Data Fusion: Resolving Conflicts from Multiple Sources

Xin Luna Dong, Laure Berti-Equille, and Divesh Srivastava

Abstract Many data management applications, such as setting up Web portals, managing enterprise data, managing community data, and sharing scientific data, require integrating data from multiple sources. Each of these sources provides a set of values, and different sources can often provide conflicting values. To present quality data to users, it is critical to resolve conflicts and discover values that reflect the real world; this task is called *data fusion*. Typically, we expect a true value to be provided by more sources than any particular false one, so we can take the value provided by the largest number of sources as the truth. Unfortunately, a false value can be spread through copying and that makes truth discovery extremely tricky. In this chapter, we consider how to find true values from conflicting information when there are a large number of sources, among which some may copy from others.

We describe a novel approach that considers *copying* between data sources in truth discovery. Intuitively, if two data sources provide a large number of common values and many of these values are unlikely to be provided by other sources (e.g., particular false values), it is very likely that one copies from the other. We apply Bayesian analysis to decide copying between sources and design an algorithm that iteratively detects dependence and discovers truth from conflicting information. We also consider *accuracy* of data sources and *similarity* between values in fusion to further improve the results. We present a case study on real-world data showing that

X.L. Dong (✉)
Google Inc., 1600 Amphitheater Pkwy, Mountain View, CA 94043, USA
e-mail: lunadong@research.att.com

L. Berti-Equille
IRD - Institut de Recherche pour le Développement, UMR 228 ESPACE-DEV, Maison de la Télédétection, 500 rue Jean-François Breton, 34093 MONTPELLIER Cedex 05, FRANCE
e-mail: Laure.Berti@ird.fr

D. Srivastava
AT&T Labs-Research, 180 Park Ave., Florham Park, NJ 07932, USA
e-mail: divesh@research.att.com

S. Sadiq (ed.), *Handbook of Data Quality*, DOI 10.1007/978-3-642-36257-6_13,

the described algorithm can significantly improve accuracy of truth discovery and is scalable when there are a large number of data sources.

1 Introduction

The amount of useful information available on the Web has been growing at a dramatic pace in recent years. In a variety of domains, such as science, business, technology, arts, entertainment, politics, government, sports, and tourism, there are a huge number of data sources that seek to provide information to a wide spectrum of information users. In addition to enabling the availability of useful information, the Web has also eased the ability to publish and spread false information across multiple sources. For example, an obituary of Apple founder Steve Jobs was published and sent to thousands of corporate clients on August 28, 2008, before it was retracted.[1] Such false information can often result in considerable damage; for example, the recent incorrect news about United Airlines filing for a second bankruptcy sent its shares tumbling, before the error was corrected.[2] The Web also makes it easy to rapidly spread rumors, which take a long time to die down. For example, the rumor from the late 1990s that the MMR vaccine given to children in Britain was harmful and linked to autism caused a significant drop in MMR coverage, leading autism experts to spend years trying to dispel the rumor.[3] Similarly, the upcoming experiments at the Large Hadron Collider (LHC) have sparked fears among the public that the LHC particle collisions might produce dangerous microscopic black holes that may mean the end of the world.[4]

Widespread availability of conflicting information (some true, some false) makes it hard to separate the wheat from the chaff. Simply using the information that is asserted by the largest number of data sources is clearly inadequate since biased (and even malicious) sources abound, and plagiarism (i.e., copying without proper attribution) between sources may be widespread. How can one find good answers to queries in such a "bad world?" Due to the evident need for practical solutions, topics such as lineage tracking [6–8] and source attribution [1, 3, 5, 9, 17, 20, 21, 27] have been widely studied. *Data fusion* is a promising approach in this space that aims at resolving conflicts from different sources and finds values that reflect the real world.

In this chapter, we describe how we find true values from conflicting information when there are a large number of sources, among which some may copy from others. First, our techniques consider trustworthiness of the sources and give more trust to sources that are more accurate. Second, we determine copying between data sources

[1] http://www.telegraph.co.uk/news/newstopics/howaboutthat/2638481/Steve-Jobs-obituary\-published-by-Bloomberg.html.

[2] http://gawker.com/5047763/how-robots-destroyed-united-airlines.

[3] http://www.guardian.co.uk/society/2008/apr/12/health.children.

[4] http://en.wikipedia.org/wiki/Large_Hadron_Collider#Safety_of_particle_collisions.

Table 1 The motivating example: five data sources provide information on the affiliations of five researchers

	S_1	S_2	S_3	S_4	S_5
Stonebraker	MIT	Berkeley	MIT	MIT	MS
Dewitt	MSR	MSR	UWisc	UWisc	UWisc
Bernstein	MSR	MSR	MSR	MSR	MSR
Carey	UCI	AT&T	BEA	BEA	BEA
Halevy	Google	Google	UW	UW	UW

Only S_1 provides all true values

and downweight copied values in truth discovery. We next illustrate our solution using an example.

Example 1. Consider the five data sources in Table 1. They provide information on affiliations of five researchers, and only S_1 provides all correct data. Sources S_4 and S_5 copy their data from S_3, and S_5 introduces certain errors during copying.

First consider the three sources S_1, S_2, and S_3. For all researchers except *Carey*, a majority voting on data provided by these three sources can find the correct affiliations. For *Carey*, these sources provide three different affiliations, resulting in a tie. However, if we take into account that the data provided by S_1 is more accurate (among the rest of the four researchers, S_1 provides all correct affiliations, whereas S_2 provides 3 and S_3 provides only 2 correct affiliations), we will consider *UCI* as most likely to be the correct value.

Now consider in addition sources S_4 and S_5. Since the affiliations provided by S_3 are copied by S_4 and S_5, naive voting would consider them as the majority and so make wrong decisions for three researchers. Only if we ignore the values provided by S_4 and S_5, we will be able to again decide the correct affiliations. Note however that identifying the copying relationships is not easy: while S_3 shares five values with S_4 and four values with S_5, S_1 and S_2 also share three values, more than half of all values.

2 Challenges and Overview of the Solution

Ideally, when applying voting, we would like to give a higher vote to more trustworthy sources and ignore copied information; however, this raises many challenges.

First, we often do not know a priori the trustworthiness of a source, and that depends on how much of its provided data are correct, but the correctness of data, on the other hand, needs to be decided by considering the number and trustworthiness of the providers; thus, it is a chicken-and-egg problem. Indeed, as we show soon, copy detection and truth discovery can also be a chicken-and-egg problem.

Second, in many applications we do not know how each source obtains its data, so we have to discover copiers from a snapshot of data. The discovery is nontrivial: sharing common data does not in itself imply copying—accurate sources can also

share a lot of independently provided correct data. Not sharing a lot of common data does not in itself imply no copying—a copier may copy only a small fraction of data from the original source; even when we decide that two sources are dependent, it is not always obvious which one is a copier.

Third, a copier can also provide some data by itself or verify the correctness of some of the copied data, so it is inappropriate to ignore all data it provides.

In this chapter, we present novel approaches for data fusion. First, we consider *copying* between data sources in truth discovery. Our technique considers not only whether two sources share the same values but also whether the shared values are true or false. Intuitively, for a particular object, there are often multiple distinct false values but usually only one true value. Sharing the same true value does not necessarily imply copying between sources; however, sharing the same false value is typically a low-probability event when the sources are fully independent. Thus, if two data sources share a lot of false values, copying is more likely. In the motivating example (Table 1), if we knew which values are true and which are false, we would suspect copying between S_3, S_4, and S_5, because they provide the same false values. On the other hand, we would suspect the copying between S_1 and S_2 much less, as they share only true values. Based on this analysis, we describe Bayesian models that compute the probability of copying between pairs of data sources and take the result into consideration in truth discovery.

We also consider *accuracy* in voting: we trust an accurate data source more and give values that it provides a higher weight. This method requires identifying not only if two sources are dependent but also which source is the copier. Indeed, accuracy in itself is a clue of direction of copying: given two data sources, if the accuracy of their common data is highly different from that of one of the sources, that source is more likely to be a copier.

