dataset when new data arrives. On the other hand, for iPTDCorr, DynaTD and iCRH, the elapsed running times in each timestamp do not deviate greatly from each other and they are significantly faster than the two batch algorithms. This is because these three methods are able to find truths incrementally without re-visiting the historical data. Compared with DynaTD and iCRH, the running time of iPTDCorr is slightly slower because it needs to conduct iterative processes and compute the weighted correlated object truths to obtain accurate object truth.

The efficiency experiment conducted on the synthetic dataset is plotted in Figure 3.4. As the running times of DynaTD and iCRH are very similar, they overlap in the figure. Both iPTDCorr and TDCorr need to compute the sum of weighted correlated object truths to update the inferred truths. iPTDCorr is able to divide such computation to each timestamp, but TDCorr has to compute the sum of weighted correlated object truths for the whole dataset all at once. This also makes its running time slower than that of GTM. As more computation is required for iPTDCorr, it runs slower than DynaTD and iCRH, but it improves the effectiveness significantly by sacrificing some efficiency as demonstrated in Section 3.7.2.

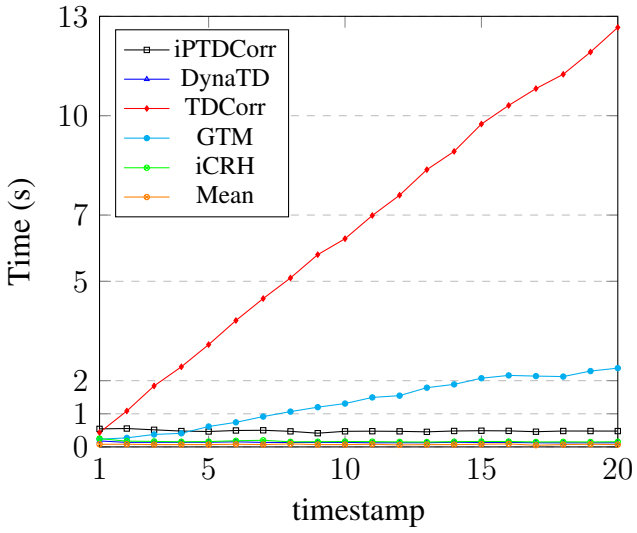


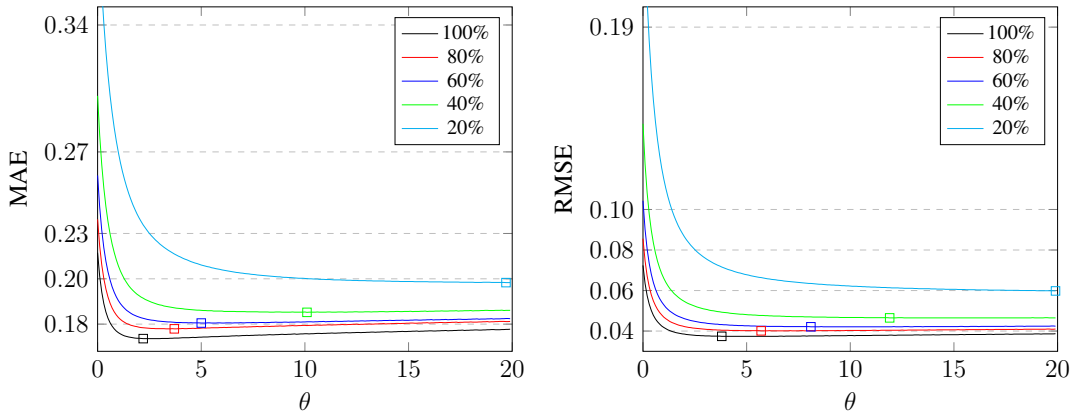
Figure 3.4: iPTDCorr Running Time on Synthetic Dataset

3.7.3 Influence of the Weighting Factor on the Errors

Weighting factor θ is introduced in Section 3.4 and is used to adjust the weight of correlated object truths that contributed to the inferred truth. Figure 3.5 shows the effect of different weighting factors on MAE and RMSE for the three datasets with different coverages. The optimal θ that leads to the smallest MAE and RMSE under each coverage is marked by a square.

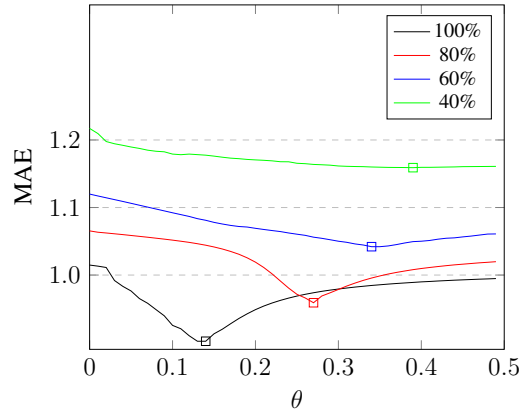
When $\theta = 0$, it corresponds to the case where correlated object truths do not contribute to inferred truths in the proposed method. For all the three datasets, the performance of PTDCorr and iPTDCorr increases as soon as θ begins to increase from 0, and this demonstrates that the developed methods are able to use object correlations information. Indeed, object correlations benefit the performance of truth discovery.

For the weather dataset, a relatively small θ leads to the best performance. The reason is that each object has few correlated objects, averagely 1.1, but each object is reported by 5 sources. If correlated object truths contribute too much to the inferred truth, it would make the weighted claims insignificant to the inferred truth. For the gas price dataset and synthetic dataset, as the average number of correlated objects

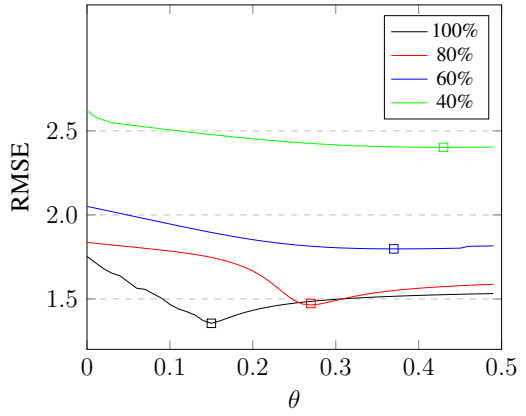


(a) Gas Price Dataset MAE

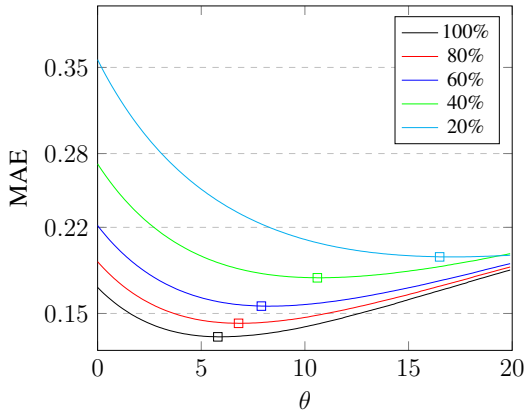
(b) Gas Price Dataset RMSE



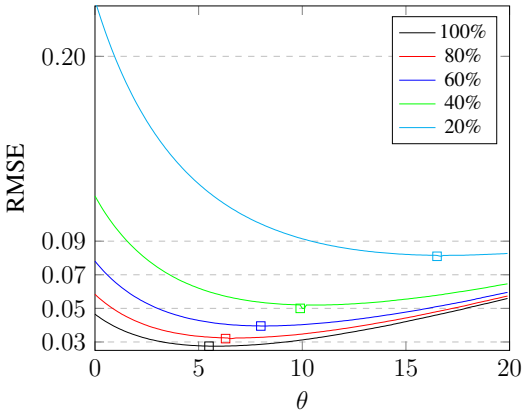
(c) Weather Dataset MAE



(d) Weather Dataset RMSE



(e) Synthetic Dataset MAE



(f) Synthetic Dataset RMSE

Figure 3.5: Effects of θ on MAE and $RMSE$

per object is big, a relatively big θ would benefit the performance. It is interesting to find that the errors of the gas price dataset do not increase significantly as θ passes the optimal value. The reason is that the gas prices of two correlated gas stations are very

close to each other, thus, a larger weight of correlated object truths in the inferred truth does not result in a big error.

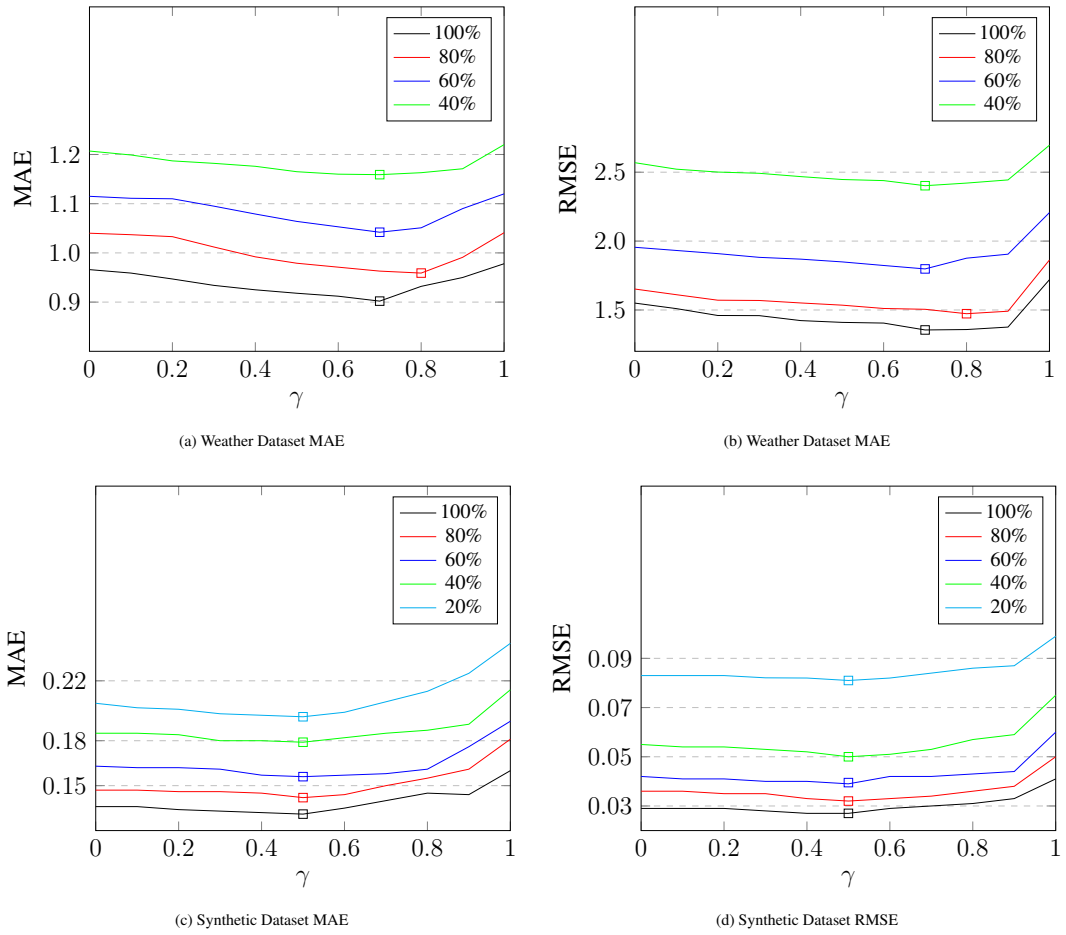
From Figure 3.5, it can be seen that the optimal value increases as the coverage decreases for all datasets. The reason is that when the coverage is low, each object is claimed by few or unreliable sources. Thus, increasing the contribution of correlated object truths to the inferred truth is able to benefit the performance of truth discovery in these scenarios.

Different θs lead to different optimal errors evaluated by MAE and RMSE, but the two best θs for MAE and RMSE are very close. For example, for the gas price dataset, when the coverage is 100%, the best θ that leads to optimal MAE is 2.2 while the best θ that leads to the optimal RMSE is 3.8. MAE and RMSE evaluate the errors from different perspectives whereas RMSE penalizes heavily on large errors. If the truth discovery application is more concerned with the large errors of inferred truths, one can choose to use the best θ that leads to the optimal RMSE to run the truth discovery algorithm.

3.7.4 Influence of the Decay Factor on the Errors

Decay factor γ is introduced in Section 3.6 and it is used to exponentially shrink the effect of historical claim counts and errors to infer object truths at the current timestamp for incremental truth inference. Figure 3.6 shows the effect of different decay factors on MAE and RMSE for the weather and synthetic datasets under different coverages. The optimal γ that leads to the smallest MAE and RMSE under each coverage is marked by a square.

For the weather dataset, after γ is decreased below the optimal value, the errors begin to increase slowly. This implies that the source weights in this dataset evolve smoothly. Indeed, using the historical claims counts and errors affects the source



(a) Weather Dataset MAE

(b) Weather Dataset MAE

(c) Synthetic Dataset MAE

(d) Synthetic Dataset RMSE

Figure 3.6: Effects of γ on MAE and $RMSE$

weights estimation. After γ is decreased below the optimal value for the synthetic dataset, the errors also increase, but the increment is very small. The reason is that it did not enforce the source weights to be changed smoothly over time when the synthetic dataset is generated. Therefore, the source weights in the synthetic dataset are less affected by the historical claims counts and errors.

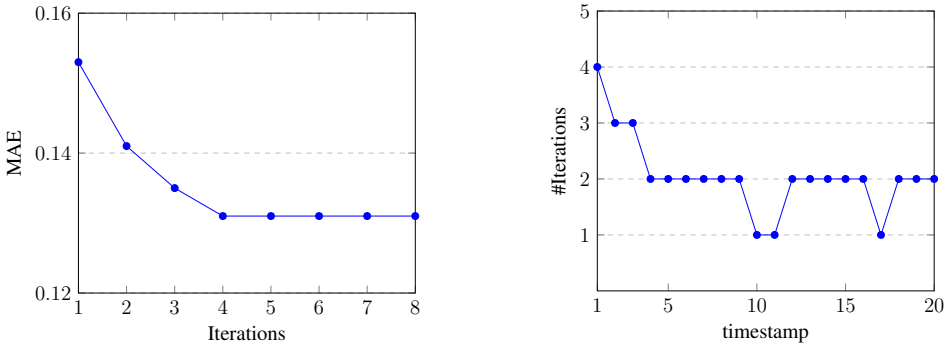
It can also be observed that both MAE and RMSE reduced significantly as soon as γ was decreased from 1 in both datasets. The reason is that when $\gamma = 1$, it corresponds to the case that it uses all the historical claim counts and errors to compute the source weight in Equation (3.15). As time goes on, the accumulated historical claims counts

and errors become very large and dominate the source weight computation at the current-timestamp. This causes the source weight rarely changes over time. Thus, using a decay factor smaller than 1 will immediately remedy this problem.

3.7.5 Convergence Analysis

To show the convergence of PTDCorr and iPTDCorr, I take the synthetic dataset as an example shown in Figure 3.7. In Figure 3.7(a), it shows the MAE w.r.t the number of iterations in the first timestamp. It can be seen that iPTDCorr converges within 4 iterations. The MAE is reduced significantly in the first 3 iterations. This is because iPTDCorr can estimate source weights within few iterations, which makes the truth discovery process more efficiently. Figure 3.7(b) shows the required number of iterations for reaching convergence. From this figure it can be observed that it uses the most iterations to reach convergence in the first timestamp, and the subsequent timestamps require less iterations to converge. The reason for this phenomenon is that the source weights are initialized uniformly in the first timestamp. The initialized source weights might be largely discrepant from the estimated source weights leading to convergence. Therefore, it requires more iterations to reach convergence for the first timestamp. For the subsequent timestamps, the posterior distribution of source weight estimated from previous timestamp is used as the prior distribution to estimate the source weights at the current timestamp. Over data streams, the source weights usually evolve smoothly over time. This implies that the weights of a source at two adjacent timestamps are usually very close. Thus, it requires less iterations to converge.

