Discovering Multiple Truths with a Hybrid Model

Furong Li† Xin Luna Dong‡
†National University of Singapore
furongli@comp.nus.edu.sg
‡Google Inc., Mountain View, CA, USA
{lunadong, arl, ngli}@google.com

ABSTRACT
Many data management applications require integrating information from multiple sources. The sources may not be accurate and provide erroneous values. We thus have to identify the true values from conflicting observations made by the sources. The problem is further complicated when there may exist multiple truths (e.g., a book written by several authors). In this paper we propose a model called HYBRID that jointly makes two decisions: how many truths there are, and what they are. It considers the conflicts between values as important evidence for ruling out wrong values, while keeps the flexibility of allowing multiple truths. In this way, HYBRID is able to achieve both high precision and high recall.

1. INTRODUCTION
When consolidating information from different sources, we may observe different values provided for the same entity. Consequently, we need to identify the correct values from conflicting observations made by multiple sources, which is known as the data fusion (or truth discovery) problem [2, 5]. We illustrate the problem using the example below.

Example 1.1. Table 1 shows the information collected from three sources regarding equipments of two winter sports: ice hockey and snowboarding. We can see that four different values are provided for the entity ice hockey (helmet, stick, boots and skis), while only the first two are correct. The goal of the truth discovery problem is to identify the correct values from Table 1.

Table 1: Information collected from different sources regarding equipments of various winter sports. ✓/× indicates the correctness.

| entity          | attribute     | value     | sources |
|-----------------|---------------|-----------|---------|
| ✓ o1            | ice hockey    | helmet    | s1, s3  |
| ✓ o2            | ice hockey    | stick     | s1, s2  |
| × o3            | ice hockey    | boots     | s2      |
| × o4            | ice hockey    | skis      | s3      |
| ✓ o5            | snowboarding  | board     | s2, s3  |
| × o6            | snowboarding  | neck guard| s1      |

The limitation of the above methods is that when multiple truths exist, they at best find one of them while missing the rest. We thus refer to them as single-truth approaches. While truth discovery algorithms usually compute a probability \( p(v) \) of each value \( v \) being true, in single-truth approaches, the probabilities of all values sum up to 1 since they assume there is only one true value.

Example 1.2. We use ACCU [1] as a representative of single-truth approaches, and then compute the probabilities of the values provided from ice hockey equipments. Assuming all sources have the same accuracy 0.6, we obtain the probabilities in Table 2 (see the first line). We observe that the probabilities of the four values add up to 1, so even the true values (helmet and stick) have rather low probabilities.

To address the above problem, multi-truth approaches [9, 7] have been proposed recently. They compute the probability for each value separately, and thus do not require the probabilities of all values sum up to 1. Instead, they compute both the probability \( p(v) \) of \( v \) being true, and the probability \( p(\neg v) \) of \( v \) being false, where \( p(v) + p(\neg v) = 1 \). Then a value \( v \) is considered true if \( p(v) > p(\neg v) \), that is, \( p(v) > 0.5 \).

An unknown semantics is used to capture the nature of multi-truth: if a source \( s \) does not provide the value \( v \), \( s \) means that it does not know whether or not \( v \) is correct (instead of saying \( v \) is incorrect). Thus, apart from accuracy (also called precision in some methods), multi-truth approaches also measure the quality of a source \( s \) by its recall, the probability that a truth is provided by \( s \).

Table 2: Value probabilities computed by different approaches for (ice hockey, equipments).

|                  | helmet | stick | boots | skis  |
|------------------|--------|-------|-------|-------|
| Single-truth [1]  | 0.47   | 0.47  | 0.03  | 0.03  |
| Multi-truth [7]   | 0.63   | 0.63  | 0.54  | 0.54  |
| HYBRID            | 0.92   | 0.92  | 0.08  | 0.08  |
Intuitively, values provided by high-precision sources are likely to be true (i.e., a higher $p(v)$), and values not provided by high-recall sources are likely to be false (i.e., a higher $p(\neg v)$). In this way, they derive $p(v)$ and $p(\neg v)$ from the precision and recall of the sources, and then normalize with the equation $p(v) + p(\neg v) = 1$.

**Example 1.3.** Table 2 also shows the value probabilities computed by the multi-truth approach PRECREC [7]. Assuming a precision of 0.6 and recall of 0.5 for each source, PRECREC will decide that all provided values are true, resulting in false positives (i.e., boots and board).

In practice, even multi-truth items often have only a few true values instead of infinite number of truths. Existing methods [6, 7] cannot capture this because they decide the truthfulness of each value independently, without considering other values provided for the entity and thus lack a global view of the entity. As a result, they suffer from low precision when the sources have low coverage or noisy observations (as shown later in Section 5).

In this paper we introduce a new solution to the truth discovery problem, called HYBRID, which works for multi-truth applications. Based on the values provided for an entity, HYBRID makes two decisions: (i) how many truths there are, and (ii) what they are. Essentially it interleaves the two decisions and finds the truths one by one. Conditioning on a sequence of true values that have been selected previously, it computes (1) the probability of a value $v$ being the next truth, and (2) the probability that there is no more truth. In this way, HYBRID combines the flexibility of the multi-truth approaches of allowing multiple truths for an entity, and the inherent strength of the single-truth approaches of considering conflicts between values as important evidence for ruling out wrong values. Therefore, it obtains both high precision and high recall.

Note that the multi-truth setting should be considered more general than the single-truth setting, since it allows for the existence of multiple truths (but not necessarily). Our proposed method also works for entities with a single-truth because it can automatically decide the number of truths. Although one can easily extend a single-truth approach to handle multi-truth applications by setting a threshold (i.e., consider all values with predicted probabilities over $\lambda$ as true), it is hard to find a threshold that works for all entities. We discuss a slightly more sophisticated extension in Section 5.

