Truth Finding with Attribute Partitioning

M. Lamine Ba
Institut Mines–Télécom
Télécom ParisTech; CNRS LTCI
Paris, France
ba@telecom-paristech.fr

Roxana Horincar
Institut Mines–Télécom
Télécom ParisTech; CNRS LTCI
Paris, France
horincar@telecom-paristech.fr

Pierre Senellart
Télécom ParisTech; CNRS LTCI
& NUS; CNRS IPAL
Paris, France & Singapore
senellar@telecom-paristech.fr

Huayu Wu
A*STAR
I2R
Singapore
huwu@i2r.a-star.edu.sg

ABSTRACT

Truth finding is the problem of determining which of the statements made by contradictory sources is correct, in the absence of prior information on the trustworthiness of the sources. A number of approaches to truth finding have been proposed, from simple majority voting to elaborate iterative algorithms that estimate the quality of sources by corroborating their statements. In this paper, we consider the case where there is an inherent structure in the statements made by sources about real-world objects, that imply different quality levels of a given source on different group of attributes of an object. We do not assume this structuring given, but instead find it automatically, by exploring and weighting the partitions of the sets of attributes of an object, and applying a reference truth finding algorithm on each subset of the optimal partition. Our experimental results on synthetic and real-world datasets show that we obtain better precision at truth finding than baselines in cases where data has an inherent structure.

1. INTRODUCTION

Many real-world applications, such as multi-source Web data integration systems or online crowdsourcing platforms, face the problem of *discovering the truth* when integrating conflicting information from a collection of sources with different (and often unknown) levels of accuracy. Such applications can use *truth finding algorithms* [1,7,9,13,16,17], which aim at discovering true facts by determining the likelihood that a given data instance describes the reality using estimated values of source accuracies.

We consider in this work the situation where conflicting statements are made about objects with inherent *structure*, and where the reliablity of sources is correlated following this structure: for example, examinations can naturally be decomposed into separate exercises per subject, and students may score quite differently in each subject; complex crowdsourcing tasks may be subdivided into subtasks where different workers have different reliability levels based on their expertise in this subtask; information about ships found on the Web [2] can be divided into groups of attributes such as position, technical data, ownership, and each Web source may have different precisions on these groups. In many of these scenarios, though data items have an inherent structure, the corresponding grouping of attributes may be unknown (for example, in data integration scenarios, the semantics of attributes may not be available).

We see this inherent structure of data as both a challenge and an opportunity for truth finding algorithms: on the one hand, traditional truth finding methods that assign a *global quality level* to each source are not able to identify sources that have varying reliabilities on different parts of the data; on the other hand, exploiting the structure of data, when it is relevant, should help us improve the quality of truth finding.

In this paper, we investigate the truth finding problem when data can be structured as an (unknown) partition of attributes with sources having a uniform quality level on each subset. We consider a *one-truth* scenario where each attribute of an object has only one true value and several possible wrong values. We focus, as a running example, on the evaluation of the truthfulness of answers provided by workers on a list of tasks which encompass questions from various fields. As a concrete example, see workers as students undergoing a multidisciplinary examination where each given test includes a set of questions from different subjects. The goal is to automatically discover the correct answers from those given by students, using truth finding. Certainly, we need a way to distinguish among the different levels of students' knowledge on subsets of correlated subjects based on the quality of their answers separately for each subset, as considering an average quality measure can degrade the precision of the truth finding process.

We define a technique, called *AccuPartition*, which extends *any* base truth finding algorithm by searching for an optimal partition of the set of attributes into subsets of correlated attributes, over which different data quality levels of sources are learned. Using a weight function, built from accuracy values themselves estimated by the truth finding algorithm, we define an optimal partition for this particular algorithm; this optimal partition is deemed to properly describe the distribution of the source qualities. In order to compute such an optimal partition, we first devise a general exhaustive exploration algorithm. As this becomes unfeasible when the number of attributes grows, we then propose a sampling-based algorithm that returns a near-optimal partition in reasonable running time. Finally, we present results about the effectiveness of our approach through experiments conducted on synthetic and real-world datasets.

We start by reviewing related work in Section 2. We then give preliminary material, along with the definition of the *AccuPartition* problem, in Section 3. In Section 4, we introduce our model for truth finding under structural correlations among data attributes by detailing our weight function, the proposed exploration algorithms, and an extension of the approach when sources have partial data coverage. Finally, we demonstrate in Section 5 the effectiveness of our approach with respect to existing algorithms through experiments.

2. RELATED WORK

There has been much research on finding the truth among statements made about object instances shared by multiple conflicting sources. This is known under various distinct names such as truth discovery [6, 16–18], fact finding, data fusion [3, 7, 12, 14], etc.; see [7, 13] for early surveys of the state-of-the-art and [1] for a more recent deeper comparative evaluation. The simplest truth finding approach is *majority voting*, which trusts the answers provided by the largest number of equally reliable sources. This naïve technique, however, disregards the fact that sources, in particular on the Web, come with different reliability levels. As a consequence, most truth finding techniques have adopted a weighted voting process with weights for the source quality. As it is unknown in general, quality is estimated based on the level of truthfulness of the object instances. Some domain-specific characteristics, especially those leading to correlations between sources or object instances, motivated the search for different weighted voting models.