In summary, the proposed PTDCorr and iPTDCorr methods can effectively infer object truths by using object correlations, and they outperform all the state-of-art truth discovery methods in terms of effectiveness. The proposed methods are especially effective when information is provided by few sources. By processing the data only



(a) MAE w.r.t. iterations in the 1st timestamp

(b) iterations for convergence

Figure 3.7: Convergence Analysis

once, iPTDCorr can efficiently infer object truths in a dynamic environment, and the experiments have shown that the proposed method has a significant improvement in efficiency over the existing truth discovery method that consider object correlation in a dynamic environment.

3.8 Conclusion

In this chapter, the truth discovery problem with object correlations was investigated. The object correlations are modeled as edges in a chain graph model and a probabilistic truth inference algorithm is proposed to infer object truths with correlations. In order to handle data arriving sequentially in a dynamic environment, incremental truth inference algorithm is develop. It is able to incorporate time-invariant correlations between different objects as well as temporal correlations for the same object to effectively infer object truths. It is also able to retain source reliability information over time as it does not need to re-process the historical data. The experiments demonstrated that the proposed methods outperformed some existing state-of-the-art truth discovery methods both effectively and efficiently.

The models developed in this chapter aims at answering research question 1. The work in this chapter has been published in (Yang et al., 2019c).

Chapter 4

Dynamic Source Weight Computation for Truth Inference over Data Streams

4.1 Overview

As reviewed in Chapter 2, most existing truth discovery methods designed for static data can achieve high accuracy but are computationally expensive to be applied to stream data. On the other hand, many existing truth discovery methods designed for streaming data can achieve high efficiency but sacrifice the accuracy. A method is needed to improve the accuracy of truth discovery over data streams, and still guarantees the efficiency. To address the accuracy and efficiency issues of streaming data truth inference, this chapter presents 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, *unit error* is defined that captures the truth discovery error caused by not changing source weights at certain timestamps. It analyzes the relationship between unit error and source weight evolution, and proves that the unit error is limited if the source weight evolution satisfies a certain condition. If the unit error is under a threshold, it skips the expensive

iterative process and computes object truths directly. As the source weight evolution is unknown before the source weights are computed, a prediction model is developed to predict source weight evolution over time. Finally, DSWC algorithm flow is presented by integrating the error analysis and the prediction model to present. In summary, the contributions in this chapter are detailed as follows.

- It theoretically analyzes the relationship between unit error and source weight evolution when each object is only claimed or reported by a subset of sources/agents. It is proved that the unit error can be limited within a threshold if the source weight evolution satisfies a certain condition.
- A prediction model is developed that is able to accurately predict source weight evolution and report the posterior distribution of source weight and source weight evolution.
- By integrating the error analysis and the prediction model, it presents 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 developed method.

The rest of this chapter is organized as follows. In Section 4.2 it defines the truth discovery studied in this chapter and discusses the related work. In Section 4.3, it presents the preliminary of this chapter. In Section 4.4, it theoretically analyzes the relationship between unit error and source weight evolution. Section 4.5 describes a prediction model that predicts source weight evolution. The DSWC algorithm flow is presented in Section 4.6. Section 4.7 presents the experimental results. Finally, it concludes in Section 4.8.

Notation	Description
j	an object
i	an source
J	set of all the objects
J_i^p	the set of objects that are claimed by source i at timestamp p
I	set of the sources
I_j^p	set of sources that claim object j at time-stamp p
x_{ij}^p	the claim of object j provided by a source i at timestamp p
z_j^p	the truth of object j at timestamp p
m_j^p	the maximum value of claims of object j at timestamp p
a_i^p	the weight of source i at timestamp p
$\phi_j^{p/q}$	the unit error of object j between timestamps p and q
$z_j^{p/q}$	the truth of j at timestamp q estimated by source weights computed at timestamp p
λ	a smoothing factor
$\delta_i^{p/q}$	the source weight evolution of source i between timestamp p and q

Table 4.1: Notations in Chapter 4

4.2 Truth Discovery & Related Work

In this section, the problem definition of truth discovery on data streams is given below, and important notations that will be used throughout this chapter are summarized in Table 4.1.

Problem Definition. Suppose there are a set of objects J and a set of agents/sources I. Each object j at a timestamp p can be claimed by a set of sources I_j^p where $I_j^p \subseteq I$. The claims of an object j by a source i at timestamp p is denoted as x_{ij}^p. The goal of truth discovery is to estimate the truth for each object at each timestamp, i.e. $\{z_j^p\}$ where z_j^p is defined as the truth of object j at timestamp p.

As reviewed in Chapter 2, many truth discovery methods have iterative algorithm flows which update source weights and object truths alternatively and iteratively. A well-known truth discovery strategy is **weighted aggregation** (Q. Li, Li, Gao, Su et al., 2014; Q. Li, Li, Gao, Zhao et al., 2014; Y. Li, Li et al., 2016; B. Zhao & Han, 2012; Y. Li et al., 2015; Yao et al., 2018). The weighted aggregation can be derived by using coordinate descent if the truth inference problem is modeled as an optimization task

(Q. Li, Li, Gao, Zhao et al., 2014), or expectation maximization if the truth inference problem is modeled by a probabilistic graphical model (B. Zhao & Han, 2012). The high-level view of weighted aggregation is given below.

$$z_j^p = \frac{\sum_{i \in I_j^p} a_i^p \times x_{ij}^p + c}{\sum_{i \in I_j^p} a_i^p + b} \tag{4.1}$$

In Equation (4.1), a_i^p is the weight of source i at timestamp p. It is modeled as a positive number which reflects the reliability of source i. A source's weight is higher if its claims are closer to the truths. c and b can be set differently to capture different characteristics when inferring object truths. For example, c_1 and c_2 are set to 0 for basic weighted aggregation (Q. Li, Li, Gao, Zhao et al., 2014). If PGM is used to estimate object truths where the object truths are generated from Gaussian distributions (B. Zhao & Han, 2012), then $c = \frac{\mu_j^p}{\sigma_j^{p2}}$ and $b = (\frac{1}{\sigma_j^p})^2$, where μ_j^p and σ_j^p are the mean and variance parameters of the Gaussian distribution that generates the truth of object j at timestamp p. 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 the temporal smoothness (Y. Li et al., 2015), c and b can be set to λz_j^{p-1} and λ, respectively, where λ is a smooth factor (hyperparameter), and z_j^{p-1} is the estimated truth of object j at the previous timestamp $p-1$. A larger λ enforces the truth at current timestamp to be very close to the estimated truth at the previous timestamp. If a claim at timestamp p is significantly different from z_j^{p-1}, this claim can be treated as an outlier and discarded.

The **source weights computation strategy** can be derived differently by different methods. For example, CRH (Q. Li, Li, Gao, Zhao et al., 2014) and DyOP (Y. Li et al., 2015) use the following equations to compute source weights.

$$\text{CRH: } a_i^p = -\log \frac{\sum_{j \in J_i^p} (x_{ij}^p - z_j^p)^2}{\sum_{i' \in I} \sum_{j \in J_{i'}^p} (x_{i'j}^p - z_j^p)^2} \tag{4.2}$$

$$\text{DyOP: } a_i^p = \frac{|J_i^p|}{\sum_{j \in J_i^p} (x_{ij}^p - z_j^p)^2} \tag{4.3}$$

In the above two equations, J_i^p denotes the objects that are claimed by source i at timestamp p. $\sum_{j \in J_i^p} (x_{ij}^p - z_j^p)^2$ represents the error that i makes on claiming the objects at timestamp p. By incorporating prior beliefs, it can be assumed the source weight is generated from an Inverse-Gamm distribution, and GTM can be applied to compute source weights (Zhao & Han, 2012):

$$\text{GTM: } a_i^p = \frac{2(\beta_1 + 1) + |J_i^p|}{2\beta_2 + \sum_{j \in J_i^p} (x_{ij}^p - z_j^p)^2} \tag{4.4}$$

In Equation (4.4), β_1 and β_2 are the hyperparameters of an Inverse-Gamma distribution which encode the prior beliefs of a_i^p. Although the source weights are computed differently by different methods, it can be observed that all the methods assign high weights to the reliable sources whose claims are closer to the object truths.

Normally, the iterative based[1] methods can achieve high accuracy. However, iterative processes are 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, Y. Li et al. (2015) 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 and DynaTD+s was proposed by adding a smoothness constraint to infer object truths. iCRH (Y. Li, Li et al., 2016) was developed to infer truths of heterogeneous data incrementally over data streams. These methods are efficient

[1] In this chapter, the truth discovery methods having an iterative algorithm flow are referred as iterative-based methods. This is different from the methods that are designed with an iterative truth discovery framework presented in Section 2.1.1. For example, CRH is an optimization based truth discovery method, but it has an iterative algorithm flow, so CRH is treated as an iterative based method in this chapter.

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 (T. Li et al., 2017) 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 claimed by all the agents at every timestamp, i.e., $\forall t \in [1, T]$, $|I_j^t| = |I|$. 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 developed method, DSWC, aims at balancing 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 4.4 is based on Taylor expansion, it only requires each source claims a subset of objects, which is more practical for real-world applications. The source weight prediction model developed in Section 4.5 is able to capture the covariance of source weights over time, which ensures the accuracy and efficiency of DSWC. In the next section, it will present the preliminaries of this chapter.

4.3 Preliminary

This chapter studies numerical truth inference problem over data streams. The weighted aggregation in Equation (4.1) is adopted to estimate object truths. From Equation (4.1) it can be seen that the truth of an object at timestamp i is determined by the the weights of sources who claim it at timestamp i. The weighted aggregation can also be written as a function of source weights given below

$$f_j^p(\{a_i^p\}) = z_j^p = \frac{\sum_{i \in I_j^p} a_i^p \times x_{ij}^p + c}{\sum_{i \in I_j^p} a_i^p + b} \tag{4.5}$$

where $\{a_i^p\}$ are the weights of sources that claim object j at timestamp p, and $\{x_{ij}^p\}$, c and b are all constants. By Equation (4.5), it can be observed that the estimated object truth is sensitive to the change of the source weights. If the values of source weights are varied, then the estimated object truth is changed. In real-world applications, source weights usually change smoothly over time (Y. Li et al., 2015). At timestamp q, if it uses the source weights computed at a previous timestamp p, where $p < q$, to estimate the truth directly without computing the source weights iteratively, it will produce a small error on the estimated truth. Whereas, the efficiency can be improved by skipping the iterative process. Inspired by this idea, I develop a novel method, **D**ynamic **S**ource

Weight **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, the *unit error* $\phi_j^{p/q}$ is defined in Equation (4.6) which measures the deviation of estimated object truth at timestamp q by using source weights computed at timestamp p.

$$\phi_j^{p/q} = \Big(\frac{z_j^p - z_j^{p/q}}{m_j^q}\Big)^2 = \Big(\frac{f_j^q(\{a_i^q\}) - f_j^q(\{a_i^p\})}{m_j^q}\Big)^2 \tag{4.6}$$

In Equation (4.6), z_j^q is the truth estimated by the source weights $\{a_i^q\}$ computed at timestamp q, i.e., $f_j^q(\{a_i^q\})$, and $z_j^{p/q}$ is the approximate truth of object j at timestamp q estimated by the source weights $\{a_i^p\}$ computed at timestamp p, i.e., $f_j^q(\{a_i^p\})$. m_j^q is a scaling factor and defined as the absolute maximum value of claims for j at timestamp q, i.e. $m_j^q = \max\{x_{ij}^q\}_{i \in I_j^q}$. If the unit error is under a user-defined tolerable threshold ϵ, then it chooses to use $\{a_i^p\}$ to approximate the object truths at timestamp q without conducting an expensive iterative process.

At timestamp q, the unit error is determined by the change of source weights from timestamp p to q. The source weight evolution $\delta_i^{p/q}$, given in Equation (4.7), can be used to capture the absolute difference of source weights from timestamp p to q. Without loss of generality, it assumes the source weights at each timestamp are scaled and summed up to 1, i.e., $\forall t \in \{1, \ldots, T\}, \sum_{i \in I} a_i^t = 1$.

$$\delta_i^{p/q} = |a_i^q - a_i^p| \tag{4.7}$$

In the next section, I will discuss the relationship between unit error and source weight evolution, and present the source weight evolution upper bound for limiting unit error.

4.4 Error Analysis

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

The approximate truth $z_j^{p/q}$, or $f_j^q(\{a_i^p\})$, is sensitive to the the source weights $\{a_i^p\}$. The change rate of $f_j^q(\{a_i^p\})$ can be captured by its derivative:

$$\begin{aligned}\frac{\partial f(\{a_i^p\})}{\partial a_i^p} &= \frac{\frac{x_{ij}^q \times (\sum_{i' \in I_j^q} a_{i'}^p + b) - (\sum_{i \in I_j^q} a_{i'}^p \times x_{i'j}^q + c)}{\sum_{i' \in I_j^q} a_{i'}^p + b}}{\frac{(\sum_{i' \in I_j^q} a_{i'}^p + b)^2}{\sum_{i' \in I_j^q} a_{i'}^p + b}} \\ &= \frac{x_{ij}^p - f(\{a_i^p\})}{\sum_{i' \in I_j^p} a_{i'}^p + b}\end{aligned} \tag{4.8}$$

To keep the notation uncluttered, Equation (4.8) uses f to denote f_j^q. Next, it presents a theorem to show the high order derivative of weight aggregation in Equation (4.5).

Theorem 4.1. *The n^{th} order partial derivative of $f(\{a_i^p\})$ w.r.t. n source weights (i.e. $a_{i1}^p, \ldots, a_{in}^p$) is:*

$$\frac{\partial^n f(\{a_i^p\})}{\partial a_{i1}^p \ldots \partial a_{in}^p} = (-1)^{n-1}(n-1)! \frac{\sum_{k=1}^n x_{ikj}^q - f(\{a_i^q\})}{(\sum_{i' \in Ij^q} a_{i'}^p + b)^n} \tag{4.9}$$

Proof. For any integers $n \geq 1$, let $P(n)$ denotes the statement $\frac{\partial^n f(\{a_i^p\})}{\partial a_{i_1}^p \ldots \partial a_{i_n}^p} = (-1)^{n-1}(n-1)! \frac{e_{i_1} + e_{i_2} + \cdots + e_{i_n}}{(\sum_{i' \in I_j^q} a_{i'}^p + b)^n}$ where $e_i = x_{ij}^q - f(\{a_i^p\})$. Theorem 4.1 can be proved by induction.

Base step ($n = 1$)**:** $P(1)$ is true as shown by Equation(4.8).