### 2. Definitions and Notations

**Data item, value, source, observation.** We call an (entity, attribute) pair a data item, denoted by $d$. Then we have a set $S$ of sources provides values on $d$. Let $v$ be a value provided by a source $s \in S$ for the data item $d$, the pair $(d, v)$ is then called an observation of $s$. For instance, there are two data items in Table 1, $d_1 = \text{(ice hockey, equipments)}$ and $d_2 = \text{(snowboarding, equipments)}$; there are 4 values provided for $d_1$ and 2 values provided for $d_2$. In total we have 6 observations made by 3 sources $\{s_1, s_2, s_3\}$.

Given a data item $d$, we use $\Psi$ to denote the set of observations made on $d$ (we dismiss $\delta$ in the notation for simplicity); then $\Psi(s)$ denotes the values from $s$. For example in Table 1 for the item (ice hockey, equipments), $\Psi(s_1) = \{\text{helmet, stick}\}$. Table 3 summarizes the notations we use in this chapter.

**Problem definition.** Given a data item $d$ and a set $S$ of sources, let $\mathcal{V}$ denote the set of values provided by $S$ on $d$. Our goal is to compute a probability $p(v|\Psi)$ for each value $v \in \mathcal{V}$ being true based on the source observations.

In this chapter we focus on the case where the sources are independent of each other; we can extend our model with techniques from [1, 7] to address correlations between sources.

### 3. A Hybrid Model

This section presents a truth discovery model, called HYBRID, which allows for the existence of multiple truths. Essentially, HYBRID makes two decisions for a data item $d$: (i) how many truths are there, and (ii) what they are.

One can imagine a natural solution that processes in two steps: (1) decide the number of truths $k$ with a single-truth method, treating “the number of truths for $d$” as a data item and $|\Psi(s)|$ as the value provided by $s$; (2) apply the single-truth method on $\mathcal{V}$ and select the values with top-$k$ probabilities as the truths. Although this approach often outperforms both the existing single-truth and multi-truth approaches (as we shall show later in Section 5), it has two problems. First, it does not update the value probabilities according to its belief of the number of truths (all probabilities still sum up to 1). Second, separating the decisions into two steps may hurt precision when many sources provide more values than the truths: once the first step decides the number of truths $k$, the second step will fill in $k$ values, possibly with values lacking strong support from the sources.

Different from the above baseline approach, HYBRID combines the two steps and finds the truths one by one. Conditioning on a sequence $\mathcal{O}$ of true values that have been selected previously, it decides (1) the probability of a value $v$ being the next truth, denoted by $p(v|\mathcal{O}, \Psi)$, and (2) the probability that there is no more truth, denoted by $p(\bot|\mathcal{O}, \Psi)$. These are disjoint decisions so their probabilities sum up to 1. Thus, when selecting the next truth, HYBRID basically applies a single-truth method.

However, when deciding whether there is any more truth (i.e., $p(\bot|\mathcal{O}, \Psi)$), HYBRID incorporates the unknown semantics used in multi-truth approaches: if a source provides 2 values for an item $d$, it claims that it knows 2 values of $d$, instead of claiming that $d$ has only 2 values.

In this way, HYBRID combines the flexibility of the multi-truth methods of allowing multiple truths for a data item, and the inherent strength of the single-truth methods of considering conflicts between values as important evidence for ruling out wrong values. Therefore, it obtains both high precision and high recall.

Moreover, HYBRID leverages the typical number of truths for each type of data items; for example, a person typically has 2 parents and 1-5 children. HYBRID allows incorporating such knowledge as the a priori probability of $p(\bot|\mathcal{O}, \Psi)$, which further improves performance. Bear in mind that a priori probabilities have much less effect than observations on computing a posteriori probabilities, so HYBRID applies a soft constraint rather than a hard one.

We next describe the HYBRID model in more details, and answer the following question: since there should not exist any ordering between the truths, how would HYBRID avoid the consequence of finding the truths one by one?

### Table 3: Table of notations.

| Notation | Description |
|----------|-------------|
| $d$      | a data item |
| $v$      | a value     |
| $s$      | a source that provides values |
| $\mathcal{S}_v$ | the set of sources that provide the value $v$ |
| $\Psi$   | the mapping between sources and their provided values |
| $\Psi(s)$ | the set of values provided by $s$ on a data item |
| $n$      | the number of wrong values in the domain |
| $\mathcal{O}$ | a sequence of values that have been selected as truths |
| $\bot$   | “there is no more truth” |
3.1 Overall Probability of a Value

Consider computing $p(v | \Psi)$ for a value $v \in \mathcal{V}$. As we select truths one by one, there can be various sequences of truths (of any length below $|V|$) that are selected before $v$. We call each sequence $O$ a possible world and denote by $\Omega$ all possible worlds. Then the probability of $v$ is the weighted sum of its probability in each possible world:

$$p(v | \Psi) = \sum_{O \in \Omega} p(v | O, \Psi) \cdot p(O | \Psi).$$  \hspace{1cm} (1)

where $p(O | \Psi)$ is the probability of entering the possible world $O$. Let $O = v_1 v_2 \ldots v_j$, $v \notin O$, denote a possible world with the sequence $v_1, v_2, \ldots, v_j$ of values selected as truths. Let $O_j$ denote a prefix of $O$ with length $j$ and $O_0 = \emptyset$. Applying the chain rule leads us to:

$$p(O | \Psi) = \prod_{j=1}^{|O|} p(v_j | O_{j-1}, \Psi).$$ \hspace{1cm} (2)

Now the only piece missing from Eqs (1, 2) is the conditional probability $p(v | O, \Psi)$, which we write in the next subsection.

Back to the question we asked previously, even though HYBRID finds truths one by one, it is order-independent as it considers all possible ways to select a value and computes an overall probability. Apparently enumerating all possible worlds is prohibitively expensive; we describe a polynomial-time approximation in Section 3.2.

