CLC_____ Number_____

UDC_____ Available for reference ☐ Yes ☐ No

A Dissertation for Bachelor's Degree

Thesis Title:MULTI-SOURCES TRUTH DISCOVERY WITH PROBABILISTIC MODEL

Student Name: YI SHANGRU

Student ID: 11510099

Department: Electrical and Electric Engineering

Program: Information Engineering

Thesis Advisor: Prof. Bo Tang

Date: May 16, 2019

COMMITMENT OF HONESTY
诚信承诺书

1. I solemnly promise that the paper presented comes from my independent research work under my supervisor's supervision. All statistics and images are real and reliable.

2. Except for the annotated reference, the paper contents no other published work or achievement by person or group. All people making important contributions to the study of the paper have been indicated clearly in the paper.

3. I promise that I did not plagiarize other people's research achievement or forge related data in the process of designing topic and research content.

4. If there is violation of any intellectual property right, I will take legal responsibility myself.

1. 本人郑重承诺所呈交的毕业设计（论文），是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料均真实可靠。

2. 除文中已经注明引用的内容外，本论文不包含任何其他人或集体已经发表或撰写过的作品或成果。对本论文的研究作出重要贡献的个人和集体，均已在文中以明确的方式标明。

3. 本人承诺在毕业论文（设计）选题和研究内容过程中没有抄袭他人研究成果和伪造相关数据等行为。

4. 在毕业论文（设计）中对侵犯任何方面知识产权的行为，由本人承担相应的法律责任。

Signature 签名: _____

Date 日期:_____

Contents

摘 要

在大数据时代的背景下，数据量面临着前所未有的爆炸性增长。由于信息来源的多样性和信息源的异构可信度，从冲突信息中推断出被预测对象的真实信息已经不再容易，导致真实发现成为一项用于解决冲突的有前途的技术。这项任务面临的最大挑战是信息源的可信度评估和高质量答案选择。但是，现有的工作无法得到更加精细的可信度，从而导致可能的性能损失。此外，现有模型在初始化中没有考虑信息源天然的差异，而是使用统一的参数。为了获得对于被预测对象的信息源可信度，我们整合领域专业知识并基于对象和信息源的相似性拓展成可信度度量，这个度量可以用作进一步的信息源筛选。我们还提出了合理的初始化方法。此外，我们提出了一种基于信息源属性的无监督的信息源可信度估计方法，用来实现快速的信息源可信度估计。在多真实信息背景下，先前工作中提出的置信得分也同样在模型中被考虑。我们提出了一个集成的两阶段模型，使用图形概率建模和贝叶斯方法来结合对象和信息源相似性和初始化，旨在找到可能的多个真理而无需任何监督。两个真实世界数据集的实验结果证明了我们提出的模型的可行性和有效性。

关键词： 真实发现, 贝叶斯模型, 相似度衡量

ABSTRACT

In the light of Big Data era, data quantities encounter unprecedented explosive growth. With various information sources and heterogeneous credibility, inferring truth from conflicting information is no longer easy, leading truth discovery serving as a promising technique to resolve conflicts. The most significant challenge for this task is the source reliability estimation and high-quality answer selection. However, existing works fail to identify the reliability in a refined level, leading possible performance loss. Also, natural differences of sources is not considered in the initialization, but using uniform parameters. To obtain object-oriented source qualities, we integrate domain expertise and extend reliability measure based on object-source similarity, which can be further used as metrics for source selection. We also propose a reasonable initialization method. Moreover, we propose an unsupervised method to enable fast source quality estimation based on source attributes. The confidence score in multi-truth context proposed in previous works is also considered. We propose an integrated two-stage model using graphical probability modeling and Bayesian approach to incorporate object-source similarity and initialization, aiming to find the possible multiple truths without any supervision. Experimental results on two real-world datasets demonstrate the feasibility and effectiveness of our proposed model.

Keywords: truth discovery, Bayesian method, similarity measurement

Notations

O	Set of objects
S	Set of sources
$object\ (o)$	Object to perform truth discovery
$\mathcal{DB}_{raw}$	Fundamental database
$attribute_o^f\ (a_o^f)$	Focusing attribute of object o
$attribute_o^n\ (a_o^n)$	Other attribute of object o
$\mathcal{DB}_{oa}$	Object attribute database
$source\ (s)$	Data provider
$attribute_s^m\ (a_s^m)$	A (statistical) attribute of source s
$\mathcal{DB}_{sa}$	Source attribute database
$V_s(o)$	Value set provided by source s for object o
$\sigma(v)$	Probability that a value is true
d^a	Domains set for a $attribute_o^n$
D_a	Domain set for all $attribute_o^n$
O^d	Object set for object that is associated with domain d
$O(s)$	Object set for object that source s claims value about
$\mathcal{DB}_t$	Inferred truth database
f	Fact generated from DB_{raw}
$\mathcal{DB}_f$	Fact database
c	Claim generated from DB_{raw}
$\mathcal{DB}_c$	Claim database
S^{re}	Output *recall* estimation of stage 1 for s in S
S^{sp}	Output *specificity* estimation of stage 1 for s in S

1. INTRODUCTION

In light of information era, the data quantities face the great explosion. Since the data providers are no longer under rigorous control to guarantee the absolute correctness of data, it is possible that there are conflicts between the answers for same query provided by different data sources. Thus it is a nontrivial task to identify or generate the correct answer from these possibly conflicting information. Consequently, how to perform a rational data integration for objects to infer the most trustworthy answer to the greatest extent becomes the key challenge. This challenge is termed as truth finding problem in [1].

One of the simplest and most straightforward strategy is to conduct majority-vote to select the most provided answer as the correct one. However, this simple strategy fails to take the reliability difference of sources into consideration. It is possible that the correct data is shaded by some erroneous ones provided by a bunch of vicious data sources. Moreover, this strategy tends to provide the most "trustworthy" one. It is acceptable when there is only one correct answer, e.g. the answer of a true-or-false question. However, if there are multiple correct answers, e.g. the authors of a book, this simple method could lead a great number of correct answers to be omitted.

Therefore, a better solution is to perform data integration for objects after evaluating the source reliability. Given the trustworthiness of sources as prior knowledge, contradictory data can be reconciled effectively by eliminating those data provided by untrustworthy sources. But it requires tons of laborious work to identify the source quality manually. Thus how to automatically infer the possible source reliability based on given data is a more perspective and attractive direction. Many works have been proposed to derive the correct answers with source quality estimated in the same time [1–7]. Meanwhile, some works consider the reliability of data as a sum of source reliability when performing truth inference [8, 9]. Some works also integrate some features or attributes of sources to the truth inference for better estimating source reliability [9]. Present truth inference problem have two different subsets, i.e. the single-truth and multi-truth. In general, there are two major differences between the single-truth and multi-truth inference. One of the differences is the source quality measurement and is first identified in [1], measuring source quality in two side, i.e. *recall* and *specificity*, instead of merely *precision* or *accuracy*. And the other difference is the answer confidence and is first examined in [7], identifying the partial rejection between answers, instead of previous total objection.

However, it is unfair to assign an uniform reliability for a source without considering its reliability difference in different data classification. None of sources is promised to demonstrate same reliability among different domains. This domain expertise difference is proposed and examined in a heuristic manner in [6]. Inspired by this work, we extend

the domain difference to a similarity measure between the sources and the objects when performing truth inference. Also, present works are initialized in a uniform manner, due to the lack of prior knowledge. Naturally, some sources are much better than the other sources and it is improper to set all the sources on the same starting line, which is intuitively performance deteriorating. Meanwhile, some source attributes have strong connection with the source reliability and they are easy to retrieve. It is desired that we can bridge these attributes to the source reliability estimation in an unsupervised manner. Moreover, for the final truth selection, present works select the truth based on a preset threshold, which is against the nature of truth selection under the context of multiple sources.

We summarize the major challenges of accomplishing above task as follows:

1. *Unknown quality of sources for different objects.* The source might demonstrate different reliability for different objects when providing claims. Since there is no quantitative method to capture this difference, it is a nontrivial task to measure that difference and plug it in truth inference model.

2. *Unknown relationship between the source quality and source attributes.* Measure the relationship between the source attributes and reliability is relatively straightforward when it comes to the supervised context. However, it is not the same case under unsupervised manner. Figure out a way to measure contribution of source attributes to its reliability could eliminate tons of work for collecting data, since we can approximate the source reliability using only several attribute values.

3. *Missing proper initialization to capture the nature reliability of sources.* Due to the lack of prior knowledge and difficulties in manually labeling, present works are started from a uniform prior, which is not consistent with the nature of sources. In some cases, this uniform initialization for source quality would lead to performance loss for truth inference.

4. *Missing quantitative criteria when selecting the good sources which are specific to an object.* It is common that we want to select some sources to predict values or answer query for some specific objects and we want the selected sources to be "good". The lack of quantitative measurement of how "good" a source for an object make this selection impossible.

In this thesis, we address above problems and propose a truth inference model with following listed points as our main contributions:

1. We recognize that a source might demonstrate different reliability for different objects when providing claims and develop a similarity measurement to address this problem.

2. We develop a method to bridge the source attributes to source reliability in unsupervised manner, which enable fast source quality estimation and truth inference initialization.

3. We identify the shortness of uniform initialization and propose an source-aware method for source reliability initialization in our truth inference model.

4. We propose a quantitative measurement of how "good" a source for an object and make the source selection for a specific object become possible.

In the following sections, the end-to-end data model and formal problem definition are given in Section 2. In Section 3, the way to model source quality and the integration of source attributes are given. Our proposed two-stage truth inference model is illustrated in details in Section 4. Section 5 presents the experiment parts. The discussion of related works is in Section 6 and conclusion is in Section 7.

2. PROBLEM FORMULATION

In general, a data source provides information about several attributes relating with objects. For each attributes, we separate several domains. For example, an movie website provides attribute information about *category* and *released year* for movies. For attribute *category*, it can be separated to several domains, such as *action*, *drama*, *comedy* and *romance*; For *released year*, it can be separated to several domains, such as "1971 to 1985" and "after 2010". The quality of information provided by a source, or namely the quality of the source may be different for each attribute, inferring the underlying independence. Thus, each attribute type may be considered individually. In this thesis, we assume the inter-independence between each attributes relating with objects and thus can be dealt with separately, in order to simplify the relation-aware problem. Also, the sources under consideration are assumed to be independent, i.e. without copying or correlation between each other. Aside from the attributes relating with objects, it is also proper to identify some attributes relating with sources and these attributes could have tight connection with the quality of the sources, rendering us help when inferring truths from conflicting data. For example, a movie website, providing information about movies, might have attributes, such as "daily visiting number" and "total sites links". These attributes are intuitively linked with the source quality.

We now illustrate the details of data model in our proposed solution and formally give definitions about the two-stage truth finding problem.

2.1 End-to-end Data Model

We now consider a single attribute type with multiple values and perform truth inference model to identify the claimed value is true or false, saying this attribute as *focusing attribute*. In this context, the considered attribute for book is *Author* and for movie is *Cast*. There are other attributes relating with objects, such as *Released Year* and *Category* for movie. We treat *focusing attribute* and other attributes in separate manner. The fundamental input data we consume is in the form of triples $(object, attribute_o^f, source)$, where *object* servers as the identifying key of an object, e.g. ISBN10 for a book, $attribute_o^f$ is one possible value of the multi-valued *focusing attribute* of an object, e.g. an author of a book, and *source* is the originate of this triples. Aside from the *focusing attribute*, the other object attributes are taken as input in the form of tuples $(object, attribute_o^1, ..., attribute_o^n)$, where $attribute_o^m$ is one of the other attributes provided for objects and is ready to be divided into domains, e.g. the *Category* of movies. Since we also consider the attributes of source that are (possibly) related with the source quality, there is additional input data in the

form of tuples $(source, attribute_s^1, ..., attribute_s^m)$, where $attribute_s^m$ is a numeric value of an source attribute, e.g. a website source's statistical value of "daily visiting number", if the source attributes are available. We also denote $V_s(o)$ as the set of values claimed by source s for object o. And one important definition for the thesis is introduced.

Definition 2.1.1. *The veracity score of a value v is the probability that this value is true, denoted as $\sigma(v)$.*

Moreover, we extract the set of sources, denoted as S, and the set of objects, denoted as O, from the fundamental database. Since there are domains for each $Attribute_o$ a_i, we thus generate a domain set for a_i, denoted as $d^{a_i} = \{d_1^{a_i}, d_2^{a_i}, ..., d_L^{a_i}\}$. We combine the generated domain sets together, denoted as $D_a = \{d^{a_1}, d^{a_2}, ..., d^{a_n}\}$. For each object o, it is associated with one or some domains for each domain set in the D_a and we denote O^d as the set of object in domain d.

The formal definitions of the data input and generated two tables are given as follows.

Definition 2.1.2. *Let $\mathcal{DB}_{raw} = \{data_1, data_2, ..., data_N\}$ be the fundamental database, where every* data *is in the format of (o, a_o^f, s), where o is the object, a_o^f is the value of focusing attribute, and s is the source. Each data is unique in the fundamental database.*

Table 2.1 is an example of a fundamental database.

Table 2.1: Fundamental database of movies

Object (Movie)	$Attribute_o^f$ *(Cast)*	*Source (Website)*
Harry Potter	Daniel Radcliffe	IMDB
Harry Potter	Emma Waston	IMDB
Harry Potter	Rupert Grint	IMDB
Harry Potter	Daniel Radcliffe	Filmcrave
Harry Potter	Johnny Depp	BadSource.com
Harry Potter	Daniel Radcliffe	BadSource.com
Harry Potter	Emma Waston	BadSource.com
500 Days with Summer	Joseph Gordon-Levitt	Top250tv
...	...	...

Definition 2.1.3. *Let $\mathcal{DB}_{oa} = \{oa_1, oa_2, ..., oa_{OA}\}$ be the object attribute database, where every* oa *is in the format of $(o, a_o^1, a_o^2, ..., a_o^n)$, where o is the object and a_o^n is the value of an object attribute. Each o is unique in the object attribute database.*

Table 2.2 is an example of an object attribute database.

Definition 2.1.4. *Let $\mathcal{DB}_{sa} = \{sa_1, sa_2, ..., sa_{SA}\}$ be the source attribute database, where every* sa *is in the format of $(s, a_s^1, a_s^2, ..., a_s^m)$, where s is the source and a_s^m is the value of an source attribute. Each s is unique in the source attribute database.*

Table 2.2: Object attribute database of movies

Object (Movie)	$Attribute_o^1$ *(Category)*	...	$Attribute_o^n$ *(Released Year)*
Harry Potter	(drama, magic)	...	2001
500 Days with Summer	(romance)	...	2009
...	...	...	...

Table 2.3: Additional attribute database of sources

Source (Website)	$Attribute_s^1$*(#Total Link-in)*	...	$Attribute_s^m$*(Country Rank)*
IMDB	216672	...	27
Letterboxd	1074	...	1807
...	...	...	...

Table 2.3 is an example of a source attribute database.

Once finish defining the input data model, we now turn to the data model of output. The output data is table containing all focusing attribute values given by sources, i.e. selecting unique *focusing attribute* value from fundamental database and grouping by $object$. Since we aim to perform truth inference to identify whether the *focusing attribute* value is True or False, each row in this table are associated with a Boolean value to indicate whether it is True or False after performing truth inference model. This indication has no relationship with the actual True or False and the goal of our model is to approximate the true situation as close as possible. During following model iteration, we use the intermediate inferred truth table to guide the update process and output a final inferred truth table as output. The model evaluation is performed by comparing the final inferred truth table and the labeled truth, i.e. the true representation. The formal definition of output data and generated table are given as follows.

Definition 2.1.5. *Let $\mathcal{DB}_t$ = {t_1, t_2, ..., t_T} be the inferred truth database, where every* t *is in the format of (o, a_o^f, t), where o is the object, a_o^f is a value of the focusing attribute and t is the inferred truth as a Boolean value. Each combination of o and a_o^f is unique in the inferred truth database.*

Table 2.4 is an example of an inferred truth database.

2.2 Problem Definition

Given the end-to-end data model, we now define the target problem of this thesis.

Truth inference The input raw database $\mathcal{DB}_{raw}$ is utilized to infer the truth and generate inferred truth database $\mathcal{DB}_t$.

Source quality inference In the process of truth inference, the quality of each source presenting in the $\mathcal{DB}_{raw}$ is estimated and thus rendering help when measure the reliability of

Table 2.4: Inferred truth database

Object (Movie)	$Attribute_o^f$ *(Cast)*	Truth
Harry Potter	Rupert Grint	True
Harry Potter	Emma Waston	True
Harry Potter	Daniel Radcliffe	True
Harry Potter	Johnny Depp	False
...	...	...

provided information. The source quality reflects how reliable of the source, given the *focusing attribute*. By estimating and comparing source quality, good and creditable sources can be selected and thus produce more accurate information.