Inductive step $P(k) \rightarrow P(k+1)$**:** Fix some integer $k \geq 2$. Assume that $P(k)$ holds. It needs to show that $P(k+1)$:

$$\frac{\partial^{k+1} f(\{a_i^p\})}{\partial a_{i_1}^p \ldots \partial a_{i_k}^p \partial a_{i_{k+1}}^p} = (-1)^k (k)! \frac{e_{i_1} + \cdots + e_{i_k} + e_{i_{k+1}}}{(\sum_{i' \in I_j^q} a_{i'}^p + b)^{k+1}}$$

Let $\Omega_j^{p/q} = \sum_{i \in I_j^q} a_i^p + b$, by the assumption it can derive

$$\frac{\partial^k f(\{a_i^p\})}{\partial a_{i_1}^p \dots \partial a_{i_k}^p} = (-1)^{k-1}(k-1)! \frac{(\sum_{y=1}^k x_{i_y j}^q) - k \times f(\{a_i^p\})}{(\Omega_j^{p/q})^k}$$

because $e_{i_y} = x_{i_y j}^p - f(\{a_i^p\})$. Rearrange the above equation, it can get:

$$k f(\{a_i^p\}) = \sum_{y=1}^{k} x_{i_y j}^q - \frac{1}{(-1)^{k-1}(k-1)!} \times \frac{\partial^k f(\{a_i^p\})}{\partial a_{i_1}^p \dots \partial a_{i_k}^p} (\Omega_j^{p/q})^k$$

Taking the derivative w.r.t. $w_i^{s_{k+1}}$ on both sides of the above equation:

$$\begin{aligned} k \frac{\partial f(\{a_i^p\})}{\partial a_{i_{k+1}}^p} &= \frac{\partial}{\partial a_{i_{k+1}}^p} \Big(\sum_{y=1}^{k} x_{i_y j}^q - \frac{1}{(-1)^{k-1}(k-1)!} \times \frac{\partial^k f(\{a_i^p\})}{\partial a_{i_1}^p \dots \partial a_{i_k}} {}^{p} \times (\Omega_j^{p/q})^k \Big) \\ &= -\frac{1}{(-1)^{k-1}(k-1)!} \Big(\frac{\partial^{k+1} f(\{a_i^p\})}{\partial a_{i_1}^p \dots \partial a_{i_{k+1}}^p} \times (\Omega_j^{p/q})^k + k(\Omega_j^{p/q})^{k-1} \times \frac{\partial^k f(\{a_i^p\})}{\partial a_{i_1}^p \dots \partial a_{i_k}^p} \Big) \end{aligned}$$

Rearrange the above equation, it can show that:

$$\begin{aligned} \frac{\partial^{k+1} f(\{a_i^p\})}{\partial a_{i_1}^p \dots \partial a_{i_{k+1}}^p} &= \frac{-1}{(\Omega_j^{p/q})^k} \Big((-1)^{k-1}(k-1)! k \frac{\partial f(\{a_i^p\})}{\partial a_{i_{k+1}}^p} + k(\Omega_j^{p/q})^{k-1} \frac{\partial^k f(\{a_i^p\})}{\partial a_{i_1}^p \dots \partial a_{i_k}^p} \Big) \\ &= (-1)^k k! \frac{(x_{i_1 j}^q - f(\{a_i^p\})) + \dots + (x_{i_k j}^q - f(\{a_i^p\})) + (x_{i_{k+1} j}^q - f(\{a_i^p\}))}{(\Omega_j^{p/q})^{k+1}} \\ &= (-1)^k (k)! \frac{e_{i_1} + \dots + e_{i_{k+1}}}{(\sum_{i' \in I_j^q} a_{i'}^p + b)^{k+1}} \end{aligned}$$

Conclusion: By induction, it is proved that for all integers $n \geq 1$, $P(n)$ is true. Therefore, Equation (4.9) holds. □

Next, it analyzes the unit error by using Taylor Expansion:

$$\begin{aligned} \sqrt{\phi_j^{p/q}} = \frac{|f(\{a_i^q\}) - f(\{a_i^p\})|}{m_j^q} &= \frac{1}{m_j^q} \Big| \sum_{i_1 \in I_j^q} \frac{\partial f}{\partial a_{i_1}^p} \Delta a_{i_1}^{p/q} \\ &\quad + \frac{1}{2!} \sum_{i_1 \in I_j^q} \sum_{i_2 \in I_j^q} \frac{\partial^2 f}{\partial a_{i_1}^p \partial a_{i_2}^p} \Delta a_{i_1}^{p/q} \Delta a_{i_2}^{p/q} + \dots \Big| \end{aligned} \tag{4.10}$$

where $\Delta a_{i_1}^{p/q} = a_{i_1}^q - a_{i_1}^p$. Based on the Equations (4.8 - 4.10), the following proposition is proposed to show the upper bound of source weight evolution to ensure $\phi_j^{p/q} \leq \epsilon$.

Proposition 4.1. *Given a unit error threshold ϵ and an object j, $\phi_j^{p/q} \leq \epsilon$ if the source weight evolution $\delta_i^{p/q}$ for each source $i \in I_j^q$ satisfies the following condition:*

$$\delta_i^{p/q} \leq \frac{\sqrt{\epsilon} \times \Omega_j^{p/q}}{|I_j^q| \times (\xi_j^{p/q} + \sqrt{\epsilon})} \tag{4.11}$$

where $\Omega_j^{p/q} = \sum_{i \in I_j^q} a_i^p + b$, and $\xi_j^{p/q} = \max\left\{\frac{|x_{ij}^q - f(\{a_i^p\})|}{m_j^q}\right\}_{i \in I_j^q}$.

Proof. By definition, $\delta_i^{p/q} = |\Delta a_i^{p/q}|$. Substituting the derivatives (Equations (4.8) and (4.9)) into Equation (4.10), it can infer:

$$\sqrt{\phi_j^{p/q}} \leq \sum_{i_1 \in I_j^q} \frac{\xi_j^{p/q}}{\Omega_j^{p/q}} \delta_{i_1}^{p/q} + \frac{1}{2} \sum_{i_1 \in I_j^q} \sum_{i_2 \in I_j^q} \frac{2\xi_j^{p/q}}{(\Omega_j^{p/q})^2} \delta_{i_1}^{p/q} \delta_{i_2}^{p/q} + \dots$$

Substituting Formula (4.11) in the above inequation, by the sum of geometric series it can show that:

$$\begin{aligned} \sqrt{\phi_j^{p/q}} &\leq \xi_j^{p/q} \times \left(\frac{\sqrt{\epsilon}}{\xi_j^{p/q} + \sqrt{\epsilon}} + \left(\frac{\sqrt{\epsilon}}{\xi_j^{p/q} + \sqrt{\epsilon}}\right)^2 + \dots \right) \\ &= \xi_j^{p/q} \times \frac{\sqrt{\epsilon}}{\xi_j^{p/q} + \sqrt{\epsilon} - \sqrt{\epsilon}} = \sqrt{\epsilon} \end{aligned}$$

Hence, $\phi_j^{p/q} \leq \epsilon$. □

Proposition 4.1 states that for an object j, if every source $i \in I_j^q$ satisfies the condition given in Formula (4.11), then $\{a_i^p\}$ can be used to approximate x_{ij}^q and ensure $\phi_j^{p/q} \leq \epsilon$ at the same time. Based on Proposition 4.1, Proposition 4.2 is proposed to define the upper bound of source weight evolution which guarantees that all the objects' unit errors are under ϵ.

Proposition 4.2. *For each source i, if $\delta_i^{p/q} \leq r_i^{p/q}$ where $r_i^{p/q} = \min(\{\frac{\sqrt{\epsilon} \times \Omega_j^{p/q}}{|I_j^q| \times (\xi_j^{p/q} + \sqrt{\epsilon})}\}_{j \in J_i^q})$, then for each object $j \in J$, $\phi_j^{p/q} \leq \epsilon$.*

Proposition 4.2 states that for a source $i \in I$, the upper bound of its source weight evolution should be no more than $r_i^{p/q}$ to ensure the unit errors of its claimed objects under ϵ. Hence, for each object j, ensuring $P(\phi_j^{p/q} \leq \epsilon) \geq \alpha$ is equivalent to ensure $P(\delta_i^{p/q} \leq r_i^{p/q}) \geq \alpha$ for all the sources.

4.5 Prediction Model

The previous section presents 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 q. In order to avoid the iterative process at each timestamp, this section presents a source weight prediction model to predict the source weights $\{a_i^q\}$ instead of computing them iteratively. Specifically, the prediction model predicts the probability of $\phi_j^{p/q} \leq \epsilon$. Given a user-defined confidence threshold α, if $p(\phi_j^{p/q} \leq \epsilon) \geq \alpha$, then it chooses to estimate object truths at timestamp q by source weight computed at timestamp p. Otherwise, it conducts the iterative process at timestamp q to obtain accurate object truths and source weights. Next, it will describe the prediction model in details.

The source weight a_i^q is computed differently by different methods. Thus, a_i^q can be treated as a random function $g_i(q)$. Similarly, given a vector of timestamps $\boldsymbol{t} = [1, \ldots, j]^T$, let $g_i(\boldsymbol{t})$ denotes the vector of weights of i over $\boldsymbol{t}$, i.e., $g_i(\boldsymbol{t}) = [a_i^1, \ldots, a_i^p]^T$. Then, $g_i(\boldsymbol{t})$ can be modeled as a Gaussian Process (GP) $g_i(\boldsymbol{t}) \sim \mathcal{N}(m(\boldsymbol{t}), \boldsymbol{K^{(i)}})$ where $m(\boldsymbol{t})$ is a prior mean function for $g_i(\boldsymbol{t})$, $\boldsymbol{K^{(i)}}$ is a $p \times p$ covariance matrix at timestamp p. The (x, y) entry in $\boldsymbol{K^{(p)}}$ stores the covariance between $g_i(x)$

and $g_i(y)$, which is measured by a kernel function $k(x, y)$. Then a_i^q can be predicted by $P(g_i(q)|q, \boldsymbol{t}, g_i(\boldsymbol{t})) = \mathcal{N}(\mu_q, \sigma_q^2)$ with mean μ_q and variance σ_q^2 given below (Rasmussen, 2004).

$$\mu_q = m(q) + \boldsymbol{k_q}(\boldsymbol{K^{(p)}})^{-1}(g_i(\boldsymbol{t}) - m(\boldsymbol{t}))$$
$$\sigma_q^2 = k(q, q) - \boldsymbol{q}(\boldsymbol{K^{(p)}})^{-1}(\boldsymbol{k_q})^T$$
$$\boldsymbol{k_q} = [k(q, 1), \ldots, k(q, p)]$$

By definition, $\delta_i^{p/q} = |a_i^q - a_i^p|$. Therefore, the probability $P(\delta_i^{p/q} \leq r_i^{p/q})$ can be computed by evaluating $P(-r_i^{p/q} \leq g_i(q) - a_i^p \leq r_i^{p/q})$, which can be calculated by using the cumulative probability of Normal distribution $\mathcal{N}(\mu_q - a_i^p, \sigma_q^2)$.

The developed 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 a_i^q, which is suitable for evaluating $P(\delta_i^{p/q} \leq r_i^{p/q})$. (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, $\boldsymbol{K^{(p)}}$ needs to be updated for future prediction. The update procedures into the following two cases:

Case 1: $P(\delta_i^{p/q} \leq r_i^{p/q}) \geq \alpha$: In this case, it uses $\{a_i^p\}$ to approximate x_{ij}^q, and does not update $\boldsymbol{K^{(p)}}$.

Case 2: $P(\delta_i^{p/q} \leq r_i^{p/q}) < \alpha$: In this case, it needs to compute $\{a_i^q\}$. The procedure of updating $\boldsymbol{K^{(p)}}$ for this case is summarized in Algorithm 4.1. In Algorithm 4.1, it will augment $\boldsymbol{K^{(p)}}$ $(q - p)$ times. In each augment, it first computes the covariance

Algorithm 4.1: Update Covariance Matrix $\boldsymbol{K}$

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

1 **for** $t = p + 1 \ldots q$ **do**
2 $\quad \boldsymbol{K_t} = [k(t,1), \ldots, k(t,t-1)]$
3 $\quad$ **if** $t = q$ **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** $\boldsymbol{K}^{(j)}$

between the source weights at timestamp t and the previous ones (Line 2). If the source weight a_i^t 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, $\boldsymbol{K^{(p)}}$ is augmented by adding new covariances and error of a_i^t in it.

Gaussian Process needs to retain all the historical information which measures the covariance between each source weight at different timestamps in $\boldsymbol{K^{(p)}}$. From Algorithm 4.1, it can be seen that the size of $\boldsymbol{K^{(p)}}$ is increased by $2p + 1$ for each augment. As $\boldsymbol{K^{(p)}}$ 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, a sliding window can be used to maintain the covariances of L most recently source weights in $\boldsymbol{K^{(p)}}$. By using the sliding window technique to update $\boldsymbol{K^{(p)}}$, the size of $\boldsymbol{K^{(p)}}$ will be at most L^2 and the inversing $\boldsymbol{K^{(p)}}$ is not an issue.

4.6 DSWC Algorithm Flow

By integrating the error analysis and prediction model, the DSWC algorithm in presented Algorithm 4.2.

In Algorithm 4.2, p is the last timestamp at which the source weights are computed

by an iterative process, q is the current timestamp, and L is the size of the sliding window. X^q is the set of claims at timestamp q and Z^q is the set of truths at timestamp q. 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 $P(\delta_i^{p/q} \leq r_i^{p/q}) \geq \alpha$ is satisfied (Line 6), it uses $\{a_i^p\}$ to approximate object truths at timestamp q (Line 7). Otherwise, iterative_process() will be conducted at the current timestamp to compute source weights and object truths.

4.7 Experiments

This section presents the experimental results conducted by using four real-world datasets to evaluate the performance of DSWC algorithm. All the methods are implemented in Java. Experiments are conducted on a Windows PC with Intel i7 CPU and 16 GBs RAM.

4.7.1 Experiment Setup

Datasets. The dataset descriptions are given below.

- **Weather** (Dong et al., 2010): This dataset contains 18 sources that record daily

Algorithm 4.2: DSWC Truth Inference

Input : Claims $\{X^q\}_{q=[1,T]}$, ϵ and α
Output : Truths at each timestamp $\{Z^q\}_{j=[q,T]}$

```
p ← 1;
for q = 1 → T do
    if q ≤ L then
        iterative_process();
    else
        if all sources satisfy P(δ_i^{p/q} ≤ r_i^{p/q}) ≥ α then
            Z^q ← {z_i^{p/q}}
        else
            iterative_process();
        end
    end
end
return {Z^q}
Procedure iterative_process()
    Initialize the truths Z^q;
    repeat
        Compute source weights;
        Compute truths;
    until Convergence condition satisfied
    Scale source weights;
    Update K for sources;
    p = q;
```

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** (X. Li, Dong, Lyons, Meng & Srivastava, 2012): 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**: Hourly weather forecast data are collected from five sources (Aeris[3],

[2]The only source that is not used because it does not contain temperature data.
[3]www.aerisweather.com

Apixu[4], Darksky[5], World Weather Online[6] and Wunderground[7]) for 42 different locations (objects) in New York city over 180 hours. The ground truths are collected 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. Claims 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 claims is high.

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

Baselines. The source weight computation strategies of CRH, DyOP and GTM, as shown in Equations (4.2), (4.3) and (4.4), are applied 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), these methods are denoted 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. Squared exponential kernel is used to measure the covariance between source weights at different timestamps.

The baseline truth inference methods include the iterative-based methods: DyOP (Y. Li et al., 2015), GTM (B. Zhao & Han, 2012), CRH (Q. Li, Li, Gao, Zhao et al., 2014), LFC (Raykar et al., 2010) and OTD (Yao et al., 2018). The descriptions of the baseline methods can be seen in Section 2.5 on page 45. As GTM can incorporate

[4]www.apixu.com
[5]darksky.net/about/
[6]www.worldweatheronline.com
[7]www.wunderground.com
[8]Data collected from https://fixer.io/.

prior beliefs, in the experiment I 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 (Y. Li et al., 2015), DynaTD+s (Y. Li et al., 2015) and iCRH (Y. Li, Li et al., 2016). 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.

In the stock dataset, each source averagely claims 897 objects, two sources claim less than 200 objects at each day, and no source claim 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.

4.7.2 Prediction Model Evaluation

This subsection presents the results of experiments which compared the effectiveness of the developed prediction model with 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, $\delta_i^{p/q} \leq r_i^{p/q}$, to ensure $\phi_i^{p/q} \leq \epsilon$ 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.