3.2 Conditional Probability of a Value

Now consider computing $p(v | O, \Psi)$. Under a possible world $O$, we either choose one of the remaining values as the next truth or decide that there is no more truth, thus $\sum_{v' \in \mathcal{V} \setminus O} p(v' | O, \Psi) + p(\bot | O, \Psi) = 1$; this is similar to what we have in single-truth approaches. Then according to the Bayes Rule, we have

$$p(v | O, \Psi) = \frac{p(O | v, \Psi)p(v | \Psi)}{\sum_{v' \in \mathcal{V} \setminus O} p(O | v', \Psi)p(v' | \Psi) + p(\bot | O, \Psi)p(\bot | \Psi)}.$$ \hspace{1cm} (3)

Here the inverse probability $p(\Psi | O, v)$ is the probability of observing $\Psi$ if $v$ is the next truth. The a priori probability $p(v | \Psi)$ is the probability of $v$ being the next truth regardless of the observations $\Psi$. The same applies to $p(\Psi | O, \bot)$ and $p(\bot | \Psi)$.

Before we can compute these two sets of probabilities, we first define the metrics that are used to measure the quality of a source.

3.2.1 Source-quality metrics

Imagine that there are $m$ latent slots for truths of a data item, and a source $s$ is asked to fill the slots. The number of slots is unknown to $s$, so it iteratively performs two tasks: predict whether there exists another slot (i.e., another truth), and if so, fill the slot with a value. We thus capture the quality of a source with two sets of metrics: that for deciding whether there exists a truth, and that for deciding the true values.

The first set of metrics enables the unknown semantics for multi-truth, and it includes two measures:

- **Precision** $P(s)$, the probability that when $s$ provides a value, there indeed exists a truth;
- **Recall** $R(s)$, the probability that when there exists a truth, $s$ provides a value.

Note that our $P(s)$ and $R(s)$ are different from the same notions in previous work [7]; we only measure how well $s$ predicts whether or not there exists a truth, but not how well $s$ predicts what the truth is; in other words, we do not require the value provided by $s$ to be the same as the truth. To facilitate later computations, we next derive the false positive rate of $s$, denoted by $Q(s)$, from $P(s)$ and $R(s)$ by applying the Bayes Rule (see [7] for details):

$$Q(s) = \frac{\alpha}{1 - \alpha} \cdot \frac{1 - P(s)}{P(s)} \cdot R(s),$$ \hspace{1cm} (4)

where $\alpha$ is the a priori probability that a provided value corresponds to a truth slot. Intuitively, $Q(s)$ is the probability that $s$ still provides a value when there is no truth slot.

The second set of metrics follows single-truth models to address the conflicts between values. It contains one measure: accuracy $A(s)$, the probability that a value provided by $s$ for a "real" truth slot is true (i.e., $s$ provides a true value after it has correctly predicted the existence of a truth slot). Note that values provided for non-existing truth slots, which are absolutely false, are not counted here, as they have been captured by $P(s)$.

We describe how we compute these metrics in the next subsection, and demonstrate the basic idea of them in the example below.

**Example 3.1.** Consider the source $s_2$ and the data item $d_1 = (\text{ice hockey, equipments})$ in Table 1. Suppose ice hockey requires 3 equipments. We observe that $s_2$ provides 2 values on $d_1$, meaning that it predicts that there are 2 slots for truths; among the provided values one is true. Therefore for this particular data item, $s_2$ has precision $2/2 = 1$, recall $2/3 = 0.67$, and accuracy $1/2 = 0.5$.

Now consider data item $d_2 = (\text{snowboarding, equipments})$, which has 1 truth. As $s_2$ provides 1 correct value, its precision, recall, and accuracy for this item are all 1.

If $s_2$ provides only these 2 data items, on average, we have $P(s_2) = \frac{1 + 1}{2} = 1$, $R(s_2) = \frac{0.67 + 1}{2} = 0.83$, and $A(s_2) = \frac{0.5 + 1}{2} = 0.75$.

3.2.2 Inverse probabilities

We are now ready to derive the inverse probabilities $p(\Psi | O, v)$ and $p(\Psi | O, \bot)$ in Eq. (3). Assuming that the set of sources are independent, we have

$$p(\Psi | O, v) = \prod_{s \in S} p(\Psi | O, v),$$ \hspace{1cm} (5)

and similar for $p(\Psi | O, \bot)$. In the following computations, when conditioning on $(O, v)$, we think that $O \cup \{v\}$ are the only set of truths; similarly, when conditioning on $(O, \bot)$, we think $O$ is the only set of truths. This is known as the closed-world assumption, and according to [6], it should give the same results as the open-world assumption where the truths form a superset of $O \cup \{v\}$.

Let $T$ be the truths of the item $d$, that is, $T = \mathcal{O}$ (when computing $p(\Psi | O, \bot)$) or $T = O \cup \{v\}$ (when computing $p(\Psi | O, v)$). Accordingly, we can partition $\Psi(s)$, values provided by $s$ on $d$, into four categories: consistent values, inconsistent values, extra values and missing values. We denote respectively the size of each category as $N_{c}, N_{w}, N_{e}, N_{m}$, and the probability that a value falls into each category as $P_{c}, P_{w}, P_{e}, P_{m}$. Then $p(\Psi(s) | O, v)$ is given by:

$$p(\Psi(s) | O, v) = P_{c}^{N_{c}} \cdot P_{w}^{N_{w}} \cdot P_{e}^{N_{e}} \cdot P_{m}^{N_{m}}.$$ \hspace{1cm} (6)

When deriving $p(\Psi(s) | O, \bot)$, the only difference is that we will award a source $s$ if it does not provide any extra value; otherwise, we re-use Eq. (6). The probability of not providing extra values is $P_{\neg e} = 1 - Q(s)$, and recall that $O$ is the (estimated) set of truths for the data item. Thus we have:

$$p(\Psi(s) | O, \bot) = \begin{cases} P_{c}^{N_{c}} \cdot P_{w}^{N_{w}} \cdot P_{e}^{N_{e}} \cdot P_{m}^{N_{m}} & |\Psi(s)| > |O|; \\ P_{c}^{N_{c}} \cdot P_{w}^{N_{w}} \cdot P_{e}^{N_{e}} \cdot P_{m}^{N_{m}} \cdot P_{\neg e} & |\Psi(s)| \leq |O|. \end{cases}$$ \hspace{1cm} (7)

We next define each category and describe how we compute their sizes and probabilities. Following [1], we assume that there are $n$ false values in the domain of $d$ and they are uniformly distributed (note that the false values may not appear in $V$).
• **Consistent value:** A consistent value is a value in \(T \cap \Psi(s)\); thus, \(N_c = |T \cap \Psi(s)|\). To provide a consistent value, \(s\) needs to correctly predict that there exists a slot for truth, and fills the slot with a true value, so \(P_c = R(s) \cdot A(s)\).