[16] accounts for the similarity between data values from different sources in its weighted voting model by leveraging the intuition that the levels of truthfulness of two similar values should influence each other in a certain sense. [5, 6] explores and detects, based on a Bayesian analysis [4], positive source correlations caused by copying relationships. The authors devise a truth finding approach that does not significantly increase the belief about the correctness level of information impacted by copying relationships as false data can be propagated by the copying. Instead, it gives more credit to data from independent providers. [14] proposes to support a broader class of correlations between sources including positive correlations (e.g., similar extraction patterns, implementing similar algorithms) beyond source copying and negative correlations (e.g., the fact that two sources cover complementary information). The authors model those correlations between sources using conditional probability theory in order to use them for the computation of the truthfulness scores of data instances through a Bayesian analysis within a *multi-truth* setting. In their technique, positive correlations are translated similarity as in [5], while negative correlations, in contrast, are handled in such a way that they do not significantly decrease the belief about the correctness level of information.

Besides correlations, other challenging aspects related to truth finding have also investigated, e.g., hardness of facts [9], heterogeneous data types [12], long-tail phenomenon [11], and dynamic or streaming data [6, 18].

The present work is to be seen as complementary to these different dimensions. As we shall see, *any* truth finding method based on source quality can be used in conjunction with our approach. To our knowledge, no previously proposed truth finding algorithm has tackled the issue of harnessing possible structural correlations among data attributes.

3. PRELIMINARIES

This section first introduces general relevant concepts for the truth finding problem and then formally states the *AccuPartition problem*.

We restrict ourselves in this work to the common *one-truth* framework where any attribute of every object has only *one correct value* and *many possible wrong values*. We consider fixed finite sets of *attribute labels* $\mathscr{A}$ and *values* $\mathscr{V}$, as well as a finite set of *objects* $\mathscr{O}$. A source makes statements about the values of *some* attributes of *some* objects:

Test 1: 1. Provide the set of prime numbers smaller than 10.
2. What is the capital city of Romania?

Test 2: 1. Give a natural number x satisfying $x \bmod 4 = 0$.
2. What is the largest country in the European Union?

(a) Test questions

	Test	Math	Geography
student 1	Test 1	$\{2,3,5,7\}$	Budapest
student 2	Test 1	{2, 4, 6, 8}	*Bucharest*
student 3	Test 1	$\{2,3,5,7\}$	Belgrade
student 1	Test 2	*24*	Spain
student 2	Test 2	26	*France*
student 3	Test 2	41	*France*

(b) Student's answers

Figure 1: Example truth finding setting: examination

DEFINITION 1. *A* source *is a partial function* $S : \mathscr{O} \times \mathscr{A} \to \mathscr{V}$ *with non-empty domain. A (candidate)* ground truth *is a total function* $G : \mathscr{O} \times \mathscr{A} \to \mathscr{V}$.

The overall objective of the truth finding problem is to determine the actual ground truth based on the statements of a number of sources. Sources are specifically defined as possibly incomplete; in particular, they may not cover all attributes:

DEFINITION 2. *The* attribute coverage *of a source* S *with respect to* $X \subseteq \mathscr{A}$ *is the proportion of attributes* $a \in \mathscr{A}$ *for which there exists at least one object* $o \in \mathscr{O}$ *such that* $S(o,a)$ *is defined:* $Cov(S,X) := \frac{|\{a \in X \mid \exists o \in \mathscr{O}\ S(o,a) \text{ defined}\}|}{|X|}$.

Two sources S and S' are *contradicting* each other whenever there exists $o \in \mathscr{O}$, $a \in \mathscr{A}$ such that $S(o,a)$ and $S'(o,a)$ are both defined and $S(o,a) \neq S'(o,a)$. A source is *correct* with respect to the ground truth G on $o \in \mathscr{O}$, $a \in \mathscr{A}$ when $S(o,a) = G(o,a)$, and *wrong* when $S(o,a)$ is defined but $S(o,a) \neq G(o,a)$.

EXAMPLE 3. *Figure 1 shows the results of three students (sources) in a multidisciplinary examination, consisting of two tests (objects), each test including two questions (attributes) respectively in mathematics and geography, as given in Figure 1(a). Figure 1(b) shows a table containing the answers provided by the three students to each question. The first and second column of each row in the table correspond to the student name and the identifier of the test (its name here). The remaining columns represent the different questions (attributes) of a test object on two different subjects: math and geography. To have a gold standard against which we can evaluate the precision of any truth finding technique on this dataset, we obtain from an expert the correct answers for questions in each test; those are given in italics in the table:* $\{2,3,5,7\}$ *and Bucharest;* 24 *and France.*

Contradictions occur in this example: for instance, the first student states that the capital city of Romania is Budapest (and is wrong) while the second student claims Bucharest (and is correct).