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 4.2: 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 4.3: Prediction Model Evaluation for Rates Dataset

- 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, *accuracy* ($accuracy = \frac{TP + TN}{TP + TN + FP + FN}$) is used to measure the effectiveness of the prediction model. Two parameters, ϵ and α are varied to evaluate the performance with different settings.

The results of experiments conducted on weather and rates datasets are shown in Tables 4.2 and 4.3. It can observed that the prediction model in DSWC outperforms the probabilistic model in ASRA under all parameter settings. The reason is that the

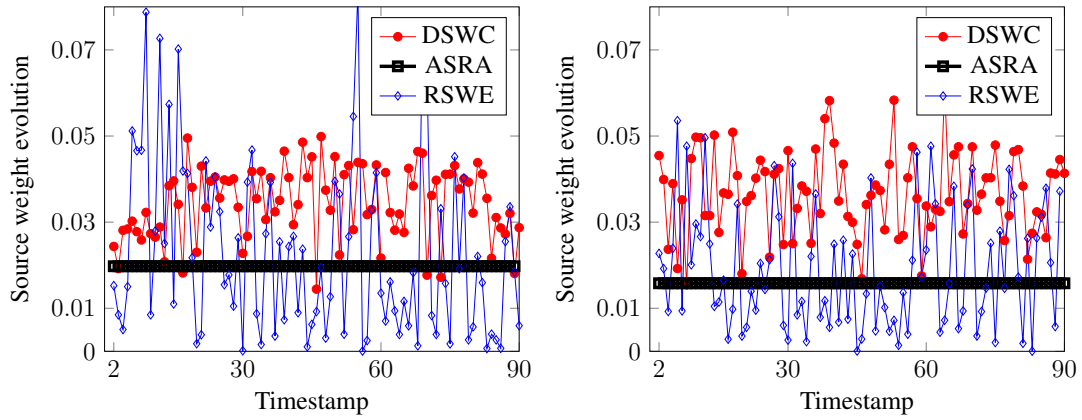
prediction model of DSWC evaluates the covariances between the source weights over time. However, ASRA models the satisfaction of source weight evolution as a Bernoulli random variable. It overlooks the correlation of the source weights over the data streams, which results in less accurate prediction results.

Furthermore, it can be observed 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 α = 0.9. The reason is that the variance of the posterior distribution is not small enough to assert $P(\delta_i^{p/q} \leq r_i^{p/q}) \geq \alpha$, 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 α = 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.

4.7.3 Source Weight Evolution Condition

The weather and rates datasets are used to to test the source weight evolution condition that satisfies $\phi_i^{q-1/q} \leq \epsilon$ 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 $r_i^{p/q}$ in Proposition 4.2 for DSWC, a random source is chosen 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 4.1. It can be observed



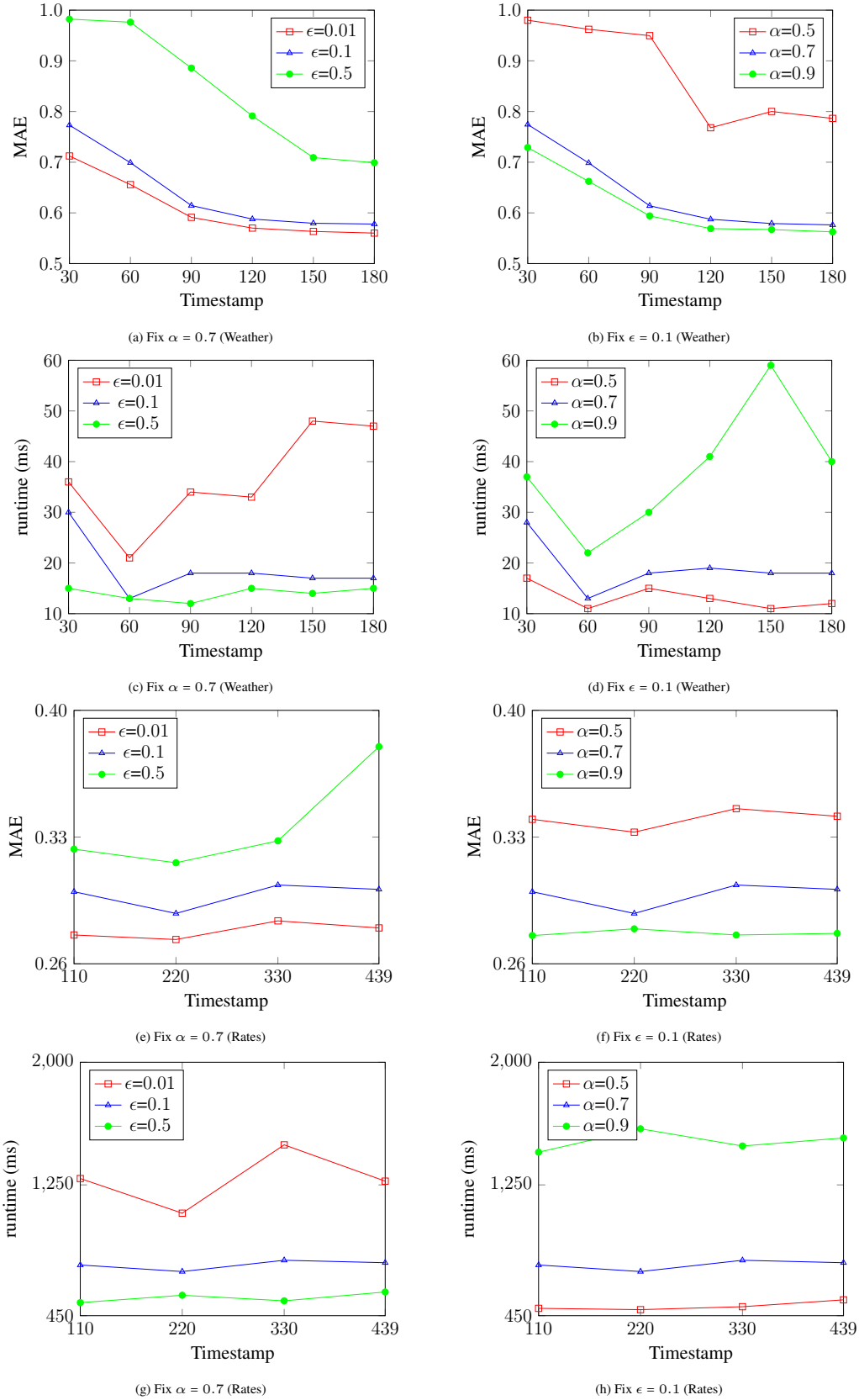
(a) Weather (b) Rates

Figure 4.1: Source Weight Evolution Condition Comparison

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 $\phi_i^{q-1/q} \leq \epsilon$. 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 $\phi_i^{q-1/q}$ but ASRA cannot. For the moments when the blue dot is above the red one, it means the source weight evolution $\delta_i^{q-1/q}$ cannot guarantee $\phi_i^{q-1/q} \leq \epsilon$. 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.

4.7.4 Parameters Analysis

This subsection presents the results of experiments that tested the effect of parameters ϵ and α on the performance of DSWC. The experiments were conducted by fixing one parameter and varying the other. The results of weather and rates datasets run by DSWC(DyOP) are illustrated in Figure 4.2. In the figure, on one hand, it can be seen that as ϵ increases, MAE increases but the runtime decreases for both datasets. The reason is that larger ϵ increases the probability $P(\delta_i^{p/q} \leq r_i^{p/q})$, which results in less iterative processes conducted over time. In this case, the truth inference process



(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 4.2: 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 4.4: Accuracy and Efficiency Comparison

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 $P(\delta_i^{p/q} \leq r_i^{p/q}) \geq \alpha$ less likely to hold. In this case, the truth inference is configured to tolerate a tiny error, which requires more iterative processes conducted.

4.7.5 Performance Comparison

The performance of DSWC is compared 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 4.4 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 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 $P(\delta_i^{p/q} \leq r_i^{p/q}) \geq \alpha$.

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.

4.8 Conclusion

This chapter introduced presents 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 claim all objects at each timestamp. Thus, it fits into more application scenarios. Experiments on four datasets demonstrate that the developed method is both accurate and efficient for truth inference over data streams.

The models developed in this chapter aims at answering Research Question 2. The work in this chapter has been published in (Yang et al., 2019a).

Chapter 5

Modeling Random Guessing and Task Difficulty for Truth Discovery in Crowdsourcing

5.1 Overview

This chapter and the next chapter study the truth discovery problem in crowdsourcing systems, and focus on the single-choice crowdsourcing tasks. In the context of crowdsourcing, the objects are the crowdsourcing tasks, the data sources are the crowd workers, and the truth discovery is usually referred as truth inference in the context of crowdsourcing. A single-choice crowdsourcing task has several choices for the workers to label. The available choices of a task is mutual exclusive and there is only one true label for each single-choice task, the true label is also known as the truth of the task. Each worker can label a task by choosing one choice from the candidate choices of the single-choice task as her response to the crowdsourcing system. As the crowd workers are not experts and the abilities of workers are different, the workers' labels to the same task can be conflicting. Thus, truth discovery techniques can be applied to estimate

the truths of each task. As the crowd workers are human, it brings more challenges to the truth discovery task. In this chapter, it considers two important phenomenons in crowdsourcing applications for crowdsourcing truth discovery. (1) The difficulties of tasks are usually different. A worker who can frequently label an easy task correctly does not mean that her labels to the hard tasks are also trustworthy. Thus, by modeling and estimating tasks' difficulties, the performance of truth discovery in crowdsourcing applications is expected to be improved. (2) As the workers are human and each single-choice task has several candidate choices to choose from, when a worker does not know the true answer of a task, she may choose to guess and submit a random choice as her label.

Motivated by the two phenomenons described above, this chapter presents a novel method, called ***C**rowdsourced **T**ruth **D**iscovery modeling **G**uessing and task **D**ifficulty* (CTDGD), that estimates the truth of single-choice tasks by jointly modeling tasks' difficulties and workers' abilities and guessing behavior. Specifically, the workers' abilities and labels and the tasks' true labels and difficulties are modeled as random variables in a probabilistic generative model. A worker's ability and the task's true label and difficulty jointly determine if the worker knows the true label of the task. If the worker does not know the truth, she submits a guessed label from the candidate choices. By modeling guessing, the workers' abilities can be estimated without overestimation. By modeling tasks' difficulties, the truths of the hard tasks can be estimated more accurately.

The rest of this chapter is organized as follows. Section 5.2 reviews related work. Section 5.3 presents the worker label modeling. In Section 5.4, it describes the probabilistic representation of CTDGD. Section 5.5 describes the inference algorithm that infers the unknown worker ability, tasks' difficulties and truths. The experiments are presented in Section 5.6. Finally, it concludes this chapter in Section 5.7.

5.2 Related Work

There are some existing approaches which estimate task' truths by considering tasks' difficulties, thus, they are relevant to the proposed method. HA-EM (Marshall, Syed & Wang, 2016) applies NLP technique to analyze the tasks' difficulties from text descriptions, and then uses the analyzed difficulty as input to its truth discovery model. UTD (Y. Wang, Ma, Su & Gao, 2017) models each true label as a distribution, and measures a task' difficulty by the variance of the true label's distribution. MistakeLCA (Pasternack & Roth, 2013) models the difficulty as the probability of a worker making mistakes on all the tasks. The difficulty is analyzed from the perspective of the workers instead of the questions, which cannot capture the real difficulties of tasks. FaitCrowd (Ma et al., 2015) and GLAD (Whitehill et al., 2009) both use probabilistic graphical model (PGM) to model the task's difficulty as a parameter. The methods discussed in this paragraph so far considers the difficulty on the task level, a task is more difficult if more workers label it wrongly, and the workers are rewarded with higher quality if they label hard tasks correctly. However, they do not consider the relationship between the fine-grained answer level difficult and the quality of workers. In (Galland et al., 2010), the authors proposed 3estimates in which the difficulty is captured by introducing the error factor of each answers. If a worker gives a wrong answer of a question, the penalty is distributed to both the worker's ability and the error of the chosen answer. However, it ignores the uncertainties the workers expressed on the answers with different difficulties.

The only work that has made progress in capturing the worker's guessing behavior is GuessLCA (Pasternack & Roth, 2013). However, GuessLCA assumes that the worker's guessing distribution is known as an input and it does not account for the tasks' difficulty.

5.3 Worker Label Modeling

Suppose there are m workers $\{w_i\}_{i=0}^{m-1}$, and n tasks $\{t_j\}_{j=0}^{n-1}$. Each task has K mutual exclusive choices indexed from 1 to K. Each worker w_i can label a task t_j by choosing a choice as her response x_{ij} for the task. The goal of truth discovery is to find the true labels $\{z_j\}$ for each task in $\{t_j\}$ from the observed labels $\{x_{ij}\}$. At the same time, the proposed CTDGD outputs the estimated workers' abilities $\{a_i\}$ and tasks' difficulties $\{d_j\}$.

A worker's ability a_i and a task's difficulty d_j are modeled as real numbers taken from $(-\infty, +\infty)$. Using the logistic function, the probability ϕ_{ij} of worker w_i knowing the true label of t_j is

$$\phi_{ij} = \sigma(a_i - d_j) = \frac{1}{1 + \exp(-(a_i - d_j))} \tag{5.1}$$

where σ is the logistic function. From Equation (5.1) suggests that the probability that worker w_i knows the truth of t_j is high if $a_i - d_j$ is large. Therefore, a worker is more likely to give a true label to an easy task and less likely to label a hard task correctly if her ability is smaller than the task's difficulty.

If the worker does not know the truth, she may guess and submit a random label as her label. Thus, the probability of an observed label x_{ij} being the truth z_j is:

$$p(x_{ij} = k | z_j = k, d_j, a_i) = \phi_{ij} + (1 - \phi_{ij})\frac{1}{K} \tag{5.2}$$

Here I choose to use the "one-coin model" (see Section 2.2.1) to model the cases that a worker submits a wrong label. Thus, for all $k' \neq k$, the probability of an observed label x_{ij} being wrong is:

$$p(x_{ij} = k' | z_j = k, d_j, a_i) = (1 - \phi_{ij})\frac{1}{K} \tag{5.3}$$

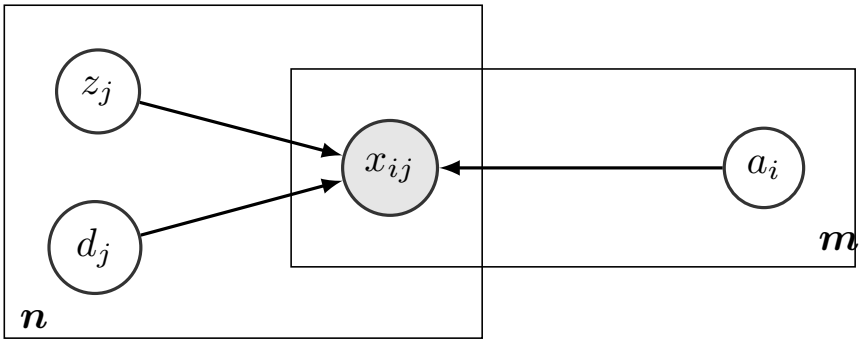


Figure 5.1: Graphical Model

Combing Equations (5.2) and (5.3), the conditional probability of an observed worker's label is:

$$p(x_{ij}|z_j, d_j, a_i) = \left(\phi_{ij} + (1-\phi_{ij})\frac{1}{K}\right)^{\delta_{ij}} \left((1-\phi_{ij})\frac{1}{K}\right)^{1-\delta_{ij}} \tag{5.4}$$

where δ_{ij} denotes the Kronecker delta function.

