Global Detection of Complex Copying Relationships Between Sources

Xin Luna Dong
AT&T Labs-Research
lunadong@research.att.com

Laure Berti-Equille*
Université de Rennes 1
berti@irisa.fr

Yifan Hu
AT&T Labs-Research
yifanhu@research.att.com

Divesh Srivastava
AT&T Labs-Research
divesh@research.att.com

ABSTRACT

Web technologies have enabled data sharing between sources but also simplified copying (and often publishing without proper attribution). The copying relationships can be complex: some sources copy from multiple sources on different subsets of data; some co-copy from the same source, and some transitively copy from another. Understanding such copying relationships is desirable both for business purposes and for improving many key components in data integration, such as resolving conflicts across various sources, reconciling distinct references to the same real-world entity, and efficiently answering queries over multiple sources. Recent works have studied how to detect copying between a pair of sources, but the techniques can fall short in the presence of complex copying relationships.

In this paper we describe techniques that discover global copying relationships between a set of structured sources. Towards this goal we make two contributions. First, we propose a global detection algorithm that identifies co-copying and transitive copying, returning only source pairs with direct copying. Second, global detection requires accurate decisions on copying direction; we significantly improve over previous techniques on this by considering various types of evidence for copying and correlation of copying on different data items. Experimental results on real-world data and synthetic data show high effectiveness and efficiency of our techniques.

1. INTRODUCTION

Web technologies have enabled data sources to publish and share their data, but also made it easy for sources to copy from each other (and often publish without proper attribution). The copying relationship can be complex: some sources act as data hubs and aggregate data from multiple sources; some provide only a small set of data independently, copying the rest of the data from their "friend" sources, who may also copy from others; some sources are well known and widely copied by many other sources.

Understanding the copying relationship between sources and the data flow has many benefits [1]. First, data are valuable and many data providers have put a lot of money and effort in collecting and cleaning their data, so they may want to understand such relationships for business purposes (and possibly protect their own rights). Second, in data integration, considering the copying relationship can help improve truth discovery, entity resolution, schema mapping, and further optimize query answering over multiple sources. Third, identifying provenance of data can be critical for in-depth data analysis and for the study of dissemination of information. Finally, independence of sources can form an important criteria in source (user) recommendation. This paper aims at discovering complex copying relationships between a set of sources, illustrated in the following example.

EXAMPLE 1.1. *We consider a data set extracted from* AbeBooks.com*; it includes 1263 CS books and 877 online bookstores (sources)*[1]*. Our copying-detection model (explained in the paper) predicted that between 465 pairs of sources the probability of copying is above .5 (visualized in Appendix Fig. 23, generated by GMap [9]).*

First, we can cluster the bookstores by the copying relationship (as GMap does) and obtain some interesting clusters. For example, cluster Departmentstoria[2] *includes many big department bookstores, such as* A1Books.com, Quartermelon.com, *and* Powell's Books*; cluster* Textbookistan *includes many textbook stores such as* www.textbooksrus.com, LGTextbooks, *and* brandnewtextbooks.

*Second, copying relationships can be complex. Some sources (*e.g.*,* Deepak Sachdeva*) seem to copy from multiple sources. Some sources (*e.g.*,* Browns Books*) seem to be copied by multiple sources (and those co-copiers often do not copy from each other). Some sources seem to transitively copy from other sources; for example,* Gunars Store *or* Gunter Koppon *(one of them is a copier of the other, but the direction is unclear) transitively copies from* WorldOfBooks *via* Books Down Under*. We found that a source can copy from up to 17 sources and be copied by up to 9 sources; and there are transitive paths (where the last source indeed copies data provided by the first) of length up to 9.* □

Discovering copying relationships between structured sources has been studied recently in [6] for static data and in [7] for dynamic data (with updates). In particular, [6] makes pairwise decisions based on common mistakes made by the sources, and [7] considers in addition similarity of update patterns. Such techniques can detect source *dependence* and improve truth discovery, but may generate inaccurate *copying* relationships in the presence of complex copying. In particular, they have the following limitations.

*Mobility research program supported by the European Commission (Grant FP6-MOIF-CT-2006-041000).

VLDB '10, September 13-17, 2010, Singapore

[1]We thank the authors of [13] for providing us the data.
[2]We named the clusters manually.

First and most importantly, these techniques consider every pair of sources in isolation of other sources and make *local* decisions; as a result, they cannot distinguish co-copying, transitive copying, and direct copying from multiple sources. Second, they neglect possible correlations on copying of data items; for example, a source that copies the name of a book tends to also copy its author list. Third, they view common mistakes as important evidence of copying but neglect other kinds of evidence such as whether the data are formatted in the same way, and whether two sources provide similar sets of real-world objects. Experimental results (Sec.6.4) show that the second and the third limitations often lead to wrong copying directions, which in turn can lead to wrong choices among co-copying, transitive copying, and multi-source copying.

This paper proposes techniques for *global* copying detection on static data and these techniques can be extended for dynamic data following the ideas in [7]. Our detection proceeds in two steps: the first step *locally* decides possibility of copying and copying direction between each pair of sources, and the second step *globally* identifies co-copying and transitive copying.

This paper makes three contributions. First, for making more accurate decisions on the copying direction, critical for global detection, we enhance the previous model by gleaning more evidence such as completeness and formatting of data (Sec.3), and considering correlated copying on data items (Sec.4). Second, as a key to global detection, we introduce the techniques for discovering co-copying and transitive copying, and distinguish them from a source indeed copying from multiple sources (Sec.5). Third, we experimented on both real-world data and synthetic data, showing effectiveness and efficiency of our techniques (Sec.6).

We note that although this paper focuses on source-level copying, in Sec.4-5 we also present techniques to decide which data items are copied between a pair of sources. Understanding instance-level copying has two more benefits: first, by applying certain data-mining techniques we can summarize which part of data are copied; second, on each data item we can better understand the data flow (presumably a source typically copies a particular item from a single source and the data flow is in the shape of a forest), so generate provenance information and further improve truth discovery.

2. OVERVIEW

This section defines the problem we solve and describes how we profile characteristics of data.

2.1 Problem definition

Consider a set of real-world objects in the same domain, denoted by $\mathcal{O}$. Each object is described by a set of attributes $\mathcal{A}$, among which we assume one uniquely identifies the objects (key)[3]; we call an attribute $A \in \mathcal{A}$ of an object $O \in \mathcal{O}$ a *data item* and denote it by $O.A$. An attribute value can be atomic (*e.g.*, string, numeric value), or a set or list of atomic values (*e.g.*, a list of authors, a set of phone numbers), which we consider as a whole. We assume for each non-key attribute an object has a *true* value that reflects the reality, and many wrong values[4], but for the key attribute there cannot be any wrong value (we assume entity resolution is already performed using known techniques [10]). We assume as input, we know the probability of each non-key value v being true, denoted by $P(v)$ (we can compute such probabilities according to [6]).

Consider a set of sources, denoted by $\mathcal{S}$, each describing a subset of objects in $\mathcal{O}$. For each object, each source can provide values for a subset of attributes in $\mathcal{A}$ and we assume a key value must be provided. Different sources may format the same value differently (*e.g.*, "John Smith" and "SMITH, John"); such formatting differences should be easily detectable according to some standardization or normalization rules[5]. For each attribute there is a set of *formatting elements* (*e.g.*, for an author list, the elements can be *list completeness, name completeness, name component ordering, capitalization,* etc.), each with a set of *options* (*e.g.*, options for *list completeness* can be *full author list, only first author*, etc.). A *formatting pattern* is a combination of options of the formatting elements; such patterns can be pre-defined by observing the data. Note that some formatting patterns may contain less information than others (*e.g.*, *only first author* vs. *full author list*) and thus they form a partial order.

[3]It is easy to extend our techniques for the case with multi-attribute keys or the case where some attributes apply to only a subset of objects.

[4]Some wrong values are partially correct (*e.g.*, misspellings and partial lists) and we can handle this case by considering value similarity as in [8].

Table 1: Sources in the motivating example.

	ISBN	name	authors
S_1	1	IPV6: Theory, Protocol, and Practice	Loshin, *Peter*
	2	Web Usability: A User-Centered Design Approach	Lazar, Jonathan
S_2	1	*IPV6*	-
	2	*Web Usability*	Jonathan Lazar
S_3	1	IPV6: Theory, Protocol, and Practice	Loshin, *Peter*
	2	*Web Usability*	Jonathan Lazar
S_4	1	IPV6: Theory, Protocol, and Practice	Loshin
	2	*Web Usability*	Lazar

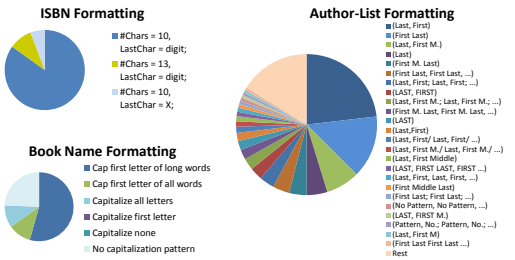


Figure 1: Formatting of attributes in the AbeBooks data set.