• **Inconsistent value:** An inconsistent value is a value that is provided for a truth slot, but differs from any true value. At most we have \(|T|\) values provided for truth slots; except the consistent values, others are inconsistent. Thus \(N_w = \min(|T|, |\Psi(s)|) - N_c\). When \(s\) provides an inconsistent value, it correctly predicts the existence of a truth slot, but fills in a **special false value**, so \(P_w = R(s) \cdot \frac{1 - A(s)}{n}\).

• **Extra value:** If \(s\) provides more than \(|T|\) values, the rest of the values are extra values; thus, \(N_e = \max(|\Psi(s)| - |T|, 0)\). When \(s\) provides an extra value, it incorrectly predicts a non-existing slot, and fills in a **particular (false) value**, so \(P_e = \frac{Q(s)}{n}\).

• **Missing value:** Alternatively when \(\Psi(s)\) contains fewer values than \(T\), \(s\) misses some truth slots (i.e., it thinks they do not exist). We have \(N_m = \max(|T| - |\Psi(s)|, 0)\) and \(P_m = 1 - R(s)\).

**Example 3.2.** Consider the data item \(d_1\) and we now compute \(p(\Psi(s_1)|o_1, o_2)\), the probability of observing the values in \(\Psi(s_1)\) if \(o_2\) is the next truth after \(o_1\) has been selected. We have \(O = o_1, \Psi(s_2) = \{o_2, o_3\}\) and \(T = \{o_1, o_2\}\). So \(\Psi(s_2)\) contains one consistent value, and one inconsistent value; there is no extra value or missing value. In other words, we have \(N_e = N_w = 1\) and \(N_c = N_m = 0\).

Supposing \(n = 10\) and \(A(s_2) = 0.6, R(s_2) = 0.9, Q(s_2) = 0.1\), we have \(P_e = 0.9 \cdot 0.6 = 0.54, P_w = 0.1 \cdot \frac{1}{10} = 0.01, P_m = 0.1\).

Then according to Eq. (8) we compute:
\[p(\Psi(s_2)|o_1, o_2) = 0.54 \cdot 1^{-0.03} \cdot 0.01^{17} = 0.019\]

We repeat the above process for the other sources and obtain:
\[p(\Psi(s_1)|o_1, o_2) = 0.292, p(\Psi(s_3)|o_1, o_2) = 0.019\]

With the source-independence assumption, we have:
\[p(\Psi(o_1, o_2) = 0.019 \cdot 0.292 \cdot 0.019 = 1.05 \times 10^{-3}\]

**3.2.3 A priori probabilities**

We then compute the probabilities \(p(\perp|O)\) and \(p(\forall|O)\) in Eq. (8). Intuitively, the chance of \(\perp\) increases when more truths are found. Let \(\beta_i\) be the a **prior probability** of \(\perp\) when we are looking for the \(i\)th truth (i.e., \(|O| = i - 1\)). So there are \(|V| - i + 1\) unselected values in \(V\); assuming they have the same a **priori probability**, the a **priori probability** \(p(\forall|O)\) of each value \(v\) would be:
\[p(\forall|O) = \frac{1 - \beta_i}{|V| - i + 1}\] (8)

We can derive \(\beta_i\) from the distribution of the number of truths for a data item. For example, among people who have children, if 30% of them have 1 child, 40% have 2 children, and so on, then \(\beta_2 = 0.3\) (with probability 30% there is not a second truth), and \(\beta_3 = 0.7\) (with probability 30%-40% there is not a third truth).

**3.2.4 Summary**

By putting the derived inverse probabilities and a **priori** probabilities into Eq. (8), we are able to obtain \(p(\forall|O, \Psi)\), and this completes the computation of Eq. (1). As the following proposition shows, HYBRID computes higher probabilities for values provided by more accurate sources; it finds more truths when high-precision sources provide more values; and it finds fewer truths when high-recall sources provide fewer values. These all conform to our intuition.

**Proposition 3.3.** Consider a value \(v\) and a source \(s \in S\) where \(v \in \Psi(s)\); we fix all sources in \(S\) except \(s\).

- If \(A(s) > \frac{1}{n-1} \cdot p(\forall|\Psi)\) increases when \(A(s)\) increases.
- If \(Q(s) < \frac{R(s) - R(s)|A(s)|}{1 - R(s)|A(s)|} \cdot p(\perp|\Psi)\) decreases when \(s\) provides more values.
- If \(R(s) > \frac{Q(s)}{1 - A(s) + A(s)|Q(s)|} \cdot p(\perp|\Psi)\) increases when \(s\) provides fewer values.

**Example 3.4.** Continuing with Example 3.2, we proceed to compute \(p(o_2|o_1, \Psi)\) using Eq. (1). This involves the reverse probability \(p(\Psi|o_1, v)\) and the a **priori probability** \(p(v|o_1)\) for every remaining value in \(V \setminus \Omega = \{o_2, o_3, o_4\}\) as well as \(\perp\).

We have obtained the inverse probability \(p(\Psi|o_1, o_2)\) in Example 3.2 we now repeat the process on \(o_3, o_4\) and \(\perp\) to compute:
\[p(\Psi|o_1, o_3) = 6.8 \times 10^{-6}\]
\[p(\Psi|o_1, \perp) = 1.05 \times 10^{-8}\]

Then assuming \(\beta_1 = 0.3, \beta_2 = 0.3\), from Eq. (8) we have \(p(o_2|o_1) = p(o_3|o_1) = p(o_4|o_1) = 0.175\).