5.4 Representation of CTDGD

CTDGD is a generative model. The worker's ability a_i, the task's difficulty d_j and truth z_j and the worker's label x_{ij} are modeled as random variables. The relationships between these random variables are depicted in Figure 5.1. The generative processes of each random variable are described as follows.

The true label is generated from a Categorical distribution: $p(z_j) = Cat(K, \boldsymbol{\alpha})$. The worker's label is generated from a Categorical distribution with the p.m.f. defined in Equation (5.4). The task difficulty d_j is generated from a Normal distribution: $p(d_j) = \mathcal{N}(\mu_j, \sigma_j^2)$. The ability of a worker is generated from a Normal distribution: $p(a_i) = \mathcal{N}(\mu_i, \sigma_i^2)$. $\boldsymbol{\alpha}$, μ_j, σ_j^2, μ_i and σ_i^2 are hyperparameters.

5.5 Inference

This section presents an Expectation-Maximization (EM) algorithm that estimates the optimal values of workers' abilities and tasks' truths and difficulties. Specifically, the true labels $Z = \{z_j\}$ are treated as the latent variables, $\Theta = \{D, A\}$ are treated as the model parameters where $D = \{d_j\}$ and $A = \{a_i\}$, and $X = \{x_{ij}\}$ are treated as the observations. The likelihood function of the model parameters is given in Equation (5.5).

$$\begin{aligned} L(\Theta; X, Z) &= p(X, Z|\Theta) \\ &= \prod_j \Big(p(z_j) \prod_i \big[\, p(a_i) p(x_{ij}|z_j, d_j, a_i) \big] \Big) \end{aligned} \tag{5.5}$$

Then EM algorithm finds the maximum likelihood of L and the optimal values of Z and Θ by iteratively performing an E-Step and a M-Step described as the followings.

E-step: In this step, it computes an auxiliary function $Q(\Theta|\Theta^{(t)})$, which is defined as the expectation of the log-likelihood function $\ln L(\Theta; X, Z)$ w.r.t. the latent variables Z given the current estimated model parameters $\Theta^{(t)}$ at iteration t and observations X:

$$\begin{aligned} &Q(\Theta|\Theta^{(t)}) = E_{Z|\Theta^{(t)},X}\Big[\ln L(\Theta; X, Z) \Big] \\ &= \sum_j \sum_{k=1}^{K} p_{jk}^{(t)} \ln p(z_j) + \sum_j p_{jk}^{(t)} \sum_i \ln p(x_{ij}|z_j, d_j, a_i) \\ &= \sum_j \sum_{k=1}^{K} p_{jk}^{(t)} \ln p(z_j) + \sum_j \sum_{k=1}^{K} p_{jk}^{(t)} \sum_i \Big[\delta_{ij} \ln(\phi_{ij} + \frac{1-\phi_{ij}}{K}) \\ &+ (1-\delta_{ij}) \ln(\frac{1-\phi_{ij}}{K}) \Big] \end{aligned} \tag{5.6}$$

In Equation (5.6), $p_{jk}^{(t)} = p(z_j = k|\Theta^{(t)}, X)$, it is defined as the conditional probability of the latent variables given the current estimated model parameters at iteration t and the observations. From the graphical model depicted in Figure 5.1, it can be observed that the workers' labels $x_{ij} \in X$ are conditional dependent given A, Z and D, i.e.

$x_{ij} \perp x_{i'j'} | \{A, Z, D\}$. Thus, the probability $p(z_j|\Theta^{(t)}, X)$ can be decomposed as:

$$p(z_j|\Theta^{(t)}, X) \propto p(z_j) \prod_i p(x_{ij}|z_j, d_j, a_i) \tag{5.7}$$

Based on the decomposition in Equation (5.7), $p_{jk}^{(t)}$ can be computed by the following equation.

$$p_{jk}^{(t)} = \frac{p(z_j = k) \prod_i p(x_{ij}|z_j = k, d_j, a_i)}{\sum_{k'=1}^{K} p(z_j = k') \prod_i p(x_{ij}|z_j = k', d_j, a_i)} \tag{5.8}$$

M-Step: M-step re-estimates the model parameters Θ at the next iteration $t + 1$ by maximizing the auxiliary function $Q(\Theta|\Theta^{(t)})$:

$$\Theta^{(t+1)} = \arg\max_{\Theta} Q(\Theta|\Theta^{(t)})$$

There is no closed form to compute a_i and d_j directly to maximize $Q(\Theta|\Theta^{(t)})$. Therefore, gradient ascent is adopted to maximize $Q(\Theta|\Theta^{(t)})$, and the gradient of $Q(\Theta|\Theta^{(t)})$ can be constructed by differentiating $Q(\Theta|\Theta^{(t)})$ w.r.t. a_i and d_j. Taking the first derivative w.r.t. a_i will make the first summation in Q vanish since the summation is a constant w.r.t. a_i. By the fact that the first derivative of logarithmic function is:

$$\frac{d \ln(x)}{x} = \frac{1}{x}$$

and the first derivative of the logistic function is:

$$\frac{d\sigma(x)}{dx} = \sigma(x)(1 - \sigma(x))$$

the first partial derivative of Q w.r.t. a_i can be computed as:

$$\begin{aligned}
\frac{\partial Q}{\partial \alpha_i} &= \sum_j \sum_{k=1}^{K} p_{jk}^{(t)} \Big[\delta_{ij} \frac{(K-1)\phi_{ij}(1-\phi_{ij})}{(K-1)\phi_{ij}+1} - (1-\delta_{ij})\phi_{ij} \Big] \\
&= \sum_j \sum_{k=1}^{K} p_{jk}^{(t)} \Big[\delta_{ij} \frac{(K-1)\phi_{ij}(1-\phi_{ij})}{(K-1)\phi_{ij}+1} - \phi_{ij} + \delta_{ij}\phi_{ij} \Big] \\
&= \sum_j \sum_{k=1}^{K} p_{jk}^{(t)} \Big[\delta_{ij} \frac{(K-1)\phi_{ij}(1-\phi_{ij}) + ((K-1)\phi_{ij}+1)\phi_{ij}}{(K-1)\phi_{ij}+1} - \phi_{ij} \Big] \\
&= \sum_j \sum_{k=1}^{K} p_{jk}^{(t)} \Big[\delta_{ij} \frac{K\phi_{ij}}{(K-1)\phi_{ij}+1} - \phi_{ij} \Big] \\
&= \sum_j \sum_{k=1}^{K} p_{jk}^{(t)} \Big[\delta_{ij} \frac{K}{(K-1)+\frac{1}{\phi_{ij}}} - \phi_{ij} \Big]
\end{aligned} \tag{5.9}$$

Similarly, the first partial derivative of Q w.r.t. d_j can be computed by Equation (5.10) given below.

$$\frac{\partial Q}{\partial d_j} = -\sum_i \sum_{k=1}^{K} p_{jk}^{(t)} \Big[\delta_{ij} \frac{K}{(K-1)+\frac{1}{\phi_{ij}}} - \phi_{ij} \Big] \tag{5.10}$$

Given the above derivations, EM algorithm iteratively conducts the E-step and M-step until convergence. The convergence analysis of the EM algorithm has been extensively studied (Wu, 1983) and it is beyond the scope of this thesis. In practice, the EM algorithm can be terminated if the change of the likelihoods between two consecutive iterations is small. After the EM algorithm terminates, the model parameters in the last iteration can be used as as the estimated worker's ability and task's difficulty. At the same time, the estimated task truth $\hat{z}_j$ can be estimated by selecting the k^{th} choice that has the highest probability among $p_{jk}^{(t)}$, i.e., $\hat{z}_j = \arg\max_k \{p_{jk}^{(t)}\}$.

5.6 Experiments

Experiments is conducted on a real-world dataset, **Game** (Aydin, Yilmaz & Demirbas, 2017), to compare CTDGD with the state-of-art truth discovery methods. This dataset is collected from a crowdsourcing platform of an Android App based on "Who Wants to Be a Millionaire". This dataset contains 1908 unique questions with 12 difficulty levels. 1891 questions are answered by 37,332 workers with 214,658 unique answers. In the experiments, each question is treated as a task, and each task has 4 choices. The performance is measured **accuracy**, and the ground truths are available for evaluation.

The baseline methods that are used for comparison include ZC (Demartini et al., 2012), GLAD (Whitehill et al., 2009), DS (Dawid & Skene, 1979), LFC (Raykar et al., 2010), CRH (Li et al., 2014), 3Estimates (Galland et al., 2010), GuessLCA (Pasternack & Roth, 2013), TruthFinder (Yin et al., 2008) and MV. The descriptions of these methods can be found in Section 2.5.

The results of experiments conducted on the Game dataset are summarized in Table 5.1. The number of tasks under each difficulty level is enclosed in the parentheses. There are only 9 questions in Level 11 and 1 question in Level 12, these questions on Level 11 and 12 are merged into Level 10, which includes the hardest tasks in this dataset. The number of questions is listed in each level in the parentheses. From Table 5.1, it can be observed that CTDGD has the best overall performance. For the easy tasks, it can be seen that all the methods have a very high accuracy, even majority voting can achieve over 90% accuracy. However, for the medium and hard level tasks, the accuracies of all the methods are dropped below 90%. This is because many workers cannot answer difficult questions correctly. Among all the methods, CTDGD has the best performance on medium and hards tasks, which demonstrates the superiority of CTDGD by jointly modeling tasks' difficulty and workers' guessing behavior.

Method	Accuracy										
	Level 1 (270)	Level 2 (265)	Level 3 (260)	Level 4 (248)	Level 5 (227)	Level 6 (191)	Level 7 (166)	Level 8 (124)	Level 9 (89)	Level 10 (51)	Overall (1891)
CTDGD	**98.15**	**96.98**	**96.54**	**95.97**	94.71	**94.24**	**96.39**	**77.42**	**71.97**	**64.71**	**93.02**
ZC	97.41	96.23	95.77	93.55	93.39	92.15	90.96	69.35	61.8	52.94	90.22
GLAD	97.41	96.23	95.38	93.55	93.39	91.1	90.36	69.35	61.8	50.98	89.95
DS	97.78	96.23	95.77	94.76	93.83	91.62	93.37	69.35	62.92	50.95	90.64
LFC	97.78	96.6	96.15	94.76	**95.15**	**94.24**	95.18	75.81	66.29	58.82	92.12
CRH	97.78	96.6	95.77	94.35	93.39	91.1	92.17	69.35	62.92	50.98	90.43
3Estimates	**98.15**	96.6	96.15	95.16	**95.15**	93.72	95.18	75.81	66.29	60.5	92.23
GuessLCA	97.78	96.6	96.15	94.35	93.83	92.67	93.98	70.16	64.04	49.02	90.9
TruthFinder	97.78	96.6	96.54	94.35	93.38	91.62	92.17	71.77	62.92	47.06	90.69
MV	97.78	96.6	96.15	94.35	93.39	91.1	92.17	69.35	64.04	47.06	90.43

Table 5.1: Experimental Results

5.7 Conclusion

This chapter presents a crowdsourcing truth discovery model, ***C**rowdsourced **T**ruth **D**iscovery modeling **G**uessing and task **D**ifficulty* (CTDGD), which jointly models tasks' difficulties, workers' guessing behavior and abilities to estimate tasks' truths. Experiments on a real-world dataset demonstrate that CTDGD is more effective to estimate the truths of crowdsourced tasks than the state-of-art truth discovery methods, especially when the tasks are difficult.

The models developed in this chapter aimed at answering Research Question 3.1. The work introduced in this chapter has been published in (Yang et al., 2019b).

Chapter 6

"This chapter is embargoed until June 16, 2021"

Chapter 7

On the Discovery of Continuous Truth: A Semi-Supervised Approach with Partial Ground Truths

7.1 Overview

The general setting of truth discovery is that source reliabilities and object ground truths are both known. Based on this setting and the principle of truth discovery, truth discovery methods are developed as an unsupervised learning algorithm that estimates the source reliabilities and object truths from the observed claims. Indeed, obtaining the entire set of ground truths from highly reliable sources is usually expensive and infeasible, but it is usually practical to acquire some ground truths for a small set of objects. For example, part of the objects' ground truths may be available from government websites, information released by governments and official sites is usually real and we can treat it as ground truths. If the partial objects' ground truths can be used by adding some supervisions in the truth discovery steps, the accuracy of truth discovery is expected to be improved. In this chapter, it presents a semi-supervised truth

discovery method, **Op**timization based **S**emi-supervised **T**ruth **D**iscovery (OpSTD) , for continuous object truths. The object truths and source reliabilities are modeled as unknown variables, and the ground truth is modeled as a regularization term to propagate its trustworthiness to the estimated truths. This chapter also presents the theoretical analysis for OpSTD and it shows a series of experiments on both real-world datasets and synthetic dataset to demonstrate the effectiveness of OpSTD.

The detailed contributions of this chapter are summarized as follows.

- The semi-supervised continuous truth discovery problem is formulated as an optimization task in which the partially observed ground truth is incorporated in an objective function by an regularization term.
- An algorithm that estimates the optimal source weights and object truths is developed
- It theoretically proves the convergence and analyze the time complexity of the developed algorithm.
- The experiment results on both real world datasets and synthetic dataset show that the proposed method outperforms the existing methods significantly.

The rest of this chapter is organized as follows. In Section 7.2, it reviews the related. Section 7.3 presents the OpSTD framework and the iterative solution. Section 7.4 presents the convergence property of the proposed method and analyze time complexity. In Section 7.5, it shows the experiments to evaluate the performance of the proposed method. Finally, Section 7.6 concludes this chapter.

7.2 Related Work

There is some work that share similarities with ours. In (Pasternack & Roth, 2013), source reliabilities are modeled as latent variables, its expectation maximization (EM) algorithm based solution can incorporate a small set of ground truths to help truth inference. But it is limited to work with categorical data only. Yin et al. (Yin & Tan, 2011) propose a truth discovery method SSTF that is specifically designed for semi-supervised truth discovery problem. SSTF is originally designed for estimating categorical object truths, it uses a graph based semi-supervised technique, label propagation, to propagate the trustworthiness of ground truths to the claims. SSTF is limited that it uses scoring technique (refer to Section 2.3.1) to output object truths. Therefore, it requires the ground truths are among claims, which is not suitable for the truth discovery applications in which the data is continuous. SSTF also uses a predefined similarity function to capture the relations among observations. This similarity function is application specific and usually hard to define in practice. In contrast, the proposed method OpSTD is specially designed for semi-supervised truth discovery over continuous data and the setting of OpSTD is much simpler. The experiments in Section 7.5 also demonstrates that OpSTD outperforms SSTF to find continuous object truths.

7.3 Semi-Supervised Truth Discovery on Continuous Data

This section presents the formulation of the problem of the semi-supervised truth discovery for continuous object truths first. Then the OpSTD framework and the truth discovery algorithm are presented.

7.3.1 Problem Formulation

Important notations related with this chapter formally defined in this subsection. In addition, related parameters are listed in Table 7.1.