EXAMPLE 2.1. *Consider the four sources in Tbl.1, each providing data on the same two books. A book is described by its* ISBN *(the key),* name*, and* authors *(S_2 does not provide* authors *for book 1). The sources may provide incorrect values (marked in italic font);* e.g.*, S_2 provides wrong (or partially correct) values for the* name *of both books. They may also provide the same value but format them differently;* e.g.*, S_1 and S_2 provide the same* authors *for book 2 but in different formats.*

Fig.1 shows the variety of observed formatting patterns on each attribute in the AbeBooks data set. We observe much higher variety on authors *than on* ISBN *and* name. □

Among the sources, some are *independent* and provide data independently, and some are *copiers* and copy all or a portion of data from other sources. A copier may verify some values and modify them when appropriate; we consider such values as independently provided, as they reflect independent observation of the real world by the copier. A copier may also reformat some copied values; we consider such values still as copied if the new format contains equal or less information (*e.g.*, copying only the first author), and as independent if the new format contains more information (*e.g.*, add more authors). Note that there is another kind of "dependence"

[5]A standardization is not necessary; even if it is performed, the formatting information should not be discarded as it helps in copying detection.

between sources–*negative correlation* (*e.g.*, data items provided by S_1 and S_2 are complementary, or S_1 chooses to provide different values or use different formats from S_2); in such cases, we consider neither source as a copier.

EXAMPLE 2.2. *Continue with the motivating example. S_1 and S_2 are independent; S_3 copies the first book from S_1 and the second one from S_2; S_4 copies from S_3 but has reformatted the values of* authors *and provides only their last names.* □

This paper aims to solve the following problem.

PROBLEM STATEMENT 1. *Given a set of objects $\mathcal{O}$ and a set of sources $\mathcal{S}$, for each pair of sources $S, S' \in \mathcal{S}$, decide*

1. *the probability of S* directly *copying from S' and vice versa.*
2. *the probability of S* directly *copying from S' on each object O and on each data item $O.A$, and vice versa.* □

We make a *closed-world* assumption on $\mathcal{O}$ and $\mathcal{S}$; thus, no source in $\mathcal{S}$ copies from a source outside $\mathcal{S}$. This assumption on $\mathcal{O}$ should not affect the results much.

We solve this problem in two steps: (1) *local detection* discovers copying for each pair of sources in isolation of other sources (Sec.3-4); and (2) *global detection* finds co-copying and transitive copying based on local-detection results and computes probabilities of direct copying (Sec.5). For local detection, we first present a basic model assuming (1) *item-wise independence*: whether source S copies an item $O.A$ from S' is independent of whether it copies $O'.A'$ from S' ($O \neq O'$ or $A \neq A'$); and (2) *no mutual copying*: there is no mutual copying (*i.e.*, S_1 copies from S_2 and S_2 copies from S_1) (Sec.3). We then relax these assumptions and present a more comprehensive model (Sec. 4).

2.2 Data and source profiling

A source is more likely to be a copier if the probability that it provides the observed data independently is very low. Judging this would require computing the a-priori probability that a particular source provides some particular data. There are several variations in data for a particular data item, including, but not limited to, whether a value is provided, which value is provided, and in which format the value is provided. We thus profile the data by *completeness, accuracy*, and *formatting style*, respectively, and one can define other measures similarly. These profiling measures typically fall in one of the three classes: *existence measure* measures whether a piece of data exists (*e.g.*, *completeness*); *correctness measure* measures correctness of data (*e.g.*, *accuracy*); and *distribution measure* measures distribution of values, formats, etc. (*e.g.*, *formatting style*).

Note that the probability that a source provides a piece of data can depend both on source-wise statistics and data-item-wise statistics; for example, S is likely to provide an object O if S has a high completeness or O is popular. Thus, we need to define each measure both for each source and for each data item.

Completeness: The *object-level completeness of a source* S, denoted by $C_O(S)$, measures the percentage of objects in $\mathcal{O}$ that S provides. The *completeness of an object* O, denoted by $C(O)$, measures the percentage of sources in $\mathcal{S}$ that provide O. Similarly, we can define attribute-level completeness.

Note that in the presence of copiers, we want to avoid being biased by them when computing completeness and other measures; for example, an object may seem popular, but most of its providers just copy data for it from a common source. We may wish to consider only independent providers; *e.g.*, we can compute $C(O)$ by

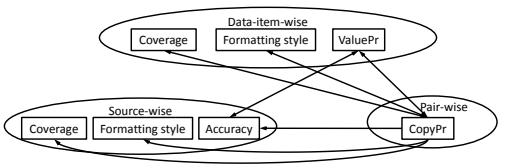


Figure 2: Relationships between measures. An arrow from measure A to B means that A's computation depends on B (*e.g.*, computing copy probability depends on source accuracy).

$$C(O) = \frac{\sum_{S \in \bar{S}(O)} P(S,O)}{\sum_{S \in \bar{S}(O)} P(S,O) + |\mathcal{S} - \bar{S}(O)|}, \quad (1)$$

where $\bar{S}(O)$ is the set of sources that provide O, $P(S,O)$ denotes the probability that S independently provides O (its computation depends on the result of copying detection), and so $\sum_{S \in \bar{S}(O)} P(S,O)$ computes the "number" of independent providers for O and $|\mathcal{S} - \bar{S}(O)|$ counts the number of sources that do not provide O.

Formatting style: The *formatting style of a source* S measures the distribution of formatting patterns used by S. For each formatting pattern f for $A \in \mathcal{A}$, we compute its popularity, $F_{A.f}(S)$, as the percentage of objects for which S uses f for the value of A. The *formatting style of an item* $O.A$ measures the distribution of formatting patterns on $O.A$ used by different sources. We compute $F_{A.f}(O.A)$ as the percentage of sources that use f for the value of $O.A$ among all providers of $O.A$.

Accuracy: The *accuracy of a source* S measures correctness of its data. We adopt techniques presented in [6] and compute accuracy by $A(S) = \text{Avg}_{v \in \bar{V}(S)} P(v)$, where $\bar{V}(S)$ is the set of values provided by S. We can easily refine this measure for each attribute. The corresponding measure for $O.A$ is the correctness of each of $O.A$'s values v and is captured by $P(v)$.

We next illustrate usage of these measures in copying detection.

EXAMPLE 2.3. *Consider S_1, S_2 and S_3 in Table 1. "Peter Loshin" has a misspelling (the correct spelling is "Pete Loshin") and has a low value probability, so indicates dependence between S_1 and S_3. Similarly, "Web Usability" is a wrong book name and has a low value probability, so indicates dependence between S_2 and S_3. It is more likely that S_3 copies from S_1 and S_2 than the opposite direction, as S_3 keeps the format of the copied data and thus formats* authors *differently for the two books, so the popularity of each formatting pattern is low (50%).* □

Fig.2 shows the relationship between these measures. We especially note that (1) as we show later, source copying probabilities depend on all measures we have defined; (2) the copying probabilities affect item-wise measures if we compute them considering only independent sources (*e.g.*, by Eq.(1)), but do not affect source-wise measures; (3) the item-wise measure and the source-wise measure are independent of each other for completeness and formatting styles; however, source accuracy and value probability are inter-dependent (unless value probabilities are given upfront as input). Therefore, there is inter-dependence between data-item-wise measures, source-wise measures, and copying probabilities; we compute them iteratively until convergence, as detailed in [6].

3. A BASIC LOCAL-DETECTION MODEL

We now present the basic model for local copying detection assuming *item-wise independence*. Consider two sources $S_1, S_2 \in \mathcal{S}, S_1 \neq S_2$. The key in deciding whether S_1 copies from S_2

$(S_1 \rightarrow S_2)$ is to decide if the probability of S_1 providing the observed data conditioned on it being independent of S_2 is much lower than that conditioned on it being a copier of S_2. Intuitively, the former probability will be much lower than the latter in two cases: first, when the two sources share low-completeness items, low-probability values, or low-popularity formats; second, when there is a big difference between the profile of the overlapping data and that of S_1's self-provided data.

Specifically, consider two sources $S_1, S_2 \in \mathcal{S}, S_1 \neq S_2$. As we assume *no mutual copying*, there are three possible relationships between them: S_1 copying from S_2 ($S_1 \rightarrow S_2$), S_2 copying from S_1 ($S_2 \rightarrow S_1$), and neither source copying from the other ($S_1 \perp S_2$). We can compute the probability for each case (they sum up to 1) by Bayesian analysis based on our observations of the data, denoted by Φ:

$$P(S_1 \rightarrow S_2|\Phi) = \frac{\alpha P(\Phi|S_1 \rightarrow S_2)}{\alpha P(\Phi|S_1 \rightarrow S_2) + \alpha P(\Phi|S_2 \rightarrow S_1) + (1-2\alpha)P(\Phi|S_1 \perp S_2)} \quad (2)$$

Here, $0 < \alpha < .5$ is the a-priori probability that a source copies from another. Thus, we need to compute the probability of Φ conditioned on different copying relationships.

Observation Φ consists of observations on each data item; *i.e.*, $\Phi = \{\Phi_{O.A} | O \in \mathcal{O}, A \in \mathcal{A}\}$. According to the *item-wise independence* assumption, we have

$$P(\Phi|cond) = \Pi_{O \in \mathcal{O}, A \in \mathcal{A}} P(\Phi_{O.A}|cond). \quad (3)$$

In local detection, we consider only data provided by S_1 and S_2; *i.e.*, $\Phi_{O.A} = \{\Phi_{O.A}(S_1), \Phi_{O.A}(S_2)\}$, where $\Phi_{O.A}(S)$ denotes data provided by S on $O.A$. We say $\Phi_{O.A}(S) = \emptyset$ if S does not provide a value for $O.A$, and $\Phi_O(S) = \emptyset$ if S does not provide a value for $O.key$ (and so not for any other attribute either). Then, we have (similar for the condition $S_2 \rightarrow S_1$)

$$P(\Phi_{O.A}|S_1 \perp S_2) = P(\Phi_{O.A}(S_1)|S_1 \not\rightarrow S_2) P(\Phi_{O.A}(S_2)|S_2 \not\rightarrow S_1); \quad (4)$$

$$P(\Phi_{O.A}|S_1 \rightarrow S_2) = P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2, \Phi_{O.A}(S_2)) P(\Phi_{O.A}(S_2)|S_2 \not\rightarrow S_1). \quad (5)$$

Thus, the key to detecting copying is to compute $P(\Phi_{O.A}(S_1)|S_1 \not\rightarrow S_2)$ and $P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2, \Phi_{O.A}(S_2))$, simplified hereafter as $P(\Phi_{O.A}(S_1))$ and $P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2)$ respectively (similar for S_2). We next describe how we compute them according to our data profiling; our methods can be easily extended when other measures are present.