We can thus obtain \(p(o_2|o_1, \Psi)\) via Eq. (1):
\[p(o_2|o_1, \Psi) = \frac{p(\Psi|o_1, o_2)p(o_2|o_1)}{p(\Psi|o_1, o_2)p(o_2|o_1) + p(\Psi|o_1, \perp)p(\perp|o_1)} = 0.88\]

Table 2 shows the probabilities obtained by enumerating all possible worlds \(\Omega\) for each value \(v\). We can see that HYBRID gives very high probabilities (0.92) for the two true values (helmet and stick) and meanwhile very low probabilities for the false ones.

**3.3 Evaluating Source Quality**

The previous subsection explains how to compute value probabilities based on the quality of sources. We do not always have such prior knowledge on source qualities, and in this case we start by assuming each source has the same quality, and then iteratively compute value probabilities and source qualities until convergence. This subsection describes how to update source quality based on the estimated truths of a set of data items.

For each source \(s\), we compute \(P(s), R(s)\) and \(A(s)\) as defined, except that we adopt the probabilistic decisions made on the truthfulness of values. We emphasize again that the computation of precision and recall do not consider the truthfulness of the values, but only the **cardinality** of the provided values (i.e., how many truth slots \(s\) thinks there are).

- The precision of \(s\) is the average of its precision on each data item \(s\) provides values for. Let \(\Psi_d(s)\) be the set of values provided by \(s\) on \(d\) and \(V_d\) be the domain of \(d\). Then \(\sum_{v \in V_d} p(v)\) is the (probabilistic) number of truths for \(d\), and we have:
\[P(s) = \frac{\text{Avg}_d \min \left( \frac{\sum_{v \in V_d} p(v)}{|V_d|}, \frac{1}{1} \right)}{1}\]

- Similarly, the recall of \(s\) is the average of its recall on each data item:
\[R(s) = \frac{\text{Avg}_d \min \left( \frac{|\Psi_d(s)|}{\sum_{v \in V_d} p(v)}, \frac{1}{1} \right)}{1}\]

- The accuracy of \(s\) can be estimated as the average probability of its values, divided by its precision, so it accounts for values provided for “real” truth slots only:
\[A(s) = \frac{\text{Avg}_{d,v \in \Psi_d(s)} p(v)}{P(s)}\]

**4. APPROXIMATION FOR HYBRID**

Computing value probabilities by enumerating all possible worlds takes exponential time. We conjecture that the value probability computation in HYBRID is \#P-complete; the proof of the conjecture remains an open problem. This section describes an approximation for probabilities under HYBRID. We start with simplifying
the computation of $p(v|O, \Psi)$ in Eq. (13), and then present our approximation algorithm.

### 4.1 Simplification of $p(v|O, \Psi)$

We can simplify the computations in Section 4.2 to a much simpler form. We start from Eqs. (10) and (17).

Given a particular source $s$, suppose $N_c = c$, $N_w = w$, $N_e = e$ and $N_m = m$ when computing $p(\Psi(s)|O, \bot)$ with Eq. (11). Then we have four cases when deciding the $N_c$, $N_w$, $N_e$ and $N_m$ for $p(\Psi(s)|O, v)$ in Eq. (11), depending on whether $|\Psi(s)| > |O|$ and whether $v \in \Psi(s)$; we illustrate them in Table 4.

| The case that $s$ belongs to | $N_c$ | $N_w$ | $N_e$ | $N_m$ |
|------------------------------|------|------|------|------|
| case 1: $|\Psi(s)| > |O|$ and $v \in \Psi(s)$ | $e + 1$ | $w$ | $e$ | $m + 1$ |
| case 2: $|\Psi(s)| > |O|$ and $v \notin \Psi(s)$ | $c$ | $w + 1$ | $e$ | $m + 1$ |
| case 3: $|\Psi(s)| \leq |O|$ and $v \in \Psi(s)$ | $e + 1$ | $w - 1$ | $e$ | $m + 1$ |
| case 4: $|\Psi(s)| \leq |O|$ and $v \notin \Psi(s)$ | $c$ | $w$ | $e$ | $m + 1$ |

We can then transform Eq. (13) into the following format:

$$p(v|O, \Psi) = \sum_{v' \in V \setminus O} L(v') + L_{|O|+1} \bot.$$

### 4.2 Approximation

Our approximation leverages three observations. First, equivalent to computing $p(v|\Psi)$ conditioning on all possible worlds, we compute $p(v|\Psi) = \sum_{v \in V} p(v|\Psi)$, where $p(v|\Psi)$ denotes the probability of $v$ being the $i$-th truth, computed by considering possible worlds $O$ with $i - 1$ values. Second, although there are multiple possible worlds with size $i - 1$, the nature of Bayesian analysis determines that one of them would have a much higher probability than the others, so we can use it for approximation. Third, once the probability of $\bot$ is above that of the $i$-th truth, it quickly increases to 1 in the following steps. Therefore if we terminate at the $i$-th step, we would not lose much. Recall that the confidence of a value $v$ does not change with $i$, but only that of $\bot$ changes, thus we can easily decide the number of steps we need before termination.

Algorithm 1 gives the details of the approximation.

- Without losing generality, let $L(v_1), L(v_2), \ldots$ be a sorted list in decreasing order, that is, $L(v_{i-1}) \geq L(v_i)$ for all $i > 1$ (Lines 10–11).
- We first initialize $p(v|\Psi) = 0$ for each $v \in V$. Then at each step $i$, we update $p(v|\Psi)$ by adding the probability $p(v|\Psi)$ for $v$ to be the $i$-th truth (Line 9). To compute $p(v|\Psi)$, we consider possible worlds where $v$ is not present yet (their probabilities sum up to $1 - p(v|\Psi)$). Assuming $v$ has the same conditional probability in all these possible worlds, denoted by $p(v|\Psi_1, \Psi)$, we have:

$$p(v|\Psi) = (1 - p(v|\Psi)) \cdot p(v|\Psi_1, \Psi).$$

- We obtain $p(v|\Psi_1, \Psi)$ from the possible world with the largest probability, which must have selected the $i - 1$ values with the highest confidence as truths; that is, $\Gamma_1 = v_1 v_2 \ldots v_{i-1}$. We thus compute $p(v|\Psi_1, \Psi)$ by normalizing the subsequence of confidence starting with $L(v_1)$ (Line 8). Note that for any possible world $O'$ with length $i - 1$, we have $p(v|O', \Psi) \leq p(v|\Psi_1, \Psi)$.