Formally, a truth finding algorithm, such as those reviewed in Section 2, is defined as follows:

DEFINITION 4. *A* truth finding algorithm F *is an algorithm that takes as input a set of sources* $\mathscr{S}$ *and returns a candidate ground truth* $F(\mathscr{S})$*, as well as an estimated* accuracy $Accuracy_F(S)$ *for each source* $S \in \mathscr{S}$.

Most truth finding processes compute a candidate ground truth based on source accuracy values; when this is the case, we just use these as $Accuracy_F(S)$. If a truth finding algorithm F does not specifically use source accuracy values (this is the case, for instance, of naïve majority voting, in short `vote`), we define source accuracy for F simply as:

$$Accuracy_F(S) := \frac{|\{o \in \mathscr{O}, a \in \mathscr{A} \mid S(o,a) = F(\mathscr{S})(o,a)\}|}{|\{o \in \mathscr{O}, a \in \mathscr{A} \mid S(o,a) \text{ defined}\}|}.$$

In Section 5, we will use the `accu` algorithm proposed in [5] (with no dependency detection, value popularity computation, or value similarity consideration) as a reference truth finding algorithm for experiments. However, we stress that our approach is independent of this choice and any other truth finding algorithm based on source accuracy values can be chosen for F. We refer the reader to [4,5] for an in-depth presentation of `accu`. It will suffice in this work to see `accu` as a black box.

EXAMPLE 5. *Reconsider the example of Figure 1 and apply* `accu` *by starting with a priori accuracy value* $\varepsilon := 0.8$. *The algorithm converges after two iterations, returning* 0.26, 0.26, *and* 0.97 *as accuracy values for students 1, 2, and 3 respectively. Concerning true answers for Test 1 and Test 2,* `accu` *wrongly concludes, e.g., that the capital city of Romania is Belgrade and* 41 *is a divisor of* 4. *The algorithm derives this truth by computing the confidence score of each possible answer for every question; for instance confidence scores estimated by the algorithm for Budapest, Bucharest, and Belgrade are respectively* 3.63, 3.63, *and* 7.5.

Algorithms similar to `accu` use global source accuracy values when computing the confidence scores of attribute values. In so doing, however, the confidence scores of certain specific attribute values can be biased, which, in turn, may drastically impact the precision of the truth finding process.

EXAMPLE 6. *Following Example 1, student 3 has a global accuracy of* 0.97, *which leads to* `accu` *selecting wrong answers, e.g., Budapest and* 41 *from this student. What actually happens is that, though student 3 is fairly accurate with respect to the gold standard, student 1 is very accurate on mathematics while being inaccurate on geography, and vice-versa for student 2. Using* local *accuracies for each subject yields better results for the truth finding as a whole: by splitting the attribute set into two independent subsets* $\{Math\}$ *and* $\{Geography\}$ *we can obtain correct answers from student 1 on mathematics and student 2 on geography.*

DEFINITION 7. *Given a source* $S : \mathscr{O} \times \mathscr{A} \to \mathscr{V}$ *and* $X \subseteq \mathscr{A}$, *we write* $S_{|X}$ *the restriction of* S *to* $\mathscr{O} \times X$.

Let $\mathscr{S}$ *be a finite set of sources. We denote* $\mathscr{S}_{|X} := \{S_{|X} \mid S \in \mathscr{S}\}$. *The* local accuracy *of* S *by* X *as estimated by truth finding algorithm* F *on* $\mathscr{S}_{|X}$ *is* $Accuracy_F(S_{|X})$, *also written* $Accuracy_F(S,X)$ *(by convention,* $Accuracy_F(S,X) = 0$ *if* $S_{|X}$ *is the empty function). We also write* $F(\mathscr{S},X)$ *for* $F(\mathscr{S}_{|X})$.

We are now ready to state our problem of interest, in an informal manner (formal definitions will come in the next section):

PROBLEM 8. *Let* F *be a truth finding algorithm and* $\mathscr{S}$ *a finite set of sources defined over fixed finite sets of objects* $\mathscr{O}$ *and attributes* $\mathscr{A}$. *The* AccuPartition problem *aims at finding an optimal partition* P *of* $\mathscr{A}$ *such that* running F on each subset of P independently maximizes the overall accuracy of F with respect to the gold standard.

4. TRUTH FINDING WITH PARTITIONING

In this section, we present our approach to the AccuPartition problem. We start by introducing a weighting function which evaluates the level of optimality of a given partitioning of input data attributes under a truth finding process. Then we present, as a reference, a general exploration algorithm which solves *AccuPartition* by considering the set of all possible partitions of the entire attribute set in order to return the best partition found based on our introduced weight function. For further efficiency, we finally devise and propose a sampling-based exploration technique that finds a near-optimal partition within a set of a fixed number of partitions.

Fix $\mathscr{O}$, $\mathscr{A}$, and $\mathscr{S}$ the finite sets of objects, attributes, and sources respectively. Let F be a truth finding algorithm.