Note that detection of copying between data sources is based on knowledge of true values and accuracy of sources, whereas correctly deciding true values requires knowledge of source copying and accuracy, and deciding accuracy of sources relies on the knowledge of which values are true and which are false. There is an interdependence between them, and we solve the problem by iteratively deciding source copying, discovering truth from conflicting information, and computing accuracy of sources, until the results converge.

In the rest of the chapter, we present how we can leverage source accuracy in data fusion in Sect. 3, present how we can leverage copying relationships in data fusion in Sect. 4, and present a case study of these techniques on a real-world data set in Sect. 5. The techniques we present in this chapter are mainly based on [14], and we shall briefly summarize other techniques in this area.

3 Fusing Sources Considering Accuracy

We first formally describe the data fusion problem and describe how we leverage the trustworthiness of sources in truth discovery. In this section we assume no copying between data sources and defer discussion on copying to the next section.

3.1 Data Fusion

We consider a set of *data sources* $\mathscr{S}$ and a set of *objects* $\mathscr{O}$. An object represents a particular aspect of a real-world entity, such as the affiliation of a researcher; in a relational database, an object corresponds to a cell in a table. For each object $O \in \mathscr{O}$, a source $S \in \mathscr{S}$ can (but not necessarily) provide a *value*. Among different values provided for an object, one correctly describes the real world and is *true*, and the rest are *false*. In this paper, we solve the following problem: given a snapshot of data sources in $\mathscr{S}$, decide the true value for each object $O \in \mathscr{O}$.

We note that a value provided by a data source can either be atomic or a set or list of atomic values (e.g., author list of a book). In the latter case, we consider the value as true if the atomic values are correct and the set or list is complete (and order preserved for a list). This setting already fits many real-world applications, and we refer our readers to [32] for solutions that treat a set or list of values as multiple values.

We start our discussion from a core case that satisfies the following two conditions, which we relax later:

- *Uniform false-value distribution:* For each object, there are multiple false values in the underlying domain, and an independent source has the same probability of providing each of them.
- *Categorical value:* For each object, values that do not match exactly are considered as completely different.

Note that this problem definition focuses on *static* information that does not evolve over time, such as authors and publishers of books; directors, actors, and actresses of movies; revenue of a company in past years; presidents of a country in the past; and capitals of countries. Data sources typically rarely update such information, and we consider a snapshot of data from different sources. There are also a lot of information that may evolve over time, such as people's contact information including phone numbers and addresses and businesses that can open or close. We refer our readers to [15] for data fusion for evolving values.

3.2 Accuracy of a Source

Let $S \in \mathscr{S}$ be a data source. The *accuracy* of S, denoted by $A(S)$, is the fraction of true values provided by S; it can also be considered as the probability that a value provided by S is the true value.

Ideally we should compute the accuracy of a source as it is defined; however, in real applications we often do not know for sure which values are true, especially among values that are provided by similar number of sources. Thus, we compute the accuracy of a source as the average probability of its values being true (we

describe how we compute such probabilities shortly). Formally, let $\bar{V}(S)$ be the values provided by S and denote by $|\bar{V}(S)|$ the size of $\bar{V}(S)$. For each $v \in \bar{V}(S)$, we denote by $P(v)$ the probability that v is true. We compute $A(S)$ as follows:

$$A(S) = \frac{\Sigma_{v \in \bar{V}(S)} P(v)}{|\bar{V}(S)|}. \tag{1}$$

We distinguish *good* sources from *bad* ones: a data source is considered to be good if for each object it is more likely to provide the true value than any *particular* false value; otherwise, it is considered to be bad. Assume for each object in $\mathscr{O}$ the number of false values in the domain is n. Then, in the core case, the probability that S provides a true value is $A(S)$ and that it provides a particular false value is $\frac{1-A(S)}{n}$. So S is good if $A(S) > \frac{1-A(S)}{n}$ (i.e., $A(S) > \frac{1}{1+n}$). We focus on good sources in the rest of this chapter, unless otherwise specified.

3.3 Probability of a Value Being True

Now we need a way to compute the probability that a value is true. Intuitively, the computation should consider both how many sources provide the value and accuracy of those sources. We apply a Bayesian analysis for this purpose.

Consider an object $O \in \mathscr{O}$. Let $\mathscr{V}(O)$ be the domain of O, including one true value and n false values. Let $\bar{S}_o$ be the sources that provide information on O. For each $v \in \mathscr{V}(O)$, we denote by $\bar{S}_o(v) \subseteq \bar{S}_o$ the set of sources that vote for v ($\bar{S}_o(v)$ can be empty). We denote by $\Psi(O)$ the observation of which value each $S \in \bar{S}_o$ votes for O.

To compute $P(v)$ for $v \in \mathscr{V}(O)$, we need to first compute the probability of $\Psi(O)$ conditioned on v being true. This probability should be that of sources in $\bar{S}_o(v)$ each providing the true value and other sources each providing a particular false value:

$$\begin{aligned} Pr(\Psi(O)|v \text{ true}) &= \Pi_{S \in \bar{S}_o(v)} A(S) \cdot \Pi_{S \in \bar{S}_o \setminus \bar{S}_o(v)} \frac{1 - A(S)}{n} \\ &= \Pi_{S \in \bar{S}_o(v)} \frac{nA(S)}{1 - A(S)} \cdot \Pi_{S \in \bar{S}_o} \frac{1 - A(S)}{n}. \end{aligned} \tag{2}$$

Among the values in $\mathscr{V}(O)$, there is one and only one true value. Assume our *a priori* belief of each value being true is the same, denoted by β. We then have

$$Pr(\Psi(O)) = \sum_{v \in \mathscr{V}(O)} \left(\beta \cdot \Pi_{S \in \bar{S}_o(v)} \frac{nA(S)}{1 - A(S)} \cdot \Pi_{S \in \bar{S}_o} \frac{1 - A(S)}{n} \right). \tag{3}$$

Applying the Bayes rule leads us to

$$P(v) = Pr(v \text{ true}|\Psi(O)) = \frac{\Pi_{S\in\bar{S}_o(v)} \frac{nA(S)}{1-A(S)}}{\sum_{v_0\in\mathscr{V}(O)} \Pi_{S\in\bar{S}_o(v_0)} \frac{nA(S)}{1-A(S)}}. \tag{4}$$

To simplify the computation, we define the *confidence* of v, denoted by $C(v)$, as[5]

$$C(v) = \sum_{S\in\bar{S}_o(v)} \log \frac{nA(S)}{1-A(S)}. \tag{5}$$

If we define the *accuracy score* of a data source S as

$$A'(S) = \log \frac{nA(S)}{1-A(S)}, \tag{6}$$

we have

$$C(v) = \sum_{S\in\bar{S}_o(v)} A'(S). \tag{7}$$

So we can compute the confidence of a value by summing up the accuracy scores of its providers. Finally, we can compute the probability of each value as follows:

$$P(v) = \frac{2^{C(v)}}{\sum_{v_0\in\mathscr{V}(O)} 2^{C(v_0)}}. \tag{8}$$

A value with a higher confidence has a higher probability to be true; thus, rather than comparing vote counts, we can just compare confidence of values. The following theorem shows three nice properties of Eq. (7):

Theorem 1. *Equation (7) has the following properties:*

1. *If all data sources are good and have the same accuracy, when the size of $\bar{S}_o(v)$ increases, $C(v)$ increases.*
2. *Fixing all sources in $\bar{S}_o(v)$ except S, when $A(S)$ increases for S, $C(v)$ increases.*
3. *If there exists $S \in \bar{S}_o(v)$ such that $A(S) = 1$ and no $S' \in \bar{S}_o(v)$ such that $A(S') = 0$, $C(v) = +\infty$; if there exists $S \in \bar{S}_o(v)$ such that $A(S) = 0$ and no $S' \in \bar{S}_o(v)$ such that $A(S') = 1$, $C(v) = -\infty$.*

[5]Note that the confidence of a value is derived from, but not equivalent to, the probability of the value.

Proof. We prove the three properties as follows:

1. When all data sources have the same accuracy, they have the same accuracy score. Let A' be the accuracy score and s be the size of $\bar{S}_o(v)$. Then $C(v) = s \cdot A'$, so $C(v)$ increases with s.
2. When $A(S)$ increases for a source S, $A'(S)$ increases as well and so $C(v)$ increases.
3. When $A(S) = 1$ for a source S, $A'(S) = \infty$ and $C(v) = \infty$. When $A(S) = 0$ for a source S, $A'(S) = -\infty$ and $C(v) = -\infty$.