Not copying: We start with $P(\Phi_{O.A}(S_1))$. Here, S_1 does not rely on S_2 and there are three cases:

1. *S_1 does not provide $O.A$ and A is the key.* Then S_1 does not provide O (*i.e.*, $\Phi_O(S_1) = \emptyset$) and
$$P(\Phi_{O.A}(S_1)) = 1 - P(\Phi_O(S_1) \neq \emptyset). \quad (6)$$
2. *S_1 does not provide $O.A$ and A is not the key.* Then
$$P(\Phi_{O.A}(S_1)|\Phi_O(S_1) \neq \emptyset) = 1 - P(\Phi_{O.A}(S_1) \neq \emptyset); \quad (7)$$
$$P(\Phi_{O.A}(S_1)|\Phi_O(S_1) = \emptyset) = 1. \quad (8)$$
3. *Otherwise*, suppose S_1 provides a value v and formats it in pattern f. Then,
$$P(\Phi_{O.A}(S_1)) = P(\Phi_{O.A}(S_1) \neq \emptyset) \cdot P(value(\Phi_{O.A}(S_1)) = v) \cdot P(format(\Phi_{O.A}(S_1)) = f). \quad (9)$$

We next consider what is the probability that a source independently provides O, provides $O.A$, provides a particular value for $O.A$, and uses a particular format for $O.A$.

We start with $P(\Phi_O(S_1) \neq \emptyset)$. Intuitively, a source S provides $|\mathcal{O}|C_O(S)$ objects, so the probabilities for providing each object should sum up to $|\mathcal{O}|C_O(S)$; similarly, the probabilities of each source providing O should sum up to $|\mathcal{S}|C(O)$. We thus shall solve the following equations:

$$\forall S \in \mathcal{S}, \sum_{O \in \mathcal{O}} P(\Phi_O(S) \neq \emptyset) = |\mathcal{O}|C_O(S); \quad (10)$$

$$\forall O \in \mathcal{O}, \sum_{S \in \mathcal{S}} P(\Phi_O(S) \neq \emptyset) = |\mathcal{S}|C(O). \quad (11)$$

There are $|\mathcal{S}| \cdot |\mathcal{O}|$ variables but only $|\mathcal{S}| + |\mathcal{O}|$ equations, so an infinite number of solutions. We choose the one with the maximum entropy [4], so has the least bias. We can prove that in most cases[6] such a solution is obtained when we assume the probability that S provides each object O is proportional to $C(O)$; thus,

$$P(\Phi_O(S_1) \neq \emptyset) = \frac{|\mathcal{O}|C_O(S_1)C(O)}{\sum_{O_0 \in \mathcal{O}} C(O_0)} = \frac{|\mathcal{S}|C(O)C_O(S_1)}{\sum_{S_0 \in \mathcal{S}} C_O(S_0)}. \quad (12)$$

Similarly we can compute $P(\Phi_{O.A}(S_1) \neq \emptyset)$ and $P(format(\Phi_{O.A}(S_1)) = f)$.

Now consider the probability of providing a particular value v. If A is the key, the probability is 1. Otherwise, assume there are m wrong values in the underlying domain. Then, S_1 provides a true value with probability $A(S_1)$ and a particular wrong value with probability $\frac{1-A(S_1)}{m}$ (we assume equal probability of providing a wrong value and relaxation of this assumption is discussed in [6]). Recall that $P(v)$ denotes the probability of value v being true, so

$$P(value(\Phi_{O.A}(S_1)) = v) = P(v)A(S_1) + (1 - P(v))\frac{1 - A(S_1)}{m}. \quad (13)$$

Copying: We next compute $P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2)$. Note that even if S_1 copies from S_2, S_1 may or may not copy a particular data item. We call the probability of copying a particular item the *selectivity*, and denote it by s. Even when S_1 copies, it can then choose to keep the original format, or to reformat it; we assume the probability of the former is k. We discuss how to set s and k in Sec.4.1.1. The computation requires comparing values and formats provided by S_1 and S_2; there are four possible cases.

1. *One of S_1 and S_2 does not provide $O.A$.* We do not penalize providing a value that the other source does not provide or vice versa (common for a copier), so
$$P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2) = P^c(\Phi_{O.A}(S_1)). \quad (14)$$
For the probability that a copier independently provides a piece of data, we mark by ^{c} and explain in more detail shortly.
2. *S_1 and S_2 provide different values on $O.A$ or S_1 uses a format with richer information.* Then, S_1 does not copy:
$$P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2) = (1-s)P^c(\Phi_{O.A}(S_1)). \quad (15)$$
3. *S_1 provides the same value as S_2 but in a different format f' (f' contains no richer information than that of S_2).* Then, S_1 might copy (w. probability s) but reformat:
$$P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2) = (1-s)P^c(\Phi_{O.A}(S_1)) + s(1-k)P^c(format(\Phi_{O.A}(S_1)) = f'). \quad (16)$$

[6]The only exception happens when $\frac{|\mathcal{O}|C_O(S_1)C(O)}{\sum_{O_0 \in \mathcal{O}} C(O_0)} > 1$ for some S_1 and O; in this case, we can estimate by setting $P(\Phi_O(S_1) \neq \emptyset) = 1$ for such S_1 and O, and compute by Eq.(12) for other sources and objects .

4. *S_1 provides the same value in the same format f.* Then, S_1 might copy (w. probability s) and might follow the original format (w. probability k):

$$P(\Phi_{O.A}(S_1)|S_1 \to S_2) = (1-s)P^c(\Phi_{O.A}(S_1)) + s(k + (1-k)P^c(format(\Phi_{O.A}(S_1)) = f)). \quad (17)$$

We note that we use $P^c(\Phi_{O.A}(S_1))$ instead of $P(\Phi_{O.A}(S_1))$ under condition of copying. We compute $P^c(\Phi_{O.A}(S_1))$ in the same way as $P(\Phi_{O.A}(S_1))$, except that we use "independent" measures computed only on S_1's data that are not copied. As we often do not know in advance which data are copied, and such computation needs to be performed for every pair of sources so needs to be very fast, we estimate these measures. As an example, we estimate S_1's "independent" object-level completeness w.r.t. S_2 as its completeness on objects not provided by S_2:

$$C_O(S_1|\neg S_2) = \frac{|\bar{O}(S_1)| - |\bar{O}(S_1) \cap \bar{O}(S_2)|}{|\mathcal{O}| - |\bar{O}(S_2)|}, \quad (18)$$

where $\bar{O}(S)$ denotes the set of objects provided by S.

Discussion: The following theorem lists positive evidence for copying, conforming to our intuitions.

THEOREM 3.1. *Given sources S_1 and S_2 and data item $O.A$, in the following cases $O.A$ forms positive evidence for $S_1 \to S_2$.*

1. *S_1 provides the same value in the same format as S_2 on $O.A$, and $P(\Phi_{O.A}(S_1)) < sk$;*
2. *S_1 provides the same value but uses a different format f', and $P(\Phi_{O.A}(S_1)) < s(1-k)P^c(format(\Phi_{O.A}(S_1)) = f')$;*
3. *$P^c(\Phi_{O.A}(S_1)) > P(\Phi_{O.A}(S_1))$ and S_2 does not provide $O.A$;*
4. *$(1-s)P^c(\Phi_{O.A}(S_1)) > P(\Phi_{O.A}(S_1))$.*

PROOF. We prove the four conditions.

1. $P(\Phi_{O.A}(S_1)) < sk < (1-s)P^c(\Phi_{O.A}(S_1)) + s(k + (1-k)P^c(format(\Phi_{O.A}(S_1)) = f))$; thus, when the two sources provide the same value in the same format, $P(\Phi_{O.A}(S_1)) < P(\Phi_{O.A}(S_1)|S_1 \to S_2)$ and $O.A$ forms positive evidence.
2. $P(\Phi_{O.A}(S_1)) < s(1-k)P^c(format(\Phi_{O.A}(S_1)) = f') < (1-s)P^c(\Phi_{O.A}(S_1)) + s(1-k)P^c(format(\Phi_{O.A}(S_1)) = f')$; thus, when the two sources provide the same value in different formats, $P(\Phi_{O.A}(S_1)) < P(\Phi_{O.A}(S_1)|S_1 \to S_2)$ and $O.A$ forms positive evidence.
3. If $P^c(\Phi_{O.A}(S_1)) > P(\Phi_{O.A}(S_1))$ and S_2 does not provide $O.A$, $P(\Phi_{O.A}(S_1)) < P(\Phi_{O.A}(S_1)|S_1 \to S_2)$ and so $O.A$ forms positive evidence.
4. If $(1-s)P^c(\Phi_{O.A}(S_1)) > P(\Phi_{O.A}(S_1))$, $P(\Phi_{O.A}(S_1)) < P(\Phi_{O.A}(S_1)|S_1 \to S_2)$ in various cases and so $O.A$ forms positive evidence.