Definition 7.1. ***Object, Source and Claim****: An* ***object****,* j*, is a thing or an event that has a continuous property. A* ***source****,* i*, is an information provider which can observe and report the property value of object* j*. A* ***Claim****,* $z_{ij} \in \mathbb{R}$*, is the continuous property value of object* j *reported by source* i*.*

Definition 7.2. ***Ground truth and estimated truth****: The* ***ground truth****,* $\bar{z}_j \in \mathbb{R}$*, of object* j *is the fractal truth that correctly describes the property value of* j*. It is usually unknown a priori. The* ***Estimated truth****,* $z_j \in \mathbb{R}$*, of object* o*, is the estimated most trustworthy information describing the property value of* j*, it is the output of a given truth discovery method.*

Definition 7.3. ***Source Weight****: The* ***source weight****,* $a_i \in \mathbb{R}^+$*, reflects the reliability of source* i*. The information provided by sources with high source weights is usually more trustworthy and closer to the truth.*

This chapter studies the semi-supervised truth discovery for continuous object truths, in which some partially available ground truths are used to supervise the truth discovery process. Let I be the set of all the sources and J be the set of all the objects. J is split into two sets J_g and J_u where J_g and J_u are disjoint and $J_g \cup J_u = J$. J_g is the set of objects whose ground truths are available, and J_u is the set of objects whose ground truths are unknown. Usually $|J_g| << |J_u|$. Next, the semi-supervised truth discovery problem is formally defined as follows.

Problem Definition. Given the Claims X where $X = \{x_{ij}\}_{j \in J, i \in I}$ and a set of ground truths $\{\bar{z}_j\}_{j \in J_g}$, semi-supervised truth discovery for continuous object truths aims at resolving conflicts among multi-source data and estimating the truths $Z_u = \{z_j\}_{j \in J_u}$ with the help of available ground truths.

Notation	Description
J	set of all the objects
J_u	set of objects whose ground truths are unknown
J_g	set of objects whose ground truths are available
J_{iu}	set of objects claimed by i, and the objects' ground truths are unknown
J_{ig}	set of objects claimed by i, and the objects' ground truths are available
I	set of all the sources
I_j	set of sources that claim object j
X	set of all the claims
Z_u	set of estimated truths for objects in J_u
A	set of all the source weights
x_{ij}	the claims for object j reported by source i
a_i	weight of source i
$\bar{z}_j$	the ground truth of object j
z_j	the estimated truth of object j

Table 7.1: Notations and parameters in Chapter 7

7.3.2 The OpSTD Framework

This subsection presents the OpSTD framework. The semi-supervised truth discovery is formulated as an optimization problem. Based on the principle of truth discovery, ground truths is used to guide the source weight estimation that can in turn impact on the truths estimation for the objects whose ground truths are unknown. Following, it is the objective function of OpSTD that aims at minimizing the overall error between the estimated object truths and source's claims.

$$\begin{aligned} \min_{Z_u, A} f(Z_u, A) = & \sum_{j \in J_u} \Big\{ \sum_{i \in I_j} a_i (z_j - x_{ij})^2 \Big\} + \theta \sum_{j \in J_g} \Big\{ \sum_{i \in I_j} a_i (\bar{z}_j - x_{ij})^2 \Big\} \\ & \sum_{i \in I} \exp(-a_i) = 1 \end{aligned} \tag{7.1}$$

In Equation (7.1), I_j is the set of sources that observe object j. In the first term $\sum_{j \in J_u} \{ \sum_{i \in I_j} a_i (z_j - x_{ij})^2 \}$, for source i, $(z_j - x_{ij})^2$ models the estimated error made by i on the claims for object j, and it computes the discrepancy between the claims provided by sources and the estimated object truths. This term itself estimates the

source weights and object truths in an unsupervised manner. In order to minimize f, the optimization process will assign high weights to sources which make small estimated errors. Similarity, if the estimated error is large, it will assign a low weight to a_i to minimize the error's contribution in the objective function.

The second term $\sum_{j \in J_g} \{\sum_{i \in I_j} a_i (\bar{z}_j - x_{ij})^2\}$ introduces supervision into the objective function to supervise source weight and object truth estimation process. For a source i, $(\bar{z}_j - x_{ij})^2$ models the discrepancy between the ground truth and the source's claims for object j. It is the real error made by i for object j. To minimize the objective function, it penalizes the unreliable sources and assigns low weights to them if the real error is large. θ is a hyper parameter which balances these two terms in the objective function. Combining these two terms makes the proposed framework semi-supervised. The source weights are determined by both estimated errors and real errors, and the ground truths supervises object truths and source weights estimation. This will be further discussed in Section 7.3.3.

The constraint function, $\sum_{i \in I} \exp(-a_i) = 1$ is required mathematically to constrain the source weights between 0 and 1. Otherwise the source weights can be set to $-\infty$ to minimize the objective function.

7.3.3 The Iterative Solution

The object truths and source weights shall be learned jointly to minimize the objective function, and the optimal values learned after the optimization process will be selected as the object truths and source weights. In order to minimize the objective function f, block coordinate descent (Bertsekas, 1999) is adopted to optimize the object function. Block coordinate descent iteratively updates one set of variables while fixing the other set to keep reducing the value of f until reaching convergence. There are two steps involved to minimize function f. Step one is to update the estimated truths Z_u while

fixing the source weights A. Step two is to update the source weights A while fixing the estimated truths Z_u. These two steps can be mathematically formulated by Formulas (7.2) and (7.3). Next, it discusses in details on how to derive the rules to update source weights and estimated truths.

$$Z_u \leftarrow \arg\min_{Z_u} f(Z_u, A) \tag{7.2}$$

$$A \leftarrow \arg\min_{A} f(Z_u, A) \quad s.t. \quad \sum_{i \in I} \exp(-a_i) = 1 \tag{7.3}$$

Object truth update rule: In this step, it updates the set of estimated object truths Z_u while fixing A. By setting $\frac{df_A(Z_u)}{dz_j} = 0$ for the object $j \in J_u$, the update rule of the estimated object truth is:

$$z_j = \frac{\sum_{i \in I_j} a_i x_{ij}}{\sum_{i \in I_j} a_i} \tag{7.4}$$

Source weight update rule: The Lagrange multiplier approach is used to solve Formula (7.3). The Lagrangian can be formulated as:

$$\mathcal{L}(A, \lambda) = f(Z_u, A) + \lambda(\sum_{i \in I} \exp(-a_i) - 1) \tag{7.5}$$

where λ is the Lagrange multiplier. By setting $\frac{d\mathcal{L}(A,\lambda)}{da_i} = 0$, from the constraint it can derive that

$$\lambda \exp(-a_i) = \sum_{j \in J_{ui}} (z_j - x_{ij})^2 + \theta \sum_{j \in J_g} (\bar{z}_j - z_{ij})^2 \tag{7.6}$$

where J_{ui} and J_{gi} are both claimed by source i, but their ground truths are unknown and available respectively. Combined with the constraint equation $\sum_{i \in I} \exp(-a_i) = 1$, the Lagrange multiplier is computed as:

$$\lambda = \sum_{i \in I} \Big\{ \sum_{j \in J_{ui}} (z_j - x_{ij})^2 + \theta \sum_{j \in J_{gi}} (\bar{z}_j - x_{ij})^2 \Big\} \tag{7.7}$$

Plugging Equation (7.7) back to Equation (7.6), it can derive the source weight update rule in Equation (7.8).

$$a_i = -\log \Big(\frac{\sum_{j \in J_{ui}} (z_j - x_{ij})^2 + \theta \sum_{j \in J_{gi}} (\bar{z}_j - x_{ij})^2}{\sum_{i \in I} \Big\{ \sum_{j \in J_{ui}} (z_j - x_{ij})^2 + \theta \sum_{j \in J_{gi}} (\bar{z}_j - x_{ij})^2 \Big\}} \Big) \tag{7.8}$$

Discussion: From Equation (7.8) of the source weight update rule, it can be seen that a source has higher weight if it makes few errors among all the sources. Specifically, the errors are determined by the estimated errors and real errors, and the proportion can be adjusted by controlling θ. If increasing θ, the source weight will be computed mostly by the real errors. In the extreme case where $\theta = \infty$, the term $\sum_{j \in J_{ui}} (z_j - x_{ij})^2$ is ignored and the source weight is totally determined by objects with the ground truths. Conversely, if decreasing θ, the source weight will be computed mostly by the estimated errors. If $\theta = 0$, this is equivalent to the truth discovery in an unsupervised setting where the ground truths do not contribute to the truth discovery process and we estimate source weights solely from the observations.

From Equation (7.4) it can be seen that the estimated object truth is computed by weighted aggregation in which all the claims for object $j \in J_u$ contribute to the estimated truth, but the contribution is discounted by the weights of the sources which provide these claims. As a result, the estimated truth will be close to the claims from sources with high weights. Furthermore, the source weights are partially computed by ground truths as in Equation (7.8). Thus, the ground truths also impact the truths estimation for

the objects whose ground truths are unknown.

The algorithm flow of the OpSTD is summarized in Algorithm 7.1. First, the source weights are initialized. If no prior knowledge is available about the reliabilities of the sources, the source weights can be initialized uniformly, i.e. $a_i = -\log(\frac{1}{|I|})$. Otherwise the source weights can be changed accordingly to reflect the initial belief of the source reliability. Then the algorithm iteratively updates object truths and source weights by Equations (7.4) and (7.8) until convergence.

Algorithm 7.1: OpSTD Algorithm Flow

Input : Claims X, ground truths X_g^* for J_g
Output : Inferred object truths z_u

```
Initialize source weights;
repeat
    for j ∈ J_u do
        Update z_j by Equation (7.4);
    end
    for i ∈ I do
        Update a_i by Equation (7.8);
    end
until Convergence
return Z_u
```

7.4 Theoretical Analysis

This section theoretically analyzes the convergence property of the OpSTD algorithm and its time complexity.

7.4.1 Convergence Analysis

The following theorem shows that OpSTD algorithm converges, and it is valid to use block coordinate descent to minimize the objective function in Equation (7.1).

Theorem 7.1. *The iterative process in OpSTD algorithm converges, and the optimal solutions, Z_u and A, is a stationary point for the objective function in Equation (7.1) to attain minimum.*

Proof. There are two blocks of variables, Z_u and A, involved in the objective function f. We use $\mathcal{Y}$ to denote the union of the two blocks of variables, i.e. $\mathcal{Y} = \{Z_u, A\}$. Let the size of $\mathcal{Y}$ be l where $l = |Z_u| + |A|$. Then the optimization problem can be rewritten as:

$$\text{minimize} \quad f(y), \quad \text{s.t.} \quad y \in \mathcal{Y}$$

According to (Bertsekas, 1999), let $\{y^r\}$ be the sequence generated by the following rule:

$$y_k^{r+1} = \arg\min_{\xi \in \mathcal{Y}_k} f(y_1^{r+1}, \ldots, y_{k-1}^{r+1}, \xi, y_{k+1}^r, \ldots, y_l^r) \quad \text{for} \quad k = 1, 2, \ldots, l$$

where r is the current iterate index, then every limit point of y^r is a stationary point and $f(\{y^r\})$ is the global minimum of f if f satisfies the following two conditions:

1. f is continuously differentiable over $\mathcal{Y}$.

2. For each $y_k \in \mathcal{Y}_k$, $f(y_1, y_2, \ldots, y_{k-1}, \xi, y_{k+1}, \ldots, y_l)$, viewed as a function of ξ while the other variables are fixed, attains a unique minimum $\bar{\xi}$ over $\mathcal{Y}_k$, and is monotonically non-increasing in the interval from y_k to $\bar{\xi}$.

Next, it shows that the objective function f satisfies the two above conditions in the following two scenarios:

- Scenario 1: Update Z_u while fixing A. In this case, $f_A(Z_u)$ is a combination of quartic functions $a_i(z_j x_{ij})^2$ where $a_i > 0$. Hence, $f_A(Z_u)$ is strictly convex and continuously differentiable and attains a unique minimum.

- Scenario 2: Update A while fixing Z_u. In this case, $f_{Z_u}(A)$ is a combination of linear functions w.r.t a_i, which is affine, strictly convex and continuous differentiable. In addition, the exponential function is strictly convex, the constraint in the objective function is also strictly convex. Thus, $f_{Z_u}(A)$ is continuously differentiable and attains a unique minimum while fixing Z_u.

Therefore, Algorithm 7.1 converges when f attains its minimum $f(y^r)$, and $\{Z_u, A\} = \{y^r\}$ is the stationary point. □

7.4.2 Time Complexity Analysis

The time complexity of OpSTD algorithm is analyzed by analyzing the computational complexity of each iteration in Algorithm 7.1. In the object truth update step, each object can be claimed by up to $|I|$ sources. The cost of updating object truths is $O(|J_u| \times |I|)$ since this step computes the sum of claims weighted by source weights. In the source weight update step, each source can claim up to $|J|$ objects. The cost of updating source weight is $O(|J| \times |I|)$ since this step computes the squared error between each source's claims and truths. Therefore, the computational complexity of each iteration in OpSTD algorithm is $O(|J| \times |I|)$. In the truth discovery application, there are at most $|J| \times |I|$. Hence, the computational complexity of each iteration is also linear with the number of observations.

7.5 Experiments

This section presents the results of experiments that compare the proposed method with the state-of-art truth discovery methods on both real and synthetic datasets. All the experiments are conducted on a PC with Intel i7 processor and 16 GB RAM.

Method	Dataset					
	Weather		Stock		Gas Price	
	MAE	RMSE	MAE	RMSE	MAE	RMSE
OpSTD	**0.7274**	**1.1546**	**0.0038**	**0.0002**	**0.2264**	**0.0781**
SSTF	N/A	N/A	N/A	N/A	0.2613	0.1057
GTM	0.8196	1.5074	0.0044	0.0004	0.2502	0.0946
CRH	0.7829	1.4518	0.0046	0.0004	0.2525	0.0987
Mean	0.9524	2.2517	0.0128	0.004	0.3156	1.1514

Table 7.2: Accuracy Comparison

7.5.1 Experiment Setup

This subsection describe the setup of the experiments.

Baseline Methods

OpSTD is compared with the following state-of-art truth discovery methods: GTM (B. Zhao & Han, 2012), CRH (Q. Li, Li, Gao, Zhao et al., 2014), SSTF (Yin & Tan, 2011) and Mean. The descriptions of these methods can be seen in Section 2.5.

Datasets

Two real-world datasets. **Weather** and **Stock**, and one synthetic dataset, **Gas Price**, are used to evaluate OpSTD. Weather and Stock datasets are also used in the experiments in Chapter 4, the descriptions of these two methods can be found in Section 4.7.1. Gas Price dataset is used in the experiments in Chapter 3, its description and generation process can be found in Section 3.7.1.

Performance Metrics

The accuracy of truth discovery methods are evaluated by MAE and RMSE (see Section 2.4). The efficiency of truth discovery methods are evaluated by running time.

7.5.2 Performance Comparison

This section reports the performance evaluation for OpSTD against the baseline methods on the three datasets. For weather and stock datasets, since the ground truths are not among the claims, it does not satisfy the condition of SSTF, SSTF is not able to estimate object truths for these two datasets. For each dataset, 20% objects are randomly chosen and the ground truths of these objects in the truth discovery process, the ground truths of the rest objects are only used for evaluation.

Accuracy Comparison

The experiment results conducted on the three datasets in terms of accuracy are summarized in Table 7.2. As shown in the table, OpSTD consistently achieves the best accuracy in terms of MAE and RMSE. Among all the methods, Mean performs worst because it simply takes the average of claims for each object as truth, which does not take source reliabilities into consideration. Compared with GTM and CRH, OpSTD's error is reduced ranging from 7% - 14% in terms of MAE and 17% - 50% in terms of RMSE over the three datasets. The reason is that these two methods explore an unsupervised approach which does not use ground truths in the truth discovery process. Therefore, their errors are larger compared to OpSTD. OpSTD also outperforms the semi-supervised method SSTF. Note that SSTF's accuracy is even lower than GTM and CRH even if it uses ground truths to estimate object truths. This is because its algorithm is designed for handling categorical data and it runs poorly on continuous data scenarios.