Note that the first property is actually a justification for the naive voting strategy when all sources have the same accuracy. The third property shows that we should be careful not to assign very high or very low accuracy to a data source, which has been avoided by defining the accuracy of a source as the average probability of its provided values.

Example 2. Consider S_1, S_2, and S_3 in Table 1 and assume their accuracies are 0.97, 0.6, and 0.4, respectively. Assuming there are 5 false values in the domain (i.e., $n = 5$), we can compute the accuracy score of each source as follows: for S_1, $A'(S_1) = \log \frac{5*0.97}{1-0.97} = 4.7$; for S_2, $A'(S_2) = \log \frac{5*0.6}{1-0.6} = 2$; and for S_3, $A'(S_3) = \log \frac{5*0.4}{1-0.4} = 1.5$.

Now consider the three values provided for *Carey*. Value *UCI* thus has confidence 8, *AT&T* has confidence 5, and *BEA* has confidence 4. Among them, *UCI* has the highest confidence and so the highest probability to be true. Indeed, its probability is $\frac{2^8}{2^8+2^5+2^4+(5-2)*2^0} = 0.9$.

3.4 Iterative Algorithm

Once we know the confidence of each value, we can choose the one with the highest confidence as the true value. However, computing value confidence requires knowing accuracy of data sources, whereas computing source accuracy requires knowing value probability. There is an interdependence between them, and we solve the problem by computing them iteratively.

In particular, we discover true values from conflicting information provided by multiple data sources as follows.

Algorithm ACCU:

1. Initialize the same accuracy (0.8) to each source.
2. For each source, compute its accuracy score by Eq. (6).
3. For each value, add up the accuracy scores of its providers as its confidence [Eq. (7)].
4. For each value, compute its probability by applying Eq. (8).
5. For each source, take the average probability of its provided values as its accuracy [Eq. (1)].
6. If the accuracies of the sources converge, for each object, output the value with the highest confidence; otherwise, go back to Step 2.

Note that ACCU may not converge; we stop the process after we detect oscillation of decided true values. In practice it has been observed that when the number of objects is much higher than the number of sources, our algorithm typically converges soon; the results generated by different rounds during oscillation have similar overall quality.

3.5 Extensions and Alternatives

Similarity of Values: We consider similarity between values. Let v and v' be two values that are similar. Intuitively, the sources that vote for v' also implicitly vote for v and should be considered when counting votes for v. For example, a source that claims *UW* as the affiliation may actually mean *UWisc* and should be considered as an implicit voter of *UWisc*.

We can extend ACCU by incorporating value similarity as follows. Formally, we denote by $sim(v, v') \in [0, 1]$ the *similarity* between v and v', which can be computed based on edit distance of strings, difference between numerical values, etc. After computing the confidence of each value of object O, we adjust them according to the similarities between them as follows:

$$C^*(v) = C(v) + \rho \cdot \sum_{v' \neq v} C(v') \cdot sim(v, v'), \tag{9}$$

where $\rho \in [0, 1]$ is a parameter controlling the influence of similar values. We then use the adjusted confidence in computation in later rounds.

Nonuniform Distribution of False Values: In reality, false values of an object may not be uniformly distributed; for example, an out-of-date value or a value similar to the true value can occur more often than others. We extend ACCU for this situation as follows.

We denote by $Pop(v|v_t)$ the *popularity* of v among all false values conditioned on v_t being true. Then, the probability that source S provides the correct value (i.e., $\Psi_o(S) = v_t$) remains $A(S)$, but the probability that S provides a particular incorrect value becomes $(1 - A(S))Pop(\Psi_o(S)|v_t)$. Thus, we have

$$\begin{aligned} &Pr(\Psi(O)|v \text{ true}) \\ &= \Pi_{S \in \bar{S}_o(v)} A(S) \cdot \Pi_{S \in \bar{S}_o \setminus \bar{S}_o(v)} (1 - A(S)) Pop(\Psi_o(S)|v) \qquad (10) \\ &= \Pi_{S \in \bar{S}_o(v)} \frac{A(S)}{1 - A(S)} \cdot \Pi_{S \in \bar{S}_o \setminus \bar{S}_o(v)} Pop(\Psi_o(S)|v) \cdot \Pi_{S \in \bar{S}_o} (1 - A(S)). \qquad (11) \end{aligned}$$

Other Ways of Measuring Trustworthiness: There have been many other ways proposed for measuring the trustworthiness of a source. For example, the measures in [24,25] consider both correctness of data and coverage of provided objects from a

source; one measure in [16,29] measures the trustworthiness as the cosine similarity between the vector of provided values and the vector of correct values; other measures in [16, 29] also take an average of value confidence but consider not only the provided values but also the values that are not provided (i.e., voted against); techniques in [30, 31] measure source trustworthiness as its accuracy as we do but apply different Bayesian analysis; finally, [32] measures source trustworthiness by specificity and sensitivity in case that there are multiple true values.

4 Fusing Sources Considering Copying

Next, we describe how we detect copiers and leverage the discovered copying relationships in data fusion.

4.1 Copying Between Sources

We say that there exists *copying* between two data sources S_1 and S_2 if they derive the same part of their data directly or transitively from a common source (can be one of S_1 and S_2). Accordingly, there are two types of data sources: *independent sources* and *copiers*.

An *independent source* provides all values independently. It may provide some erroneous values because of incorrect knowledge of the real world, misspellings, etc.

A *copier* copies a part (or all) of data from other sources (independent sources or copiers). It can copy from multiple sources by union, intersection, etc., and as we focus on a snapshot of data, cyclic copying on a particular object is impossible. In addition, a copier may revise some of the copied values or add additional values, though, such revised and added values are considered as independent contributions of the copier.

To make our models tractable, we consider only *direct* copying in copy detection and truth discovery. We discuss at the end of this section how we distinguish transitive copying and co-copying from direct copying.

4.2 Copy Detection

We start with copy detection considering only correctness of values. To make the computation tractable, we make the following assumptions in copy detection:

- *Assumption 1 (independent values).* The values that are independently provided by a data source on different objects are independent of each other.
- *Assumption 2 (independent copying).* The copying between a pair of data sources is independent of the copying between any other pair of data sources.

- *Assumption 3 (no mutual copying).* There is no mutual copying between a pair of sources; that is, S_1 copying from S_2 and S_2 copying from S_1 do not happen at the same time.

We note that the real world is complex: different sources may represent the same value in different ways, error rates on different data items can be different, errors of certain types may happen more often, copiers can have various copying behaviors, etc. Instead of modeling every possible variant, the basic model we present in detail next captures the most significant aspects of data providing and copying, so are tractable and can avoid overfitting. Indeed, our experiments on real-world data show that it already obtains high accuracy. At the end of this section, we discuss briefly how we can extend the basic model by considering other aspects of data, such as coverage and formatting of data; by considering correlation on copying, such as copying all values associated with the same real-world entity; and by considering indirect copying, including transitive copying and co-copying.

We next describe the basic copy-detection model.

Assume $\mathscr{S}$ consists of two types of data sources: good independent sources and copiers. Consider two sources $S_1, S_2 \in \mathscr{S}$. We apply Bayesian analysis to compute the probability of copying between S_1 and S_2 given observation of their data. For this purpose, we need to compute the probability of the observed data, conditioned on independence of or copying between the sources.

Our computation requires several parameters: n ($n > 1$), the number of false values in the underlying domain for each object; c ($0 < c \leq 1$), the probability that a value provided by a copier is copied; and $A(S_1), A(S_2)$, the accuracies of the sources. Note that in practice, we may not know values of these parameters *a priori* and the values may vary from object to object and from source to source. We bootstrap our algorithms by setting the parameters to default values initially and iteratively refining them by computing the estimated values according to the truth discovery and copy detection results (details given shortly).