□

In addition, recall that our goal is to find copiers rather than negative correlation, we shall omit evidence for negative correlation. There are two types of such evidence. First, S_1 and S_2 providing the same value in the same format but $P(\Phi_{O.A}(S_1)|S_1 \to S_2) < P(\Phi_{O.A}(S_1))$ shows that a "dependent" source is less likely to provide the same data and indeed implies negative correlation; we set $P(\Phi_{O.A}(S_1)|S_1 \to S_2) = P(\Phi_{O.A}(S_1))$ in this case. Second, $C_O(S_1|\neg S_2) > C_O(S_1)$ (similar for $O.A$) shows that a "dependent" source is more likely to provide an object not provided by the original source and indeed implies negative correlation; we set $C_O(S_1|\neg S_2) = C_O(S_1)$ in this case.

Table 2: Ex. 3.2 and 4.1. Each table describes data provided by two sources on 5 objects, each with 5 attributes (K is the key). "S" in the table indicates that the two sources provide the same value in the same format, and "D" indicates that they provide different values. Copying seems more likely for (b) than for (a).

	K	A_1	A_2	A_3	A_4
O_1	S	S	S	D	D
O_2	S	D	S	S	D
O_3	S	S	D	S	D
O_4	S	S	S	D	S
O_5	S	D	S	S	S

(a)

	K	A_1	A_2	A_3	A_4
O_1	S	S	S	S	S
O_2	S	S	S	S	S
O_3	S	S	S	S	S
O_4	S	D	D	D	D
O_5	S	D	D	D	D

(b)

EXAMPLE 3.2. *Consider sources S_1 and S_2, each providing 5 objects and 5 attributes for each object (shown in Tbl. 2(a)). Assume $P(\Phi_{O.K}(S_1)) = P^c(\Phi_{O.K}(S_1)) = .9$, $P(\Phi_{O.A_i}(S_1)) = P^c(\Phi_{O.A_i}(S_1)) = .5, i \in [1,4]$, and $P^c(format(\Phi_{O.A}(S_1))) = .8$ for each attribute A. We set $s = .6, k = .5$.*

We first compute $P(\Phi(S_1)|S_1 \to S_2)$. There are 5 key items on which the two sources provide the same value in the same format; the probability is $(1-.6).9 + .6*(.5 + (1-.5)*.8) = .4*.9 + .6*.9 = .9$ (Eq.(17)). There are 12 non-key items on which the two sources provide the same value in the same format; the probability is $.4*.5 + .6*.9 = .74$ (Eq.(17)). Finally, there are 8 items on which the two sources provide different values and the probability is $.4*.5 = .2$ (Eq.(15)). So $P(\Phi(S_1)|S_1 \to S_2) = .9^5 * .74^{12} * .2^8$ (Eq.(3)).*

On the other hand, it is obvious that $P(\Phi(S_1)|S_1 \not\to S_2) = .9^5 * .5^{20}$. So $\frac{P(\Phi(S_1)|S_1 \to S_2)}{P(\Phi(S_1)|S_1 \not\to S_2)} = .07$ and S_1 is unlikely to be a copier of S_2. This is reasonable because S_1 provides a lot of values differently from S_2, and for the values they share, S_1 has a relatively high probability to provide them by itself. □

Comparison with [6]: There are three differences between our basic model and the model presented in [6].

1. The basic model allows the flexibility of plugging in evidence of various types, including completeness and formatting of data in addition to correctness of data.
2. In addition to source-wise measures, we consider also item-wise measures when computing $P(\Phi_{O.A}(S))$.
3. Instead of using $P(\Phi_{O.A}(S_1))$, we use $P^c(\Phi_{O.A}(S_1))$ under condition $S_1 \to S_2$.

Note that difference 3 is a correction of the previous model; however, our experiments show that it does not necessarily improve the results when we consider only data correctness. Finally, none of the techniques in Sec.4-5 is included in [6].

4. ENHANCED LOCAL DETECTION

This section enhances the basic model by considering correlated copying (Sec.4.1) and mutual copying (Sec.4.2).

4.1 Correlated copying

The basic model assumes *item-wise independence*, which seldom holds in reality. For example, the copier may compose a SQL query and copy all returned objects; when it copies an object, it often copies all provided attributes or the attributes in its own schema. This section discusses object copying (the latter example); similar techniques can be applied for query-driven copying (the early example).

One can imagine that a copier often copies in one of two modes: 1) it copies a subset of objects on a subset of attributes, called *per-object copying*; 2) it copies on a subset of attributes for a set

of independently provided objects (or objects copied from other sources), called *per-attribute* copying. The difference is whether the copier also copies the key values or not. Thus, when we compute $P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2)$, we need to consider whether S_1 copies on O or only on A and treat them differently.

Specifically, we denote by $S_1 \overset{O}{\rightarrow} S_2$ that S_1 copies O from S_2, by $S_1 \overset{O.A}{\rightarrow} S_2$ that S_1 copies $O.A$ from S_2, and simplify $P(S_1 \overset{O}{\rightarrow} S_2|S_1 \rightarrow S_2)$ as $s(O)$ ($s(O)$ can be viewed as the selectivity on O for $S_1 \rightarrow S_2$, but we omit $S_1 \rightarrow S_2$ for simplicity). Then, we have

$$P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2) = s(O)P(\Phi_{O.A}(S_1)|S_1 \overset{O}{\rightarrow} S_2) + (1 - s(O))P(\Phi_{O.A}(S_1)|S_1 \overset{O}{\not\rightarrow} S_2, S_1 \rightarrow S_2)). \quad (19)$$

The computation of $P(\Phi_{O.A}(S_1)|S_1 \overset{O}{\rightarrow} S_2)$ and $P(\Phi_{O.A}(S_1)|S_1 \overset{O}{\not\rightarrow} S_2, S_1 \rightarrow S_2)$ is the same as $P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2)$, except that we shall set the selectivity differently. We denote the selectivity for an attribute $A \in \mathcal{A}$ of a copied object by $s_A^{\rightarrow}$ and that of an uncopied object (per-attribute copying) by $s_A^{\not\rightarrow}$. According to our definition, $s_{key}^{\rightarrow} = 1$ and $s_{key}^{\not\rightarrow} = 0$.

The next question is how to compute $s(O)$ depending on our observation of data provided on O. Let s_{obj} be the a-priori probability that a copier copies an object. Then, by Bayesian analysis,

$$s(O) = P(S_1 \overset{O}{\rightarrow} S_2|\Phi_O(S_1), S_1 \rightarrow S_2) = \frac{s_{obj}P(\Phi_O(S_1)|S_1 \overset{O}{\rightarrow} S_2)}{s_{obj}P(\Phi_O(S_1)|S_1 \overset{O}{\rightarrow} S_2) + (1 - s_{obj})P(\Phi_O|S_1 \overset{O}{\not\rightarrow} S_2, S_1 \rightarrow S_2)} \quad (20)$$

We can compute $P(\Phi_O(S_1)|S_1 \overset{O}{\rightarrow} S_2)$ from $P(\Phi_{O.A}(S_1)|S_1 \overset{O}{\rightarrow} S_2)$, $A \in \mathcal{A}$, assuming all attributes are independent (we can relax this assumption by further grouping the attributes).

EXAMPLE 4.1. *Continue with Ex.3.2 and now consider S_1 and S_2 in Tbl. 2(b). With the same calculations as in Ex.3.2, S_1 appears unlikely to be a copier of S_2. Now consider per-object copying and we set $s_{obj} = .6$ and $s_A^{\rightarrow} = .9, s_A^{\not\rightarrow} = .1$ for each attribute.*

For each non-key item in $\{O_1, O_2, O_3\}$, if S_1 copies the object, the probability that it provides the data is $.1 * .5 + .9 * .9 = .86$ (Eq.(17)); otherwise, if S_1 copies from S_2 but not on the object, the probability becomes $.9 * .5 + .1 * .9 = .54$. For each non-key item in $\{O_4, O_5\}$, if S_1 copies the object, the probability is $.1 * .5 = .05$ (Eq.(15)); otherwise, the probability is $.9 * .5 = .45$. Finally, for each key attribute, the probability that S_1 provides it is always .9.

Accordingly, the probability that S_1 copies O_1, O_2 or O_3 is $\frac{.6*(.9*.86^4)}{.6*(.9*.86^4)+.4*(.9*.54^4)} = .9$ (Eq.(20)). The probability that it provides each non-key value is thus $.9 * .86 + .1 * .54 = .83$ (Eq.(19)). Similarly, the probability that S_1 copies O_4 or O_5 is .002 and the probability that it provides each non-key value is .45. Therefore, $\frac{P(\Phi(S_1)|S_1 \rightarrow S_2)}{P(\Phi(S_1)|S_1 \not\rightarrow S_2)} = \frac{.9^5 * .83^{12} * .45^8}{.9^5 * .5^{20}} = 188$ and S_1 is likely to be a copier of S_2.

To compare, for Tbl.2(a), considering per-object copying obtains a ratio of 1.37 and still does not strongly imply copying. □

4.1.1 Parameter Setting

One big challenge for applying the enhanced model is parameter setting. The enhanced model involves parameters, $s_{obj}, s_A^{\rightarrow}, s_A^{\not\rightarrow}$ and k; they are essentially conditional probabilities and can vary from source pair to source pair. Setting them appropriately is important in achieving accurate results. We set them for each direction of each pair of sources in two steps: first, we initialize them empirically according to the data; second, we adjust them later according to copying-detection results and re-apply our model accordingly.