These two target problems are tightly connected with each other. In fact, the truth and source quality are inferred simultaneously, given that the source quality is estimated based on the correctness of given information and the information is evaluated based on its provided source.

In next section, metrics of evaluating source quality are discussed. Subsequently, the details of the truth inference and source quality estimation in our proposed model are explained.

3. SOURCE QUALITY MODELING

In this section, we examine our proposed model in the way to model source quality and the intuitions behind this model. We present the metric for source quality firstly and discuss the shortage of single metric measurement. As for compensation, the two-side quality measurement, proposed in [1] and then extended by [6], is introduced next. Inspired by [9], some attributes assumed to be related with source quality are included in the proposed model for source quality. In order to perform refined truth discovery, the relationship model between sources and the inference objects is also discussed, preparing for the qualification of source reliability contribution.

3.1 Quality Metric Review

In classical document retrieval or information search, there is a universally acknowledged quality metric, considering the correctness and coverage of provided information or retrieved document. Given the fundamental database, for a specific source, we can select all the objects that it provide information and generate an object set. Furthermore, for this object set, we can extract all the a_o^f linked with the objects in this object set. Once the ground truth, i.e. whether a a_o^f is true, is available, we can count the value provided by source is correct or not and generate confusion matrix of this source in **Table 3.1**.

Table 3.1: Confusion matrix of source s

	a_o^f correct	a_o^f wrong
s provide	True Positives (TP_s)	False Positives (FP_s)
s not provide	False Negatives (FN_s)	True Negatives (TN_s)

- *Precision* of source *s* is the fraction of its provided a_o^f being correct, i.e., $\frac{TP_s}{TP_s+FP_s}$.
- *Accuracy* of source *s* is the fraction of its information, including the provided and not provided ones, being correct, i.e., $\frac{TP_s+TN_s}{TP_s+FN_s+FP_s+TN_s}$.
- *Recall* of source *s* is the fraction of its provided true a_o^f among all the true a_o^f, i.e., $\frac{TP_s}{TP_s+FN_s}$. And $1-$ *recall* is known as false negative rate.
- *Specificity* of source *s* is the fraction of its not provided false a_o^f among all the false a_o^f, i.e., $\frac{TN_s}{FP_s+TN_s}$. And $1-$ *specificity* is known as false positive rate.

Table 3.2 demonstrates the calculated source quality metrics based on the **Table 2.1** and **Table 2.4**.

Table 3.2: Quality of sources based on **Table 2.1** and **Table 2.4**

Metric	IMDB	BadSource.com	Filmcrave
TP	3	2	1
FP	0	1	0
FN	0	1	2
TN	1	0	1
Precision	1	2/3	1
Accuracy	1	1/2	1/2
Recall	1	2/3	1/3
Specificity	1	0	1

3.2 Single Metric Measurement

The multi-value truth inference model is originally extended from the single-value version. Previous studies and proposed works focus on the data integration or fusion, retrieving or generating a single-value truth from conflicting information. Within this part, we discuss the shortages of metrics, utilized for previous single-value inference, when the inference model is extended to multi-value one. We give detailed example to illustrate these flaws.

3.2.1 Flaws of Precision

As for single-value truth inference, a single source quality metric of *precision* is utilized in works [2, 3], aiming to decide the most trustworthy answer from conflicting data. This single metric measurement only considers the provided information. Due to the practical difficulties in data collection and validation, more sources tend to provide less but most trustworthy values. This situation results the loss of support for those less provided values, even though they are provided by some well-known and trustworthy sources. Since these values are less supported, they are unlikely to be inferred as true in the single metric model, causing a great degradation in the coverage for multi-value inference. An example is given to illustrate this flaw:

Example 3.2.1. *We tend to retrieve an inferred truth table with possibly highest precision, meaning that only the value supported by most sources can be inferred as true in the output. In **Table 2.1**, the value "Daniel Radcliffe" is provided by all sources, taking this value as the only output truth for "Harry Potter". However, the casts of "Harry Potter" contain "Emma Waston" and "Rupert Grint" and they are not inferred due to less support, resulting bad performance in terms of coverage, i.e. recall.*

3.2.2 Flaws of Accuracy

In [4, 9], the metric of accuracy is utilized. However, it is inappropriate to model the two types of errors, i.e., false positives and false negatives, as a single one, since there is no necessary correlation between these two errors. A source can be strict with its provided information, resulting low false positives but high false negatives, while another source can be casual which provides more but puts less care about the correctness, resulting low false negative but high false positives. As for single *accuracy* metric, it is possible that these two types of sources are treated with same trustworthiness. Since we cannot identify the error type with only *accuracy* metric, this situation would put us in a dilemma: if we accept the relative low *accuracy* sources (some with high false positives, some with high false negatives), it is possible that lots of false values are inferred, resulting low *precision*; if we do not accept those low *accuracy* sources, it is possible that lots of true values are ignored, resulting low *recall*. An example is given to illustrate this flaw:

Example 3.2.2. *It is clear that the accuracy of Netflix and BadSource.com is identical in* ***Table 3.2****. If we set a low accuracy accepting bound, the value "Jonny Depp" might be inferred as true for "Harry Potter", resulting bad performance in terms of precision; if we set a high accuracy accepting bound, the value "Emma Waston" might be ignored in the inferred truth, resulting bad performance in terms of recall.*

3.3 Two-side Quality Measurement

Recognizing the latent flaws in the single metric measurement, a two-side source quality measurement is proposed in [1] and demonstrate practical elevations in the inferred truth evaluation. This two-side quality measurement is also utilized in the proposed model of this thesis. Within this part, we present this quality measure and give detailed example to illustrate its rationality.

3.3.1 Quality Metric: Recall and Specificity

According to previous illustrated flaws of single metric measurement, it is impossible to capture the two types of error using one metric, since these two error types measure source quality in relatively separated and independent aspect. A conservative and strict source would put more emphasis on the precision, thus limiting the number of provided values to maintain correctness, while a bold source would focus on the coverage, thus tending to provide false value more frequently.

Two separate source quality metrics, i.e.*recall* and *specificity*, are utilized in multi-value truth inference model, in order to measure the source quality in a more rational and comprehensive manner. An example is given to illustrate how the two-side metrics make compensation to previous single-metric flaws:

Example 3.3.1. *From the Table 3.2, it is clear that Netflix is a source with low recall and high specificity, IMDB is a source with both high recall and specificity, and BadSource.com is a source with low specificity and medium recall. Given this as knowledge, the Netflix-not-provided values "Emma Watson" and "Rupert Grint" will be given less penalty and we intend to less believe the value "Johnny Depp", since it is only provided by the BadSource.com. Once setting proper parameter for accepting values, it is possible to eliminate "Johnny Depp" and give ("Daniel Radcliffe", "Emma Watson", "Rupert Grint") as the inferred truth for "Harry Potter". This result achieve both the highest possible precision and recall.*

It is implicit that we assume the ground truth is the same as the inferred truth, which is not the case for the unsupervised manner of the truth inference problem. Within following truth inference model, we use the intermediate inferred truth to calculate the source quality and update the inferred truth based on the new source quality within each iteration, until reaching convergence.

3.4 Inference Supplement: Source Attributes

It is believed that there are additional attribute or features linked with the source ability. To obtain these attributes for source requires less effort, compared with labeling the ground truth. However, the contribution of these attributes or detailed relationship to the source quality remain unknown, especially for unsupervised truth inference. It is desirable that a relationship between these informative attributes and source quality can be retrieved, thus providing some direct metrics for selecting sources, instead of performing truth inference. The introduction of source attributes or features is proposed in [9], focusing on single-value truth inference and supervised learning. Moreover, due to the loss of information about source in starting stage, the initialization of source quality parameter in [1, 6, 7] are performed in a uniform manner, which is not consistent with nature of sources. Though the parameter sensitivity is checked in above works, the check is still conducted in uniform manner, without initializing each source with different quality parameter.

In this work, we extend the truth inference model to two stage, along with a supplementary stage. The supplementary stage is only conducted, if source attributes are inputted. As for supplementary stage, the inferred source qualities, i.e. *recall* and *specificity*, in the first stage are respectively modeled as a weighted combination of normalized source attributes plus an error item. This error item is used to capture the loss or insufficiency in data collection. Through conducting *Least Square* method, we can minimize the square sum of error items and calculate the source attribute weights. Those weights can be utilized for direct source selection, once the source attributes are available. Another truth inference stage, stage 2, is conducted to perform refined source quality modeling. After retrieving a adjusted source qualities based on those weights, we adjust the source quality to narrow the

source quality difference between sources. This is because we want to keep a large possible value set in the beginning of stage 2 while preserving the source difference for stage 2 initialization. Without adjusts, the values provided by those low quality sources will be directly eliminated, due to the nature multiply. We adjust the source quality q based on **Equation 3.1**. The inferred weights for source attributes can used for fast source quality estimation and thus for truth inference initialization.

$$q_{adjust} = q^{\gamma_q} \tag{3.1}$$

The γ_q can be adjusted based on natures of source. For example, we can set this parameter higher for *recall*, since most sources tend to ignore some unimportant values; we can set this parameter lower for *specificity*, since most sources tend to provide true values.

4. TWO-STAGE TRUTH INFERENCE MODEL

A two-stage truth inference model is illustrated in details in this section. This two-stage model is based on the model proposed in [1] and [6]. A sampling-based truth inference model is proposed in the first one, showing good performance in terms of capturing the underlying source quality, while the model proposed in the second one integrates some heuristic to estimate a refined source quality, i.e. quality in different domains. Inspired by these two models, a two-stage truth inference model, along with an intermediate stage for attribute-based reliability adjust, is proposed in this thesis. For the first stage, a sampling-based inference, considering the distribution of the observed values, is conducted for the source quality estimation. For the intermediate stage, the estimated source quality is adjusted, given some reliability-related source attributes. For the second stage, the retrieved source quality is used for the initialization of a Bayesian-based truth inference, which take the similarity between source and object into account.

4.1 Stage 1: Truth Distribution Model

In this section, the first stage, called ***T**ruth **D**istribution **M**odel (TDM)*, of the proposed model, which aims at discovering the latent source quality based on the distribution of the observed fact is discussed. Due to intuition that the observed facts proposed by sources or the latent truth have strong connection with the source quality, Bayesian networks are utilized to construct the possible probability-based relationship between them. This methodology is firstly proposed in [1] and demonstrate great performance for capturing the latent source quality. However, the initialization method of this work might cause problem, due to the data quantity difference between sources. An introduction to Bayesian networks will be firstly given within this section, preparing for the probability model. Since some operations on the original input data are required to support this inference stage, the data model for stage 1 is then illustrated, along with the intuitions behind the proposed probability model for justifying the rationales. Meanwhile, we explain the reason why the initialization in in [1] is unreasonable and propose our version of model initialization. And finally the model details, including a brief review about *Beta Distribution*, and source quality inference process are shown. The inferred source quality will be further utilized in the proceeding refined truth inference stage.

4.1.1 Probability Graphical Modeling

Probabilistic graphical modeling is a branch of machine learning that utilize probability distributions to describe the world and perform informative inference about it. This

model is especially useful when the relations between events involves a significant amount of uncertainty [10]. As a combination of graph theory and probability, probabilistic graphical modeling can construct the inter-relationship between the uncertainty events and thus enable the derivation of efficient machine learning algorithms for solving the uncertainty.

Through parameterizing the probability distributions with several variables or parameters and representing the rational relationship between them as *directed acyclic graphs (DAG)* [10], Bayesian network, one of the probabilistic graphical model, can model the real-world events based on prior belief and knowledge. The events with uncertainty and have influence on the other events are represented as random variables and thus the construction of Bayesian networks, i.e. directed graphical models, follows the chains rule as **Equation 4.1**, which indicates the conditional dependence between events..

$$p(x_i, x_{i-1}, x_{i-2}, ..., x_n) = p(x_1)p(x_2|x1) \cdots p(x_n|x_{n-1}, ..., x_2, x_1) \tag{4.1}$$

.

The distribution in this form with the inter-dependence is expressed as *directed acylic graphs* readily, in which nodes correspond to variables and directed edges from node a to b (a is viewed as the parent of b) stands for the conditional dependence between random variables represented by the two linked nodes. The random variables associated with the child node follows the probabilistic conditional distribution with parameter that depend on values of the parent nodes. In this context, each node is linked with a random variable that could be observed values, latent (unobserved) values or unknown parameters [1]. Consider a live case that the admission result of a student applying for graduate program. This admission decision d depends on the Grade Point Average (GPA) g, reference letter quality q, program enroll number e and program application number a. These four factors could be represented as random variables and there is conditional dependence between them and the the final admission result. Asides, the quality of reference letter has some connections or dependence relationship with GPA. Moreover, since the grading policy in different universities might differ greatly, we can add a parameter β indicating the distribution difference of the GPA in the applicant's undergraduate school. The constructed joint probability distribution over this 5 random variables can be factorized as **Equation 4.2**.

$$p(d, g, q, e, a|\beta) = p(d|g, q, e, a)p(q|g)p(g|\beta)p(e)p(a) \tag{4.2}$$

.

In practical application, the assignment of the latent variables and unknown parameters values that (approximately) maximize likelihoods for those unobserved variables can be inferred by performing *maximum a posterior* (**MAP**) estimation, given the observed data and prior and conditional distributions. Various inference algorithms are ready to perform

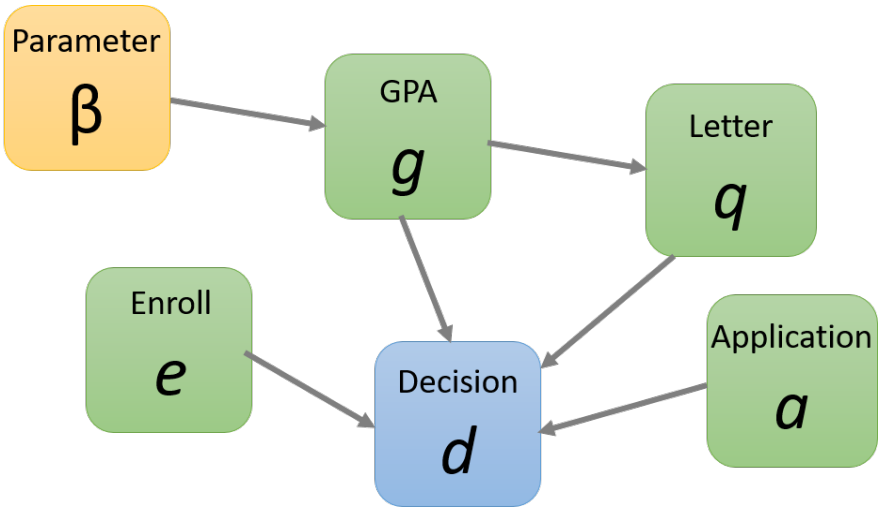

Figure 4.1: Bayesian Network Example

MAP, including *Collapsed Gibbs Sampling* and *Expectation Maximization (EM)*. The stage 1 of the proposed model, *TDM*, is a Bayesian network for inferring the source quality based on the distribution of the observation provided by sources. The inferred source quality is then taken as the input for the stage 2 to perform proceeding truth inference.

4.1.2 Data Model for Truth Distribution Model

Given the original input defined in **Section 2.1**, there are additional operations required, in order to support **TDM**. Moreover, since the output of this stage is used for the following stage, the output data model is also required to be specified.

For the sake of examining the possible-value set of $attribute_o^f$ provided by sources, we re-cast the fundamental input into a table of facts, i.e. select distinct $attribute_o^f$ from fundamental database and group by $object$, and further generate a table of claims to identify whether a source provide a fact in facts table.

Definition 4.1.1. *Let $\mathcal{DB}_f = \{f_1, f_2, ..., f_F\}$ be the fact database with a set of unique* fact*, i.e. object-attribute pair, extracted from the fundamental database, where every* f *is in the format of (F_{id}, o, a_o^f), where F_{id} severs as the primary key for this tuple, o is the object, and a_o^f is the value of focusing attribute. Each combination of o and a_o^f is unique in the fact database and it is possible that the pair may present in the fundamental database for many times.*

Table 4.1 is a fact database generated from **Table 2.1**.

After retrieving the facts from the fundamental, we obtain a value set of $Attribute_o^f$ for each *object*. In order to link the values in value set with the sources provided information for the *object*, we further generate a table of claims, i.e. a fact-source pair with a associated Boolean value to identify whether the *source* provides the fact.