In our observation, we are interested in three sets of objects: $\bar{O}_t$, denoting the set of objects on which S_1 and S_2 provide the same true value; $\bar{O}_f$, denoting the set of objects on which they provide the same false value; and $\bar{O}_d$, denoting the set of objects on which they provide different values ($\bar{O}_t \cup \bar{O}_f \cup \bar{O}_d \subseteq \mathscr{O}$). Intuitively, two independent sources providing the same false value are a low-probability event; thus, if we fix $\bar{O}_t \cup \bar{O}_f$ and $\bar{O}_d$, the more common false values that S_1 and S_2 provide, the more likely that they are dependent. On the other hand, if we fix $\bar{O}_t$ and $\bar{O}_f$, the fewer objects on which S_1 and S_2 provide different values, the more likely that they are dependent. We denote by Φ the observation of $\bar{O}_t$, $\bar{O}_f$, and $\bar{O}_d$ and by k_t, k_f, and k_d their sizes, respectively. We next describe how we compute the conditional probability of Φ based on these intuitions.

We first consider the case where S_1 and S_2 are independent, denoted by $S_1 \perp S_2$. Since there is a single true value, the probability that S_1 and S_2 provide the same true value for object O is

$$Pr(O \in \bar{O}_t | S_1 \perp S_2) = A(S_1) \cdot A(S_2). \tag{12}$$

Under the *uniform-false-value-distribution* condition, the probability that source S provides a particular false value for object O is $\frac{1-A(S)}{n}$. Thus, the probability that S_1 and S_2 provide the same false value for O is

$$Pr(O \in \bar{O}_f | S_1 \perp S_2) = n \cdot \frac{1 - A(S_1)}{n} \cdot \frac{1 - A(S_2)}{n} = \frac{(1 - A(S_1))(1 - A(S_2))}{n}. \tag{13}$$

Then, the probability that S_1 and S_2 provide different values on an object O, denoted by P_d for convenience, is

$$Pr(O \in \bar{O}_d | S_1 \perp S_2) = 1 - A(S_1)A(S_2) - \frac{(1 - A(S_1))(1 - A(S_2))}{n} = P_d. \tag{14}$$

Following the *independent-values* assumption, the conditional probability of observing Φ is

$$Pr(\Phi | S_1 \perp S_2) = \frac{A(S_1)^{k_t} A(S_2)^{k_t} (1 - A(S_1))^{k_f} (1 - A(S_2))^{k_f} P_d^{k_d}}{n^{k_f}}. \tag{15}$$

We next consider the case when S_2 copies from S_1, denoted by $S_2 \rightarrow S_1$. There are two cases where S_1 and S_2 provide the same value v for an object O. First, with probability c, S_2 copies v from S_1, and so v is true with probability $A(S_1)$ and false with probability $1 - A(S_1)$. Second, with probability $1 - c$, the two sources provide v independently, and so its probability of being true or false is the same as in the case where S_1 and S_2 are independent. Thus, we have

$$Pr(O \in \bar{O}_t | S_2 \rightarrow S_1) = A(S_1) \cdot c + A(S_1) \cdot A(S_2) \cdot (1 - c), \tag{16}$$

$$Pr(O \in \bar{O}_f | S_2 \rightarrow S_1) = (1 - A(S_1)) \cdot c + \frac{(1 - A(S_1))(1 - A(S_2))}{n} \cdot (1 - c). \tag{17}$$

Finally, the probability that S_1 and S_2 provide different values on an object is that of S_1 providing a value independently, and the value differs from that provided by S_2:

$$Pr(O \in \bar{O}_d | S_2 \rightarrow S_1) = P_d \cdot (1 - c). \tag{18}$$

We compute $Pr(\Phi | S_2 \rightarrow S_1)$ accordingly; similarly we can also compute $Pr(\Phi | S_1 \rightarrow S_2)$. Now we can compute the probability of $S_1 \perp S_2$ by applying the Bayes rule:

$$\begin{aligned} &Pr(S_1 \perp S_2 | \Phi) \\ &= \frac{\alpha Pr(\Phi | S_1 \perp S_2)}{\alpha Pr(\Phi | S_1 \perp S_2) + \frac{1-\alpha}{2} Pr(\Phi | S_1 \rightarrow S_2) + \frac{1-\alpha}{2} Pr(\Phi | S_2 \rightarrow S_1)}. \end{aligned} \tag{19}$$

Here $\alpha = Pr(S_1 \perp S_2)(0 < \alpha < 1)$ is the *a priori* probability that two data sources are independent. As we have no *a priori* preference for copy direction, we set the *a priori* probability for copying in each direction as $\frac{1-\alpha}{2}$.

Equation (19) has several nice properties that conform to the intuitions we discussed early in this section, formalized as follows:

Theorem 2. *Let $\mathscr{S}$ be a set of good independent sources and copiers. Equation (19) has the following three properties on $\mathscr{S}$.*

1. *Fixing $k_t + k_f$ and k_d, when k_f increases, the probability of copying (i.e., $Pr(S_1 \to S_2|\Phi) + Pr(S_2 \to S_1|\Phi)$) increases.*
2. *Fixing $k_t + k_f + k_d$, when $k_t + k_f$ increases and none of k_t and k_f decreases, the probability of copying increases.*
3. *Fixing k_t and k_f, when k_d decreases, the probability of copying increases.*

Proof. We prove the three properties assuming each source has accuracy $1 - \varepsilon$ (ε can be considered as the error rate) as follows, and we can extend for the case where each source has a different accuracy. We only need to prove that the opposite holds for $Pr(S_1 \perp S_2|\Phi)$.

1. Let $k_0 = k_t + k_f + k_d$. Then, $k_d = k_0 - k_t - k_f$. We have

$$Pr(S_1 \perp S_2|\Phi) = 1 - \left(1 + (\frac{1-\alpha}{\alpha})(\frac{1-\varepsilon-c+c\varepsilon}{1-\varepsilon+c\varepsilon})^{k_t}(\frac{\varepsilon-c\varepsilon}{cn+\varepsilon-c\varepsilon})^{k_f}(\frac{1}{1-c})^{k_0}\right)^{-1}.$$

   As $0 < c < 1$, we have $0 < \frac{1-\varepsilon-c+c\varepsilon}{1-c\varepsilon} < 1$ and $0 < \frac{\varepsilon-c\varepsilon}{cn+\varepsilon-c\varepsilon} < 1$. When k_t or k_f increases, $(\frac{1-\varepsilon-c+c\varepsilon}{1-c\varepsilon})^{k_t}$ or $(\frac{\varepsilon-c\varepsilon}{cn+\varepsilon-c\varepsilon})^{k_f}$ decreases. Thus, $Pr(S_1 \perp S_2|\Phi)$ decreases.
2. Let $k_c = k_t + k_f$. Then, $k_t = k_c - k_f$. We have

$$Pr(S_1 \perp S_2|\Phi) = 1 - \left(1 + (\frac{1-\alpha}{\alpha})(\frac{1-\varepsilon}{1-\varepsilon+c\varepsilon})^{k_c}(\frac{\varepsilon(1-\varepsilon+c\varepsilon)}{(1-\varepsilon)(cn+\varepsilon-c\varepsilon)})^{k_f}(\frac{1}{1-c})^{k}\right)^{-1}.$$

   Because $\varepsilon < \frac{n}{n+1}$, $\varepsilon(1-\varepsilon+c\varepsilon) < (1-\varepsilon)(cn+\varepsilon-c\varepsilon)$. Thus, when k_f increases, $(\frac{\varepsilon(1-c\varepsilon)}{(1-\varepsilon)(n-cn+c\varepsilon)})^{k_f}$ decreases and so $Pr(S_1 \perp S_2|\Phi)$ decreases.
3. Because k_d increases, $(\frac{1}{1-c})^{k_d}$ increases, and so $Pr(S_1 \perp S_2|\Phi)$ increases.

Example 3. Continue with Ex.1 and consider the possible copying relationship between S_1 and S_2. We observe that they share no false values (all values they share are correct), so copying is unlikely. With $\alpha = 0.5, c = 0.2, A(S_1) = 0.97$, *and* $A(S_2) = 0.6$, the Bayesian analysis goes as follows.