Optimal Partition Estimation. As shown in [8], the only manner to evaluate the quality of a truth finding algorithm outcome when we have no knowledge about the real data is through its estimated accuracy values for sources. We exploit the same intuition in the computation of the weight of a partition.

Given the set $\mathscr{P}$ of all partitions of $\mathscr{A}$, we introduce a *weighting function* $\Theta : \mathscr{P} \longmapsto [0,1]$ that associates to every partition P in $\mathscr{P}$ a *weight* $\Theta(P)$ whose main purpose is to model the level of optimality of the partition when using F. We define our weighting function Θ by accumulating evidence about the quality of F over the different subsets of the corresponding partition. To do so, we harness local source accuracy values returned by F and devise a *scoring function*, which describes the quality of F on a subset.

DEFINITION 9. *We define the* score $\tau(X)$ *of a subset* $X \subseteq \mathscr{A}$ *as a monotone function of local accuracy values returned by* $F(\mathscr{S},X)$.

We present next various scoring strategies and we refer to Section 5 for an experimental comparison:

- The *maximum scoring function*, in short `maxAccu`, is defined as $\tau_{\texttt{maxAccu}}(X) := \max_{S \in \mathscr{S}} Accuracy_F(S,X)$.
- The *average scoring function*, in short `avgAccu`, is defined as $\tau_{\texttt{avgAccu}}(X) := \frac{1}{|\mathscr{S}|} \sum_{S \in \mathscr{S}} Accuracy_F(S,X)$.
- The *probabilistic analysis-based method*, in short `appAccu`, was introduced in [8] as an estimator of the quality of a truth finding process over a set of sources in the same domain. It is based on source accuracy values and popularity of false values. See [8].
- As a reference point, we also introduce the *oracle function*, in short `oracle`, with respect to a gold standard ground truth G obtained, for instance, from domain experts:

$$\tau(X) := \frac{|\{o \in \mathscr{O}, a \in X \mid F(\mathscr{S},X)(o,a) = G(o,a)\}|}{|\mathscr{O}| \cdot |X|}$$

It is not realistic to assume such a function is available, as the goal of truth finding is to find this ground truth, but it is useful as a comparison with other methods.

We define the weight of a given partition based on the scores of its subsets, as follows.

DEFINITION 10. *The* weight *of a partition* P *of* $\mathscr{A}$ *is the average of the scores of its different subsets. Formally, we set:* $\Theta(P) := \frac{1}{|P|} \cdot \sum_{X \in P} \tau(X)$ *where* $|\, P \,|$ *represents the number of subsets in the partition. A partition* P *of* $\mathscr{A}$ *is* optimal *if it has the highest weight among all partitions of* $\mathscr{A}$.

The *AccuPartition* problem (see Problem 8) thus amounts to finding an optimal partition as given in Definition 10. The following is immediate:

OBSERVATION 11. *When the trivial partition with only one subset is optimal, AccuPartition amounts to the same as F; when the partition formed of all singleton subsets is optimal, AccuPartition amounts to the same as* `vote`.

We now describe a general exponential-time algorithm for *AccuPartition* and then we introduce an approximation technique for gaining in efficiency.

General Exploration Technique. A naïve but exact solution for *AccuPartition* is to explore all possible partitions of the input attribute set. The algorithm `GenAccuPartition` computes the weight of each possible partition, and derives an optimal partition among ones sharing the highest weight. The algorithm finds the truth with respect to this optimal partition, and returns a correct optimal partition when the used scoring function is accurate. However, the algorithm does not scale since the number of partitions grows *exponentially* in the size of the input attribute set (the number of partitions of a set is given by the Bell numbers, which grow exponentially fast [10]).

We thus present an approximation algorithm to reduce the search space of `GenAccuPartition` while returning a near-optimal partition.

Sampling-Based Exploration Technique. We use a random sampling approach in order to effectively restrict the search of an optimal partition to a limited number of candidates from the entire set of partitions.

Fix the number q of samples to explore. In order to efficiently sample uniformly from the entire set $\mathscr{P}$ of all partitions of an input attribute set $\mathscr{A}$ of size n, we proceed in several steps. First, we randomly draw a number k with drawing probability proportional to the Stirling number of the second kind (see Chapter 6 in [10]), $S(n,k)$, that is equal to the number of partitions having k sets. Then, we use a recursive technique to draw a partition at random among those having k sets, based on the recurrence relation obeyed by the Stirling numbers: $S(n,k) = S(n-1,k-1) + k \cdot S(n-1,k)$. For efficiency reasons, we use precomputed Stirling numbers and the time complexity to draw a random partition is $O(n \cdot k)$. The weight of each partition is then computed, and the one with the highest weight is kept. This procedure is repeated as many times as the number q of partitions to be sampled.

Extension with Partial Coverage. In the case where the truth finding process with partitioning uses sources with very diverse levels of data coverage, we revisit our approach by just revising the definition of the scoring function in order to account for the source coverage on given subsets of partitions.