**Algorithm 1:** Approximation for HYBRID

```
input : Observations $\Omega$ containing a set $V$ of values provided by a set $O$ of sources on data item $d$; prior probability $\beta$
output : Probability $p(v|\Psi)$ for each $v \in V$

1. foreach $v \in V$ do
2.  $p(v|\Psi) \leftarrow 0$;
3.  Compute $L(v)$ using Eq. (17);
4.  Let $L(v_1), L(v_2), \ldots$ be a sorted list in decreasing order;
5.  foreach $i \in [1, |V|]$ do
6.    Compute $L_i(\bot)$ using Eq. (17);
7.    foreach $v \in V$ do
8.      $p(v|\Psi_1, \Psi) = \min_{\Gamma_1} \frac{L(v)}{\sum_{v' \in V \setminus \Gamma_1} L(v') + L_{|O|+1} \bot}$;
9.      $p(v|\Psi) \leftarrow p(v|\Psi) + (1 - p(v|\Psi)) \cdot p(v|\Psi_1, \Psi)$;
10.     if $L_i(\bot) > L(v_1)$ then
11.        break;
```

**Example 4.1.** Consider again computing $p(o_2|\Psi)$. The sorted list of the value confidences is [225, 225, 15, 15], given by $o_1, o_2, o_3$ and $o_4$; the confidences of $\bot$ in different steps are [1 0.04 18033, 18033]. We thus terminate after the third step (when $i = 3$).

When $i=1$, we compute $p_1(o_2|\Psi) = 0.47$.
When $i=2$, we compute $p_2(o_2|\Gamma_2, \Psi) = p(o_2|o_3, \Psi) = 0.88$, thus $p_2(o_2|\Psi) = (1 - 0.47) \times 0.88 = 0.47$. 

When \( i=3 \), we compute 
\[
p(o_2|\Gamma_3, \Psi) = \frac{225}{1+2+180} = 0.01,
\]
thus 
\[
p_1(o_2|\Psi) = (1 - 0.47 - 0.47) \times 0.01 = 0.0006.
\]
The final probability for \( o_2 \) is 
\[
p(o_2|\Psi) = p_1(o_2|\Psi) + p_2(o_2|\Psi) + p_3(o_2|\Psi) = 0.9406,
\]very close to the probability 0.92 obtained by HYBRID.

The next theorem shows that Algorithm 1 approximates the value probabilities both efficiently and effectively.

**Theorem 4.2.** Let \( d \) be a data item and \( n \) be the number of values provided for \( d \).

- Algorithm 1 estimates the probability of each provided value in time \( O(n^2) \).
- For each value \( v \) on \( d \), we have \( |p(v) - \hat{p}(v)| < \frac{1}{k} \), where \( \hat{p}(v) \) is the exact probability computed by HYBRID, and \( p(v) \) is the probability obtained by Algorithm 1.

**Proof.** See Appendix X.

5. EXPERIMENTAL STUDY

We now present experimental results to evaluate the proposed approach. Section 5.2 describes experimental settings. Then Section 5.2.2 gives a comprehensive comparison of various fusion models on a widely used real dataset as well as synthetic data, showing that HYBRID outperforms others in general and is the most robust.

### 5.1 Experimental Settings

**Methods to compare.** We compared the following fusion algorithms.

- **ACCUC** [1], the single-truth model reviewed in Section 1. For each data item, it considers the value with the highest predicted probability as the truth.
- **PRECREC** [7], the multi-truth model reviewed in Section 1. It considers a value correct if its predicted probability is above 0.5.
- **LTM** [9], a multi-truth model using directed graphical model. It also considers a value correct if its predicted probability is above 0.5.
- **TWOSTEP**, the baseline method described in Section 3. It first decides the number of truths \( k \), and then returns the \( k \) values with top probabilities according to ACCU.
- **HYBRID**, Algorithm 1 described in Section 3. It considers the values obtained before the termination step as the truths.

**Implementations.** Whenever applicable, we set \( n = 10, \alpha = 0.25 \), and consider only “good” sources (e.g., sources on which the conditions in Proposition 3.3 hold). We initialize the source quality metrics as \( A = 0.8, R = 0.8, Q = 0.2 \), and then iteratively compute value probabilities and source qualities for up to 5 iterations. We implemented all methods in Java on a MapReduce-based framework.

**Metrics.** We report precision and recall for each method. Precision measures among all values predicted as correct, the percentage that are true. Recall measures among all true observations, the percentage that are predicted as correct. F-measure is computed as \( \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \). (Note that they are different from precision and recall of individual sources as defined in Section 5.2).

### 5.2 Experiment Results

#### 5.2.1 Results on Book data

We first use the Book data from [8], which has been widely used for knowledge-fusion experiments. As shown in Table 5, it contains 6,139 book-author triples on 1,263 books from 876 retailers. The gold standard consists of authors for 100 randomly sampled books, where the authors were manually identified from book covers. According to the gold standard, 62% of the provided authors are correct, and 98% of the true values are provided by some source. There are 57% of the books that have multiple authors.

In addition to the five fusion methods we listed before, we also compared with ACCU_LIST, which applies ACCU but considers the full list of authors as a whole [1][8].

Table 5 shows the results, and we can see that HYBRID obtains a higher F-measure than existing single-truth and multi-truth models. By considering both conflicts between values and the possibility of having multiple truths, it is able to identify more true values without sacrificing much of precision. Not surprisingly, ACCU has the highest precision but the lowest recall as it only finds one author for a book. LTM has a lower precision as it lacks a global view of the values provided for the same data item. Instead, PRECREC has a lower recall but a higher precision: in this dataset many sources provide conflicting values, which makes it hard to choose the right one.