We start with computation of $Pr(\Phi|S_1 \perp S_2)$. We have $Pr(O \in \bar{O}_t|S_1 \perp S_2) = 0.97 * 0.6 = 0.582$. There is no object in $\bar{O}_f$, and we denote by P_d the probability $Pr(O \in \bar{O}_f|S_1 \perp S_2)$. Thus, $Pr(\Phi|S_1 \perp S_2) = 0.582^3 * P_d^2 = 0.2P_d^2$.

Next consider $Pr(\Phi|S_1 \rightarrow S_2)$. We have $Pr(O \in \bar{O}_t|S_1 \perp S_2) = 0.8 * 0.6 + 0.2 * 0.582 = 0.6$ and $Pr(O \in \bar{O}_f|S_1 \rightarrow S_2) = 0.2P_d$. Thus, $Pr(\Phi|S_1 \rightarrow S_2) = 0.6^3 * (0.2P_d)^2 = 0.008P_d^2$. Similarly, $Pr(\Phi|S_2 \rightarrow S_1) = 0.028P_d^2$.

According to Eq. (19), $Pr(S_1 \perp S_2|\Phi) = \frac{0.5*0.2P_d^2}{0.5*0.2P_d^2+0.25*0.008P_d^2+0.25*0.028P_d^2} = 0.92$, so independence is very likely.

4.3 Independent Vote Count of a Value

We have described how we decide if two sources are dependent. However, even if a source copies from another, it is possible that it provides some of the values independently, so it would be inappropriate to treat these values as copied values and ignore them. We next describe how to count the *independent* vote for a particular value. We start with ideal vote count assuming all sources have the same accuracy, then describe an approximation, and finally describe how to combine the independent vote count with source accuracy.

4.3.1 Ideal Vote Count

We start from the case where we know deterministically the copying relationship between sources and discuss probabilistic copying subsequently. Consider a specific value v for a particular object O and let $\bar{S}_o(v)$ be the set of data sources that provide v on O. We can draw a *copying graph* G, where for each $S \in \bar{S}_o(v)$, there is a node and for each $S_1, S_2 \in \bar{S}_o(v)$ where S_1 copies from S_2, there is an edge from S_1 to S_2.

For each $S \in \bar{S}_o(v)$, we denote by $d(S, G)$ the out-degree of S in G, corresponding to the number of data sources from which S copies. If $d(S, G) = 0$, S is independent and its vote count for v is 1. Otherwise, for each source S' that S copies from, S provides a value independently of S' with probability $1 - c$. According to the *independent-copying* assumption, the probability that S provides v independently of any other source is $(1 - c)^{d(S,G)}$ and the total vote count of v with respect to G is

$$V(v, G) = \sum_{S \in \bar{S}_o(v)} (1 - c)^{d(S,G)}. \tag{20}$$

However, recall that Eq. (19) computes only a probability of copying in each direction. Thus, we have to enumerate all possible copying graphs and take the sum of the vote count with respect to each of them, weighted by the probability of the graph. Let $\bar{D}_o$ be the set of possible copying between sources in $\bar{S}_o(v)$, and we denote the probability of $D \in \bar{D}_o$ by $p(D)$. Consider a subset $\bar{D} \subseteq \bar{D}_o$ of m

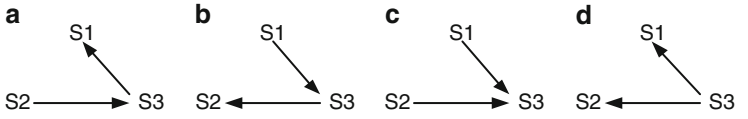

Fig. 1 Copying graphs with a copying between S_1 and S_3 and one between S_2, and S_3, where S_1, S_2, and S_3 provide the same value on an object

copyings. According to the *independent-copying* assumption, the probability that all and only copying relationships in $\bar{D}$ hold is

$$Pr(\bar{D}) = \Pi_{D \in \bar{D}} p(D) \Pi_{D \in \bar{D}_o - \bar{D}} (1 - p(D)). \tag{21}$$

As each copying can have one of the two directions, there are up to 2^m acyclic copying graphs with this set of copying relationships. Intuitively, the more independent sources in a graph, the less likely that all sources in the graph provide the same value. By applying Bayesian analysis, we can compute the probability of each graph. We skip the equations for space reasons and illustrate the computation of vote count in the following example:

Example 4. Consider three data sources S_1, S_2 and S_3 that provide the same value v on an object. Assume $c = 0.8$ and between each pair of sources the probability of copying is 0.4 (0.2 in each direction). We can compute v's vote count by enumerating all possible copying graphs:

- There is 1 graph with no copying. All sources are independent so the vote count is $1 + 1 + 1 = 3$. The probability of this graph is $(1 - 0.4)^3 = 0.216$.
- There are 6 graphs with only one copying. The total probability of graphs that contain a particular copying is $(1 - 0.4)^2 * 0.4 = 0.144$. Each copying has two directions, so the probability of each such graph is $0.144/2 = 0.072$. No matter which direction the copying is in, the vote count is $1 + 1 + 0.2 = 2.2$.
- There are 12 graphs with two copyings. Figure 1 shows the 4 that contain a copying between S_1 and S_3 and a copying between S_2 and S_3. The sum of their probabilities is $(1 - 0.4) * 0.4^2 = 0.096$. For each of the first three graphs (Fig. 1a–c, each with a single independent source), the vote count is $1 + 0.2 + 0.2 = 1.4$, and by applying the Bayes rule, we compute its probability as $0.32 * 0.096 = 0.03$. For the last one (Fig. 1d, with two independent sources), the vote count is $1 + 1 + 0.2^2 = 2.04$ and its probability is $0.04 * 0.096 = 0.004$.
- Finally, there are 6 acyclic graphs with three copyings (details ignored to save space), where each has vote count $1 + 0.2 + 0.2^2 = 1.24$ and probability $0.4^3/6 = 0.011$.

The total vote count of v, computed as the weighted sum, is 2.08.

4.3.2 Estimating Vote Count

As there are an exponential number of copying graphs, computing the vote count by enumerating all of them can be quite expensive. To make the analysis scalable, we shall find a way to estimate the vote count in polynomial time.

We estimate a vote count by considering the data sources one by one. For each source S, we denote by $\overline{Pre}(S)$ the set of sources that have already been considered and by $\overline{Post}(S)$ the set of sources that have not been considered yet. We compute the probability that the value provided by S is independent of any source in $\overline{Pre}(S)$ and take it as the vote count of S. The vote count computed in this way is not precise because if S depends only on sources in $\overline{Post}(S)$ but some of those sources depend on sources in $\overline{Pre}(S)$, our estimation still (incorrectly) counts S's vote. To minimize such error, we wish that the probability that S depends on a source $S' \in \overline{Post}(S)$ and S' depends on a source $S'' \in \overline{Pre}(S)$ be the lowest. Thus, we use a greedy algorithm and consider data sources in the following order:

1. If the probability of $S_1 \rightarrow S_2$ is much higher than that of $S_2 \rightarrow S_1$, we consider S_1 as a copier of S_2 with probability $Pr(S_1 \rightarrow S_2|\Phi) + Pr(S_2 \rightarrow S_1|\Phi)$ (recall that we assume there is no mutual copying) and order S_2 before S_1. Otherwise, we consider both directions as equally possible, and there is no particular order between S_1 and S_2; we consider such copying *undirectional*.
2. For each subset of sources between which there is no particular ordering yet, we sort them as follows: in the first round, we select a data source that is associated with the undirectional copying of the highest probability ($Pr(S_1 \rightarrow S_2|\Phi) + Pr(S_2 \rightarrow S_1|\Phi)$); in later rounds, each time we select a data source that has the copying with the maximum probability with one of the previously selected sources.

We now consider how to compute the vote count of v once we have decided an order of the data sources. Let S be a data source that votes for v. The probability that S provides v independently of a source $S_0 \in \overline{Pre}(S)$ is $1 - c(Pr(S_1 \rightarrow S_0|\Phi) + Pr(S_0 \rightarrow S_1|\Phi))$, and the probability that S provides v independently of any data source in $\overline{Pre}(S)$, denoted by $I(S)$, is

$$I(S) = \Pi_{S_0 \in \overline{Pre}(S)}(1 - c(Pr(S_1 \rightarrow S_0|\Phi) + Pr(S_0 \rightarrow S_1|\Phi))). \quad (22)$$

The total vote count of v is $\sum_{S \in \bar{S}_o(v)} I(S)$.