Table 4.1: The fact table for **Table 2.1**

F_{id}	*Object (Movie)*	$Attribute_o^f$ *(cast)*
1	Harry Potter	Emma Waston
2	Harry Potter	Daniel Radcliffe
3	Harry Potter	Rupert Grint
4	Harry Potter	Jonny Depp
5	500 Days with Summer	Joseph Gordon-Levitt
...	...	...

Definition 4.1.2. *Let $\mathcal{DB}_c$ = {c_1, c_2, ..., c_C} be the claim database with a set of unique* claim*. Each* c *extracted from the fundamental database is in the format of* (F_{id}, s, o_c)*, where* F_{id} *is the id of a fact in the fact table,* s *is a source providing information for the* object *associated with that fact, and* o_c *is the observation of the claim, taking a Boolean value.*

In details, each fact f in the fact table is generated in this manner:

1. For each source providing f in the fundamental database, we set the o_c as True and generate $(F_{id}, s, True)$.
2. For each source not providing f but providing information about the *o* associated with f in the fundamental database, we set the o_c as False and generate $(F_{id}, s, False)$.

We further group the claim by the fact it associated with. The set of claims that are linked with fact f is denoted as C_f and the set for the rest claims is denoted as C_{-f}. Moreover, let S_f be the set of sources associated with fact f and S_{-f} be the set of sources not associated with f.

Table 4.2 is a claim database generated from **Table 2.1**.

Table 4.2: The claim for **Table 2.1**

F_{id}	*Source (website)*	$Observation$
1	IMDB	True
1	Filmcrave	False
1	BadSource.com	True
2	IMDB	True
2	Filmcrave	True
2	BadSource.com	True
4	IMDB	False
4	BadSource.com	True
4	Filmcrave	False
...	...	...

As for the output of this stage, we produce two source quality estimation, i.e. *recall* and *specificity*, for each source and generate two source quality set as for output. One set is for *recall* and the form is $S^{re} = \{s_1^{re}, s_2^{re}, ..., s_{|S|}^{re}\}$, with s_i^{re} as the *recall* source quality estimation for s_i in S; One set is for *specificity* and the form is $S^{sp} = \{s_1^{sp}, s_2^{sp}, ..., s_{|S|}^{sp}\}$, with s_i^{sp} as the *specificity* source quality estimation for s_i in S.

4.1.3 Intuitions behind the Truth Distribution Model

It is intuitive that the underlying truth has dependency on source quality, the truth of facts and the fact observation of sources, which is proposed and explained in [1]. We want to model this dependency in probabilistic graphical model to capture the probabilistic relationship and thus perform truth inference. Within this part, we explain the intuitions of modeling the three factors as random variables.

- **Source Quality**
  The source quality is measured in two different and independent metrics: *recall* and *specificity*. Therefore, two separate random variables, respective for *recall* and *specificity*, are created for each sources. And we often have some prior belief or assumptions about sources, and we should be able to integrate these assumptions with sources. Generally, it is reasonable to assume the majority of sources tend to provide correct values, leading high specificity for each sources. It is also appropriate to assume missing data is common for sources, leading relatively low recall for sources. Moreover, if we have some prior assumptions for some specific sources, we should be allowed to plug in such prior belief in the truth inference model.

- **Fact Truth**
  As [1], we model the probability of each fact being true as a latent random variable. Additionally, the actual truth label of each fact, depending on the true probability, is modeled as a latent Boolean random variable. By doing so, two types of errors (false positives and false negatives) can be distinguished clearly and we can model the *recall* and *specificity* for sources in natural way. Moreover, if we have some prior assumptions for some specific facts, we should be allowed to plug in such prior belief in the truth inference model.

- **Claim Observation**
  The claim is composed of three components: the fact, the provided source, and the Boolean observation. It is clear that the claim observation depends on the fact truth and the provided source. In details, if the fact is true, the observation from a source with high *recall* is more likely to *True*, while the observation from a source with low *recall* is more likely to *False*. Meanwhile, if the fact is false, the observation

from a source with high *specificity* is more likely to *False*, while the observation from a source with low *specificity* is more likely to *True*. We thus model the claim observations as random variables, depending on source quality and the latent fact truth. Once the claim observations are available, it is possible that we can go back and infer the source quality and the latent truth.

4.1.4 Beta Distribution

The Beta distribution *Beta(a, b)* is a two-parameter distribution with range [0,1] and its *probability density function (pdf)* is shown as **Equation 4.3** [11]. In the context of Bayesian updating, *a* and *b* are referred as hyperparameters to distinguish them from the unknown parameter θ. Moreover, *a* and *b* are 'one level up' from θ since they parameterize its *pdf* [11].

$$f(\theta) = \frac{(a+b-1)!}{(a-1)!(b-1)!}\theta^{a-1}(1-\theta)^{b-1} \tag{4.3}$$

The *pdf* $f(\theta)$ has the form $c\theta^{a-1}(1-\theta)^{b-1}$, with c as a normalizing constant. And there are lots of other distributions in the form, including Binomial and Bernoulli. Also, Beta distribution is called a conjugate prior for Binomial and Bernoulli distributions. This means that if the likelihood is Binomial or Bernoulli, then a beta prior gives a beta posterior [11]. This characteristic is utilized by the truth inference model to integrate out some random variables, leading the simplification of joint distribution estimation. And practically conjugate means easy update [12].

A classic example to illustrate the intuitions of Beta distribution is the coin toss experiment, shown in **Example 4.1.1**.

Example 4.1.1. *Suppose we have a coin with unknown probability θ of heads. We want to estimate θ from the experiment results. It is reasonable to use a uniform probability for θ: $\theta \sim Uni(0,1)$, which is also $Beta(1,1)$. The $(1,1)$ means there is no experiment and no head is generated. Assume we have tossed the coin 12 times and get 8 heads and 4 tails. An intuitive way to estimate θ is to direct set θ as $\frac{8}{12}$. However, this estimation can only approximate the θ in coarse way, due to limited experiments. Since there is still uncertainty around θ, we use the experiment results as prior belief and generate a distribution to capture the uncertainty while plugging in the prior belief. After the experiments, since the likelihood function with a known $theta$ is binomial, we can get the posterior as $Beta(9,5)$, with $9 = 1+8$ and $5 = 1+4$. The $(9,5)$ means that there are $9+5-1 = 12$ experiments and 8 heads are generated. Moreover, the expectation of the generated Beta distribution is $\frac{9}{13}$, capturing our intuitions for estimating θ. The pdf of these two distributions is shown in **Figure 4.2**.*

4.1.5 Bayesian Network Model Details

The details of the **TDM** is illustrated within this section. The constructed probability graphical model is shown in **Figure 4.3**. Each node in the graph represents a random

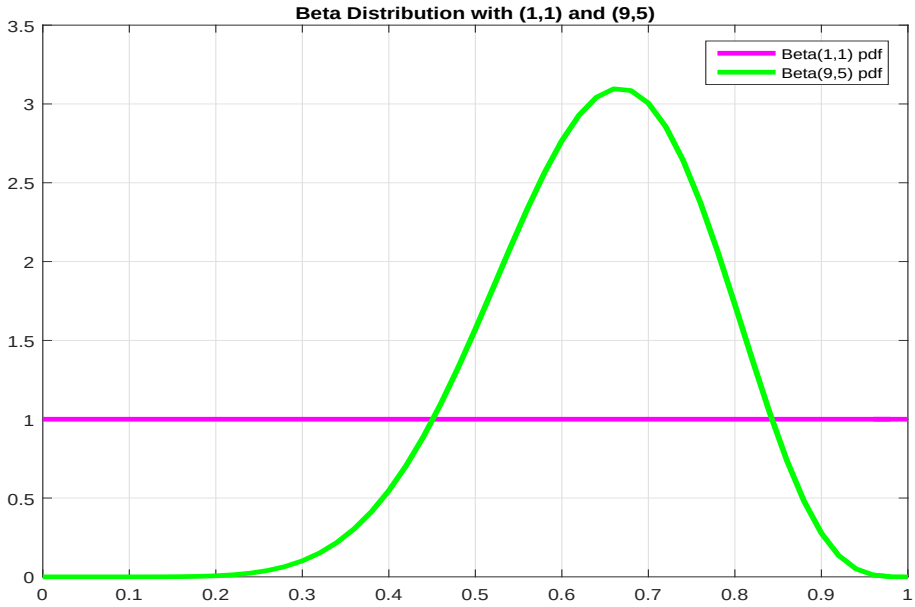

Figure 4.2: Beta(1,1) and Beta(13,9)

variable or prior parameter and the dark nodes stand for observed variables. Following the graphical model proposed in [1], a plate with a set as label indicates that the random variables represented by the nodes within the plate are replicated for each element in the set. For example, the S plate indicates that the sources have independent quality nodes. The direct edge from node a to b means the random variable related with b is generated from the distribution taking a as (one of) the parameter(s). And for each source, a special initialization for parameter ω^1 and ω^0 is discussed in Section 4.1.6. The detailed illustration for the proposed model is as follows.

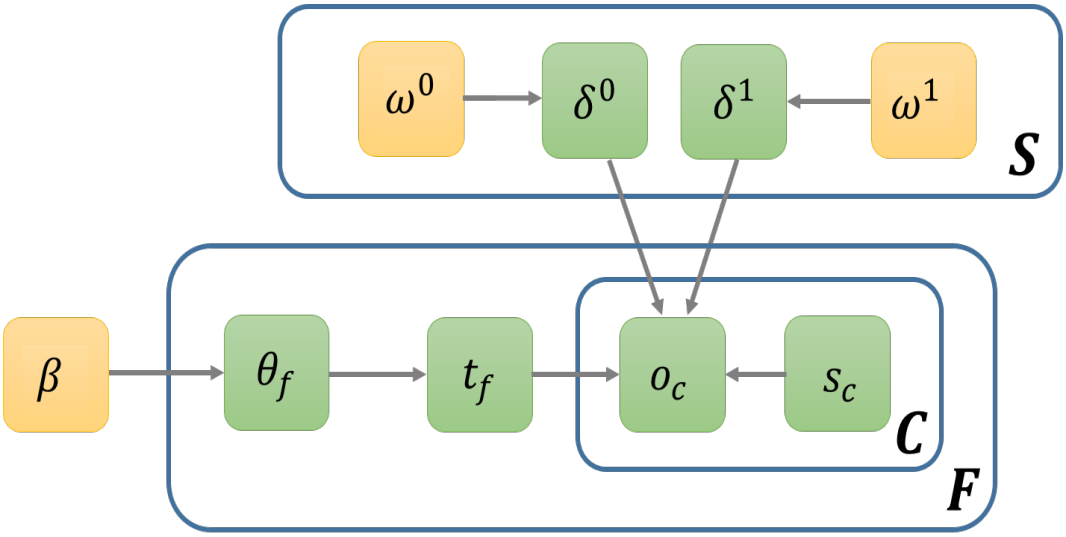

Figure 4.3: The probabilistic graphical model of **TDM**

1. **Recall.**

The recall δ^1 for each source is generated from a Beta Distribution with a source-dependent hyperparameter $\omega^1 = (\omega^{1,1}, \omega^{1,0})$, where $\omega^{1,1}$ is the prior true positive count and $\omega^{1,0}$ is the prior false negative count, as **Equation 4.4**:

$$\delta^1 \sim Beta(\omega^{1,1}, \omega^{1,0}). \tag{4.4}$$

The reason to use Beta distribution is because it is the conjugate prior of its children node distributions, i.e. Binomial and Bernoulli distributions, and the inference is more efficient. ω^1 controls the prior belief for *recall* and we assume sources tend to ignore some true facts and we do not have some strong prior for ω^1 in practice. Thus we use an uniform prior to generate δ^1. For each source, ω^1 is a hyperparameter depending on the fact number relating with the source. A specific initialization procedure for ω^1 is in Section 4.1.6.

2. **FPR.**
   The false positive rate δ^0 for each source, which is $(1 - specificity)$, is generated from a Beta Distribution with hyperparameter $\omega^0 = (\omega^{0,1}, \omega^{0,0})$, where $\omega^{0,1}$ is the prior false positive count and $\omega^{0,0}$ is the prior true negative count, as **Equation 4.5**:

$$\delta^0 \sim Beta(\omega^{0,1}, \omega^{0,0}). \tag{4.5}$$

   The reason to use Beta distribution is the same as previous. The ω^0 controls the prior belief for the specificity. In practice, we assume the overall quality of data is high and the false positive rate is relatively low. Without this assumption, it is possible that model could flip every truth due to the multiply nature. Thus we set $\omega^{0,0}$ significantly higher than $\omega^{0,1}$. Also, for each source, ω^0 is a hyperparameter depending on the fact number relating with this source. A specific initialization procedure for ω^0 is in Section 4.1.6.

3. **For each fact**

   3(a). Prior truth probability. The prior truth probability θ_f is generated from a Beta distribution with hyperparameter $\beta = (\beta_1, \beta_0)$, as **Equation 4.6**. β_1 is prior true count and β_0 is prior false count. β controls the probability that a fact is true.

$$\theta_f \sim Beta(\beta_1, \beta_0). \tag{4.6}$$

   3(b). Generated truth label. The truth label is generated from a Bernoulli distribution with θ_f, as **Equation 4.7**. The generated t_f is a Boolean variable with prior probability as θ_f.

$$t_f \sim Bernoulli(\theta_f). \tag{4.7}$$

3(c). Claim observation. The observation for each claim o_c is an observed Boolean variable. It is assumed to be generated from a Bernoulli distribution with $\delta_{s_c}^{t_f}$, as **Equation 4.8**. If the truth label $t_f = 0$, i.e. the fact is generated as False, then we use the generated false positive δ^0 of the source linked with this claim. If the truth label $t_f = 1$, i.e. the fact is generated as True, then we use the generated recall δ^1 of the source linked with this claim.

$$o_c \sim Bernoulli(\delta_{s_c}^{t_f}). \tag{4.8}$$

4.1.6 Shortages of Uniform Initialization

In the following proposed model, after we generating the *recall* and *specificity* for a source from Beta distribution, we want to update corresponding quality metrics based on the inferred truth, just as **Example 4.1.1**. The inferred truth is thus used for posterior update to estimate corresponding parameters, i.e. *recall* and *specificity*. However, this update is sensitive to both the quantity of prior counts and the quantity of the posterior increments. And the posterior increments have strong connection with the fact number that a source is associated with. For a source, the associated fact number not only include the facts it provides but also the facts provided by other sources for the common objects, as shown in **Equation 4.9**. If the prior count is too high, the updated increments would be insignificant, resulting the inferred truth becoming useless for source quality estimation. On the other side, if the prior count is too low, the updated increments would weighs too much, leading our prior knowledge to be neglected. In previous work [1], *recall* and *specificity* are assumed to be generated from Beta distributions that all source take the same hyperparameter. This uniform initialization would cause problems as illustrated above when the data quantities or the associated fact number from each source are not uniform, which is exactly the case in our datasets. The number of associated facts for sources in our **Movie Director Dataset** are demonstrated in **Table 4.3** and the detailed situation is illustrated in **Example 4.1.2**.

$$\#AssoFact_s = \sum_{o \in O(s)} |f \in \mathcal{DB}_f \; where \; o_f = o| \tag{4.9}$$

Example 4.1.2. *We use a uniform initialization for all sources.*

- *One case is setting the prior count too high: we set* $\omega^0 = (100, 10000)$ *for* specificity *to generate high specificity,* $1 - \frac{100}{10100} = 99\%$*. Assume we want to update* specificity *for IFCfilms, i.e. add the number of its false positive count to 100 and add the number of its true negative count to 10000. It is clear that it is associated with 134 facts. If it is a relatively bad source and there are lots of false positive generated, we would like to lower its* specificity*. However, even if we add 100 to 100 and 34 to 10000, its* specificity *is still high,* $1 - \frac{200}{10234} = 98\%$*, which is unreasonable and anti-intuitive. This would result a possible performance degradation.*

Table 4.3: Associated fact number for sources in **Movie Director Dataset**

Source	#Associated Fact	*Source*	#Associated Fact
Letterboxd	**32856**	Goodfilms	23632
IMDB	18035	Top250tv	10041
Flimcrave	9372	Matacritic	4334
1moviesonline	4329	Amazon	3595
dewanontons	3215	Movieinsider	2732
Allmovie	2685	Moviefone	2248
Agoodmovietowatch	213	**IFCfilms**	**134**

- *The other case is setting the prior count too low: we set* $\omega^0 = (1, 100)$ *for* specificity *to generate high specificity,* $1 - \frac{1}{101} = 99\%$*. Assume we also want to update* specificity *for Goodfilms, i.e. add the number of its false positive count to 100 and add the number of its true negative count to 10000. It is clear that it is associated with* 23632 *facts. This huge number would directly invalidate our prior assumption, which is again unreasonable. Though this might not affect the possible model performance, it could make our preliminary efforts to be useless, assuming we have some strong prior knowledge for sources.*

Instead, we propose an source-dependent initialization method. For each source, once getting the fundamental database, we can count the number of facts that each source is linked with, as **Equation 4.9**. We can have then assign these facts by ratios to the quality matrix of source to perform initialization. The ratios are predefined parameters in the format of *ratio matrix*. We illustrate this by example. Since we do not have strong prior knowledge for *precision* and *recall*, we can assign equal number to TP_s, FP_s and FN_s, saying the ratio as $1 : 1 : 1$. And we have strong prior that source have high *specificity*, setting the ratio for FP_s and TN_s as $1 : 99$.The total ratio would be $1 : 1 : 1 : 99$ for $TP_s : FN_s : FP_s : TN_s$. By setting this ratio, we can multiple it with the total facts that a source is associated with to perform initialization. Assume we use IMDB. Since it is linked with 18035 facts, we can get $prior\ TP_s = \frac{1}{1+1+1+99} \times \#AssoFact_s$, $prior\ FN_s = \frac{1}{1+1+1+99} \times \#AssoFact_s$, $prior\ FP_s = \frac{1}{1+1+1+99} \times \#AssoFact_s$, and $prior\ TN_s = \frac{99}{1+1+1+99} \times \#AssoFact_s$. Thus, we use $\omega^1 = (TP_s, FN_s)$ and $\omega^0 = (FP_s, TN_s)$ for IMDB. A example *ratio matrix* is shown in **Table 4.4**.