Efficiency

The experiment results conducted on the three datasets in terms of running times are summarized in Table 7.3. From this table, it can be seen that Mean achieves the optimal

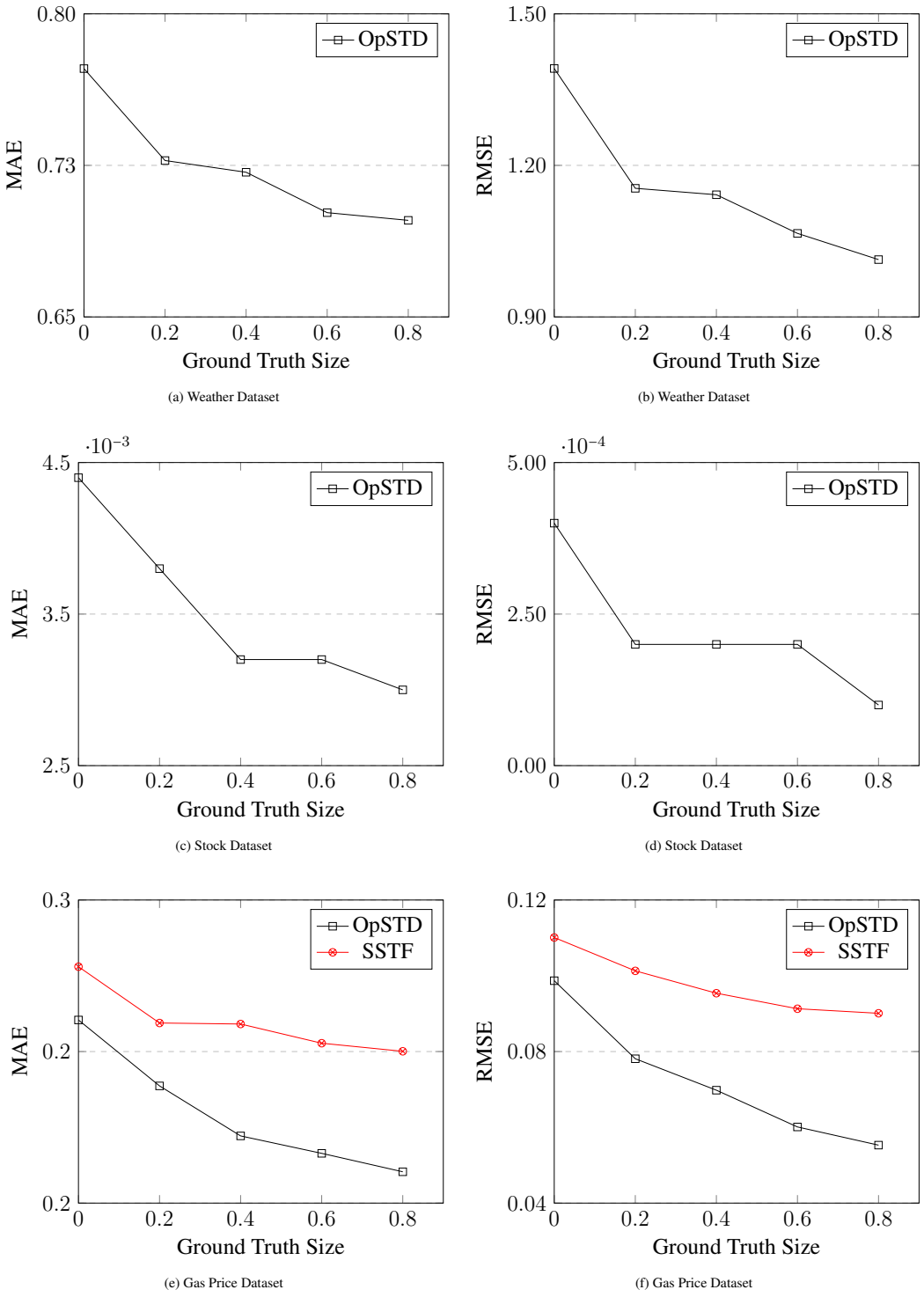
	Dataset		
Method	Weather	Stock	Gas Price
OpSTD	0.245	0.371	0.125
SSTF	N/A	N/A	7.129
GTM	0.277	0.453	0.173
CRH	0.283	0.409	0.151
Mean	0.031	0.04	0.019

Table 7.3: Running Times (Second(s))

efficiency. This is because Mean ignores source reliabilities estimation and it outputs mean of observations as truths directly. Among the baseline methods, OpSTD runs about 10% faster than GTM and CRH over the three datasets. The reason is that OpSTD uses 20% ground truths as its input and it estimates the truths for the rest 80% objects, while GTM and CRH discovers truths for the whole dataset. Compared with SSTF, OpSTD runs 57 times faster, which demonstrates the superiority of OpSTD for truth finding with ground truths.

7.5.3 Sensitivity Analysis

This subsection presents the experiments on testing the effect of ground truth size and θ to the accuracy of the proposed OpSTD. The effect of ground truth size to the accuracy of OpSTD is tested first. The θs are fixed at 20, 15 and 35 for weather, stock and gas price datasets, respectively.The size of the available ground truth is varied over the whole dataset from 0 to 0.8 with the step of 0.2. Since the accuracy of SSTF is also sensitive to the ground truth size, SSTF is also tested with different sizes of ground truths on gas price dataset in this experiment. The experiment result is plotted in Figure 7.1. From Figure 7.1, on one hand, it can be seen that both MAE and RMSE are high for all the three datasets when ground truth size is 0. This is the case when no ground truth is used in OpSTD and its accuracy is the same as CRH. As the ground truths are involved in the proposed truth discovery process, the errors begin to drop; on the other



(a) Weather Dataset

(b) Weather Dataset

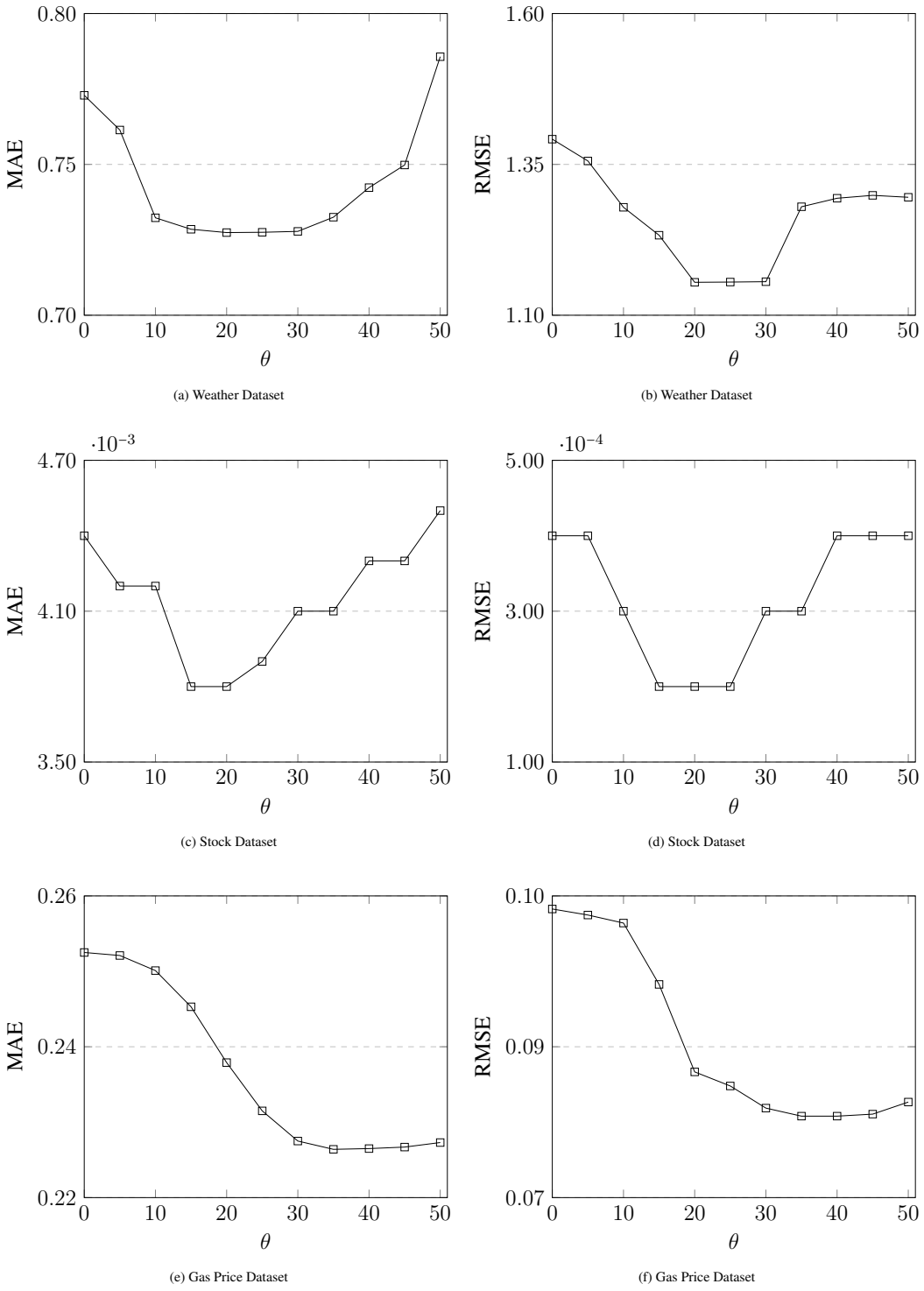
(c) Stock Dataset

(d) Stock Dataset

(e) Gas Price Dataset

(f) Gas Price Dataset

Figure 7.1: Effects of Ground Truth Size to MAE and $RMSE$



(a) Weather Dataset

(b) Weather Dataset

(c) Stock Dataset

(d) Stock Dataset

(e) Gas Price Dataset

(f) Gas Price Dataset

Figure 7.2: Effects of θ to MAE and $RMSE$

hand, it can also be seen that the errors are inverse proportional to the size of ground truths. This demonstrates that the ground truth indeed benefits the truth estimation in OpSTD. From Figures 7.1(e) and 7.1(f) it can also be observed that OpSTD outperforms SSTD in terms of MAE and RMSE under all ground truth sizes. This shows that OpSTD can utilize ground truths better for truth discovery tasks with continuous object truths.

The effect of θ to the accuracy of the proposed method is plotted in Figure 7.2. In this experiment, the ground truth size is fixed at 0.2 and θ is varied from 0 to 50. From this Figure 7.2, it can be seen that the errors begin to decrease when θs begin to increase from 0 and reach the optimal error very soon. Being different from the ground truth size, when increasing θ, the errors also begin to increase after it reaches the optimal ones. The reason is that as the θ is increased, the real errors become significant and it dominates the estimated errors in Equation (7.8). This may cause the estimated source weights overfit the objects whose ground truths are available, but less general to the rest 80% objects whose object truths are estimated. Given different datasets having different distribution and characteristics, θ is sensitive to OpSTD and we use the best θ to achieve the optimal performance.

In summary, ground truth, even a small set of ground truth, are beneficial for truth discovery. By effectively incorporating ground truths into the proposed method, the accuracy can be improved significantly. When ground truth size is small, θ is sensitive to different datasets and can be tuned to achieve optimal results.

7.6 Conclusion

In this chapter, semi-supervised truth discovery method for continuous object truths is investigated. The truth discovery problem is formulated as an optimization task in which object truths and source weights are modeled as unknown variables, and the ground truths is formulated as a regularization term to reinforce the source weights.

An iterative solution is developed to estimate object truths and source weights and its convergence property and time complexity are analyzed. A series of experiments is conducted to demonstrate that the proposed method outperforms the existing truth discovery methods in terms of both accuracy and efficiency.

The models developed in this chapter aims at answering Research Question 4. The work introduced in this chapter has been published in (Yang et al., 2018).

Chapter 8

Conclusion

This chapters summarizes the truth discovery models developed in this thesis, as well as address the limitations and the future work directions. This thesis presents five truth discovery models that address different aspects of truth discovery problems in different applications.

8.1 Summary of Thesis Contribution

There are five truth discovery models developed in this thesis.

8.1.1 Capturing Object Correlation in a Dynamic Truth Discovery Environment

Using object correlation is studied in Chapter 3. I proposed a chain graph based framework, Probabilistic Truth Discovery with Object Correlation (PTDCorr), in which source reliabilities, sources' claims and object truths are modeled as random variables. The correlation among objects are modeled as Markov Random Field in the probabilistic model, and the proposed PTDCorr model aimed at answering Research Question 1.1. Due to the modeling of object correlations, the influences of reliable sources can be

propagated to their neighbors in the chain graph model. This significantly improves the truth discovery accuracy when there are some objects claimed by few sources.

Based on PTDCorr, I developed iPTDCorr, which is the incremental version of PTDCorr that can efficiently estimate object truths over data streams, and the proposed iPTDCorr model aimed at answering Research Question 1.2. The novelty of iPTDCorr is that it can further use the temporal correlation among objects and it is able to estimate object truths at the present timestamp without re-processing the historical data.

8.1.2 Improving Accuracy and Efficiency for Truth Discovery over Data Streams

This thesis further studies how to improve both accuracy and efficiency for truth discovery over data streams. Chapter 4 presented Dynamic Source Weight Computation (DSWC) truth discovery algorithm that can apply many existing iterative truth discovery algorithms to stream data applications in order to improve both accuracy and efficiency. DSWC was proposed to answer Research Question 2. DSWC allows the users to set an error threshold, it uses the source weights computed at the previous timestamp to approximate the object truths at present if the error can be limited under the given threshold. Thus, the iterative source weight computation steps can be avoided and high efficiency can be achieved.

8.1.3 Modeling Task Difficulty and Worker Guessing Behavior

This thesis also specifically studies truth discovery in crowdsourcing applications for single-choice crowdsourcing tasks. In the context of crowdsourcing, the object is a task and the source is a worker. As the tasks have different difficulties and workers may guess a label if they do not know the truth of some tasks, Chapter 5 presented a probabilistic generative model CTDGD that jointly models task difficulty and worker's

guessing behaviors in the truth discovery step. The proposed CTDGD model aimed at answering Research Question 3.1.

8.1.4 Impact of Choice Confusion Degrees to the Workers

The choices of a crowdsourcing task may bring different levels of confusions to a crowd worker. In Chapter 6, I argued that choice confusion degrees determine task difficulty. Thus, I proposed a probabilistic generative truth discovery model CTI that considers choice confusion degrees in the truth inference process. By modeling choice confusion degrees, the accuracy of crowdsourcing truth inference is improved and it is verified by experiments. The CTI model was proposed to answer Research Question 3.2.

8.1.5 Incorporating Partially Observed Ground Truths

Many existing truth discovery methods are unsupervised learning models in which the object truths and source weights are both unknown *a priori*. It is infeasible and very expensive to obtain the ground truths for all the objects, but it is sometimes practical to acquire a small set of ground truths for some objects. Chapter 7 discussed how to use the partially observed ground truths to guide source weight estimation. In Chapter 7, I presented a semi-supervised truth discovery method, OpSTD, for continuous object truths, and the OpSTD model was proposed to answer Research Question 4. In OpSTD, the truth discovery problem is formulated as an optimization task which needs to minimize an objective function in order to minimize the overall error between sources' claims and object truths. The partially observed ground truths are modeled as an regularization term in the objective function and the strength of this regularization term can be tuned freely in different applications.

8.2 Research Limitations and Future Works

This section addresses some limitations of the developed methods in this thesis and discuss the future works.

- The PTDCorr model presented in Chapter 3 treats the object correlation information as an input. The correlation is captured by a function and it is application dependent. Thus, it requires domain knowledge to pre-define the correlation function before the truth discovery is conducted on the dataset. In the future work I would like to dive deeper into this problem and aim at developing a truth discovery method that can infer object correlations from the data instead of using it as an input.