DEFINITION 12. *Given a scoring function τ, we define the* partial coverage revision *of τ as the version of τ where every occurrence of $Accuracy_F(S,X)$ is replaced with $Accuracy_F(S,X) \times Cov(S,X)$.*

This allows biasing towards sources with high coverage. We redefine thus our `maxAccu`, `avgAccu`, and `appAccu` for partial coverage. We refer to the corresponding scoring strategies with `maxAccuCov`, `avgAccuCov`, and `appAccuCov` respectively.

5. EXPERIMENTAL EVALUATION

In this section, we report the experimental evaluation conducted on synthetic, semi-synthetic, as well as real-world data. We studied the precision of our `GenAccuPartition` algorithm – with and without sampling – against `accu` [4,5] when sources exhibit distinct accuracies on different subsets of data attributes. We are interested in the improvements generated by `GenAccuPartition` over the advanced truth finding procedure on which it is based on, i.e., `accu`. Remember that we can substitute `accu` with any other truth finding algorithm based on source accuracy values. For completeness, we also report the results of the naïve `vote` approach. As we shall see, `vote` actually often outperforms `accu`; this has been observed in [1] as well, and highlights the fact that truth finding is a hard problem. All the algorithms were implemented[1] in Java.

We used three different kinds of datasets: purely synthetic data to explore in depth the trade-off brought by AccuPartition, semi-synthetic data that exhibits the features exploited by AccuPartition, and real-world data. We present the corresponding experiments in turn.

[1]As the `accu` source code is not available, we reimplemented it ourselves and obtained results consistent with the ones reported in the original papers [1,5,13].

Table 1: Mean values for dataset configurations

Configs	m_1	m_2	m_3
config 1	1.0	0.0	1.0
config 2	1.0	0.0	0.8
config 3	1.0	0.2	0.8
config 4	0.8	0.4	0.8

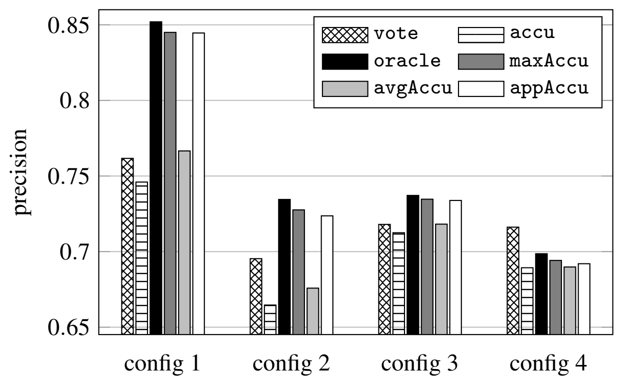
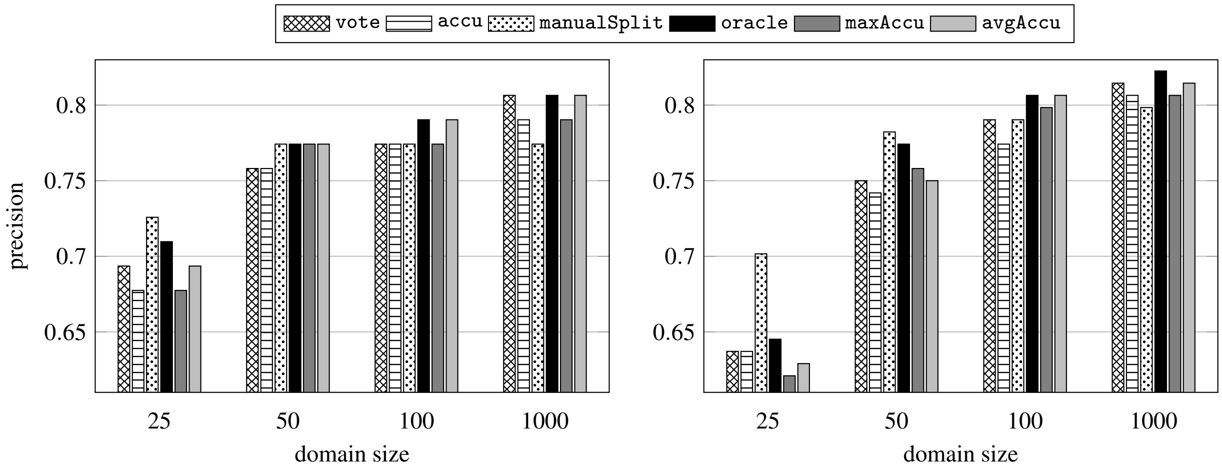


Figure 2: Precision results on synthetic data

5.1 Experiments on Synthetic Data

In order to systematically evaluate our algorithms in various conditions, many not easily found with readily available gold standard in real-world datasets, we started by experimenting on synthetic data. We set up a synthetic data generator that allows to simulate various settings in which every involved source has a level of accuracy unevenly distributed over the entire attribute set. That is, we mainly generated datasets with sources that are very accurate on some specific subsets of attributes while being less accurate on some others.