Table 4.4: Example ratio matrix

Ratio	***True***	***False***
Positive	1	1
Negative	99	1

4.1.7 Collapsed Gibbs Sampling

A major limitation of Bayesian approaches is that obtaining the posterior distribution often requires the integration of high-dimensional function [13]. *Markov Chain Monte Carlo (MCMC)* methods are thus developed to simulate the process of direct sample draws from some complex distribution. The reason calling *MCMC* is because the next sample is generated based on current sample value, forming a *Markov chain*. The original *Monte Carlo* method is developed to use random numeric sample generation to compute integral, through transforming the target function to the form of expectation calculation of some distribution. By constructing a Markov chain that has the desired distribution as its equilibrium distribution, one can obtain a sample of the desired distribution by observing the chain after a number of steps. A large number of samples are thus drawn with some probability density and the expectation is estimated, finishing estimation of integral [13]. The more steps there are, the more closely the distribution of the sample matches the actual desired distribution. One of the *MCMC* method is exactly the *Gibbs sampling*.

Gibbs Sampling, which estimate the joint distribution through drawing conditioned on other random variables. *Gibbs* is more efficient to obtain good approximation of joint distribution. Gibbs sampling only considers *univariate* conditional distribution. This means that all of the random variables are assigned with fixed values, except one. Comparing with complex joint distribution, the conditional distributions are much easier to simulate [13]. Considering n random variables, saying $\{x_0, x_1, ..., x_{n-1}\}$, n random variables are simulated sequentially from the n *univariate* conditions rather than a full joint distribution. In details, we first assign initial values for n-1 random variables and sample the unassigned random variable x_0 from this conditional distribution. After the value assignment of x_0, we turn to sample next random variable, saying x_1. We sample x_1 based on present value assignments for the other n-1 random variables, i.e. another conditional distribution. We keep this manner for sufficient number of iterations or scan enough cycles for n random variables. Then the Gibbs sequence converges to a stationary (equilibrium) distribution, which is independent of the initialization, and this stationary distribution is the target distribution that we want, i.e. the joint distribution.

Collapsed Gibbs Sampling is developed based on Gibbs Sampling. Through linking the other possible random variables or finding their dependency to some common ones, we can make our sampling to be simpler and get the joint distribution only for the common ones. Since some variables are no longer to be sampled sequentially, this sampler is called as *collapsed* Gibbs sampler. As for truth inference problem, we can treat the latent *True* or *False* of provided a_f^o as random variables. The other random variables including truth probability, source specificity and sensitivity are integrated out, due to the conjugation of exponential families [1], meaning these random variables can be integrated out within the

sampling process. We can thus use the joint distribution to generate a new set of True or False assignments for the a_f^o to infer the truth.

4.1.8 Stage 1 Inference Algorithm

In this part, we integrate the constructed probability graphical model with the data-model to illustrate the detailed truth inference process of stage 1. We first give the likelihood functions and then truth inference algorithm, based on the collapsed Gibbs sampling. The source quality estimation are discussed at last and the estimated source quality for each source is the output of this stage.

- **Likelihood functions**
  Based on the probabilistic graphical model of **TDM**, the likelihood function for the observation of each claim *c* of fact f given the dependency parameter is as **Equation 4.10**.

$$p(o_c|\theta_f, \delta^0_{s_c}, \delta^0_{s_c}) = p(o_c|\delta^0_{s_c})(1-\theta_f) + p(o_c|\delta^1_{s_c})\theta_f \tag{4.10}$$

  Since we model the sources as independent, the complete likelihood for all random variables, including the latent truth, given the hyperparamter ω_0, ω_1 and β is as **Equation 4.11**.

$$\begin{aligned} p(\mathbf{o}, \mathbf{s}, \mathbf{t}, \boldsymbol{\theta}, \boldsymbol{\delta^0}, \boldsymbol{\delta^1}|\boldsymbol{\omega^0}, \boldsymbol{\omega^1}, \beta) = \prod_{s\in S} p(\delta^0_s|\omega^0_s)p(\delta^1_s|\omega^1_s)\times \\ \times \prod_{f\in F}\left(p(\theta_f|\beta)\sum_{t_f\in 0,1}\theta_f^{t_f}(1-\theta_f)^{1-t_f}\prod_{c\in C_f}p(o_c|\delta^{t_f}_{s_c})\right) \end{aligned} \tag{4.11}$$

- **Conditional distribution for collapsed Gibbs sampling**
  Given observation data, a assignment of latent truth that maximize the joint probability, or equivalently estimate the *maximum a posterior* (**MAP**) for t as **Equation 4.12**.

$$\hat{\boldsymbol{t}} = \arg\max_t \iiint p(\boldsymbol{o}, \boldsymbol{s}, \boldsymbol{t}, \boldsymbol{\theta}, \boldsymbol{\delta^0}, \boldsymbol{\delta^1}) d\boldsymbol{\theta} d\boldsymbol{\delta^0} d\boldsymbol{\delta^1} \tag{4.12}$$

  With the help of *collapsed Gibbs sampling*, instead of using brute force search all the possible latent truth assignment, we can estimate above distribution in a more efficient way. As introduced in previous part, Gibbs sampling is a Markov chain Monte Carlo (MCMC) algorithm which estimate a joint distribution by sampling from a series of conditional distributions sequentially. And this model use the conjugacy of exponential families [1] that $\boldsymbol{\theta}$, $\boldsymbol{\delta^0}$ and $\boldsymbol{\delta^1}$ can be integrated out in the sampling. Since once the distributions of latent truth and observation are retrieved, the posterior distribution of those random variables can be directly retrieved. Moreover, since we can get the likelihood function of observation given the latent truth, we can merely focus

on the latent truth.

Given this, we can start to measure the joint distribution of the latent truth and observation through sampling from the conditional distribution of latent truth. We let $\boldsymbol{t_{-f}}$ be the truth set of all facts in F except f. For each fact, we iteratively sample it from a conditional distribution, given all the other facts, as **Equation 4.13**.

$$
\begin{aligned}
&p(t_f = i|\boldsymbol{t_{-f}}, \boldsymbol{o}, \boldsymbol{s}) \propto \beta_i \prod_{c \in C_f} \frac{n^{-f}_{s_c,i,o_c} + \omega_{i,o_c}}{n^{-f}_{s_c,i,1} + \omega_{i,q} + n^{-f}_{s_c,i,0} + \omega_{i,0}} \\
&where \quad n^{-f}_{s_c,i,j} = |\{c' \in C_{-f} | s_{c'} = s_c, t_{f_{c'}} = i, o_{c'} = j\}|
\end{aligned}
\tag{4.13}
$$

The item $n^{-f}_{s_c,i,j}$ is the number of claim provided by s_c whose associated fact is not f, the assigned latent fact truth as *i* and the observation as *j*. For example, $n^{-f}_{s_c,0,0}$ is the true negative count excluding claims relating with fact f; $n^{-f}_{s_c,1,0}$ is the false negative count excluding claims relating with fact f; $n^{-f}_{s_c,0,1}$ is the false positive count excluding claims relating with fact f; $n^{-f}_{s_c,0,0}$ is the true negative count excluding claims relating with fact f.

The detailed deduction for **Equation 4.13** is shown as follows.

Equation 4.13 deduction:

$$p(t_f = i|\boldsymbol{t_{-f}}, \boldsymbol{o}, \boldsymbol{s}) \propto p(t_f = i|t_{-f}) \prod_{c \in C_f} p(o_c, s_c|t_f = i, \boldsymbol{o_{-f}}, \boldsymbol{s_{-f}}) \tag{4.14}$$

The first term in **Equation 4.14** is rewritten as follows:

$$\begin{aligned} p(t_f = i|t_{-f}) &= \int p(t_f = i|\theta_f) p(\theta_f|t_{-f}) d\theta_f \\ &= \frac{1}{\mathcal{C}(\beta_1, \beta_0)} \int \theta_f^{\beta_1 + i - 1} (1 - \theta_f)^{\beta_0 + (1-i) - 1} d\theta_f \\ &= \frac{\mathcal{C}(\beta_1 + i, \beta_0 + 1 - i)}{\mathcal{C}(\beta_1, \beta_0)} = \frac{\beta_i}{\beta_1 + \beta_0} \propto \beta_i \\ &\quad where\, \mathcal{C}(\alpha, \beta)\, is\, the\, normalized\, factor\, for\, Beta(\alpha, \beta) \end{aligned} \tag{4.15}$$

The remaining part in **Equation 4.14** can be transformed as follows:

$$\begin{aligned} p(o_c, s_c|t_f = i, \boldsymbol{o_{-f}}, \boldsymbol{s_{-f}}) &\propto \int p(o_c|\delta^i_{s_c}) p(\delta^i_{s_c}|\boldsymbol{o_{-f}}, \boldsymbol{s_{-f}}) d\delta^i_{s_c} \\ &\propto \int p(o_c|\delta^i_{s_c}) p(\delta^i_{s_c}) \prod_{c' \notin C_f, s_{c'} = s_c} p(o_{c'}|\delta^i_{s_c}) d\delta^i_{s_c} \\ &\propto \frac{\int (\delta^i_{s_c})^{o_c + n^{-f}_{s_c,i,1} + \omega^{i,1}_{s_c} - 1} (1 - \delta^i_{s_c})^{(1-o_c) + n^{-f}_{s_c,i,0} + \omega^{i,0}_{s_c} - 1} d\delta^i_{s_c}}{\mathcal{C}(n^{-f}_{s_c,i,1} + \omega^{i,1}_{s_c}, n^{-f}_{s_c,i,0} + \omega^{i,0}_{s_c})} \\ &= \frac{n^{-f}_{s_c,i,o_c} + \omega^{i,o_c}_{s_c}}{n^{-f}_{s_c,i,1} + \omega^{i,1}_{s_c} + n^{-f}_{s_c,i,0} + \omega^{i,0}_{s_c}} \end{aligned} \tag{4.16}$$

- **Source quality estimation with collapsed Gibbs sampling**

  The pseudo code of the truth inference and source quality estimation based on collapsed Gibbs sampling is presented in **Algorithm 4.1**.

  The latent truth for each fact is randomly assigned for initialization and the latent truth is re-sampled from the conditional distribution in each iteration. The source quality is then updated for each source. We can make the final prediction of the latent truth by using the final sampled value or the expectation of truth. The truth expectation for each fact is calculated after abandoning first *m* samples, saying the burn-in samples which is needed for Gibbs sampling to reach equilibrium state. The average of latent true value and is updated every *g* iteration to prevent correlation. For the final latent truth, we can get the average latent truth value of a fact, i.e. its true probability $p(t_f = 1)$, and $p(t_f = 0) = 1 - p(t_f = 1)$ is its false probability.

  After both $p(t_f = 1)$ and $p(t_f = 0)$ of facts are retrieved, we now turn our goal to the source quality estimation. It is easy to estimate the source quality, since the posterior of source quality is still a Beta distribution. We update the source quality

based on **Equation 4.17** and **Equation 4.18**. The updated source qualities form the two output sets, i.e. S^{re} and S^{sp}. We now finish this stage and the outputs are ready for following stages.

$$s^{re} = \delta_s^1 = \frac{E[n_{s,1,1}] + \omega_s^{1,1}}{E[n_{s,1,1}] + \omega_s^{1,1} + E[n_{s,1,0}] + \omega_s^{1,0}} \tag{4.17}$$

$$s^{sp} = \delta_s^0 = \frac{E[n_{s,0,0}] + \omega_s^{0,0}}{E[n_{s,0,0}] + \omega_s^{0,0} + E[n_{s,0,1}] + \omega_s^{0,1}} \tag{4.18}$$

$$where\ E[n_{s,i,j}] = \sum_{c \in C, s_c = s, o_c = j} p(t_{f_c} = i). \tag{4.19}$$

Since this algorithm performs iteration for each claim, the time complexity of **Algorithm 4.1** is $O(|C|)$ or $O(|S| \times |F|)$, whose running time is linear to the number of claims. It is much more efficient, comparing with a brute force truth assignment algorithm with complexity as $O(2^{|F|})$ [1].

Algorithm 4.1: TDM : Collapsed Gibbs Sampling

Input: $\mathcal{DB}_{raw}$
Output: S^{re}, S^{sp}

{Initialization}
for ***all*** $s \in S$ **do**
  Initialization for ω_s^0 *and* ω_s^1*, given ratio matrix*
for ***all*** $f \in F$ **do**
  {Sampling t_f from uniform prior}
  if uniform() $\leq$ *0.5* **then**
    $t_f \leftarrow 0$
  else
    $t_f \leftarrow 1$
  for ***all*** $c \in C_f$ **do**
    $n_{s_c,t_f,o_c} \leftarrow n_{s_c,t_f,o_c} + 1$
{Sampling}
for $iteration = 1 \rightarrow K$ **do**
  $iteration \leftarrow iteration + 1$
  for ***all*** $f \in F$ **do**
    $p_{t_f} \leftarrow \beta_{t_f}, p_{1-t_f} \leftarrow \beta_{1-t_f}$
    for ***all*** $c \in C_f$ **do**
      {Calculate $p(t_f = i|\boldsymbol{t_{-f}}, \boldsymbol{o}, \boldsymbol{s})$ with $i = 0, 1$ as **Equation 4.13**}
      $p_{t_f} \leftarrow p_{t_f} \times \frac{n^{-f}_{s_c,t_f,o_c} + \omega_{t_f,o_c}}{n^{-f}_{s_c,t_f,1} + \omega_{t_f,q} + n^{-f}_{s_c,t_f,0} + \omega_{t_f,0}}$
      $p_{1-t_f} \leftarrow p_{1-t_f} \times \frac{n^{-f}_{s_c,1-t_f,o_c} + \omega_{1-t_f,o_c}}{n^{-f}_{s_c,1-t_f,1} + \omega_{1-t_f,q} + n^{-f}_{s_c,1-t_f,0} + \omega_{1-t_f,0}}$
    {Sample t_f from the calculated probability}
    if $uniform() \leq \frac{p_{t_f}}{p_{t_f} + p_{1-t_f}}$ **then**
      $t_f = 1 - t_f$
      {t_f flip, update source quality count}
      for ***all*** $c \in C_f$ **do**
        $n_{s_c,1-t_f,o_c} \leftarrow n_{s_c,1-t_f,o_c} - 1$
        $n_{s_c,t_f,o_c} \leftarrow n_{s_c,t_f,o_c} + 1$
    {Calculate expectation of t_f}
    if $iteration \leq burnin$ ***and*** $iteration\%g = 0$ **then**
      $p(t_f = 1) \leftarrow p(t_f = 1) + t_f/(K/g)$
{Estimate source quality}
$S^{re}, S^{sp} \leftarrow list()$
for ***all*** $s \in S$ **do**
  $s^{re} = \frac{E[n_{s,1,1}] + \omega_{1,1}}{E[n_{s,1,1}] + \omega_{1,1} + E[n_{s,1,0}] + \omega_{1,0}}$
  $s^{sp} = \frac{E[n_{s,0,0}] + \omega_{0,0}}{E[n_{s,0,0}] + \omega_{0,0} + E[n_{s,0,1}] + \omega_{0,1}}$
  append s^{re} *to* S^{re}, *append* s^{sp} *to* S^{sp}

4.2 Source Attributes Supplement

Within this part, we introduce the intermediate stage as a supplement to the truth inference model. If the source attributes are provided, this stage is initialized and calculated an adjusted *recall* and *specificity* for each sources. The adjusted source quality measurements are taken as the inputs of stage 2 to perform initialization. On the contrary, if there are no source attributes available, the weight estimation is skipped and the truth inference process directly proceed to the second stage, taking the adjusted output of stage 1 to perform initialization.