Example 5. Continue with Example 4. As all copyings have the same probability, we can consider the data sources in any order. We choose the order of S_1, S_2, S_3. The vote count of S_1 is 1, that of S_2 is $1 - 0.4 * 0.8 = 0.68$, and that of S_3 is $0.68^2 = 0.46$. So the estimated vote count is $1 + 0.68 + 0.46 = 2.14$, very close to the real one, 2.08.

We formalize properties of the vote-count estimation as follows, showing scalability of our estimation algorithm:

Algorithm 2: ACCUCOPY: Discover true values by considering accuracy of and copying between data sources.

Input: $\mathscr{S}, \mathscr{O}$.
Output: The true value for each object in $\mathscr{O}$.
Set the accuracy of each source as $1 - \epsilon$;
while (accuracy of sources changes && no oscillation of decided true values)
  Compute probability of copying between each pair of sources;
  Sort sources according to the copyings;
  Compute confidence of each value for each object;
  Compute accuracy of each source;
for each ($O \in \mathscr{O}$)
  Among all values of O, select the one with the highest confidence as the true value;

Theorem 3. *Our vote-count estimation has the following two properties:*

1. *Let t_0 be the ideal vote count of a value and t be the estimated vote count. Then, $t_0 \leq t \leq 1.5t_0$.*
2. *Let s be the number of sources that provide information on an object. We can estimate the vote count of all values of this object in time $O(s^2 \log s)$.*

4.3.3 Combining with Source Accuracy

Finally, when we consider the accuracy of sources, we compute the confidence of v as follows:

$$C(v) = \sum_{S \in \bar{S}_o(v)} A'(S) I(S). \tag{23}$$

In the equation, $I(S)$ is computed by Eq. (22). In other words, we take only the "independent fraction" of the original vote count (decided by source accuracy) from each source.

4.4 Iterative Algorithm

We now extend the ACCU algorithm to incorporate analysis of source copying. We need to compute three measures: accuracy of sources, copying between sources, and confidence of values. Accuracy of a source depends on confidence of values, copying between sources depends on accuracy of sources and the true values selected according to the confidence of values, and confidence of values depends on both accuracy of and copying between data sources.

We conduct analysis of both accuracy and copying in each round. Specifically, Algorithm ACCUCOPY starts by setting the same accuracy for each source and the same probability for each value ; then iteratively (1) computes copying based on

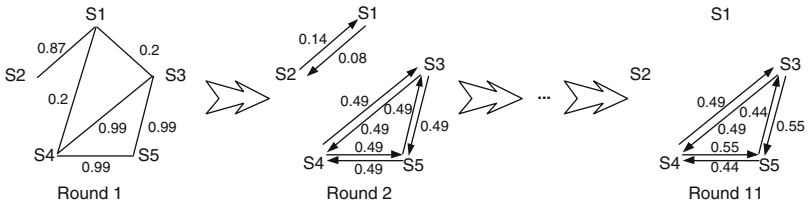

Fig. 2 Probabilities of copyings computed by ACCUCOPY on the motivating example. We only show copyings where the sum of the probabilities in both directions is over 0.1

Table 2 Accuracy of data sources computed by ACCUCOPY on the motivating example

	S_1	S_2	S_3	S_4	S_5
Round 1	0.52	0.42	0.53	0.53	0.53
Round 2	0.63	0.46	0.55	0.55	0.41
Round 3	0.71	0.52	0.53	0.53	0.37
Round 4	0.79	0.57	0.48	0.48	0.31
...	...	...	...	...	...
Round 11	0.97	0.61	0.40	0.40	0.21

the confidence of values computed in the previous round, (2) updates confidence of values accordingly, and (3) updates accuracy of sources accordingly, and stops when the accuracy of the sources becomes stable. Note that it is crucial to consider copying between sources from the beginning; otherwise, a data source that has been duplicated many times can dominate the vote results in the first round and make it hard to detect the copying between it and its copiers (as they share only "true" values). Our initial decision on copying is similar to Eq. (19) except considering both the possibility of a value being true and that of the value being false, and we skip details here.

We can prove that if we ignore source accuracy (i.e., assuming all sources have the same accuracy) and there are a finite number of objects in $\mathscr{O}$, Algorithm ACCUCOPY cannot change the decision for an object O back and forth between two different values forever; thus, the algorithm converges.

Theorem 4. *Let $\mathscr{S}$ be a set of good independent sources and copiers that provide information on objects in $\mathscr{O}$. Let l be the number of objects in $\mathscr{O}$ and n_0 be the maximum number of values provided for an object by $\mathscr{S}$. The* ACCUVOTE *algorithm converges in at most $2ln_0$ rounds on $\mathscr{S}$ and $\mathscr{O}$ if it ignores source accuracy.*

Once we consider accuracy of sources, ACCUCOPY may not converge: when we select different values as the true values, the direction of the copying between two sources can change and in turn suggest different true values. As in ACCU, we stop the process after we detect oscillation of decided true values. Finally, we note that the complexity of each round is $O(|\mathscr{O}||\mathscr{S}|^2 \log |\mathscr{S}|)$.

Example 6. Continue with the motivating example. Figure 2 shows the probability of copying, Table 2 shows the computed accuracy of each data source, and Table 3 shows the confidence of affiliations computed for *Carey* and *Halevy*.

Table 3 Confidence of affiliations computed for *Carey* and *Halevy* in the motivating example

	Carey			*Halevy*	
	UCI	AT&T	BEA	Google	UW
Round 1	1.61	1.61	2.0	2.1	2.0
Round 2	1.68	1.3	2.12	2.74	2.12
Round 3	2.12	1.47	2.24	3.59	2.24
Round 4	2.51	1.68	2.14	4.01	2.14
...	...	...	...	...	...
Round 11	4.73	2.08	1.47	6.67	1.47

Initially, Line 1 of Algorithm AccuCopy sets the accuracy of each source to 0.8. Accordingly, Line 3 computes the probability of copying between sources as shown on the left of Fig. 2. Taking the copying into consideration, Line 5 computes confidence of the values; for example, for *Carey* it computes 1.61 as the confidence of value *UCI* and *AT&T* and 2.0 as the confidence of value *BEA*. Then, Line 6 updates the accuracy of each source to 0.52, 0.42, 0.53, 0.53, and 0.53, respectively, according to the computed value confidence; the updated accuracy is used in the next round.

Starting from the second round, S_1 is considered more accurate and its values are given higher weight. In later rounds, AccuCopy gradually increases the accuracy of S_1 and decreases that of S_3, S_4, and S_5. At the fourth round, AccuCopy decides that *UCI* is the correct affiliation for *Carey* and finds the right affiliations for all researchers. Finally, AccuCopy terminates at the eleventh round, and the source accuracy it computes converges close to the expected ones (1, 0.6, 0.4, 0.4, 0.2, respectively).

4.5 Extensions for Copy Detection

We next describe several extensions for copy detection.

Considering Other Aspects of Data [13]: In addition to the values provided by each source, we can also obtain evidence for copying from other aspects of data, such as coverage of the data and formatting of the data. Copying is considered likely if two sources share a lot of objects that are rarely provided by others, if they use common rare formats, and so on.

Correlated Copying [2, 13]: The basic model assumes *item-wise independence*, which seldom holds in reality. One can imagine that a copier often copies in one of two modes: (1) it copies data for a subset of entities on a subset of attributes (e.g., title, author list, and publisher of a book), called *per-entity copying*; (2) it copies on a subset of attributes for a set of entities that it provides independently (or entities copied from other sources), called *per-attribute* copying. We can distinguish these two modes in copy detection.

Global Copy Detection [13]: The copying discovered by local detection may be due to co-copying or transitive copying. For example, if S_3 copies from S_1 and S_2 and S_4 copies from S_3, local detection may conclude with $S_4 \rightarrow S_1$ and $S_4 \rightarrow S_2$. The goal of global detection is to fix this problem. The key intuition employed in global detection is that since co-copying and transitive copying can often be inferred from direct copying, we first find a set of copying relationships **R** that significantly influence the rest of the relationships and take them as direct copyings. Then, for each of the remaining copyings, we judge if it is indirect conditioned on **R**; in other words, in global detection we compute $Pr(S_1 \rightarrow S_2|\Phi, \mathbf{R})$ instead of $Pr(S_1 \rightarrow S_2|\Phi)$ for pairs outside **R**.