Synthetic Data Generation. We set up a synthetic dataset generation process which requires five parameters: number of attributes (*na*), number of objects (*no*), number of sources (*ns*), and two mean values m_1 and m_2 for uniform probability distribution functions U_1 and U_2. These probability functions enable to randomly assign high and low source accuracy values to given subsets of attributes. Our generator proceeds with this input as follows. First, we generate sets $\mathscr{A}$, $\mathscr{O}$, and $\mathscr{S}$ of the chosen number of attributes, objects (each object having *na* attributes), and sources. Each source has full coverage. Next, the generator randomly selects a partition P of $\mathscr{A}$. For each subset in P, we randomly choose a source from $\mathscr{S}$ which is deemed to be highly accurate on this subset. Let us denote by $\mathscr{S}'$ the subset of chosen sources for subsets in P. Let $\mathscr{S}'' := \mathscr{S} - \mathscr{S}'$ be the set of non-selected sources. For every source, we generate the attribute values of each object as follows. For every subset X_1 in P together with the corresponding chosen source S in $\mathscr{S}'$, we uniformly set using our distribution functions U_1 and U_2: (i) $Accuracy(S,X_1)$ to a high local accuracy value; and (ii) $Accuracy(S,X_2)$ to a low accuracy value for any other $X_2 \neq X_1$ from P. Moreover, some sources in $\mathscr{S}''$, say $\frac{|\mathscr{S}''|}{2}$ sources, are also chosen to tell the truth on attributes in X_1 for at least a certain percentage of objects depending on how the accuracy value of S on X_1 is distributed over the object set. Any two distinct attributes may contribute with different weights to the accuracy value of a source on the subset that contains them, given its object instances. As a result, we devise a finer source accuracy value at the level of an attribute when knowing the local source accuracy



(a) 62 attributes

(b) 124 attributes

Figure 3: Precision results on semi-synthetic *Exam*

on a subset and its covered number of objects; we select attribute weights via a uniform probability distribution function whose mean, denoted by m_3, is set to 1.0 or 0.8. This computation is done so that the average of the accuracy values of a source on a subset of attributes is exactly equal to its local accuracy on this subset. The accuracy value for a particular attribute gives an indication of the proportion of objects for which the source provides a correct value for that attribute. Finally, the generator – based on these accuracy values for attributes – defines the attributes of each object for which a source tells the truth. In particular, for source S and for every object, mostly correct values are generated for attributes in X_1 and false ones for attributes in X_2, which leads to the construction of the object instances of S. The object instances of the other sources are set in the same way. Observe that the generation process uses a very large domain of false values for attributes in order to prevent having very popular false values within the generated datasets. Furthermore, object instances of sources in $\mathscr{S}''$ are defined such that they have neither significant local accuracy values, nor significant global accuracy values. At last, our generator produces the corresponding ground truth data.

Experimental Setup. We considered four distinct configurations for our tests over synthetic data. For each of these configurations, we set the number of attributes, objects, and sources as follows: $na = 6$, $no = 1000$, and $ns = 10$. For each source, we thus obtained 6,000 attribute values, in other terms, 60,000 data items in total. The distinct configurations are obtained by varying the distribution means m_1, m_2, and m_3, as presented in Table 1. A different synthetic dataset is generated for each given configuration.

Precision Results. We studied the precision of `GenAccuPartition` (without sampling) with the various scoring functions introduced in Section 4. Figure 2 presents a comparison of the precisions of `GenAccuPartition` – with scoring functions `oracle`, `maxAccu`, `avgAccu`, and `appAccu` – against those of `accu` and `vote`. The reported precision values represent the results averaged over 10 random data generations given the same configuration parameters.

For all tested synthetic datasets, we observe that `GenAccuPartition` (without sampling) outperforms `accu` in terms of precision, under the different scoring functions. This highlights and experimentally proves the importance of the partitioning approach and its ability to improve `accu` in scenarios where sources have different levels of accuracy on the given data attribute set. The explanation for the striking improvement (10% precision increase in the best case) that `GenAccuPartition` brings for the first configurations relies on the way these configurations were generated: the differences between the local source accuracy values on the different subsets of the same partition are the strongest in the first configuration, having been created with mean values m_1, m_2, and m_3 respectively equal to 1.0, 0.0 and 1.0, which boosts the performance of `GenAccuPartition`. These differences are then smoothed out due to the choice of the mean values that create homogeneous sources with similar accuracies on different attribute subsets.

5.2 Experiments on Semi-Synthetic Data

Our semi-synthetic datasets are obtained from a real-world *Exam* dataset that has partial attribute coverage, using different ways to generate the missing data attribute values.

Experimental Setup. We first describe the real-world *Exam* dataset and then our different techniques for filling the missing data attribute values within this dataset.