Initialization: Consider the copying relationship $S_1 \rightarrow S_2$. We start with s_{obj}. We first examine overlapping objects; for each attribute, we compute the ratio of common values. Accordingly we generate the histogram for each range of ratio, find the peak range, and use its middle value as the selectivity for overlapping objects, $s_{overlap-obj}$. Then, the overall selectivity is

$$s_{obj} = \frac{s_{overlap-obj} \cdot |\bar{O}(S_1) \cap \bar{O}(S_2)|}{|\bar{O}(S_2)|}. \quad (21)$$

For each $A \in \mathcal{A}$, we denote by $\overline{O.A}_v$ the items on which the two sources provide the same value. Then, we compute $s_A^{\rightarrow}$ and $s_A^{\not\rightarrow}$ as

$$s_A^{\rightarrow} = \frac{|\overline{O.A}_v|}{s_{overlap-obj} \cdot |\bar{O}(S_1) \cap \bar{O}(S_2)|}; \quad (22)$$

$$s_A^{\not\rightarrow} = \frac{|\overline{O.A}_v| - s_A^{\rightarrow} \cdot s_{overlap-obj} \cdot |\bar{O}(S_1) \cap \bar{O}(S_2)|}{(1 - s_{overlap-obj})|\bar{O}(S_1) \cap \bar{O}(S_2)|}. \quad (23)$$

Now consider k and we set it for each attribute. Let $\overline{O.A}_f$ be the items of A on which the two sources provide the same value in the same format. For the copied objects, the format keeping rate is $\frac{|\overline{O.A}_f|}{|\overline{O.A}_v|}$; for the rest of the objects, we use a default rate k_0. So

$$s_A = s_{overlap-obj} \cdot s_A^{\rightarrow} + (1 - s_{overlap-obj})s_A^{\not\rightarrow}; \quad (24)$$

$$k_A = s_A \cdot \frac{|\overline{O.A}_f|}{|\overline{O.A}_v|} + (1 - s_A) \cdot k_0. \quad (25)$$

Finally, note that we want to avoid extreme values for the parameters and so set them only in a certain range. In our experiments we use range $[.1, .9]$ and truncate values outside this range.

Adjustment: According to our copying detection results, we can adjust the parameters and re-do the detection. In particular, if S_1 copies from S_2 (with probability $P(S_1 \rightarrow S_2)$), we shall use the percentage of copied objects or data items (or preserved formatting) observed from the data; otherwise, we shall use the initial settings. Specifically, we adjust the parameters according to the following equations (similar for $s'_{overlap-obj}$ and $s_A^{\not\rightarrow\prime}$):

$$s'_{obj} = \frac{\sum_{O \in \bar{O}(S_2)} s(O)}{|\bar{O}(S_2)|} P(S_1 \rightarrow S_2) + s_{obj}(1 - P(S_1 \rightarrow S_2)) \quad (26)$$

$$s_A^{\rightarrow\prime} = \frac{\sum_{O \in \bar{O}(S_1) \cap \bar{O}(S_2)} s(O)P(S_1 \overset{O.A}{\rightarrow} S_2|\Phi_{O.A}, S_1 \overset{O}{\rightarrow} S_2)}{\sum_{O \in \bar{O}(S_1) \cap \bar{O}(S_2)} s(O)} \cdot P(S_1 \rightarrow S_2) + s_A^{\rightarrow}(1 - P(S_1 \rightarrow S_2)); \quad (27)$$

$$k'_A = \frac{\sum_{O.A \in \overline{O.A}_f} P(S_1 \overset{O.A}{\rightarrow} S_2|\Phi_{O.A}, S_1 \rightarrow S_2)}{\sum_{O \in \bar{O}(S_1) \cap \bar{O}(S_2)} P(S_1 \overset{O.A}{\rightarrow} S_2|\Phi_{O.A}, S_1 \rightarrow S_2)} \cdot P(S_1 \rightarrow S_2) + k(1 - P(S_1 \rightarrow S_2)). \quad (28)$$

Here, $P(S_1 \overset{O.A}{\rightarrow} S_2|\Phi_{O.A}, S_1 \rightarrow S_2)$ denotes the probability that a copier copies a data item $O.A$. We compute it in a similar way as we compute $s(O)$:

$$P(S_1 \overset{O.A}{\rightarrow} S_2|\Phi_{O.A}, S_1 \rightarrow S_2) = s(O)P(S_1 \overset{O.A}{\rightarrow} S_2|\Phi_{O.A}, S_1 \overset{O}{\rightarrow} S_2) + (1 - s(O))P(S_1 \overset{O.A}{\rightarrow} S_2|\Phi_{O.A}, S_1 \overset{O}{\not\rightarrow} S_2, S_1 \rightarrow S_2); \quad (29)$$

$$P(S_1 \overset{O.A}{\rightarrow} S_2|\Phi_{O.A}, S_1 \overset{O}{\rightarrow} S_2) = \frac{s_A^{\rightarrow}P(\Phi_{O.A}(S_1)|S_1 \overset{O.A}{\rightarrow} S_2)}{P(\Phi_{O.A}(S_1)|S_1 \overset{O}{\rightarrow} S_2)} = \frac{P(\Phi_{O.A}(S_1)|S_1 \overset{O}{\rightarrow} S_2) - (1 - s_A^{\rightarrow})P^c(\Phi_{O.A}(S_1))}{P(\Phi_{O.A}(S_1)|S_1 \overset{O}{\rightarrow} S_2)}. \quad (30)$$

We compute $P(S_1 \overset{O.A}{\rightarrow} S_2|\Phi_{O.A}, S_1 \overset{O}{\not\rightarrow} S_2, S_1 \rightarrow S_2)$ in a similar way to Eq.(30) and skip the equation.

Algorithm 1: MutualDetection($\mathcal{S}, \mathcal{O}$)

Input : $\mathcal{S}$ sources, $\mathcal{O}$ objects
Output : $\mathcal{D}$ Copying probabilities between each pair of sources in $\mathcal{S}$
// $\mathcal{D}^0(S_1, S_2) = \{P(S_1 \to S_2), P(S_2 \to S_1)\}$
// Pairwise copying detection
$\mathcal{D}^0 \leftarrow$ PairwiseDetection($\mathcal{S}, \mathcal{O}$);
// Detect mutual copying
$\overline{OA}' = \emptyset$;
foreach $S_1, S_2 \in \mathcal{S}, S_1 \neq S_2$ **do**
  while $P(S_1 \to S_2) + P(S_2 \to S_1) > .5$ **do**
    Compute $\overline{O.A}(S_1 \to S_2)$ and $\overline{O.A}(S_2 \to S_1)$ by Eq.(29);
    $\overline{OA} \leftarrow \overline{O.A}(S_1 \to S_2) \cap \overline{O.A}(S_2 \to S_1)$;
    if $\frac{|\overline{OA}|}{\overline{O.A}(S_1 \to S_2)} \geq \delta$ && $\frac{|\overline{OA}|}{\overline{O.A}(S_2 \to S_1)} \geq \delta$ **then**
      $\mathcal{D}(S_1, S_2) = \mathcal{D}^0(S_1, S_2)$; **break**;
    else if $\frac{|\overline{OA}' \cap \overline{OA}|}{\overline{OA}'} > \eta$ *(η is set to be close to 1)* **then**
      recompute $P(S_1 \to S_2)$ and $P(S_2 \to S_1)$ by Eq.(31)
      and update $\mathcal{D}(S_1, S_2)$; **break**;
    else
      update "independent" measures of S_1 by considering
      also $\overline{O.A}(S_2 \to S_1) - \overline{OA}$ and similar for S_2;
      recompute $P(S_1 \to S_2)$ and $P(S_2 \to S_1)$ by Eq.(2);
      $\overline{OA}' \leftarrow \overline{OA}$;

Discussions: Experimental results show that (1) using $s_{overlap-obj}$ for overlapping objects and s_{obj} for the rest of the objects obtains better results than using s_{obj} everywhere; (2) setting $s_A^{\rightarrow}, s_A^{\not\rightarrow}$ and k differently for different attributes obtains better results; and (3) setting the parameters empirically can significantly improve over setting arbitrary values, but adjusting the parameters does not show much of further benefit.

4.2 Mutual copying

In practice, S_1 and S_2 can be specialists in different areas and mutually copy (on different items). In such cases, our detection considers all common items and is more likely to be able to detect copying, so we just need to re-examine if mutual copying exists.

It is often hard to distinguish mutual copying from non-mutual copying. Our key intuition is that if the copied data look neither fully like S_1's own data (have similar profiles) nor fully like S_2's, but we can partition them such that one partition looks like S_1's and the other like S_2's, mutual copying is more likely. However, partitioning is not easy; we approximate by finding objects copied by S_1 (conditioned on S_1 being a copier) and those copied by S_2, then decide if the overlap is small.

In particular, we detect mutual copying in four steps: (see Algorithm MUTUALDETECTION).

1. Detect copying between S_1 and S_2 and apply Eq.(29) to find data items S_1 copies from S_2, denoted by $\overline{O.A}(S_1 \to S_2)$, and those S_2 copies from S_1, denoted by $\overline{O.A}(S_2 \to S_1)$.

2. If the copy probabilities in both directions are low, or one of $\overline{O.A}(S_1 \to S_2)$ and $\overline{O.A}(S_2 \to S_1)$ is subsumed by the other (*i.e.*, $\frac{|\overline{O.A}(S_1 \to S_2) \cap \overline{O.A}(S_2 \to S_1)|}{\min\{|\overline{O.A}(S_1 \to S_2)|, |\overline{O.A}(S_2 \to S_1)|\}} \geq \delta$, where $0 < \delta < 1$ is a threshold for highly overlapping), return the current results (no mutual copying).

3. If $|\overline{O.A}(S_1 \to S_2) \cap \overline{O.A}(S_2 \to S_1)|$ does not change much from the last round, modify Eq.(2) as

$$P'(S_1 \to S_2) = \frac{\alpha P(\Phi | S_1 \to S_2)}{\alpha P(\Phi | S_1 \to S_2) + (1-\alpha) P(\Phi | S_1 \not\to S_2)} = \frac{2P(S_1 \to S_2)}{1 + P(S_1 \to S_2) - P(S_2 \to S_1)}. \quad (31)$$

and similarly for $S_2 \to S_1$ (the sum can be above 1).