Dynamic Data [15]: When we know the update history, we employ a Hidden Markov Model (HMM) to decide whether a source copies from another source and at which moments it copies, exploiting the intuition that the copying relationships can evolve over time, but frequent back-and-forth changes are unlikely.

5 A Case Study

We now describe a case study on a real-world data set extracted by searching computer-science books on *AbeBooks.com*. For each book, *AbeBooks.com* returns information provided by a set of online bookstores. Our goal is to find the list of authors for each book. In the data set there are 877 bookstores, 1,263 books, and 24,364 listings (each listing contains a list of authors on a book provided by a bookstore).

We did a normalization of author names and generated a normalized form that preserves the order of the authors and the first name and last name (ignoring the middle name) of each author. On average, each book has 19 listings; the number of different author lists after cleaning varies from 1 to 23 and is 4 on average.

We used a golden standard that contains 100 randomly selected books and the list of authors found on the cover of each book. We compared the fusion results with the golden standard, considering missing or additional authors, misordering, misspelling, and missing first name or last name as errors; however, we do not report missing or misspelled middle names. Table 4 shows the number of errors of different types on the selected books if we apply a naive voting (note that the result author lists on some books may contain multiple types of errors).

We define *precision* of the results as the fraction of objects on which we select the true values (as the number of true values we return and the real number of true values are both the same as the number of objects, the *recall* of the results is the same as the precision). Note that this definition is different from that of accuracy of sources.

Precision and Efficiency

We compared the following data fusion models on this data set:

- VOTE conducts naive voting.
- SIM conducts naive voting but considers similarity between values.

Table 4 Different types of errors by naive voting

Missing authors	Additional authors	Misordering	Misspelling	Incomplete names
23	4	3	2	2

Table 5 Results on the book data set

Model	Precision	Rounds	Time (s)
VOTE	0.71	1	0.2
SIM	0.74	1	0.2
ACCU	0.79	23	1.1
COPY	0.83	3	28.3
ACCUCOPY	0.87	22	185.8
ACCUCOPYSIM	0.89	18	197.5

For each method, we report the precision of the results, the run time, and the number of rounds for convergence. ACCUCOPY and COPY obtain a high precision

- ACCU considers accuracy of sources as we described in Sect. 3 but assumes all sources are independent.
- COPY considers copying between sources as we described in Sect. 4 but assumes all sources have the same accuracy.
- ACCUCOPY applies the ACCUCOPY algorithm described in Sect. 4, considering both source accuracy and copying.
- ACCUCOPYSIM applies the ACCUCOPY algorithm and considers in addition similarity between values.

When applicable, we set $\alpha = 0.2, c = 0.8, \varepsilon = 0.2$, and $n = 100$, though, we observed that ranging α from 0.05 to 0.5, ranging c from 0.5 to 0.95, and ranging ε from 0.05 to 0.3 did not change the results much. We compared similarity of two author lists using 2-g Jaccard distance.

Table 5 lists the precision of results of each algorithm. ACCUCOPYSIM obtained the best results and improved over VOTE by 25.4 %. SIM, ACCU, and COPY each extends VOTE on a different aspect; while all of them increased the precision, COPY increased it the most.

To further understand how considering copying and accuracy of sources can affect our results, we looked at the books on which ACCUCOPY and VOTE generated different results and manually found the correct authors. There are 143 such books, among which ACCUCOPY gave correct authors for 119 books, VOTE gave correct authors for 15 books, and both gave incorrect authors for 9 books.

Finally, COPY was quite efficient and finished in 28.3 seconds. It took ACCUCOPY and ACCUCOPYSIM longer time to converge (3.1 and 3.3 min, respectively), though, truth discovery is often a one-time process, and so taking a few minutes is reasonable.

Table 6 Bookstores that are likely to be copied by more than ten other bookstores

Bookstore	#Copiers	#Books	Accuracy
Caiman	17.5	1024	0.55
MildredsBooks	14.5	123	0.88
COBU GmbH & Co. KG	13.5	131	0.91
THESAINTBOOKSTORE	13.5	321	0.84
Limelight Bookshop	12	921	0.54
Revaluation Books	12	1091	0.76
Players Quest	11.5	212	0.82
AshleyJohnson	11.5	77	0.79
Powell's Books	11	547	0.55
AlphaCraze.com	10.5	157	0.85
Avg	12.8	460	0.75

For each bookstore, we show the number of books it lists and its accuracy computed by ACCUCOPYSIM

Table 7 Difference between accuracy of sources computed by our algorithms and the sampled accuracy on the golden standard

	Sampled	ACCUCOPYSIM	ACCUCOPY	ACCU
Average source accuracy	0.542	0.607	0.614	0.623
Average difference	–	0.082	0.087	0.096

The accuracy computed by ACCUCOPYSIM is the closest to the sampled accuracy

Copying and Source Accuracy:

Out of the 385,000 pairs of bookstores, 2,916 pairs provide information on at least the same 10 books, and among them ACCUCOPYSIM found 508 pairs that are likely to be dependent. Among each such pair S_1 and S_2, if the probability of S_1 depending on S_2 is over 2/3 of the probability of S_1 and S_2 being dependent, we consider S_1 as a *copier* of S_2; otherwise, we consider S_1 and S_2 each has 0.5 probability to be a *copier*. Table 6 shows the bookstores whose information is likely to be copied by more than ten bookstores. On average each of them provides information on 460 books and has accuracy 0.75. Note that among all bookstores, on average each provides information on 28 books, conforming to the intuition that small bookstores are more likely to copy data from large ones. Interestingly, when we applied VOTE on only the information provided by bookstores in Table 6, we obtained a precision of only 0.58, showing that bookstores that are large and copied often actually can make a lot of mistakes.

Finally, we compare the source accuracy computed by our algorithms with that sampled on the 100 books in the golden standard. Specifically, there were 46 bookstores that provide information on more than 10 books in the golden standard. For each of them we computed the *sampled accuracy* as the fraction of the books on which the bookstore provides the same author list as the golden standard. Then, for each bookstore we computed the difference between its accuracy computed by one of our algorithms and the sampled accuracy (Table 7). The source accuracy computed by ACCUCOPYSIM is the closest to the sampled accuracy, indicating

the effectiveness of our model on computing source accuracy and showing that considering copying between sources helps obtain better source accuracy.

6 Related Work

Our work is closely related to two research areas: (a) data provenance and (b) trust and authoritativeness of data sources.

Data Provenance. Representing and analyzing provenance has been a topic of research since a decade ago [4, 6]. In the literature (e.g., *Open Provenance Model* [23]), provenance is classically modeled as a directed acyclic graph (DAG): the nodes in the DAG represent objects such as files, processes, tuples, and data sets; the edges between two nodes indicate a dependency between the objects. Simmhan et al. [26] provide a taxonomy of provenance characteristics and classify the approaches into data-oriented approaches and process-oriented approaches. Whereas data-oriented approaches focus on data items, process-oriented approaches emphasize information about the processes that produce or consume the data. Bunemann et al. [6–8] identify several open issues for data provenance in the Web era such as (a) obtaining provenance information, (b) citing components of a data resource that may be (components of) another resource in another context, and (c) ensuring integrity of citations under the assumption that cited data resources evolve. Our work can be beneficial to both data-oriented and process-oriented approaches since it collects provenance information, determines copying relationships between dependent data sources, and can be used for DAG generation.

In the context of databases [28] and scientific workflows [11, 12, 33], provenance research usually focuses on the transactions of creation and update of data items by examining data lineage in the query results and data products. In the majority of cases, these approaches consider the sources of a data item that are directly related to the creation process without taking into account possible copying relationships that the data providers may have with each other. With the goal to address this limitation in the context of the Semantic Web, Da Silva et al. [10] propose the *Inference Web* project and describe a provenance infrastructure that supports "the extraction, maintenance and usage of knowledge provenance related to answers of web applications and services." The term knowledge provenance refers to information about the origin of knowledge and about the reasoning processes used to produce answers. [19] also propose additional dimensions related to the creation and access of data for characterizing provenance information.