The *Exam* dataset was obtained by aggregating anonymized entrance examination results for overseas students applying to the ParisTech program in 2014. ParisTech, the Paris Institute of Technology, is a collegiate university gathering graduate schools in the Paris area. The exam, seen as one object, is a multiple-choice questionnaire where each question has 5 possible answers, out of which only one is correct. We had access to the answers of 247 students (the set of sources), from 3 different countries. They had to answer 124 questions (the set of attributes) in total from 9 different domains: Math 1A, Chemistry 1, Math 1B, Physics, Electrical Engineering, Computer Science, Chemistry 2, Life Sciences, and Math 2. The set of correct answers to all the questions represents the ground truth.

This dataset has very low attribute coverage, i.e., 36% on average, and this sparsity is explained by the examination conditions: students were required to answer the questions of two domains (Math 1A and Physics), they had to pick one extra domain out of two (Chemistry 1 or Math 1B), and answering questions from all the other five domains was entirely optional. Furthermore, the students were discouraged to answer questions they did not master, since wrong answers were penalized. Therefore, the coverage is bigger on domains like Math 1A, Chemistry 1, Math 1B or Physics than on the optional ones.

In order to avoid the heterogeneity introduced by the low dataset coverage, for every unanswered question we synthetically generated a false answer, randomly chosen in a value domain of size equal to 25, 50, 100, or 1000. Using false value domains of bigger sizes gives similar results as the ones reported for domain size 1000. Working with false value domains of smaller sizes drastically decreases the truth finding accuracy and generates very similar performances for all tested algorithms.

Precision Results. As we did on the synthetic datasets, we also studied the precision of `GenAccuPartition` against the `accu` algorithm. As explained in Section 4, since the number of partitions of a set grows exponentially with the set size, we are no longer able to generate all possible partitions of the attribute set when working on the *Exam* dataset. We thus use `GenAccuPartition` with the *sampling* technique which explores 1000 randomly generated partitions and we report the precision results obtained for the partition with the best weight. Moreover, we report results separately on a *manually split partition* obtained by grouping together questions that belong to the same subject.

We focus on two different scenarios. In Figure 3(a), we consider only the 62 attributes that come from the four domains (Math 1A, Chemistry 1, Math 1B, Physics) on which, as we explained above, original coverage is bigger. In Figure 3(b), we present results for all 124 attributes from all domains. We also experimented on the 32 attributes of the two compulsory domains, Math 1A and Physics. In this case, the original coverage is high enough that synthetically increasing coverage does not impact the results obtained on raw data (presented in Section 5.3, Figure 4(a)).

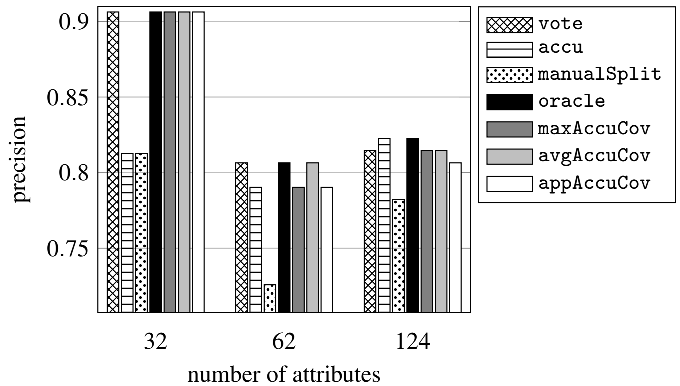
We observe that the precision of `GenAccuPartition` evaluated on the *manually split partition* is significantly higher than the one obtained with `accu` when filling missing answers with false values from small value domains. In this case, some false values become dominant for certain attributes, which drastically decreases the performances of `accu` and `vote`, increasing the difference between algorithm performances. Observe that the *manually split partition* is not generated while *sampling* the 1000 partitions we worked on. The results obtained by `GenAccuPartition` with sampling, with different subset score functions improve the results found by `accu`, even though the improvements are milder for bigger false value domains. We also observe that results obtained for the false value domain size of 1000 are similar to the ones obtained on raw data, as presented in Section 5.3, Figure 4(a).

5.3 Experiments on Real-World Data

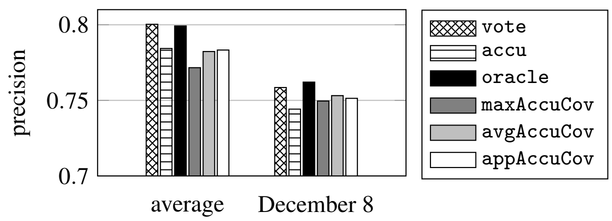
Finally, we run tests on two real-world datasets: the *Exam* and the *Flights* dataset.

Experimental Setup. We use the raw data from the *Exam* dataset introduced in the previous section. Moreover, we experiment on the *Flights* dataset[2], previously used in [1, 8, 13, 15], over which a manual cleaning was required. It contains information over 1200 flights with 6 attributes each, collected from 38 sources over 31 consecutive days. Both real-world datasets used in this section have partial data coverage. As introduced in Section 4, we use subset score functions that take partial coverage into account. Therefore, we use `maxAccuCov`, `avgAccuCov`, and `appAccuCov`, instead of `maxAccu`, `avgAccu`, and `appAccu`.