4. Update S_1's "independent" measures (*e.g.*, $C_O(S_1|S_2)$) by considering also data in $\overline{O.A}(S_2 \to S_1) - \overline{O.A}(S_1 \to S_2)$ (similar for S_2) and go to step 1.

We observed that the algorithm always converges in a couple of rounds. Its complexity is $O(n|\mathcal{O}||\mathcal{A}||\mathcal{S}|^2)$, where n is the maximum number of rounds for convergence.

Table 3: Three sets of data sources. In each one, S_1 and S_2 share 50 values, S_1 and S_3 share 50 values, and S_2 and S_3 share 30 values.

Src	$\mathcal{D}_1$ (Multi-copy)	$\mathcal{D}_2$ (Co-copy)	$\mathcal{D}_3$ (Transitive)
S_1	$v_1 \sim v_{100}$, where $v_{81} \sim v_{100}$ are popular values		
S_2	$v_1 \sim v_{50}, v_{101} \sim v_{130}$	$v_1 \sim v_{50}$	$v_1 \sim v_{50}$
S_3	$v_{51} \sim v_{130}$	$v_{21} \sim v_{70}$	$v_{21} \sim v_{50}, v_{81} \sim v_{100}$

5. GLOBAL COPYING DETECTION

Local detection aims at discovering (positive) dependence between sources; however, such dependence is not always caused by direct copying, but can also be due to co-copying or transitive copying. In the motivating example (Tbl.1), local detection may conclude with $S_4 \to S_1$ and $S_4 \to S_2$, although S_4 only transitively copies from S_1 and S_2. Global detection tries to fix this problem. However, identifying co-copying and transitive copying is non-trivial, as the following example illustrates.

EXAMPLE 5.1. *Consider S_1, S_2 and S_3, where local detection decides that $S_2 \to S_1, S_3 \to S_1$, and $S_3 \to S_2$. We need to decide if S_3 copies only from S_1 (co-copying with S_2), only from S_2 (transitively copying from S_1), or from both (multi-source copying).*

One may consider finding a clue from the copying probabilities, but this often does not work when each pair of sources share a lot of values and thus have a copying probability of 1.

One may then consider comparing the numbers of overlapping values, but this is insufficient. Consider $\mathcal{D}_1$ and $\mathcal{D}_2$ in Tbl.3. Each pair of sources share the same number of values for the two cases; however, because of the different distribution of the shared values, $\mathcal{D}_1$ seems more likely to have multi-source copying, whereas $\mathcal{D}_2$ seems more likely to have co-copying.

One may next consider comparing the sets of overlapping values, but this is still insufficient. Consider $\mathcal{D}_2$ and $\mathcal{D}_3$ in Tbl.3. Values that all of S_1, S_2 and S_3 share are the same ($v_{21} \sim v_{50}$) in the two cases; however, $\mathcal{D}_2$ seems more likely to have a co-copying, whereas $\mathcal{D}_3$ seems more likely to have a transitive copying, because the rest of the 20 values shared between S_1 and S_3 in $\mathcal{D}_3$ are popular values and S_3 may provide them independently. Thus, we need to reason for each data item in a more principled way. □

Our key intuition is that since co-copying and transitive copying can often be inferred from direct copying, we first find a set of copying relationships $\mathbf{R}$ that significantly influence the rest of the relationships, and then adjust the rest accordingly and decide if each is a direct or indirect copying (the results are denoted by $P(S_1 \to S_2|\mathbf{R}), (S_1, S_2) \notin \mathbf{R}$). In this process we need to solve two problems: (1) how to select the set $\mathbf{R}$; and (2) how to compute $P(S_1 \to S_2|\mathbf{R})$. The first problem relies on the second, so we start from our solution for the second problem.

5.1 Computing $P(S_1 \to S_2|\mathbf{R})$

As we have illustrated, we cannot derive this probability directly from the copying probabilities in $\mathbf{R}$, but have to reason for each data item if S_1 is likely to copy it from S_2 even in the presence of the copying between S_1 and other sources. Thus, when we compute $P(S_1 \to S_2|\mathbf{R})$, we replace $P(\Phi_{O.A}(S_1))$ everywhere with $P(\Phi_{O.A}(S_1)|\mathbf{R})$, the probability that S_1 independently provides the data or copies the data from other sources according to $\mathbf{R}$. We next illustrate how we compute $P(\Phi_{O.A}(S_1)|\mathbf{R})$ using the case of $\Phi_{O.A}(S_1) \neq \emptyset$.

Consider the set of sources that are associated with S_1 by some copying relationship in $\mathbf{R}$. Consider two subsets: $\bar{S}_f(O.A)$, those providing the same value in the same format on $O.A$ as S_1; $\bar{S}_v(O.A)$, those providing the same value in a different format. The probability that S_1 does not copy $O.A$ from any source in $\bar{S}_f(O.A)$ is

$$P_f = \sum_{S \in \bar{S}_f(O.A)} (1 - P(S_1 \to S)P(S_1 \overset{O.A}{\longrightarrow} S) \cdot (k + (1-k)P(format(\Phi_{O.A}(S_1)) = f)), \quad (32)$$

where $P(S_1 \to S)$ and $P(S_1 \overset{O.A}{\longrightarrow} S)$ are inferred from $\mathbf{R}$. Similarly, we can compute the probability that S_1 does not copy $O.A$ from any source in $\bar{S}_v(O.A)$ and provide the observed format, denoted by P_v. Then S_1 either provides the data by itself (with probability $P_v P_f$), or copies $O.A$ from $\bar{S}_v(O.A)$ or $\bar{S}_f(O.A)$, so

$$P(\Phi_{O.A}(S_1)|\mathbf{R}) = (1 - P_v P_f) + P_v P_f P(\Phi_{O.A}(S_1)). \quad (33)$$

EXAMPLE 5.2. *Continue with Ex.5.1. Consider $\mathcal{D}_1$ and suppose $\mathbf{R} = \{(S_1, S_3)\}$. For each $v \in \{v_{101}, \ldots, v_{130}\}$, $P(\Phi_v(S_3)) = P(\Phi_v(S_3|\mathbf{R}))$, so S_3 still looks like a copier of S_2.*

Consider $\mathcal{D}_2$ and suppose $\mathbf{R} = \{(S_1, S_3)\}$. For each $v \in \{v_{21}, \ldots, v_{50}\}$, $P(\Phi_v(S_3|\mathbf{R}))$ is much larger than $P(\Phi_v(S_3))$, so S_3 looks much less likely a copier of S_2.

Finally, consider $\mathcal{D}_3$ and suppose $\mathbf{R} = \{(S_2, S_3)\}$. For each $v \in \{v_{21}, \ldots, v_{50}\}$, again $P(\Phi_v(S_3|\mathbf{R}))$ is much larger than $P(\Phi_v(S_3))$; for each $v \in \{v_{81}, \ldots, v_{100}\}$, $P(\Phi_v(S_3|\mathbf{R})) = P(\Phi_v(S_3))$ but is high. Thus, S_3 looks less likely a copier of S_1. □

5.2 Finding R

Finding a proper set of relationships for $\mathbf{R}$ is crucial. As an example, for $\mathcal{D}_2$ in Tbl.3, if we include $S_3 \to S_2$ in $\mathbf{R}$, we will not be able to detect the real copying $S_3 \to S_1$. We wish to include in $\mathbf{R}$ the most influential copying relationships; that is, our goal is *to find the set $\mathbf{R}$ that maximizes*

$$\psi(\mathbf{R}) = \sum_{\langle S_1, S_2 \rangle \notin \mathbf{R}} (P(S_1 \to S_2) - P(S_1 \to S_2|\mathbf{R})). \quad (34)$$

(We shall consider only positive influence and if $P(S_1 \to S_2) - P(S_1 \to S_2|\mathbf{R}) < 0$, we set it to 0.) We can prove the NP-hardness of this problem by a reduction from the HITTING SET problem.

THEOREM 5.3. *The problem of finding the most influential copying relationships $\mathbf{R}$ is NP-complete.* □

PROOF. It is obvious that if we guess a set $\mathbf{R}$, we can verify in polynomial time whether $\psi(\mathbf{R})$ is above a given value. So the problem is in NP.

We now prove the NP-hardness by a reduction from the MINIMUM HITTING SET problem, defined as follows. Let $H = (V, E)$ be a hypergraph with vertex set V and hyperedge set E. Then a set $V' \subseteq V$ is called a *hitting set* of H if for all edges $e \in E$, it holds that $V' \cap e \neq \emptyset$. V' is called a *minimum hitting set* if there does not exist another hitting set that has a smaller size.

We reduce the minimum hitting set problem to our problem as follows. For each $v \in V$, introduce a source S_v; in addition, introduce a source S_0 that has an edge to each S_v (copying from S_v) with probability 1. For each $v, v' \in V$, there is no edge between S_v and $S_{v'}$ (the copying probability is 0). For each $e \in E$, and each $v, v' \in e$, set $P(S_0 \to S_v) - P(S_0 \to S_v|\{S_0 \to S_{v'}\})) = 0$. Obviously, the construction is in polynomial time.

Now we show that we have a hitting set of size n if and only if $\psi(\mathbf{R}) = |V| - n$ (in other words, the minimum hitting set corresponds to the most influential $\mathbf{R}$).