Through viewing the quality measurement as a linear combination of source attributes, the *Least Square* method is utilized to infer the weights of each source attributes. The quality measurements for each source are then recalculated based on the weights and adjusted based on **Equation 3.1**. We first review the *Least Square* method and then give the algorithm for this intermediate stage. The input attribute values are normalized as **Equation 4.20**

$$\begin{aligned} &x_{normalized} = \frac{x-u}{\sigma} \\ &where\ u\ as\ average\ value\ and\ \sigma\ as\ std \end{aligned} \tag{4.20}$$

4.2.1 Least Square Method

The method of *Least Square* is a procedure to determine the best fit line to data, which capture the linear relationship between input data and the output [14]. Assume the input observation is in following form:

$$\begin{aligned} &\{(\boldsymbol{x^1}, y^1), (\boldsymbol{x^2}, y^2), ..., (\boldsymbol{x^N}, y^N)\} \\ &where\ \boldsymbol{x^i}\ is\ a\ m\ dimensional\ vector\ = (x_1^i, x_2^i, ..., x_m^i)^{'}. \end{aligned} \tag{4.21}$$

We believe that there is a approximately linear relationship between x^i and y^i, i.e. $y = \boldsymbol{w}^{'}\boldsymbol{x}+b$, and $\boldsymbol{w}$ is also m-dimensional vector to measure the weight of each dimensional in x. Ideally, $y - (\boldsymbol{x}^{'}\boldsymbol{w} + b)$ should be zero. Then we turn to look at following:

$$\{y^1 - ((\boldsymbol{x^1})^{'}\boldsymbol{w} + b), y_2 - ((\boldsymbol{x^2})^{'}\boldsymbol{w} + b), ..., y_N - ((\boldsymbol{x^N})^{'}\boldsymbol{w} + b)\}. \tag{4.22}$$

We can extend both $\boldsymbol{w}$ and $\boldsymbol{x^i}$ as following, to include the term b:

$$\begin{aligned} &\boldsymbol{x^i} = (1, (\boldsymbol{x^i})^{'})^{'}, \boldsymbol{w} = (b, \boldsymbol{w})^{'} \\ &we\ can\ get\ \{y^1 - (\boldsymbol{x^1})^{'}\boldsymbol{w}, y_2 - (\boldsymbol{x^2})^{'}\boldsymbol{w}, ..., y_N - (\boldsymbol{x^N})^{'}\boldsymbol{w}\}. \end{aligned} \tag{4.23}$$

To be a good approximation linear fitting, the square sum of above items should be

small. We write the square sum of above items in function of $\boldsymbol{w}$ as follows:

$$E(\boldsymbol{w}) = \sum_{n=1}^{N} (y^i - (\boldsymbol{x^i})^{'} \boldsymbol{w})^2. \tag{4.24}$$

Given this, our goad is to find the $\boldsymbol{w}$ that minimize the $E(\boldsymbol{w})$. If we write $\boldsymbol{y}$, X and $\boldsymbol{\varepsilon}$ as following:

$$\boldsymbol{y} = (y^1, y^2, ..., y^N)', X = (\boldsymbol{x^1}, \boldsymbol{x^2}, ..., \boldsymbol{x^N})', \boldsymbol{\varepsilon} = \boldsymbol{y} - X\boldsymbol{w} \tag{4.25}$$

We can get $E(\boldsymbol{w})$ is the $\boldsymbol{\varepsilon}^{'}\boldsymbol{\varepsilon}$. Then the question can be written as follows:

$$\min_{\boldsymbol{w}} \boldsymbol{\varepsilon}^{'}\boldsymbol{\varepsilon} \tag{4.26}$$

We can extent above equation in the form of $\boldsymbol{y}$, $\boldsymbol{w}$ and X:

$$\begin{aligned} \boldsymbol{\varepsilon}^{'}\boldsymbol{\varepsilon} &= (\boldsymbol{y} - X\boldsymbol{w})^{'}(\boldsymbol{y} - X\boldsymbol{w}) \\ &= \boldsymbol{y}^{'}\boldsymbol{y} - 2\boldsymbol{w}^{'}X^{'}\boldsymbol{y} - \boldsymbol{w}^{'}X^{'}X\boldsymbol{w} \end{aligned} \tag{4.27}$$

The gradient $\triangledown$ of $\boldsymbol{\varepsilon}^{'}\boldsymbol{\varepsilon}$ can be expressed as:

$$\triangledown \boldsymbol{\varepsilon}^{'}\boldsymbol{\varepsilon} = [\frac{\partial \boldsymbol{\varepsilon}^{'}\boldsymbol{\varepsilon}}{\partial w_i}]' \tag{4.28}$$

We rewrite previous expression as following, using the matrix derivatives:

$$\triangledown \boldsymbol{\varepsilon}^{'}\boldsymbol{\varepsilon} = 2X^{'}Xw - 2X^{'}y \tag{4.29}$$

When the gradient equals 0, we can get the desired $\hat{w}$ as:

$$\begin{aligned} 2X^{'}X\hat{w} - 2X^{'}y &= 0 \\ X^{'}X\hat{w} &= X^{'}y \\ \hat{w} &= (X^{'}X)^{-1}X^{'}y \end{aligned} \tag{4.30}$$

In order to retrieve the inverse matrix of $X^{'}X$, we required the data in X to be linearly independent.

4.2.2 Source Quality Adjust Algorithm

We retrieve the estimated source quality sets, i.e. S^{re} and S^{sp}, from the stage 1. If there are source attributes available, we start this intermediate stage to adjust the source quality. The algorithm of this intermediate stage is presented in **Algorithm 4.2**.

Algorithm 4.2: Adjusted Source Quality

Input: $S^{re}, S^{sp}, (\mathcal{DB}_{sa})$
Output: S^{re}, S^{sp}, ($\hat{\boldsymbol{w}^{re}}$, $\hat{\boldsymbol{w}^{sp}}$)

if *Available Source Attribute* **then**
  {Initialization}
  for $i = 1 \rightarrow |\mathcal{DB}_{sa}|$ **do**
    Normalize all source attribute values
    $\boldsymbol{x^i} \leftarrow (a^m_{s_{sa_i}})'$
  $X^{re}, X^{sp} \leftarrow (\boldsymbol{x^1}, ..., \boldsymbol{x}^{|\mathcal{DB}_{sa}|})'$
  $\boldsymbol{y}^{re} \leftarrow (s^{re} \in S^{re})'$, $\boldsymbol{y}^{sp} \leftarrow (s^{sp} \in S^{sp})'$
  Calculate $\hat{\boldsymbol{w}^{re}}$ *and* $\hat{\boldsymbol{w}^{sp}}$
  {Adjust $s^{re} \in S^{re}$ and $s^{sp} \in S^{sp}$}
  for $s \in S$ **do**
    $s^{re} = ((\boldsymbol{x^s})'\hat{\boldsymbol{w}^{re}})$
    $s^{sp} = ((\boldsymbol{x^s})'\hat{\boldsymbol{w}^{sp}})$
$S^{re}, S^{sp} \leftarrow list()$
for $s \in S$ **do**
  $s^{re} = (s^{re})^{\gamma_{re}}$
  $s^{sp} = (s^{sp})^{\gamma_{sp}}$
  append s^{re} *to* S^{re}, *append* s^{sp} *to* S^{sp}

4.3 Stage 2: Similarity-Aware Truth Inference Model

In this section, the second stage, called ***S**imilarity-**A**ware **T**ruth Inference **M**odel (SATM)*, of the proposed model, which aims at discovering the latent truth, is introduced. The source quality inferred from previous stage is utilized for initialization of this stage. In [6], the data richness of source in specific domain is shown as a good prior belief for the source quality estimation in that domain. It is based on the intuitions that sources demonstrates different expertise extent in different domains, i.e, one with higher data richness in a domain tends to shown great expertise in that specific domain and verse vice. Inspired by this work, we tend to extend this richness measurement to a distance-based metric, which measure the closeness or similarity between a source and an object that it gives data about. Meanwhile, a huge difference between the single-truth and multi-truth is that the conflicting values no longer stand for total exclusion for multi-truth model, leading a confidence measurement for the value a source provided. This is first proposed in [7] and extended by [6], as a factor when performing truth inference. As for our model, we also plug this in. For the rest of this section, we first illustrate the intuitions and details of this similarity measurement. Then the truth confidence is introduced as a factor in our truth inference model. Finally the source quality modeling and model inference algorithm are given. As for this final stage, there are two outputs, i.e. source quality and the inferred truth.

4.3.1 Source-object Similarity Modeling

In this part, we introduce our method to measure the similarity between a source and an object that this source claims values. The idea is originated from a heuristic idea that estimate the domain expertise of sources based on the data richness of a source in a domain and it is introduced in [6], showing great performance enhancement. In that work, each object is associated with only one domain during calculation, though objects might be associated with multiple domains, e.g. the categories of a book. As a source providing more data in the domain associated with that object, the expertise in that domain is assumed to be higher, which is integrated as a factor to the proposed truth inference model. This idea can be viewed as a one-dimensional similarity measure between the source and the object, due to the object-domain association.

Within our proposed model, we extent this similarity measurement to multi-dimension in order to dealing with the possible domains that the object is associated with. The object attribute database $\mathcal{DB}_{oa}$ is used for the similarity measurement. Given an object and its attributes, we can retrieve the attribute values and separate each attribute to domains. Once the domain separations are completed, we can establish the relationship between source and domain, due to the object-source association. Thus the domain expertise for source in

each domain can be calculated. We further represent both the source and object in vectorized manner and measure their similarity. The retrieved similarities are then integrated to our truth inference model. We first introduce several basic components of this similarity modeling.

1. **Domain separation**

   The domain separation is trivial for the categorized attributes, such as *Category* of book and *Country* of movie. We can naturally use the categorized value as domains. However, it is not the same case for the numeric data, such as *Price* of books and *Released Year* of movies. We have to select a good separation criteria to include those data in our similarity measurement.

   For an attribute, an intuitive and rational separation is to separate those numeric values into different intervals to form domains and we might set the number of interval to some predefined number according to our needs. For example, if we want a fine-grained separation, we can set a bigger number. And now the remaining question is the size selection for interval. Instead of a predefined size of intervals, we want to dynamically adjust the number of interval based on the situation of attributes. In order to accomplish that, we retrieve the histogram for numeric attribute with fineness as the basic unit of that attribute, e.g. 1 dollar for *Price* and 1 year for *Released Year*. We then choose the interval size that distribute same quantity of data to different domains of an attribute. This domain selection maintains a uniform distribution of data quantity and the intra-domain unevenness of sources can be shown better. An example to illustrate above idea is given in **Example 4.3.1**.

   Example 4.3.1. *Assume now we want retrieve the domain separation for the Released Year attribute for movies. As for Released Year, the data range is from 1890 to 2021 and the histogram with 1-year fineness is shown in* ***Figure 4.4****. It is clear that the data distribution is not uniform. If we set the number of interval of 6, we can approximate separate the data to 6 parts with (approximately) same data quantities. The interval separation line is also shown in* ***Figure 4.4****.*

2. **Source domain expertise**

   The heuristic that infer the initial domain expertise from the data richness shows a performance enhancement in [6] and we extend our model following the same manner. A factor termed as *source domain percentage*, denoted as $p_d(s)$, is calculated for each source s in each domain d. $p_d(s)$ is the percentage of data quantity provided by source s to the total data quantity in domain d, computed in **Equation 4.31**. Once the $p_d(s)$ for each source is retrieved, we can start to evaluate the *source domain expertise*, denoted as $e_d(s)$. There are two aspects that we should give concerns about. The one is the role of $p_d(s)$ when calculating $e_d(s)$: we expect monotonicity, i.e. higher

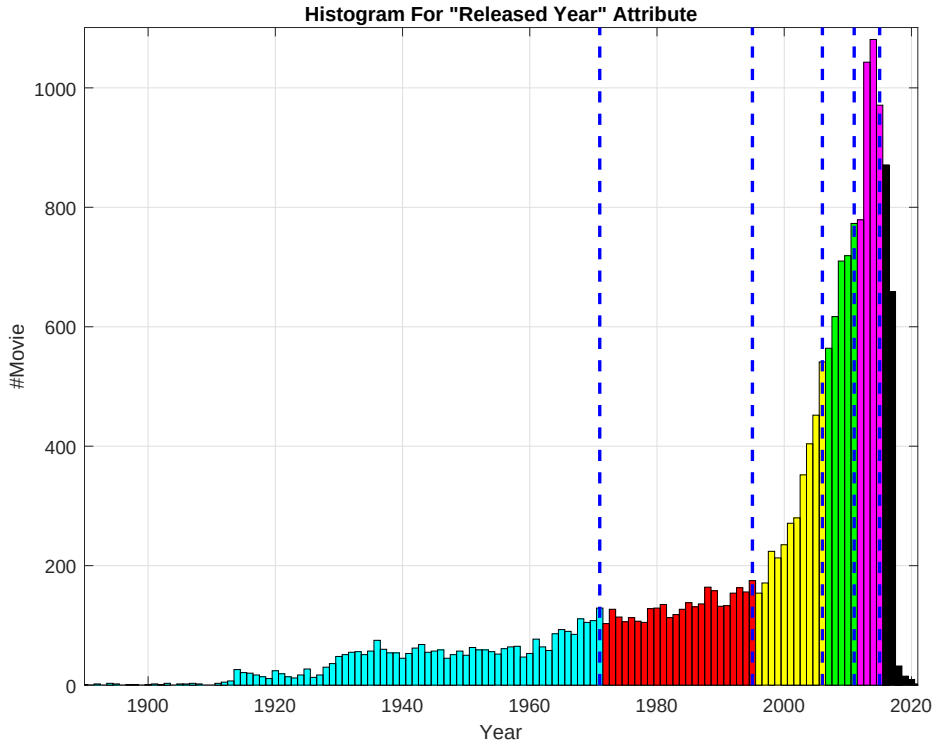

Figure 4.4: Histogram for *Released Year* and domain separation

$p_d(s)$ is correspond to greater $e_d(s)$ and verse vice. The other is the distribution information of $p_d(s)$: if all sources have similar percentage in one domain, the $p_d(s)$ tends to more or less useless; if there is great percentage difference between sources in one domain, we tend to magnify that influence. Considering above factors, the $e_d(s)$ is calculated as **Equation 4.32**. α_d is a predefined adjust factor to emphasize and distinguish the percentage difference in domain d.

$$p_d(s) = \frac{|O^d(s)|}{\sum_{s \in S} |O^d(s)|} \tag{4.31}$$

$$e_d(s) = \sqrt{1 - (\alpha_d \cdot p_d(s) - 1)^2} \tag{4.32}$$

3. **Vectorized representation**

   We now give definitions about how to vectorize the objects and sources to domain dimension.

   Definition 4.3.1. *The source vector of a source s is a vector with dimension element as $e_d(s)$, denoted as SV_s. The SV_s is a $|D_a| = \sum_{d^{a_i} \in D_a} |d^{a_i}|$ dimensional vector with the form as:*

$$SV_s = (e_{d_1}(s), e_{d_1}(s), ..., e_{d_{|D_a|}}(s))'$$

   Definition 4.3.2. *The object vector of an object o is a vector with dimension element as 0 and 1, denoted as OV_o. For OV_o, 1 on the i^{th} dimension indicates that the object is associated with domain d_i, and 0 on the i^{th} dimension indicates that the object is not associated with domain d_i. The OV_o is also a $|D_a|$ dimensional vector with the*

form as:

$$OV_o = (\mathbb{I}_{o \in O^{d_1}}, \mathbb{I}_{o \in O^{d_1}}, ..., \mathbb{I}_{o \in O^{d_{|D_a|}}})'$$

4. **Similarity measurement**

   As mentioned before, each attribute is treated independently. Thus, we assume that there is no dependency between the domains from different attributes. As for domains from the same attribute, it is possible that there are correlations. However, since we have factorized all possible domains of objects, instead of merely considering one, the possible correlations between domains from one attribute are implicitly taken into account. The similarity between a source s and an object o is measured as **Equation 4.33** and a example is given in **Example 4.3.2**. The retrieved similarity can be used for a quantitative measurement for source selection. If the similarity between s and o is high, it means that s provides more objects that is similar to o, i.e. tends to be more authoritative for o. A three dimensional illustration for similarity-based selection is shown in **Figure 4.5**. The sources that are close to the object are shaded.

$$I(o, s) = \frac{OV_o' \cdot SV_s}{||OV_o||} \tag{4.33}$$

Example 4.3.2. *We now measure the similarity between a source and a book. Assume there are 4 domains in total, saying "Category: Science", "Price in [$10, $20]", "Published in [1990, 1999]" and "Category: Arts". We can calculate* $e_d(s)$ *for this source and generate* SV_s *as* $(e_{science}(s), e_{[\$10,\$20]}(s), e_{[1990,1999]}(s), e_{arts}(s))'$. *We assume the book is a "Science" book, published in 1995, with a price of $19.99. Then we generate* OV_o *as* $(1, 1, 1, 0)'$.