Precision Results. For the *Exam* dataset, we present results for different subsets of attributes. We experiment on 32, 62, and all 124 attributes, that correspond respectively to the two compulsory domains (Math 1A and Physics), to four domains (the two compulsory ones and two more from which students have to choose only one, Chemistry 1 or Math 1B), and to all nine domains. The results, presented in Figure 4(a), follow the same trend as the ones presented before, showing improvements of `GenAccuPartition` (with sampling) over `accu`. However, partial coverage tend to decrease the overall quality of the experimental results. Therefore



(a) *Exam*



(b) *Flights*

Figure 4: Precision results on real-world datasets

these improvements are more pronounced when experimenting over a high coverage dataset (81% coverage for the 32 attributes dataset) than on lower coverage ones (36% coverage for all 124 attributes).

In Figure 4(b), we present the precision results of `GenAccuPartition` (witout sampling) on the *Flights* dataset averaged over the entire one month period, after having evaluated our algorithms on each day separately. We also present results for the 8th of December, the same random date on which results are reported in [13]. Since the attributes of the *Flights* dataset (scheduled and actual departure date, scheduled and actual arrival date, departure and arrival gate) do not have any specific inherent structure, this dataset is not particularly well suited to emphasize the benefits of using `GenAccuPartition` over `accu`. Nevertheless, `GenAccuPartition` tested with different subset score functions outperformes `accu` for the particular date of 8th of December.

6. CONCLUSION AND FUTURE WORK

Our approach searches first for an optimal partition of the attribute set into subsets of correlated attributes, over which sources have different local accuracies, and then applies truth finding algorithms locally over each partition subset. Experimental results over synthetic, semi-synthetic, and real-world datasets show that our proposed method can significantly improve the accuracy of the truth discovery process.

There are many interesting challenges in this problem for further development. First, we are experimenting with new subset scores and partition weighting functions. Second, we are considering a bottom-up algorithm that constructs the optimal partition, starting from singletons. Third, we aim to combine our attribute partitioning approach with source selection methods in order to further leverage the inherent data structure and associated domain experts.

[2]Available at `http://lunadong.com/fusionDataSets.htm`.

7. REFERENCES

[1] D. Attia Waguih and L. Berti-Équille. Truth discovery algorithms: An experimental evaluation. *CoRR*, abs/1409.6428, 2014.

[2] M. L. Ba, S. Montenez, R. Tang, and T. Abdessalem. Integration of web sources under uncertainty and dependencies. In *UnCrowd*, 2014.

[3] J. Bleiholder, K. Draba, and F. Naumann. FuSem: exploring different semantics of data fusion. In *VLDB*, 2007.

[4] X. L. Dong, L. Berti-Equille, Y. Hu, and D. Srivastava. Global detection of complex copying relationships between sources. *VLDB Endow.*, 3(1), 2010.

[5] X. L. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: the role of source dependence. *PVLDB*, 2(1), 2009.

[6] X. L. Dong, L. Berti-Equille, and D. Srivastava. Truth discovery and copying detection in a dynamic world. *PVLDB*, 2(1), 2009.

[7] X. L. Dong and F. Naumann. Data fusion: Resolving data conflicts for integration. *PVLDB*, 2(1), 2009.

[8] X. L. Dong, B. Saha, and D. Srivastava. Less is more: Selecting sources wisely for integration. *PVLDB*, 6(2), 2012.

[9] A. Galland, S. Abiteboul, A. Marian, and P. Senellart. Corroborating information from disagreeing views. In *WSDM*, 2010.

[10] R. L. Graham, D. E. Knuth, and O. Patashnik. *Concrete Mathematics: A Foundation for Computer Science*. Addison-Wesley, 1994.

[11] Q. Li, Y. Li, J. Gao, L. Su, B. Zhao, M. Demirbas, W. Fan, and J. Han. A confidence-aware approach for truth discovery on long-tail data. *PLDB*, 8(4), 2015.

[12] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han. Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation. In *SIGMOD*, 2014.

[13] X. Li, X. L. Dong, K. Lyons, W. Meng, and D. Srivastava. Truth finding on the deep web: Is the problem solved? *PVLDB*, 6(2), 2012.

[14] R. Pochampally, A. Das Sarma, X. L. Dong, A. Meliou, and D. Srivastava. Fusing data with correlations. In *SIGMOD*, 2014.

[15] D. A. Waguih, N. Goel, H. M. Hammady, and L. Berti-Equille. AllegatorTrack: Combining and reporting results of truth discovery from multi-source data. In *ICDE*, 2015.

[16] X. Yin, J. Han, and P. S. Yu. Truth discovery with multiple conflicting information providers on the web. *IEEE TKDE*, 2008.

[17] B. Zhao, B. I. P. Rubinstein, J. Gemmell, and J. Han. A Bayesian approach to discovering truth from conflicting sources for data integration. *PVLDB*, 5(6), 2012.

[18] Z. Zhao, J. Cheng, and W. Ng. Truth discovery in data streams: A single-pass probabilistic approach. In *CIKM*, 2014.