### 5.2.2 Results on Synthetic Data

To better understand the performance of different approaches in various situations, we compare them on synthetic data where we vary the number of truths and the quality of sources.

We generated 10 data sources providing values on 100 data items, where wrong values were randomly selected from a domain of 100 values. We varied the following parameters when generating the data.

- **Number of truths** for each data item: ranges from 1 to 10, and by default follows a Gaussian distribution with mean=6 and std=1.
- **Source accuracy**: ranges from 0.2 to 1, and 0.7 by default.
- **Source recall**: ranges from 0.2 to 1, and 0.7 by default.
- **Extra ratio**: equals to \( \frac{N_e}{N_e + N_w} \) (see Eq. (6)); ranges from 0.2 to 1, and 0.2 by default.

All experiments were repeated 100 times and we report the average.

Figure 1 shows the results when we vary the number of truths in data generation. HYBRID can fairly well “guess” the number of truths and consistently outperforms the others. As the number of truths increases, the precision of HYBRID remains high, while the precision of PRECREC drops. This is because the extra ratio is fixed; when there are more truths, there will be more wrong values and PRECREC is more sensitive to noise. Not surprisingly, ACCU always has the highest precision. LTM and TWOSTEP have low...
Figure 1: Varying the number of truths on synthetic data. **HYBRID** improves over other models when the number of truths is large.

Figure 2: Varying source accuracy. **HYBRID** obtains a significantly higher precision when source accuracy is low.

Figure 3: Varying source recall. **HYBRID** gives the highest precision and F1 when the sources have medium recall.

Figure 4: Varying extra ratio. **HYBRID** is the most robust and outperforms the others.
precision: the former lacks a global view of values for the same data item, while the latter may return false values with weak support.

Figures 2 and 3 plot the results of different methods when we vary source qualities. As expected, as the quality of the sources drops, the quality of the fusion results drops as well. However, we observe that HYBRID has the highest F-measure in general and it is the most robust. It usually gives significantly higher precision than PRECREC since it considers the conflicts between provided values as evidence to eliminate wrong values. While most methods perform better when the source quality increases, PRECREC eliminates wrong values. Extensive experiments show that HYBRID obtains the worst results when the source quality is medium (0.4-0.6). This is because, when the sources have similar probability of providing a true value and providing a false value, PRECREC is unable to distinguish them.

6. RELATED WORK

Data fusion [2,4] refers to the problem of identifying the truths from different values provided by various sources. We have provided a high-level review of different approaches in Section 2 and presented a comprehensive experimental study in Section 5.

A model called TEM [10] considers in addition whether the truth for a date item exists at all (i.e., date-of-death does not exist for an alive person). It is mainly designed for single-truth scenarios. The method in [3] considers the case where most sources provide only a few triples, thus source quality cannot be reliably estimated. The HYBRID model can be enhanced by these approaches.

7. CONCLUSION

In this paper we present an approach to find true values for an entity from information provided by different sources. It jointly makes two decisions on an entity: how many truths there are, and what they are. In this way, it allows the existence of multiple truths, while considering the conflicts between different values as important evidence for ruling out wrong values. Extensive experiments on both real-world and synthetic data show that the proposed outperforms the state-of-the-art techniques, and it is able to obtain a high precision without sacrificing the recall much.

8. REFERENCES

[1] X. L. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: the role of source dependence. PVLDB, 2009.
[2] M. Gupta and J. Han. Heterogeneous network-based trust analysis: a survey. ACM SIGKDD Explorations Newsletter, 2011.
[3] 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. PVLDB, 2014.
[4] X. Li, X. L. Dong, K. Lyons, W. Meng, and D. Srivastava. Truth finding on the deep web: Is the problem solved? PVLDB, 2013.
[5] Y. Li, J. Gao, C. Meng, Q. Li, L. Su, B. Zhao, W. Fan, and J. Han. A survey on truth discovery. arXiv preprint arXiv:1505.02463, 2015.
[6] X. Liu, X. L. Dong, B. C. Ooi, and D. Srivastava. Online data fusion. PVLDB, 2011.
[7] R. Pochampally, A. Das Sarma, X. L. Dong, A. Meliou, and D. Srivastava. Fusing data with correlations. In SIGMOD, 2014.
[8] X. Yin, J. Han, and P. S. Yu. Truth discovery with multiple conflicting information providers on the web. In KDD, 2007.
[9] B. Zhao, B. I. P. Rubinstein, J. Gemmell, and J. Han. A bayesian approach to discovering truth from conflicting sources for data integration. PVLDB, 2012.
[10] S. Zhi, B. Zhao, W. Tong, J. Gao, D. Yu, H. Ji, and J. Han. Modeling truth existence in truth discovery. In KDD, 2015.

APPENDIX

A. PROOF OF THEOREM 4.2

Theorem 4.2. Let d be a data item and n be the number of values provided for d.

- Algorithm 1 estimates the probability of each provided value in time O(n^2).
- For each value v on d, we have |p(v) − p̂(v)| < 1/n, where p̂(v) is the exact probability computed by HYBRID, and p(v) is the probability obtained by Algorithm 1.

Proof. We first consider the time complexity of Algorithm 1. Lines 1-11 have a complexity of O(n log n). The loops in Lines 5 and 7 take O(n^2) time. Therefore the overall complexity is O(n^2).

Next we prove the approximation bound of Algorithm 1. In this proof, we use a tree structure to illustrate the computations made by the full HYBRID model; see Figure 5 as an example. The root of the tree represents having not selected any value. A path from the root to a node represents a possible way of selecting v; for instance, the path v_1-v_2-v_3 corresponds to the case where we select v_3 after selecting v_1 and v_2 sequentially (i.e., O = v_1v_2). The children of a node represent candidates for the next truth. The number under each node v is the conditional probability p(v|O, Ψ). By multiplying the numbers along a path, we obtain the probability of the path. The overall p(v|Ψ) is thus the sum of the probabilities of all paths ending with v.