- **"only if"**: For a hitting set V' with size n, we define $\mathbf{R} = \{S_0 \to S_v | v \in V'\}$. Then, for each $S_0 \to S_v \notin \mathbf{R}$, we have $P(S_0 \to S_v|\mathbf{R}) = 0$, so $\psi(\mathbf{R}) = |V| - n$.
- **"if"**: Given $\mathbf{R}$ that satisfies $\psi(\mathbf{R}) = |V| - n$, we define $V' = \{v | S_0 \to S_v \in \mathbf{R}\}$. Given the construction, $\mathbf{R}$ must contain n relationships, and for each $S_0 \to S_v \notin \mathbf{R}$, we have $P(S_0 \to S_v|\mathbf{R}) = 0$. Thus, V' must be a hitting set and its size is n.

This concludes our NP-hardness proof. □

We now present several observations, based on which we propose a fast greedy algorithm. The first observation is that according to Eq.(33), $P(S_1 \to S_2|\mathbf{R})$ relies on only relationships involving S_1. Thus, we can construct $\mathbf{R}$ by finding for each source the most "influential" sources among those it may copy from.

PROPOSITION 5.4. *If we denote by $\bar{D}(S_1), S_1 \in \mathcal{S}$, the sources with which S_1 has a copying relationship in $\mathbf{R}$, by $\mathbf{R}(S_1)$ the relationships in $\mathbf{R}$ involving S_1, and by $\Delta(S_1 \to S_2|\bar{D}(S_1)) = P(S_1 \to S_2) - P(S_1 \to S_2|\mathbf{R}(S_1))$, we have $\psi(\mathbf{R}) = \sum_{S_1 \in \mathcal{S}} \sum_{S_2 \notin \bar{D}(S_1)} \Delta(S_1 \to S_2|\bar{D}(S_1))$.* □

PROOF. We compute $P(S_1 \to S_2|\mathbf{R})$ by Eq.(33), which considers only the relationships associated with S_1. Thus, the equation in the proposition holds. □

The second observation reveals the relationship between the joint influence of sources in $\bar{D}(S_1)$ and the individual influence of each of them. Accordingly, we can simplify our algorithm by considering influence of an individual copying relationship on another.

PROPOSITION 5.5. *If we denote by $\Delta(S_1 \to S_2|S) = P(S_1 \to S_2) - P(S_1 \to S_2|\{S_1 \to S\})$, then, (1) $\Delta(S_1 \to S_2|\bar{D}(S_1)) \geq \Delta(S_1 \to S_2|S)$ for each $S \in \bar{D}(S_1)$; and (2) $\Delta(S_1 \to S_2|\bar{D}(S_1)) \leq \sum_{S \in \bar{D}(S_1)} \Delta(S_1 \to S_2|S)$.* □

PROOF. (1) As $S \in \bar{D}(S_1)$, for each $O.A$, the $P_v P_f$ conditioned on S (*i.e.*, $S_1 \to S$) must be no larger than the $P_v P_f$ conditioned on $\bar{D}(S_1)$; thus, $P(\Phi_{O.A}(S_1)|\bar{D}(S_1)) \geq \Delta(S_1 \to S_2|S)$, so $P(S_1 \to S_2|\bar{D}(S_1)) \leq P(S_1 \to S_2|\bar{D}(S_1))$ and $\Delta(S_1 \to S_2|\bar{D}(S_1)) \geq \Delta(S_1 \to S_2|S)$.

(2) It is easy to prove that when $\bar{D}(S_1)$ contains a single source, the claim holds. We now consider the case when there are at least two sources. Suppose there exist $S, S' \in \bar{D}(S_1)$ such that $\Delta(S_1 \to S_2|S) + \Delta(S_1 \to S_2|S') \geq P(S_1 \to S_2)$. Then, $\Delta(S_1 \to S_2|\bar{D}(S_1)) \leq P(S_1 \to S_2) \leq \sum_{S \in \bar{D}(S_1)} \Delta(S_1 \to S_2|S)$.

Otherwise, for any $S, S' \in \bar{D}(S_1)$, we have $P(S_1 \to S_2|S) + P(S_1 \to S_2|S') < P(S_1 \to S_2) < P(S_1 \to S_2) + P(S_1 \to S_2|\bar{D}(S_1))$. Assume there are k sources in $D(S_1)$. Then,

$$\sum_{S \in \bar{D}(S_1)} \Delta(S_1 \to S_2|S) < (k-1)P(S_1 \to S_2) + P(S_1 \to S_2|\bar{D}(S_1)),$$

so $\Delta(S_1 \to S_2|\bar{D}(S_1)) \leq \sum_{S \in \bar{D}(S_1)} \Delta(S_1 \to S_2|S)$. □

Based on Proposition 5.5, we wish to greedily select relationships that have the highest accumulated effect on others. The next two observations state which relationships should be pruned (not added to $\mathbf{R}$). The third observation shows that we want to prune relationships that cause less accumulated changes on others than being affected by others; its proof is based on Proposition 5.5.

PROPOSITION 5.6. *For any $S \in \mathcal{S}$, if there exist S_1 and S_2, $S_1 \neq S_2$, such that (1) $\Delta(S \to S_1|S_2) > \sum_{S_0 \neq S, S_1, S_2} \Delta(S \to S_0|S_1)$, (2) $(S, S_2) \in \mathbf{R}$, and (3) $\mathbf{R}' = \mathbf{R} \cup \{(S, S_1)\}$, then $\psi(\mathbf{R}) > \psi(\mathbf{R}')$.* □

PROOF. According to Proposition 5.5, we have

$$\begin{aligned}
&\psi(\mathbf{R}) - \psi(\mathbf{R}') \\
&= \Delta(S \to S_1|\mathbf{R}) - \sum_{\langle S_1, S_2\rangle \notin \mathbf{R}} (\Delta(S_1 \to S_2|\mathbf{R}') - \Delta(S_1 \to S_2|\mathbf{R})) \\
&\geq \Delta(S \to S_1|S_2) - \sum_{S_0 \neq S, S_1, S_2} \Delta(S \to S_0|S_1) > 0.
\end{aligned}$$

□

The final observation shows that we should prune a relationship if it can be significantly affected by those already selected into $\mathbf{R}$, because it is more likely to be a co-copying or transitive copying and its effect on others will be dominated by the relationships in $\mathbf{R}$.

OBSERVATION 5.7. *For any $S \in \mathcal{S}$, if there exist S_1 and S_2, $S_1 \neq S_2$, such that (1) $\Delta(S \to S_1|\{S \to S_2\}) > .5$, (2) $(S, S_2) \in \mathbf{R}$, and (3) $\mathbf{R}' = \mathbf{R} \cup \{(S, S_1)\}$, then typically $\psi(\mathbf{R}) > \psi(\mathbf{R}')$.* □

REASONING. Because $\Delta(S \to S_1|\{S \to S_2\}) > .5$, $P(S \to S_1|\{S \to S_2\}) < .5$ and S is unlikely to be a direct copier of S_1. We can categorize the common items between S and S_1 into two types, those that are also shared between S and S_2 and those that are not strong evidence of copying. Thus, $\Delta(S_1 \to S_2|\mathbf{R}') - \Delta(S_1 \to S_2|\mathbf{R}))$ is small for most $S_1 \to S_2$. □

Based on these observations, for each source $S \in \mathcal{S}$ our algorithm proceeds in four steps.

1. Find all sources from which S is likely to copy or copying is likely but the direction is unclear, denoted by $\bar{R}(S)$.
2. For each $S_1, S_2 \in \bar{R}(S), S_1 \neq S_2$, compute $\Delta(S \to S_1|S_2)$ and $\Delta(S \to S_2|S_1)$ (effect). For each $S' \in \bar{R}(S)$, compute $\sigma(S') = \sum_{S_0 \neq S, S'} \Delta(S \to S_0|S')$ (effects on others) and $\Lambda(S') = \max_{S_0 \neq S, S'} \Delta(S \to S'|S_0)$ (maximum effect by others).
3. Find the source S' with the highest $\sigma(S')$ (most influential) and remove it from $\bar{R}(S)$. If $\sigma(S') > \Lambda(S')$ (affecting others more than being affected), (1) add (S, S') to $\mathbf{R}$, (2) for each $S_0 \in \bar{R}(S)$, if $P(S \to S_0|\{S \to S'\}) < .5$, remove S_0 from $\bar{R}(S)$ (pruning); (3) update $\sigma(S_0), S_0 \in \bar{R}(S)$, by ignoring the removed sources.
4. Go to step 3 until $\bar{R}(S) = \emptyset$.

EXAMPLE 5.8. *Consider S_4 in the motivating example (Tbl.1). Assume $\Delta(S_4 \to S_1|S_3) = .8, \Delta(S_4 \to S_2|S_3) = .8, \Delta(S_4 \to S_3|S_1) = .5, \Delta(S_4 \to S_3|S_2) = .5$. $S_4 \to S_3$ has the highest accumulated effect (1.6) and is less affected by others (.5), so we add it to $\mathbf{R}$. Since both $S_4 \to S_1$ and $S_4 \to S_2$ are significantly affected by $S_4 \to S_3$, we can prune them and terminate.* □

5.3 Improving the efficiency

Algorithm GLOBALDETECTION) gives details for global detection. The following theorem shows its complexity.