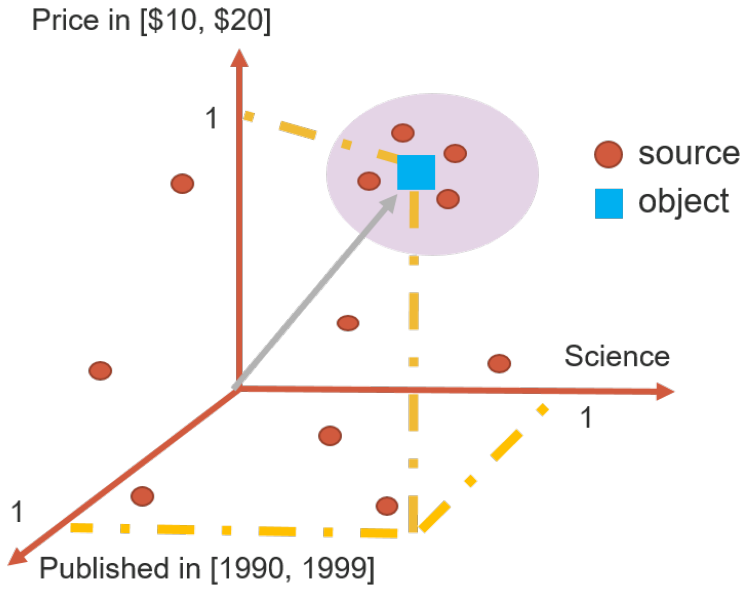

Figure 4.5: Three dimensional similarity example

4.3.2 Multi-Truth Confidence

There are two add-on factors in our proposed model: one is the object-source similarity and the other is the partial exclusions between claimed values. We have discussed the object-source similarity in previous section and we present the partial exclusion in this section.

In the context of multi-truth inference, the conflicting values are no longer that mutually exclusive as the single-truth one. However, they are still against each other to some extent. There are two aspects for modeling the partial exclusion: one is the intra-exclusion between the values claimed by a source and the other is the inter-exclusion between the values claimed by various sources. For the first one, an intuitive idea is that we might vary the confidence of the given values source by source. If a source claims lots of values for an object, we tend to lower the confidence of the provided values as for this source; if a source claims a small number of values for an object, we tend to elevate the confidence of the claimed values as for this source. This model is consistent with the nature of source, i.e. a source providing more values then the other is assumed to be more adventurous. As for the other aspect, if a value that is not provided by a source is provided by the other sources, there are still a partial supports for this value from this source, since there is no direct objection. Considering these two aspects, a partial exclusion modeling is first modeled in [7] and extended in [6]. We now present this model.

The $V(o)$ is the claimed value set for object o. For a source and its $V_s(o)$, we should assign confidence score for both the claimed values and the unclaimed values. Notice that $V_s(o) \subseteq V(o)$, we want the sum of confidence score for value in $V_s(o)$ to be 1 and we want some confidence difference between the $v \in V_s(o)$ and $v \notin V_s(o)$. The confidence score is calculated as **Equation 4.34**.

$$c_s(v) = \begin{cases} (1 - \frac{|V(o) \setminus V_s(o)|}{V(o)^2}) \frac{1}{|V_s(o)|}, v \in V_s(o) \\ \frac{1}{|V(o)|^2}, v \notin V_s(o) \end{cases} \tag{4.34}$$

A example for calculating the confidence score is given as follows. Assume $V(s) = a, b, c$ and $V_o(s) = a, c$. Then the confidence score for the claimed value is $(1 - \frac{1}{9}) \cdot \frac{1}{2} = \frac{4}{9}$ and the confidence score for the unclaimed value is $\frac{1}{3^2} = \frac{1}{9}$. As for the truth inference model ignoring this partial exclusion, they implicitly set the confidence score of unclaimed value as 0.

The confidence score for claimed value with varying $|V(o)|$ is shown in **Figure 4.6**. It is clear that as the number of claimed value becoming larger, the confidence score becomes smaller, reflecting the enhancement of intra-exclusion. The ratio of confidence score for claimed value and unclaimed value with varying $|V(o)|$ is shown in **Figure 4.7**. It is clear

Figure 4.6: Confidence score variation

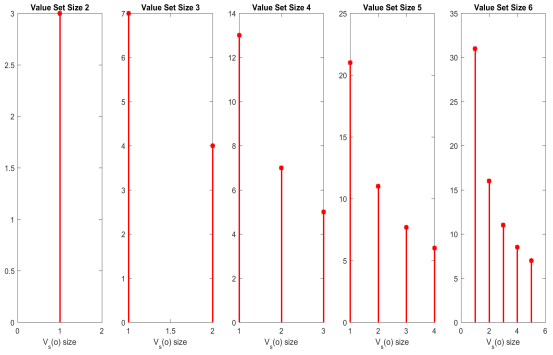

Figure 4.7: Confidence score ratio

that as the number of claimed value becoming larger, the confidence score ratio becomes smaller, reflecting the reduction of inter-exclusion.

4.3.3 Bayesian Inference

Within this part, a Bayesian model is presented to perform the truth inference task. To infer the truth, we need to compute the probability of a claimed value to be true, i.e. $\sigma(v)$. The intuitive idea to formulate $\sigma(v)$ is that $\sigma(v)$ depends on both supports and oppositions. As discussed before, we model the source quality in two different measurement, which is ready to express the support or opposition to a value. We first give detailed illustration about the proposed Bayesian model and then present the pseudo code of this model, along with complexity analysis. Once we finish this stage, the inferred truth database $\mathcal{DB}_t$ is generated and we complete the whole truth inference process.

We use $\psi(o)$ as the observation of claims from the sources for object o. $\sigma(v)$ is used to represent *a prior* veracity for v. Our target now turns to estimating the probability that v is true, given $\psi(o)$, i.e. $Pr(v|\psi(o))$. Based on Bayesian rules, we can express $Pr(v|\psi(o))$ as **Equation 4.35**.

$$\begin{aligned} Pr(v|\psi(o)) &= \frac{Pr(\psi(o)|v)Pr(v)}{Pr(\psi(o))} \\ &= \frac{Pr(\psi(o)|v)\sigma(v)}{Pr(\psi(o)|v)\sigma(v) + Pr(\psi(o)|\overline{v})(1-\sigma(v))} \\ &= \frac{1}{1 + \frac{1-\sigma(o)}{\sigma(o)} \cdot \frac{Pr(\psi(o)|\overline{v})}{Pr(\psi(o)|v)}} \end{aligned} \tag{4.35}$$

We can now consider $Pr(\psi(o)|v)$ and $Pr(\psi(o)|\overline{v})$ by representing them in a the form of sources reliability and we tend to include both the support and opposition of sources. We first model the two-side quality of source in this truth inference model. We use $O(s)$ to represent a set of objects that source s claims value about. Following the expression of

recall and *specificity*, we can estimate *TP*, *FP*, *FN* and *TN* as **Equation 4.36** and **Equation 4.37**.

$$\hat{TP}(s) = \sum_{o \in O(s)} \sum_{v \in V_s(o)} \sigma(v),\ \hat{FN}(s) = \sum_{o \in O(s)} \sum_{v \notin V_s(o)} \sigma(v) \tag{4.36}$$

$$\hat{FP}(s) = \sum_{o \in O(s)} \sum_{v \in V_s(o)} (1 - \sigma(v)),\ \hat{TN}(s) = \sum_{o \in O(s)} \sum_{v \notin V_s(o)} (1 - \sigma(v)) \tag{4.37}$$

We can then express *recall* of source s, denoted as $\tau^{re}(s)$, as **Equation 4.38**, and we can express *specificity* of source s, denoted as $\tau^{sp}(s)$, as **Equation 4.39**.

$$\tau^{re}(s) = \frac{\hat{TP}(s)}{\hat{TP}(s) + \hat{FN}(s)} \tag{4.38}$$

$$\tau^{sp}(s) = 1 - \frac{\hat{FP}(s)}{\hat{FP}(s) + \hat{TN}(s)} \tag{4.39}$$

We use $S_o(v)$ to represent a set of source that claim value v for object o, and we use $S_o(\overline{v})$ to represent a set of source that do not claim value v for object o. Then, we can express the probability that the sources supporting v are correct as $\prod_{s \in S_o(v)} \tau^{re}(s)$ and the probability that the sources not supporting v are wrong as $\prod_{s \in S_o(\overline{v})} (1 - \tau^{sp}(s))$. Given this, we can express $Pr(\psi(o)|v)$ as **Equation 4.40** and $Pr(\psi(o)|\overline{v})$ as **Equation 4.41**.

$$Pr(\psi(o)|v) = \prod_{s \in S_o(v)} \tau^{re}(s) \prod_{s \in S_o(\overline{v})} (1 - \tau^{sp}(s)) \tag{4.40}$$

$$Pr(\psi(o)|\overline{v}) = \prod_{s \in S_o(\overline{v})} \tau^{re}(s) \prod_{s \in S_o(v)} (1 - \tau^{sp}(s)) \tag{4.41}$$

We then extend this Bayesian model by involving the source-object similarity, introduced in Section 4.3.1, and multi-truth confidence, introduced in Section 4.3.2. In order to avoid the elimination of these factors during the iteration, we model them as powers of the model, as the proposed model in [6]. The final expression for $Pr(\psi(o)|v)$ and $Pr(\psi(o)|\overline{v})$ are shown as **Equation 4.42** and **Equation 4.43** respectively.

$$Pr(\psi(o)|v) = \prod_{s \in S_o(v)} \tau^{re}(s)^{I(o,s)c_s(v)} \prod_{s \in S_o(\overline{v})} (1 - \tau^{sp}(s))^{I(o,s)c_s(v)} \tag{4.42}$$

$$Pr(\psi(o)|\overline{v}) = \prod_{s \in S_o(\overline{v})} \tau^{re}(s)^{I(o,s)c_s(v)} \prod_{s \in S_o(v)} (1 - \tau^{sp}(s))^{I(o,s)c_s(v)} \tag{4.43}$$

4.3.4 Stage 2 Inference Algorithm

The pseudo code of the Bayesian truth inference model is presented in **Algorithm 4.3**.

Algorithm 4.3: SATM : Bayesian inference

Input: $\mathcal{DB}_{raw}, \mathcal{DB}_{oa}, S^{re}, S^{sp}$
Output: $\mathcal{DB}_t$

{Initialization}
for *all* $s \in S$ **do**
  for *all* $o \in O(s)$ **do**
    Calculate $I(o, s) \leftarrow$ **Equation 4.33**
  $\tau^{re}(s) \leftarrow s^{re} \in S^{re}$
  $\tau^{sp}(s) \leftarrow s^{sp} \in S^{re}$
for *all* $o \in O$, s *and* $v \in V_s(o)$ **do**
  $\sigma(v) \leftarrow$ **default value**
  $c_s(v) \leftarrow$ **Equation 4.34**
{Updating Bayesian probability}
while *unconverge* **do**
  for *all* $o \in O$, s *and* $v \in V_s(o)$ **do**
    $Pr(\psi(o)|v) \leftarrow$ **Equation 4.42**
    $Pr(\psi(o)|\overline{v}) \leftarrow$ **Equation 4.43**
    $\sigma(v) \leftarrow$ **Equation 4.35**
  for *all* $s \in S$ **do**
    $\tau^{re}(s) \leftarrow$ **Equation 4.38**
    $\tau^{sp}(s) \leftarrow$ **Equation 4.39**
for *all* $o \in O$ **do**
  for *all* $v \in V(o)$ **do**
    if $\sigma(v) \geq \theta$ **then**
      Insert v *to* $\mathcal{DB}_t$

The source-object similarity is calculated for each source and its associated objects and this value maintains unchanged within truth inference. We then use the source quality retrieved from previous stages for initialization for $\tau^{rec}(s)$ and $\tau^{sp}(s)$ and calculate the corresponding confidence score of values for each source. The veracity score is initialized with predefined value. Within the recursive updating, we recalculate the source quality based on an updated veracity score until the change of veracity score remaining in an interval. As for final output, we select those values with veracity score higher than a predefined threshold θ to be true for corresponding objects.

We now analyze the time complexity of **Algorithm 4.3**. The time complexity to calculate the source-object similarity is $O(|C|)$, if we assume the vector dimension to be a constant. Also, since we calculate the confidence score for each value in $V(o)$ for each

source, the corresponding cost is $O(|C|)$. Meanwhile, the time complexity for each run of recursive update is also $O(|C|)$. Thus, to summarize, the time complexity is $O(|C|)$ for this stage, which is linear to the total number of claims.

4.4 Combined Truth Inference Model

In this section, we present pseudo code of the proposed integrated two-stage truth inference model in **Algorithm 4.4**. For stage 1 and stage 2, we have analyzed the time complexity, which are both linear to the number of claims. As for the proposed intermediate stage, since there is a matrix inverse operation, the possible time complexity could be cubic to the dimensions of source attribute. However, in practical, even cubic to source attribute dimensions is much smaller than the number of claims. Thus, we conclude the time complexity of our proposed truth inference model as $O(|C|)$, which is linear to the total claim number.

Algorithm 4.4: Two-stage Truth Inference

Input: $\mathcal{DB}_{raw}, \mathcal{DB}_{oa}, (\mathcal{DB}_{sa})$
Output: $\mathcal{DB}_t$, ($\hat{\boldsymbol{w}^{re}}$, $\hat{\boldsymbol{w}^{re}}$)

{Retriving S^{re} and S^{sp}}
Perform TDM, given parameters

{Updating S^{re} and S^{sp}}
Perform intermediate stage, given parameters

{Inferring truth and generating $\mathcal{DB}_t$}
Perform SATM, given parameters

5. EXPERIMENTS

Two real-world datasets are used to test the performance of our model. We first describe the meta data of these two datasets in Section 5.1. The performance comparisons with state-of-art approach and model efficiency are demonstrated in Section 5.2. In Section 5.3, we analyze the parameters sensitivity. Finally, several further discussions to exploit the model and dataset are conducted in Section 5.4.

All the experiments presented are conducted on a server with 128GB RAM, 3.60GHz CPU, with CentOS Linux Release 7.6.1810 installed. All the algorithms including data cleaning were implemented in Python 3.6.1.

5.1 Data Description

We conduct experiments on two real-world datasets to evaluate our framework. These two datasets were originally created by [6], crawled from websites. As for source-object similarity measurements, we utilized some existing object attributes in the datasets, e.g. *"Published Year"* and *"Price"* and collected extra information from website, e.g. *"Country"*. However, there is no source attribute information available. Thus we obtained traffic statistics for source from *Alexa.com* for sources in one of the dataset. The detailed illustration for datasets are shown as follows.

1. **BOOK**
   This dataset is originally collected by [6] from *AbeBooks.com*, which is a book transaction website, in April 2017. The original dataset contains 54,591 different registered book-sellers, providing 2,338,559 listing information for 210,206 books. Each book is identified by its ISBN-13. The object attributes include *"Price"*, *"Published Year"* and *"Category"*. We did not collect new object attributes for *BOOK*. The original data contains noise and we preformed pre-cleaning on both the provided author claims and the object attributes for this dataset and we illustrates as below.

   - For the provided author claims, we remove the book with only one source or seller providing information about. And we cleaned some extra information presenting in some author claims, such as roles, and unify the name representation to generate a dataset with only books having conflicting value sets provided by sources. This pre-cleaning is conducted in coarse manner. It is possible that each source have its own naming conventions. Since the source number is great, the cleaning quality is low and it is relatively hard to extract claimed values from data, which is not the same case for **MOVIE** dataset.

- For object attributes, we also did some treatment to guarantee consistency. As for *"Published Year"*, we eliminated possible conflicting information to a unified one. Also, since this website is a transaction one, there are lots of different prices for one book. Thus, as for *"Price"*, we collected all listed prices of a book and use their average value as its *"Price"* attribute. As for *"Category"*, there are 18 categories in original dataset and it is possible that a book have multiple categories. The 18 categories are respectively *arts*, *crime*, *children*, *cookbook*, *romance*, *religion*, *reference*, *craft book*, *history*, *science*, *horror*, *travel*, *literature*, *science fiction*, *business*, *self-help*, *social* and *biography*. We maintained all possible categories for a book and extend them to different dimensions in source-object similarity measurements.