- The DSWC algorithm developed in Chapter 4 assumes the object truth is computed by weighted aggregation. A more general method is required to be developed in the future which can be applied to a wider range of iterative methods.

- The CTDGD algorithm developed in Chapter 5 assumes the work guesses a label randomly if she does not know the truth. However, some workers may choose some more sophisticated guessing methods and this needs to be explored in the future work.

Besides, each truth discovery method either developed in this thesis or proposed by other researches has different kinds of assumptions. For example, all the methods developed in this thesis have the source dependency assumption. There is no single method that work well on all the exiting truth discovery datasets because the real-world problems are very complex and some assumptions are usually not held. Truth discovery is still at the emerging phase, more sophisticated truth discovery methods are required to relax more assumptions in the existing methods and can be applied to more complex real-world applications.

References

Aydin, B. I., Yilmaz, Y. S. & Demirbas, M. (2017). A crowdsourced "who wants to be a millionaire?" player. *Concurrency and Computation: Practice and Experience*, e4168.

Aydin, B. I., Yilmaz, Y. S., Li, Y., Li, Q., Gao, J. & Demirbas, M. (2014). Crowdsourcing for multiple-choice question answering. In *Aaai* (pp. 2946–2953).

Beck, A. & Tetruashvili, L. (2013). On the convergence of block coordinate descent type methods. *SIAM journal on Optimization*, *23*(4), 2037–2060.

Bertsekas, D. P. (1999). *Nonlinear programming*. Athena scientific Belmont.

Dawid, A. P. & Skene, A. M. (1979). Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, *28*(1), 20–28.

Demartini, G., Difallah, D. E. & Cudré-Mauroux, P. (2012). Zencrowd: leveraging probabilistic reasoning and crowdsourcing techniques for large-scale entity linking. In *Proceedings of the 21st international conference on world wide web* (pp. 469–478).

Dong, X. L., Berti-Equille, L., Hu, Y. & Srivastava, D. (2010). Global detection of complex copying relationships between sources. *Proceedings of the VLDB Endowment*, *3*(1-2), 1358–1369.

Dong, X. L., Berti-Equille, L. & Srivastava, D. (2009a). Integrating conflicting data: the role of source dependence. *Proceedings of the VLDB Endowment*, *2*(1), 550–561.

Dong, X. L., Berti-Equille, L. & Srivastava, D. (2009b). Truth discovery and copying detection in a dynamic world. *Proceedings of the VLDB Endowment*, *2*(1), 562–573.

Dong, X. L., Gabrilovich, E., Murphy, K., Dang, V., Horn, W., Lugaresi, C., ... Zhang, W. (2015). Knowledge-based trust: Estimating the trustworthiness of web sources. *arXiv preprint arXiv:1502.03519*.

Gadiraju, U., Kawase, R., Dietze, S. & Demartini, G. (2015). Understanding malicious behavior in crowdsourcing platforms: The case of online surveys. In *Proceedings of the 33rd annual acm conference on human factors in computing systems* (pp. 1631–1640).

Galland, A., Abiteboul, S., Marian, A. & Senellart, P. (2010). Corroborating information from disagreeing views. In *Proceedings of the third acm international conference on web search and data mining* (pp. 131–140).

Ghosh, A., Kale, S. & McAfee, P. (2011). Who moderates the moderators?: crowdsourcing abuse detection in user-generated content. In *Proceedings of the 12th acm conference on electronic commerce* (pp. 167–176).

Gupta, A., Lamba, H., Kumaraguru, P. & Joshi, A. (2013). Faking sandy: characterizing and identifying fake images on twitter during hurricane sandy. In *Proceedings of the 22nd international conference on world wide web* (pp. 729–736).

Han, J., Pei, J. & Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Ipeirotis, P. G., Provost, F. & Wang, J. (2010). Quality management on amazon mechanical turk. In *Proceedings of the acm sigkdd workshop on human computation* (pp. 64–67).

Karger, D. R., Oh, S. & Shah, D. (2011). Iterative learning for reliable crowdsourcing systems. In *Advances in neural information processing systems* (pp. 1953–1961).

Kim, H.-C. & Ghahramani, Z. (2012). Bayesian classifier combination. In *Artificial intelligence and statistics* (pp. 619–627).

Koller, D. & Friedman, N. (2009). *Probabilistic graphical models: principles and techniques*. MIT press.

Kurve, A., Miller, D. J. & Kesidis, G. (2014). Multicategory crowdsourcing accounting for variable task difficulty, worker skill, and worker intention. *IEEE Transactions on Knowledge and Data Engineering*, *27*(3), 794–809.

Le, H., Wang, D., Ahmadi, H., Uddin, Y. S., Szymanski, B., Ganti, R. & Abdelzaher, T. (2011). Distilling likely truth from noisy streaming data with apollo. In *Proceedings of the 9th acm conference on embedded networked sensor systems* (pp. 417–418).

Li, F., Lee, M. L. & Hsu, W. (2014). Entity profiling with varying source reliabilities. In *Proceedings of the 20th acm sigkdd international conference on knowledge discovery and data mining* (pp. 1146–1155).

Li, H., Zhao, B. & Fuxman, A. (2014). The wisdom of minority: Discovering and targeting the right group of workers for crowdsourcing. In *Proceedings of the 23rd international conference on world wide web* (pp. 165–176).

Li, Q., Li, Y., Gao, J., Su, L., Zhao, B., Demirbas, M., ... Han, J. (2014). A confidence-aware approach for truth discovery on long-tail data. *Proceedings of the VLDB Endowment*, *8*(4), 425–436.

Li, Q., Li, Y., Gao, J., Zhao, B., Fan, W. & Han, J. (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* (pp. 1187–1198).

Li, T., Gu, Y., Zhou, X., Ma, Q. & Yu, G. (2017). An effective and efficient truth discovery framework over data streams. In *Edbt* (pp. 180–191).

Li, X., Dong, X. L., Lyons, K., Meng, W. & Srivastava, D. (2012). Truth finding on the deep web: Is the problem solved? In *Proceedings of the vldb endowment* (Vol. 6, pp. 97–108).

Li, Y., Du, N., Liu, C., Xie, Y., Fan, W., Li, Q., ... Sun, H. (2017). Reliable medical diagnosis from crowdsourcing: Discover trustworthy answers from non-experts. In *Proceedings of the tenth acm international conference on web search and data*

mining (pp. 253–261).

Li, Y., Gao, J., Meng, C., Li, Q., Su, L., Zhao, B., ... Han, J. (2016). A survey on truth discovery. *ACM Sigkdd Explorations Newsletter*, *17*(2), 1–16.

Li, Y., Li, Q., Gao, J., Su, L., Zhao, B., Fan, W. & Han, J. (2015). On the discovery of evolving truth. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining* (pp. 675–684).

Li, Y., Li, Q., Gao, J., Su, L., Zhao, B., Fan, W. & Han, J. (2016). Conflicts to harmony: A framework for resolving conflicts in heterogeneous data by truth discovery. *IEEE Transactions on Knowledge and Data Engineering*, *28*(8), 1986–1999.

Li, Z., Han, J., Yu, Y. et al. (2016). Aggregating crowd wisdom with instance grouping methods. In *Asia-pacific web conference* (pp. 468–479).

Liu, W., Liu, J., Duan, H., Hu, W. & Wei, B. (2017). Exploiting source-object networks to object conflicts in linked data. In *European semantic web conference* (pp. 53–67).

Ma, F., Li, Y., Li, Q., Qiu, M., Gao, J., Zhi, S., ... Han, J. (2015). Faitcrowd: Fine grained truth discovery for crowdsourced data aggregation. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining* (pp. 745–754).

Marshall, J., Syed, M. & Wang, D. (2016). Hardness-aware truth discovery in social sensing applications. In *Distributed computing in sensor systems (dcoss), 2016 international conference on* (pp. 143–152).

Meng, C., Jiang, W., Li, Y., Gao, J., Su, L., Ding, H. & Cheng, Y. (2015). Truth discovery on crowd sensing of correlated entities. In *Proceedings of the 13th acm conference on embedded networked sensor systems* (pp. 169–182).

Mukherjee, S., Weikum, G. & Danescu-Niculescu-Mizil, C. (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* (pp. 65–74).

Ouyang, R. W., Srivastava, M., Toniolo, A. & Norman, T. J. (2015). Truth discovery in crowdsourced detection of spatial events. *IEEE transactions on knowledge and data engineering*, *28*(4), 1047–1060.

Pal, A., Rastogi, V., Machanavajjhala, A. & Bohannon, P. (2012). Information integration over time in unreliable and uncertain environments. In *Proceedings of the 21st international conference on world wide web* (pp. 789–798).

Pasternack, J. & Roth, D. (2010). Knowing what to believe (when you already know something). In *Proceedings of the 23rd international conference on computational linguistics* (pp. 877–885).

Pasternack, J. & Roth, D. (2013). Latent credibility analysis. In *Proceedings of the 22nd international conference on world wide web* (pp. 1009–1020).

Pochampally, R., Das Sarma, A., Dong, X. L., Meliou, A. & Srivastava, D. (2014). Fusing data with correlations. In *Proceedings of the 2014 acm sigmod international conference on management of data* (pp. 433–444).

Rasmussen, C. E. (2004). Gaussian processes in machine learning. In *Advanced lectures on machine learning* (pp. 63–71). Springer.

Raykar, V. C. & Yu, S. (2012). Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *Journal of Machine Learning Research*, *13*(Feb), 491–518.

Raykar, V. C., Yu, S., Zhao, L. H., Valadez, G. H., Florin, C., Bogoni, L. & Moy, L. (2010). Learning from crowds. *Journal of Machine Learning Research*, *11*(Apr), 1297–1322.

Su, L., Li, Q., Hu, S., Wang, S., Gao, J., Liu, H., ... others (2014). Generalized decision aggregation in distributed sensing systems. In *2014 ieee real-time systems symposium* (pp. 1–10).

Venanzi, M., Guiver, J., Kazai, G., Kohli, P. & Shokouhi, M. (2014). Community-based bayesian aggregation models for crowdsourcing. In *Proceedings of the 23rd international conference on world wide web* (pp. 155–164).

Wang, D., Abdelzaher, T. & Kaplan, L. (2015). *Social sensing: building reliable systems on unreliable data*. Morgan Kaufmann.

Wang, D., Abdelzaher, T., Kaplan, L. & Aggarwal, C. C. (2013). Recursive fact-finding: A streaming approach to truth estimation in crowdsourcing applications. In *Distributed computing systems (icdcs), 2013 ieee 33rd international conference on* (pp. 530–539).

Wang, D., Abdelzaher, T., Kaplan, L., Ganti, R., Hu, S. & Liu, H. (2013). Exploitation of physical constraints for reliable social sensing. In *Real-time systems symposium (rtss), 2013 ieee 34th* (pp. 212–223).

Wang, D., Kaplan, L. & Abdelzaher, T. F. (2014). Maximum likelihood analysis of conflicting observations in social sensing. *ACM Transactions on Sensor Networks (ToSN)*, *10*(2), 30.

Wang, S., Su, L., Li, S., Hu, S., Amin, T., Wang, H., ... Abdelzaher, T. (2015). Scalable social sensing of interdependent phenomena. In *Proceedings of the 14th international conference on information processing in sensor networks* (pp. 202–213).

Wang, S., Wang, D., Su, L., Kaplan, L. & Abdelzaher, T. F. (2014). Towards cyber-physical systems in social spaces: The data reliability challenge. In *2014 ieee real-time systems symposium* (pp. 74–85).

Wang, X., Sheng, Q. Z., Yao, L., Li, X., Fang, X. S., Xu, X. & Benatallah, B. (2016a). Empowering truth discovery with multi-truth prediction. In *Proceedings of the 25th acm international on conference on information and knowledge management* (pp. 881–890).

Wang, X., Sheng, Q. Z., Yao, L., Li, X., Fang, X. S., Xu, X. & Benatallah, B. (2016b). Truth discovery via exploiting implications from multi-source data. In *Proceedings of the 25th acm international on conference on information and knowledge management* (pp. 861–870).

Wang, Y., Ma, F., Su, L. & Gao, J. (2017). Discovering truths from distributed data. In *2017 ieee international conference on data mining (icdm)* (pp. 505–514).

Welinder, P., Branson, S., Perona, P. & Belongie, S. J. (2010). The multidimensional wisdom of crowds. In *Advances in neural information processing systems* (pp. 2424–2432).

Whitehill, J., Wu, T.-f., Bergsma, J., Movellan, J. R. & Ruvolo, P. L. (2009). Whose vote should count more: Optimal integration of labels from labelers of unknown expertise. In *Advances in neural information processing systems* (pp. 2035–2043).

Wu, C. J. (1983). On the convergence properties of the em algorithm. *The Annals of statistics*, 95–103.

Yang, Y., Bai, Q. & Liu, Q. (2018). On the discovery of continuous truth: A semi-supervised approach with partial ground truths. In *International conference on web information systems engineering* (pp. 424–438).

Yang, Y., Bai, Q. & Liu, Q. (2019a). Dynamic source weight computation for truth inference over data streams. In *Proceedings of the 18th international conference on autonomous agents and multiagent systems* (pp. 277–285).

Yang, Y., Bai, Q. & Liu, Q. (2019b). Modeling random guessing and task difficulty for truth inference in crowdsourcing. In *Proceedings of the 18th international conference on autonomous agents and multiagent systems* (pp. 2288–2290).

Yang, Y., Bai, Q. & Liu, Q. (2019c). A probabilistic model for truth discovery with object correlations. *Knowledge-Based Systems*, *165*, 360–373.

Yao, L., Su, L., Li, Q., Li, Y., Ma, F., Gao, J. & Zhang, A. (2018). Online truth discovery on time series data. In *Proceedings of the 2018 siam international conference on data mining* (pp. 162–170).

Yin, X., Han, J. & Philip, S. Y. (2008). Truth discovery with multiple conflicting information providers on the web. *IEEE Transactions on Knowledge and Data Engineering*, *20*(6), 796–808.

Yin, X. & Tan, W. (2011). Semi-supervised truth discovery. In *Proceedings of the 20th international conference on world wide web* (pp. 217–226).

Yu, D., Huang, H., Cassidy, T., Ji, H., Wang, C., Zhi, S., ... Magdon-Ismail, M. (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* (pp. 1567–1578).

Yuen, M.-C., King, I. & Leung, K.-S. (2011). A survey of crowdsourcing systems. In *2011 ieee third international conference on privacy, security, risk and trust and 2011 ieee third international conference on social computing* (pp. 766–773).

Zhao, B. & Han, J. (2012). A probabilistic model for estimating real-valued truth from conflicting sources. *Proc. of QDB*.

Zhao, B., Rubinstein, B. I., Gemmell, J. & Han, J. (2012). A bayesian approach to discovering truth from conflicting sources for data integration. *Proceedings of the VLDB Endowment*, *5*(6), 550–561.

Zhao, Z., Wei, F., Zhou, M., Chen, W. & Ng, W. (2015). Crowd-selection query processing in crowdsourcing databases: A task-driven approach. In *Edbt* (pp. 397–408).

Zheng, Y., Li, G. & Cheng, R. (2016). Docs: a domain-aware crowdsourcing system using knowledge bases. *Proceedings of the VLDB Endowment*, *10*(4), 361–372.

Zhi, S., Zhao, B., Tong, W., Gao, J., Yu, D., Ji, H. & Han, J. (2015). Modeling truth existence in truth discovery. In *Proceedings of the 21th acm sigkdd international*

conference on knowledge discovery and data mining (pp. 1543–1552).