Trust and Authoritativeness of Sources. Provenance and trust are closely related research topics for many years [9]. Various trust models have been developed emphasizing different characteristics of trust. Artz and Gil [1] provide a comprehensive overview of existing trust models. The most common approach to address trustworthiness in the Web is trust infrastructures that are based on a *Web of*

Trust [17]. Approaches such as *PageRank* [5] and *Authorityhub* analysis [21] decide authority based on link analysis [3]. *EigenTrust* [20] and *TrustMe* [27] assign a global trust rating to each data source based on its behavior in a P2P network. While the majority of current approaches consider trustworthiness of data sources, their trustworthiness is not directly related to source accuracy. In addition, they do not consider cases where a data set may have multiple sources, where information providers (re-)publish data aggregated from the original sources, or where inference engines discover implicit facts (or ownership statements) from different sources.

7 Summary

In this chapter we present how to improve truth discovery by analyzing accuracy of sources and detecting copying between sources. We describe Bayesian models that discover copiers by analyzing values shared between sources. The results of our models can be considered as a probabilistic database, where each object is associated with a probability distribution of various values in the underlying domain. We described a case study showing that the presented algorithms can significantly improve accuracy of truth discovery and are scalable when there are a large number of data sources.

There are still many open problems for data integration, and here we list a few:

- *Complex fusion functions*: Often, the fusion decision is not based on the conflicting values themselves, but possibly on other data values of the affected tuples, such as a time stamp. In addition, fusion decisions on different attributes of the same tuples often need to coordinate, for instance, in an effort to keep associations between first and last names and not to mix them from different tuples. Providing a language to express such fusion functions and developing algorithms for their efficient execution are open problems.
- *Incremental fusion*: Fusion functions such as voting or average are subject to incorrect results if new conflicting values appear. Techniques, such as retaining data lineage and maintaining simple metadata or statistics, need to be developed to facilitate incremental fusion.
- *Online fusion*: In some applications, it is infeasible to fuse data from different sources in advance either because it is impossible to obtain all data from some sources or because the total amount of data from various sources is huge. In such cases we need to efficiently perform data fusion in an online fashion at the time of query answering. There has been preliminary work in this direction [22], but the work can be extended by considering more types of queries and quality measures.
- *Data lineage*: Database administrators and data owners are notoriously hesitant to merge data and thus lose the original values, in particular if the merged result is not the same as at least one of the original values. Retaining data lineage despite merging is similar to the problem of data lineage through aggregation operators.

Effective and efficient management of data lineage in the context of fusion is yet to be examined.

- *Combining truth discovery and other integration tasks*: The results of data fusion can often benefit other data-integration tasks, such as schema mapping and record linkage. For example, correcting wrong values in some records can help link these records with records that represent the same entity [18]. To obtain the best results in schema mapping, record linkage, and data fusion, we may need to combine them and perform them iteratively.

References

1. Artz D, Gil Y (2010) A survey of trust in computer science and the semantic web. J Web Semantics 5(2)
2. Blanco L, Crescenzi V, Merialdo P, Papotti P (2010) Probabilistic models to reconcile complex data from inaccurate data sources. In: Proceedings of CAiSE
3. Borodin A, Roberts G, Rosenthal J, Tsaparas P (2005) Link analysis ranking: algorithms, theory, and experiments. ACM TOIT 5:231–297
4. Bose R, Frew J (2005) Lineage retrieval for scientific data processing: a survey. ACM Comput Surv 37(1):1–28
5. Brin S, Page L (1998) The anatomy of a large-scale hypertextual Web search engine. Comput Netw ISDN Syst 30(1–7):107–117
6. Buneman P, Cheney J, Tan WC (2008) Curated databases. In: Proceedings of PODS
7. Buneman P, Khanna S, Tan WC (2000) Data provenance: some basic issues. In: Proceedings of the 20th conference on foundations of software technology and theoretical computer science (FST TCS). Springer
8. Buneman P, Khanna S, Tan WC (2001) Why and where: a characterization of data provenance. In: Proceedings of the 8th international conference on database theory (ICDT). Springer
9. Carroll JJ, Bizer C, Hayes P, Stickler P (2005) Named graphs, provenance and trust. In: Proceedings of WWW
10. da Silva PP, McGuinness DL, McCool R (2003) Knowledge provenance infrastructure. Data Eng Bull 26(4):26–32
11. Davidson SB, Boulakia SC, Eyal A, Ludaescher B, McPhillips TM, Bowers S, Anand MK, Freire J (2007) Provenance in scientific workflow systems. IEEE Data Eng Bull 30(4):44–50
12. Deelman E, Berriman GB, Chervenak A, Corcho O, Groth P, Moreau L (2010) Metadata and provenance management. In: Shoshani A, Rotem D (eds) Scientific data management: challenges, existing technology, and deployment. CRC/Taylor and Francis Books (Chapter 12)
13. Dong XL, Berti-Equille L, Hu Y, Srivastava D (2010) Global detection of complex copying relationships between sources. PVLDB 3(1):1358–1369
14. Dong XL, Berti-Equille L, Srivastava D (2009) Integrating conflicting data: the role of source dependence. PVLDB 2(1):550–561
15. Dong XL, Berti-Equille L, Srivastava D (2009) Truth discovery and copying detection in a dynamic world. PVLDB 2(1):562–573
16. Galland A, Abiteboul S, Marian A, Senellart P (2010) Corroborating information from disagreeing views. In: Proceedings of WSDM
17. Golbeck J, Parsia B, Hendler JA (2003) Trust networks on the semantic web. In: Proceedings of the 7th international workshop on cooperative information agents (CIA)
18. Guo S, Dong XD, Srivastava D, Zajac R (2010) Record linkage with uniqueness constraints and erroneous values. PVLDB 3(1):417–428

19. Hartig O (2009) Provenance information in the web of data. In: Proceedings of the linked data on the web (LDOW'09), workshop of the world wide web conference (WWW), Madrid
20. Kamvar S, Schlosser M, Garcia-Molina H (2003) The Eigentrust algorithm for reputation management in P2P networks. In: Proceedings of WWW
21. Kleinberg JM (1998) Authoritative sources in a hyperlinked environment. In: SODA
22. Liu X, Dong XL, Ooi BC, Srivastava D (2011) Online data fusion. PVLDB 4(11):932–943
23. Moreau L, Clifford B, Freire J, Futrelle J, Gil Y, Groth P, Kwasnikowska N, Miles S, Missier P, Myers J, Plale B, Simmhan Y, Stephan E, Van den Bussche J (2010) The open provenance model core specification (v1.1). Future Generation Computer Systems
24. Pasternack J, Roth D (2010) Knowing what to believe (when you already know something). In: Proceedings of COLING
25. Pasternack J, Roth D (2011) Making better informed trust decisions with generalized fact-finding. In: Proceedings of IJCAI
26. Simmhan Y, Plale B, Gannon D (2005) A survey of data provenance in e-Science. SIGMOD Rec 34(3):31–36
27. Singh A, Liu L (2003) TrustMe: anonymous management of trust relationships in decentralized P2P systems (2003). In: IEEE international conference on peer-to-peer computing
28. Tan WC (2007) Provenance in databases: past, current, and future. IEEE Data Eng Bull 30(4):3–12
29. Wu M, Marian A (2011) A framework for corroborating answers from multiple web sources. Inf Syst 36(2):431–449
30. Yin X, Han J, Yu PS (2007) Truth discovery with multiple conflicting information providers on the Web. In: Proceedings of SIGKDD
31. Yin X, Han J, Yu PS (2008) Truth discovery with multiple conflicting information providers on the web. IEEE Trans Knowl Data Eng 20:796–808
32. Zhao B, Rubinstein BIP, Gemmell J, Han J (2012) A Bayesian approach to discovering truth from conflicting sources for data integration. PVLDB 5(6):550–561
33. Zhao J (2007) A conceptual model for e-Science provenance. PhD thesis, University of Manchester