After pre-cleaning, we generate a dataset with 6,480 sources, providing information for 65,826 books. Since sources might provide no information for book and we did not generate fact for "no", each sources provide 0.0%(0 fact) to 69.68%(140,058 facts) of books. Also, each source is associated with 1 to 140,058 facts. On average, there are 26.71 sources providing information and 3.38 different sets of authors for a book. We randomly select 400 books from the generated conflicting dataset as our test database. For books in test database, we manually labeled the truth of their authors. For every round of experiment, 120 books are randomly chosen as a test set and we repeat for 10 times.

2. **MOVIE**

   This dataset is also originally collected by [6] from 15 movie-related website, including *imdb*, *amazon*, *goodfilms*, *metacritic*, *letterboxd*, *movieinsider*, *amazon*, *moviefone*, *dewaontons*, *filmcrave*, *top250tv*, *allmovie*, *instantwatcher*, *1moviesonline* and *agoodmovietowatch*. We identify a movie by the combination of its released year and title. The original dataset contains *"Released Year"* and *"Genres"* as object attributes and we adding *"Country"* as extended object attribute. The *"Country"* information for movies is collected from *themoviedb.org*. There is noise in original dataset and we performed pre-cleaning on both the object attributes and provided director information. We illustrate our pre-cleaning as follows.

   - For the provided director claims, we remove the movie with only one website source providing information about. And we cleaned some garbled characters and unify the name representation to generate a dataset with only movies having conflicting value sets provided by sources. This pre-cleaning is also conducted in coarse manner. It is still possible that each source have its own naming convention. However, since the source number is limited, the cleaning quality is

high and it is easy to extract claimed values from data.

- For object attributes, some treatment to guarantee consistency is performed. As for *"Genres"*, since some provided *"Genres"* values have some similar meaning but different illustrations, we unify representation of *"Genres"*. As a result, there are in total 37 genres. As for *"Country"*, there are in total 108 countries.

After pre-cleaning, we generate a dataset with 15 sources, providing information for 18,446 books. Each sources provide 0.29%(53 fact) to 82.86%(15285 facts) of books. Also, each source is associated with 134 to 32,856 facts. On average, there are 3.15 sources providing information and 2.09 different sets of authors for a book. We randomly select 320 movies from the generated conflicting dataset as our test database. For movies in test database, we manually labeled the truth of their directors. For every round of experiment, 100 movies are randomly chosen as a test set and we repeat for 10 times.

3. **SOURCE ATTRIBUTE**
   We obtained traffic statistics from *Alexa.com* for sources in **MOVIE** dataset. The traffic statistics contains: (i) global ranks, (ii) total sites linking in, (iii) search visits, (iv) country rank, (v) daily page views per visitor, (vi) bounce rate, and (vii) daily time on site. All attributes take numeric values. Also, we identify that the claimed values of some sources in **MOVIE** are provided by users, instead of the officials, such as *Letterboxd*. Thus we generate an extra tag, (viii) official, to identify this difference, with 1 as *not official* and -1 as *official*. Moreover, the functions of some sources are not specific for movies or other related information. For example, *Amazon* is known as a online-shopping website, instead of a movie website as *IMDB*. Again, we generate another tag, (ix) specific, to capture this difference, with 1 as *specific* and -1 as *not specific*.

5.2 Model Performance Validation

5.2.1 Baselines and Metrics

We compare our models with several state-of-arts techniques and some naive voting strategies. These techniques are briefly summarized as follows, and refer readers to the original publications for details.

Majority Voting treats the value with maximum number of supporters as truth for each object.

LTM[1] presents a graphical model and utilize Gibbs sampling to estimate the source quality and infer the truth. It models two types of source quality under multi-truth context: *recall* and *specificity*.

DART[6] incorporates both the domain expertise score and confidence score in truth inference. For each source, a domain-dependent quality is utilized to determine the value truth for objects in that domain. We use *"Genres"* for **MOVIE** and *"Category"* for **BOOK**.

SRV is our naive model. It utilize the source quality inferred from the first stage, **TDM**, of our two-stage truth inference model and calculate the veracity score for each value using **Equation 4.35**. We use $\sigma(v) = 0.5$ and calculate one time. The value with veracity score ≥ 0.4, instead of 0.5, is inferred as true, since the estimation is coarse.

TSTM is our proposed **T**wo **S**tage **T**ruth Inference **M**odel, which integrates attributes and source-object similarity into truth inference. For both **MOVIE** and **BOOK**, we run the model without attributes but merely adjust *recall* and *specificity*.

TSTM-Attr is **TSTM** with source attributes. For **MOVIE**, since there are source attributes available, we run the model with source attributes and demonstrate their performance. For **BOOK**, since there is no attribute available, we do not run this model.

TSTM-Top only selects the sources with highest similarity to be involved in **TSTM**. In **BOOK**, for each book, we select top 50% sources as value providers. In **MOVIE**, for each movie, we select top 60% sources as value provider. This effect of portion for top sources selection is examined in Section 5.3. We run the model without including source attributes.

We set the parameters for the model above according to the optimal settings suggested by their authors. For our method, we set γ_{re} and γ_{sp} for **Equation 3.1** as 0.8 and 0.9 to narrow the quality difference since the source estimation is only approximate. And we set α for **Equation 4.32** as 1.5 for **MOVIE** and 2 for **BOOK**. This setting is because the data quantity distribution for domains is uneven in both datasets and some sources provide very-low percentage of data. Without elevating their domain expertise, their contribution in Bayesian probability inference will be eliminated. The *a prior* veracity for **Equation 4.35** is set as 0.5. Meanwhile, for our model, the iteration number for **TDM**, the first stage, is set as 7, with burn-in as 2 and sample gap as 1. The iteration number for **SATM**, the second stage, is set as 10.

5.2.2 Method Comparison

Table 5.1 shows the performance of different algorithm on two datasets in terms of *precision*, *recall* and *F1-measure*. Our method achieves relatively high *recall*, while maintaining high *precision*. The *F-measure* is the harmonic mean of *recall* and *precision* (i.e. $F1 = \frac{2*precision*recall}{precision+recall}$). The high *F1* indicates the good performance of our proposed model. Through selecting most similar sources to provide claims about objects, the *recall* and overall *F1* increments a little, due to the relatively high quality of those sources. For our model

involving source attributes, the final quality is the same as our raw model, indicating the feasibility of directly using source attributes to estimate source quality for initialization.

Since we only select the most trustworthy answers after source qaulity estimation, our naive model **SRV** reach high *precision* in both datasets, which is rational, which is the same case as **Majority Vote**.

As for **DART**, our experiments indicate that it tends to underestimate the *specificity*, leading high *recall* and low *precision*. This method performs good when the overall data quality is high, which is the case of **MOVIE**. However, as for low-quality data, which is the case of **BOOK**, the model shows a great *precision* degradation. This trend is also reflected in our experiments on noise-added data in Section 5.4.3.

LTM demonstrates good *precision*. However, due to the initialization flaws that we illustrated above, the method tends to overestimate *specificity* of sources, leading low *recall*. The inferred source quality and the comparison with our model are shown in Section 5.4.2.

Table 5.1: Model performance evaluation on datasets.

Methods	***BOOK dataset***			***MOVIE dataset***		
	precision	**recall**	**F1-measure**	**precision**	**recall**	**F1-measure**
Majority Vote	**0.8425**	0.7399	0.7873	0.8348	0.5776	0.6815
LTM	0.7951	0.8835	0.8368	0.8559	0.78	0.8094
DART	0.4411	**0.9539**	0.6031	0.7838	**0.9262**	0.8487
SRV	**0.8952**	0.6691	0.7657	**0.9651**	0.3655	0.5296
TSTM	0.7959	0.8971	**0.8433**	**0.8561**	0.8947	**0.8747**
TSTM-Attr	NA	NA	NA	**0.8561**	0.8947	**0.8747**
TSTM-Top	0.7934	**0.8983**	**0.8424**	0.8556	**0.8975**	**0.8758**

5.2.3 Model Efficiency

We examine the execution time of each algorithm. We create 5 small datasets using **MOVIE** by randomly sampling 2K, 4K, 6K, 8K and 10K movies from the entire dataset. We run each algorithms for 10 times and take the average. The time cost for above methods is shown in **Table 5.2.**

To further verify the proposed time complexity which is linear to claims number or objects number, we run linear regression on the execution time of **TSTM**, which yields an R^2 of 0.9964, demonstrating the scalability of **TSTM**. The regression is shown as **Figure 5.1**.

Moreover, since our proposed truth inference model is an iterative algorithm, we need to estimate the required iteration to achieve good performance, i.e. F-measure. We examine our model using iteration 1, 2, 3, 5, 10, 15, 20, 50, 100, 200, 300 for merely stage 2 and test

Table 5.2: Average execution time of all algorithms (in seconds)

	Runtime (seconds) vs. #Objects				
#Objects	2000	4000	6000	8000	10000
Majority Vote	0.0061	0.0201	0.0207	0.03037	0.0656
SRV	0.533	1.0724	1.6059	2.1875	2.7754
LTM	2.4478	4.8992	7.3295	10.027	12.6134
DART	**0.4238**	1.2736	2.6193	3.8795	4.7505
TSTM	0.5966	**1.2231**	**1.8759**	**2.6626**	**3.4722**
TSTM-Top	**0.4375**	**0.8870**	**1.3614**	**1.9535**	**2.4860**

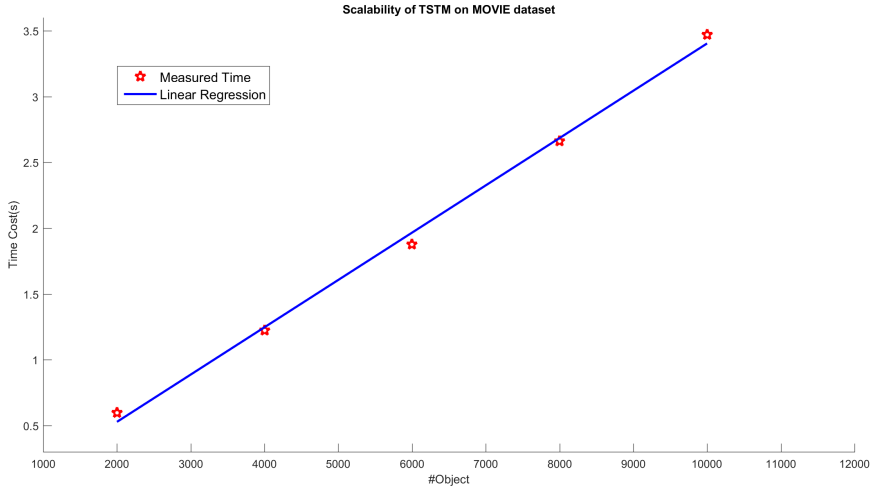

Figure 5.1: Linear regression for model execution time

its performance on our test database. We repeat each for 10 times and the result shows that our method reach the best performance after merely 5 iterations and additional iterations provide no performance elevation, indicating a fast convergence after a small number of iterations.

5.3 Parameter Sensitivity

We explore the impact of the different parameter settings of our proposed model. Since we propose a source-dependent initialization method for source quality and this initialization depends on the given *ratio matrix*, we want to capture the relationship between *ratio matrix* and the inferred truth. By varying the ratio in ratio matrix, the prior belief for source is changed. We maintain the relative proportion for (TP_s, FN_s) and (FP_s, TN_s) and vary the ratio for (TP_s, FN_s). We respectively set *ratio matrix* as (1, 1, 1, 99), (10, 10, 1, 99), (20, 20, 1, 99), (30, 30, 1, 99), (40, 40, 1, 99) and (50, 50, 1, 99). The first two items in tuple represent TP_s and FN_s, and the other two items represent FP_s and TN_s. We main-

tain other parameter setting as illustrated above and conduct experiments on **MOVIE** and **BOOK**. The *recall*, *precision* and *F1-measure* are shown in **Figure 5.2** and **Figure 5.3**.

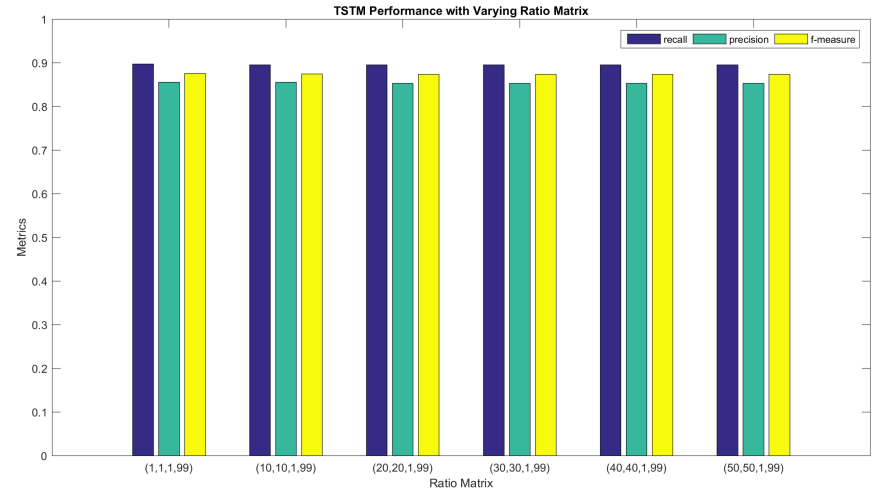

Figure 5.2: *ratio matrix* on **MOVIE** dataset

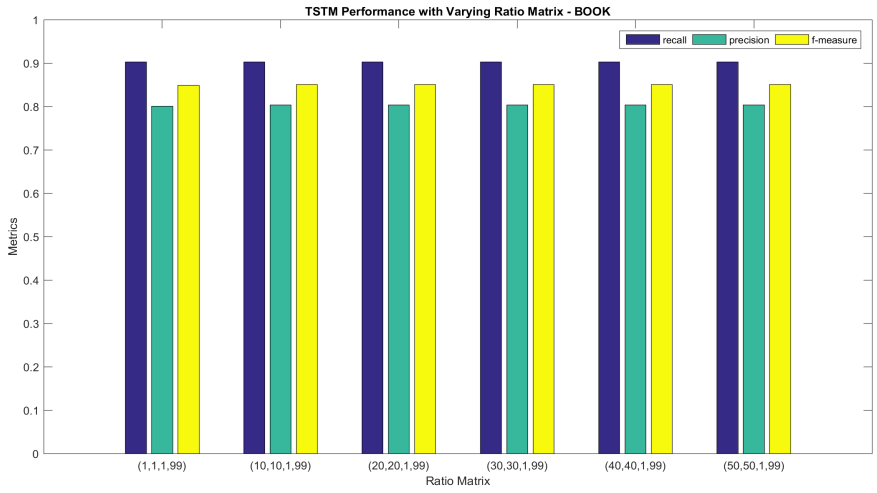

Figure 5.3: *ratio matrix* on **BOOK** dataset

The effect of the selected source portion for **TSTM-Top** is also examined. The plotted *F1-measure* in **Figure 5.4** indicates the best setting is around 50% for **BOOK** and 60% for **MOVIE**.

5.4 Further Exploitation On Model and Datasets

Some further discussions on both datasets and the proposed model are illustrated as follows. In this section, we first discuss an adjusted model performance, inspired by the missing values in our fundamental database. We then evaluate the retrieved source qualities

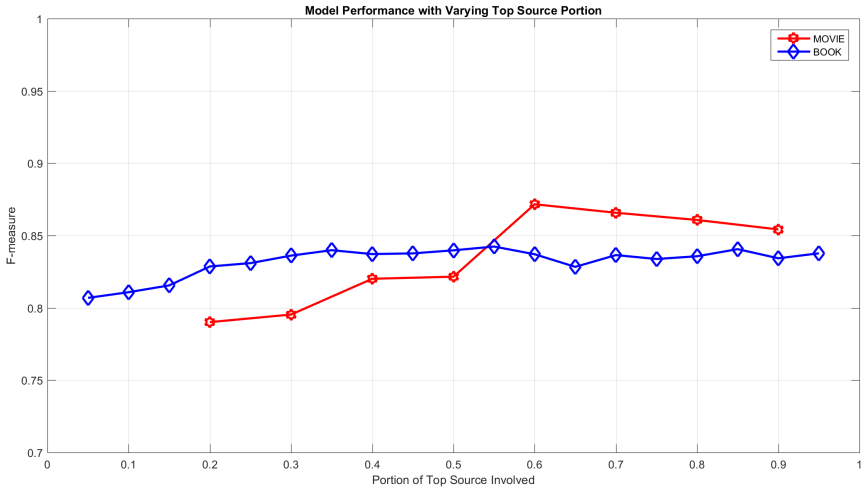

Figure 5.4: Effect of top source portions

and the generality of retrieved source attributes. We present a experiment with low-quality data to check our model performance in the last.