Algorithm 1 differs from the full HYBRID model in two places: (1) it terminates early when L_0(L) > L(v) without enumerating all possible worlds (Line 11), and (2) it assumes that v has the same conditional probability p(v|I_j, Ψ) under all possible worlds with size i - 1 (Line 5). The first one will make the approximated probability p(v) lower than the exact probability p̂(v) under the full model, while the second one will lead to a higher p(v). We next prove by constructing the worst case for each of them.

Case 1: Algorithm 1 terminates early such that p(v) < p̂(v). In this case the approximation error is due to the early termination: Algorithm 1 terminates at step i - 1 without increasing p(v) by p_j(v) (j ≥ i) in future steps. It is easy to check that Algorithm 1 outputs the same probabilities as the full model if the number of provided values is less than 3. So we require at least three values in the domain. The earliest possible termination is at step 2 (i.e., i = 3).

Next we construct a case with three values and Algorithm 1 terminates at step 2, such that the approximation error is reflected in one step (which is p_3(v)). By definition we have L(v_3) ≤ L(v_2) and L(v_3) ≥ L(v_1); to terminate at step 2, we need L(v_2) < L(v_1). It is easy to see that among the tree values, v_3 has the largest probability at step 3. To maximize the approximation error...
case by \( p_3(v_3) \), we need \( L(v_3) \) to be maximized and \( L_2(\perp) \) as well as \( L_3(\perp) \) to be minimized.

Suppose \( L(v_2) = a \), we then have \( L(v_3) = L(v_2) = a \); let \( \gamma \) be a real number where \( \gamma \geq 1 \), we have \( L(v_1) = \gamma a \). Further, let \( L_2(\perp) = L_3(\perp) = a + \epsilon \) where \( \epsilon \) is a very small constant. As usual, we have \( L_1(\perp) = 0 \). With the above setting, we compute all conditional probabilities following Section 3.2 and illustrate in Figure 5 (we omit \( \epsilon \)).

Next we compute the overall probability for \( v_3 \) using the full HYBRID model and Algorithm 1 respectively.

For the full HYBRID model, we find all paths ending with \( v_3 \) at each level of the tree:

- **Level 1:** \( \hat{p}_1(v_3) = \frac{1}{\gamma + 2} \);
- **Level 2:** \( \hat{p}_2(v_3) = \frac{\gamma}{\gamma + 2} \cdot \frac{1}{\gamma + 2} + \frac{1}{\gamma + 2} \cdot \frac{1}{\gamma + 2} \);
- **Level 3:** \( \hat{p}_3(v_3) = \frac{\gamma}{\gamma + 2} \cdot \frac{1}{\gamma + 2} \cdot \frac{1}{\gamma + 2} \cdot \frac{1}{\gamma + 2} \).

Finally, \( \hat{p}(v_3) = \hat{p}_1(v_3) + \hat{p}_2(v_3) + \hat{p}_3(v_3) = \frac{1}{\gamma + 2} + \frac{\gamma + 1}{2(\gamma + 2)} \).

For Algorithm 1 it terminates after 2 steps:

- **When \( i = 1 \):** \( p_1(v_3) = \frac{1}{\gamma + 2} \);
- **When \( i = 2 \):** \( p_2(v_3) = (1 - \frac{1}{\gamma + 2}) \times \frac{1}{\gamma + 2} = \frac{\gamma + 1}{3(\gamma + 2)} \);
- **Finally:** \( p(v_3) = p_1(v_3) + p_2(v_3) = \frac{1}{\gamma + 2} + \frac{\gamma + 1}{3(\gamma + 2)} \).

Therefore \( \hat{p}(v_3) - p(v_3) = \frac{1}{\gamma + 2} - \frac{\gamma + 1}{3(\gamma + 2)} < \frac{1}{\delta} \).

**Case II:** We assume the same conditional probability under all possible worlds such that \( p(v) > \hat{p}(v) \).

Recall that Eq. (19) assumes that in each step \( i \), \( v \) has the same conditional probability \( p(v|\Gamma_i, \Psi) \) under all possible worlds where \( v \) is not present yet. This \( p(v|\Gamma_i, \Psi) \) is an upper bound of the real conditional probability; we obtain the largest difference between \( p(v|\Gamma_i, \Psi) \) and the real conditional probability when the possible worlds end with \( \perp \) (the real probability is 0). We thus construct a case where \( p_2(\perp)p(v_3|\Gamma_3, \Psi) \) is maximized, leading to an over-estimation of \( p_3(v_3) \).

Similar to the previous case, we assume \( L(v_1) = \gamma a \) and \( L(v_2) = a \). In this case we want Algorithm 1 to continue when \( i = 3 \), so that \( p_3(v_3) \) is added to the overall probability of \( v_3 \). Therefore we require \( L_2(\perp) \leq L(v_2) \); to make \( p_3(\perp) \) larger, we have \( L_2(\perp) = L(v_2) = a \). Given that \( L(v_3) \geq L(v_2) \), to maximize \( p(v_3|\Gamma_3, \Psi) \), we need \( L(v_3) = L(v_2) = a \).

With the above setting, the probabilities computed by the full model remain the same, but Algorithm 1 continues when \( i = 3 \):

- \( p_3(v_3) = (1 - \frac{1}{\gamma + 2} - (1 - \frac{1}{\gamma + 2}) \times \frac{1}{\gamma + 2}) \times \frac{1}{\gamma + 2} = \frac{\gamma + 1}{3(\gamma + 2)} \);
- \( p(v_3) = p_1(v_3) + p_2(v_3) + p_3(v_3) = \frac{1}{\gamma + 2} + \frac{\gamma + 1}{3(\gamma + 2)} + \frac{\gamma + 1}{3(\gamma + 2)} = \frac{1}{\gamma + 2} + \frac{2(\gamma + 1)}{3(\gamma + 2)} \).

Therefore \( p(v_3) - \hat{p}(v_3) = \frac{1}{\delta} \cdot \frac{\gamma + 1}{\gamma + 2} < \frac{1}{\delta} \).

Combining the two cases, we have \( |p(v) - \hat{p}(v)| < \frac{1}{\delta} \). \( \square \)