5.4.1 Data Refinement Check

As for the objects selected to test database, we perform a refined check on the claimed values in our generated database. An intuition to conduct this refined check is that if some values in truth are totally unavailable in our dataset, i.e. not provided by any sources, it is impossible for us to perform truth inference to infer that values. We want to capture this gap and perform a more reasonable performance check. We manually check whether the truth values present in our fundamental database for **MOVIE** and **BOOK**. We use a strict evaluation method, i.e. the inferred truth can only be correct if it is exactly the same as the labeled truth. In our database, there are some values are indeed missing. After we subtracting the number of these values from our *FNs*, we can retrieve an adjusted model performance, which is more proper to represent the actual situation.

5.4.2 Case study: Source Quality Prediction & Role of Source Attributes

As a side output, the source quality, including *recall* and *specificity*, are also estimated in our model, based on the inferred truth of provided values. Since we have realized the shortage of uniform initialization, we compare the source quality inferred from [1] with ours. The source quality inferred from **LTM** and our **TSTM** is shown in **Table 5.3**. It is clear that **LTM** tends to overestimate the *specificity* of sources, especially for those sources that are associated with relatively few facts. This would result low recall in its inferred truth, which is the same as our experiments. As for our model, since we identify the nature

difference between sources, the inferred source qualities tend to more effective and show better inference performance.

Table 5.3: Inferred source quality and comparison

	TSTM		*LTM*	
source	***recall***	***specificity***	***recall***	***specificity***
imdb	0.8064	0.8621	0.8139	0.8502
dewanontons	0.8218	0.9436	0.4771	0.9848
top250tv	0.8391	0.9425	0.917	0.9632
matacritic	0.864	0.9497	0.8908	0.9814
letterboxd	0.453	0.4743	0.4995	0.8275
goodfilms	0.7335	0.8289	0.7431	0.8433
instantwatcher	0.7168	0.6941	0.7769	0.9861
filmcrave	0.7682	0.8526	0.7927	0.9363
moviefone	0.6058	0.5342	0.6668	0.9664
amazon	0.4314	0.2475	0.5491	0.9372
movieinsider	0.4808	0.4587	0.5421	0.9606
1movieonline	0.7825	0.9257	0.8306	0.9819
allmovie	0.3997	0.4303	0.4946	0.9659
ifcfilms	0.355	0.4793	0.4771	0.9892
agoodmovietowatch	0.6536	0.719	0.6725	0.9891

Also, since we have included source attributes in our inference model, we want test generality to see whether the inferred weight can be directly used to estimate source quality in a relatively accurate manner. We use **MOVIE** for attribute generality validation. The weights for source attributes retrieved by using 12 of 15 sources in **MOVIE**. Using these weights, we can generate source qualities, i.e. *recall* and *specificity*, for the other 3 sources. Then we compare them with the results of first-stage source quality estimation using full database. The source qualities of the generated ones and full estimated ones are shown in **Table 5.4**. There is small gap between the them. Thus once we retrieving the attributes weights, the stage 1 can be eliminated or run in batch manner.

Table 5.4: Source quality estimation comparison

	TDM (stage 1) Inferred		*Generated*	
source	***recall***	***specificity***	***recall***	***specificity***
goodfilms	0.7323	0.9457	0.6836	0.9245
dewanontons	0.8608	0.9708	0.8695	0.9881
movieinsider	0.5303	0.8846	0.5972	0.8461

5.4.3 Data with Low Overall Quality

In practical, there are noises in data. In order to measure the applicability of our proposed model in real-world circumstance, we conduct experiment on **MOVIE** with different portions of extra noise added on the provided values. With added noise, there are gabbled character in the provided claims, causing difficulties for truth inference. The experiment results is shown in **Figure 5.5**, **Figure 5.6** and **Figure 5.7**. Our model demonstrate relatively good performance with increasing noise. And it is clear that **DART** encounters fast *precision* degradation when the noise portions increasing, while **LTM** maintains high *precision*.

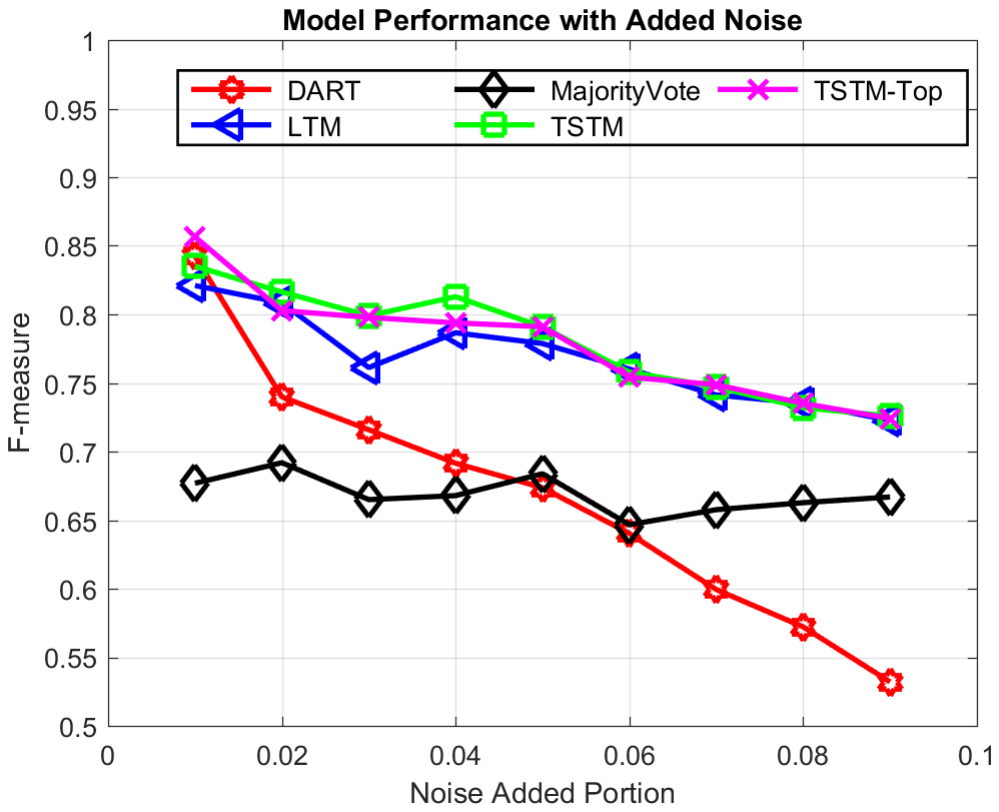

Figure 5.5: Model performance with added noise

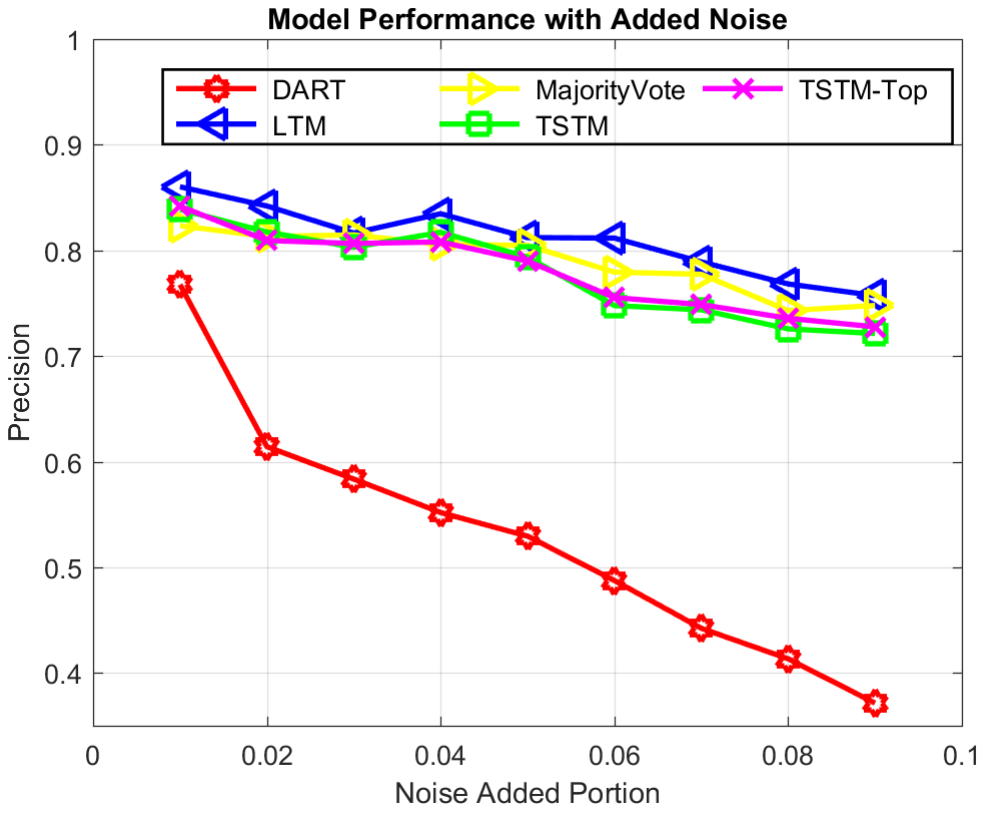

Figure 5.6: Model performance with added noise

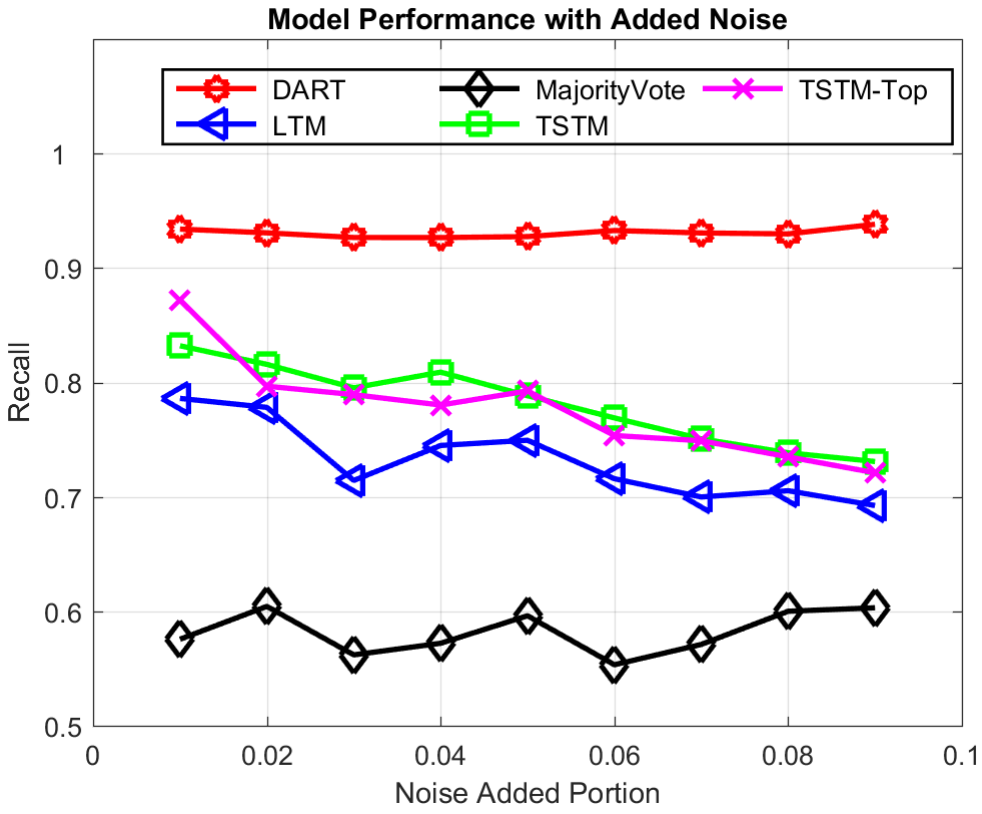

Figure 5.7: Model performance with added noise

6. RELATED WORKS

Extensive works have exploited the area of data fusion, targeting on resolving possible conflicts and determining the underlying truth. This problem is first developed under the context of single-truth assumption and then extended to a multi-truth one. Some works devote to identify the single truth with source quality estimation [5, 15, 16]. The Bayesian based algorithm is first proposed by [16], in which the truth finding problem is formally formulated as probability model. And [1, 6, 7] identify the difference between single-truth and multi-truth inference and model the source quality considering two side. [1] model the source quality considering two type of errors and proposed a graphical probabilistic model. Moreover, some works incorporates the domain knowledge and some features to infer the truth. [9] utilize *EM* and *ERM* algorithm to infer the weights of those reliability-related features and thus perform truth inference. Also, some past works focus on other aspects of truth inference problem, e.g. detecting the copy relationship between sources.

7. CONCLUSION AND FUTURE WORK

We examine the problem of multi-truth inference with object-source similarity considered. To obtain object-oriented source qualities, we integrate domain expertise and extend reliability measure based on object-source similarity, which can be further used as metrics for source selection. We also propose a reasonable initialization method . Moreover, we propose a unsupervised method to enable fast source quality estimation based on source attributes. The confidence score in multi-truth context proposed in previous works is also considered. We propose a integrated two-stage model using graphical probability modeling and Bayesian approach to incorporate object-source similarity and initialization, aiming to find the possible multiple truths without any supervision. Experimental results on two real-world datasets demonstrate the feasibility and effectiveness of our proposed model.

However, there are still challenges in our problems. Tremendous efforts are devoted to data cleaning job, which is tedious and exhausted. Lots of special treatments for characters and unnecessary information including people titles cause annoying problems. Since different sources might use a different naming convention, how to perform automatic and high-quality data cleaning becomes another problem for multi-source datasets. Once the truth inference model is decided, the model performance depends on the quality of provided data. Thus truth discovery model can be used as metrics to measure the data cleaning quality. Moreover, the proposed weight estimation for source attributes might have some nonlinear relationships, which is not captured by our model. In future we will try to model the source quality estimation in a more general manner and apply NLP techniques for data cleaning.

References

[1] Zhao B, Rubinstein B I, Gemmell J, et al. A bayesian approach to discovering truth from conflicting sources for data integration. Proceedings of the VLDB Endowment, 2012, 5(6):550–561.

[2] Pasternack J, Roth D. Making better informed trust decisions with generalized fact-finding. Proceedings of Twenty-Second International Joint Conference on Artificial Intelligence, 2011.

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

[4] Galland A, Abiteboul S, Marian A, et al. Corroborating information from disagreeing views. Proceedings of Proceedings of the third ACM international conference on Web search and data mining. ACM, 2010. 131–140.

[5] Yin X, Tan W. Semi-supervised truth discovery. Proceedings of Proceedings of the 20th international conference on World wide web. ACM, 2011. 217–226.

[6] Lin X, Chen L. Domain-aware multi-truth discovery from conflicting sources. Proceedings of the VLDB Endowment, 2018, 11(5):635–647.

[7] Wang X, Sheng Q Z, Fang X S, et al. An integrated bayesian approach for effective multi-truth discovery. Proceedings of Proceedings of the 24th ACM International on Conference on Information and Knowledge Management. ACM, 2015. 493–502.

[8] Li Q, Li Y, Gao J, et al. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. Proceedings of Proceedings of the 2014 ACM SIGMOD international conference on Management of data. ACM, 2014. 1187–1198.

[9] Joglekar M, Rekatsinas T, Garcia-Molina H, et al. SLiMFast: guaranteed results for data fusion and source reliability. arXiv preprint arXiv:1512.06474, 2015..

[10] Kuleshov V, Ermon S. Stanford CS228 - Probabilistic Graphical Models. https://ermongroup.github.io/cs228-notes/.

[11] Bayesian Updating: Continous Priors. https://ocw.mit.edu/courses/mathematics/18-05-introduction-to-probability-and-statistics-spring-2014/class-slides/MIT18_05S14_class14slides.pdf.

[12] Piech C. Beta Distribution. https://web.stanford.edu/class/archive/cs/cs109/cs109.1176/lectureHandouts/15%20Beta.pdf.

[13] B W. Markov Chain Monte Carlo and Gibbs Sampling. https://www3.ime.usp.br/~jstern/miscellanea/LabSimulacao/Walsh04.pdf.

[14] Miller S J. The Method of Least Squares. https://web.williams.edu/Mathematics/sjmiller/public_html/BrownClasses/54/handouts/MethodLeastSquares.pdf.

[15] Zhao B, Han J. A probabilistic model for estimating real-valued truth from conflicting sources. Proc. of QDB, 2012..

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