Algorithm 2: GlobalDetection($\mathcal{S}, \mathcal{O}$)

```
Input  : S sources, O objects
Output : D Copying probabilities between each pair of sources in S
// Pairwise copying detection
1  D^0 ← PairwiseDetection(S, O);
   // Find R
2  R ← ∅;
3  foreach S ∈ S do
4      R̄ ← FindOriginals(S, D^0); // Find sources S copies from
5      foreach S_1 ∈ R̄ do
6          σ(S_1) ← 0; Λ(S_1) ← 0;
7      foreach ⟨S_1, S_2⟩ ∈ R̄, S_1 ≠ S_2 do
8          Δ(S_1|S_2) ← P(S → S_1) − P(S → S_1|{S → S_2});
9          if Δ(S_1|S_2) < 0 then
10             Δ(S_1|S_2) ← 0;
11         σ(S_2) ← σ(S_2) + Δ(S_1|S_2);
           Λ(S_1) ← max(Λ(S_1), Δ(S_1|S_2));
12     while R̄ ≠ ∅ do
13         find S' with the max σ(S');
14         R̄ ← R̄ − {S'};
15         if σ(S') > Λ(S') then
16             R ← R ∪ {(S, S')};
17             foreach S_0 ∈ R̄ do
18                 if P(S → S_0) − Δ(S_0|S') < .5 then
19                     R̄ ← R̄ − {S_0};
20                     foreach S_1 ∈ R̄ do
21                         σ(S_1) ← σ(S_1) − Δ(S_0|S_1);
22                 else
23                     σ(S_0) ← σ(S_0) − Δ(S'|S_0);
   // Recompute global copying probabilities
24 foreach S_1, S_2 ∈ S, S_1 ≠ S_2 do
25     if (S_1, S_2) ∈ R then
26         D(S_1, S_2) = D^0(S_1, S_2);
27     else
28         D(S_1, S_2) = GlobalPr(S_1, S_2, R);
```

THEOREM 5.9. *Let $m = \max_{S \in \mathcal{S}} |\bar{R}(S)|$ and r be the maximum number of sources related to a source in $\mathbf{R}$. The complexity of* GLOBALDETECTION *is $O(m^2|\mathcal{S}||\mathcal{O}| + r|\mathcal{S}|^2|\mathcal{O}|)$.* □

PROOF. Computing Δ spends time $O(|\mathcal{O}|)$. We compute it for each pair of sources in $\bar{R}$, so Line 7-11 spends time $O(m^2|\mathcal{O}|)$. On the other hand, Line 12-23 spends time $O(m^2)$. So Line 1-23 spends time $O(m^2|\mathcal{S}||\mathcal{O}|)$.

Similarly, Line 28 spends time $O(r|\mathcal{O}|)$ and we run it for every pair of sources. So Line 24-28 spends time $O(r|\mathcal{S}|^2|\mathcal{O}|)$. This proves the claim. □

Note that the algorithm can be quite expensive when there are a large number of sources. We can further simplify the algorithm in three ways.

1. When we generate $\mathbf{R}$, we can include in $\bar{R}(S)$ only sources that are likely to cause significant changes or be significantly changed; in particular, those that share a lot of common values with S (not only a high copying probability). In this way, we can reduce m. (In experiments we consider sources with which S shares 20% of its values.)
2. When we compute $P(S_1 \to S_2|\mathbf{R})$, we can compute $P(\Phi_{O.A}(S_1)|\mathbf{R})$ only on objects that are strong indicators of copying. Specifically, we start with computing $P(\Phi_{O.A}(S_1))$;

then, only if $\frac{P(\Phi_{O.A}(S_1)|S_1 \to S_2)}{P(\Phi_{O.A}(S_1)|S_1 \not\to S_2)} > \tau$ or $\frac{P(\Phi_O(S_1)|S_1 \to S_2)}{P(\Phi_O(S_1)|S_1 \not\to S_2)} > \tau$, we compute $P(\Phi_{O.A}(S_1)|\mathbf{R})$ for each attribute of O. Here, τ is a threshold indicating strong evidence for copying and controls the level of approximation (in experiments we set $\tau = 2$). In this way, we usually need to compute $P(\Phi_{O.A}(S_1)|\mathbf{R})$ for significantly less data items, but where we do compute $P(\Phi_{O.A}(S_1)|\mathbf{R})$, we also need to compute $P(\Phi_{O.A}(S_1))$.

3. When we apply Eq.(32), instead of using $P(S_1 \overset{O.A}{\longrightarrow} S)$, we can use the default selectivity $s_{obj}\vec{s_A} + (1 - s_{obj})\vec{s_A}$. In this way, computing $P(\Phi_{O.A}(S_1)|\mathbf{R})$ is much cheaper.

5.4 Instance-level copying detection

Finally, we re-visit instance-level copying detection. Eq.(20) and Eq.(29) compute the probability of copying a particular object or data item, but they are *local* and consider only a pair of sources. In the real life, a source typically copies an object or an item from a single source rather than from multiple sources. With this assumption, we can apply *global* instance-level copying detection.

We start with the probability of S_1 copying a particular object O from S_2. There are three possibilities: S_1 copying O from S_2 (with a-priori probability $P(S_1 \overset{O}{\longrightarrow} S_2|S_1 \to S_2)P(S_1 \to S_2)$), S_1 copying O from another source $S \neq S_1, S_2$ (with a-priori probability $P(S_1 \overset{O}{\longrightarrow} S|S_1 \to S)P(S_1 \to S)$), and S_1 providing values on O independently (with a-priori probability $(1 - \Pi_{S \neq S_1}(1 - P(S_1 \overset{O}{\longrightarrow} S|S_1 \to S)P(S_1 \to S))))$. In the third case, for each attribute $A \in \mathcal{A}$, S_1 can copy $O.A$ from another source (with probability $P_A = 1 - \Pi_{S \neq S_1}(1 - s_A^{\not\to} P(S_1 \overset{O}{\not\to} S|S_1 \to S)P(S_1 \to S))$), or provide the value independently. We denote by $\mathcal{D}(S)$ the copying relationships related to S. By Bayesian analysis, we have

$$
\begin{aligned}
&P(S_1 \overset{O}{\longrightarrow} S_2|\Phi_O(S_1), \mathcal{D}(S)) \\
&= \frac{P(\Phi_O(S_1)|S_1 \overset{O}{\longrightarrow} S_2)P(S_1 \overset{O}{\longrightarrow} S_2|\mathcal{D}(S))}{P(\Phi_O(S_1)|\mathcal{D}(S))} \\
&= \frac{P(\Phi_O(S_1)|S_1 \overset{O}{\longrightarrow} S_2)P(S_1 \overset{O}{\longrightarrow} S_2|S_1 \to S_2)P(S_1 \to S_2)}{P(\Phi_O(S_1)|\mathcal{D}(S))} \quad (35) \\
&P(\Phi_O(S_1)|\mathcal{D}(S)) \\
&= \sum_{S \neq S_1} P(\Phi_O(S_1)|S_1 \overset{O}{\longrightarrow} S)P(S_1 \overset{O}{\longrightarrow} S|S_1 \to S)P(S_1 \to S) \\
&+ (1 - \Pi_{S \neq S_1}(1 - P(S_1 \overset{O}{\longrightarrow} S|S_1 \to S)P(S_1 \to S))) \cdot \\
&\Pi_{A \in \mathcal{A}}(P_A + (1 - P_A)P(\Phi_{O.A}(S_1))). \quad (36)
\end{aligned}
$$

Similarly, we can compute the probability of copying on $O.A$; we skip the details.

6. EXPERIMENTAL RESULTS

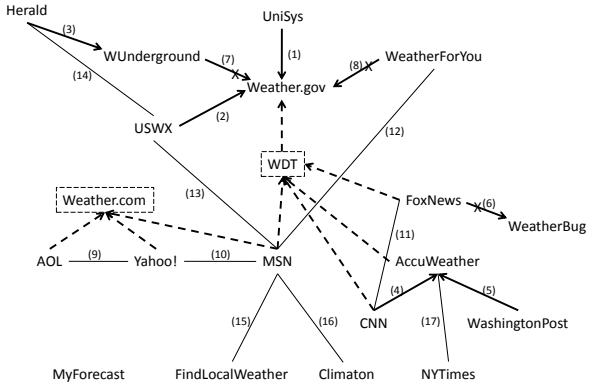
We conducted experiments on two real-world data sets: the weather data and the AbeBooks data. They are very different in the size of the domain, the number of the sources, and the characteristics of the data. Together with synthetic data generated from the AbeBooks data (Sec.6.4), we are able to test various aspects of our models, showing their effectiveness, stability, and efficiency.

6.1 Experiment settings

We implemented the model in [6] (ACCU), the basic model (BASIC), the correlated-copying model (LOCAL), and global detection (GLOBAL). We set the parameters as described in Sec.4.1.1, with initial values $\alpha = .25$ and $k_0 = .8$. We did not assume knowledge of correctness or popularity of values and conducted truth finding and copying detection iteratively (Sec.2). We used Java and experimented on a WindowsXP machine with 2.66GHz Intel CPU and 3.48GB of RAM.

We experimented on two data sets, the weather data set (Sec.6.2) and the AbeBooks data set (Sec.6.3). We have a partial golden standard for the first but no golden standard for the second; thus, we focus on the correctness of our results on the first and the efficiency of our algorithm on the second. For a better understanding of how our methods perform in detecting various types of copying, we generated synthetic data from the AbeBooks data set and report correctness of our results (Sec.6.4).

We measured *precision*, the proportion of identified copying that are real, and *recall*, the proportion of real copying that are identified. *F-measure* is computed as $\frac{2PR}{P+R}$, where P is the precision and R is the recall. On the synthetic data, to better quantify how we detect transitive and co-copying, we report sensitivity, the proportion of real copying that are identified with the correct direction (a strict version of *recall*), and *specificity*, the proportion of non-copyings that are identified as such.



	Rel	Reason
G	(1)	*UniSys* lists *www.nws.noaa.gov* (a mirror of *Weather.gov*)
O		as a resource.
L	(2)	*USWX* links to *Weather.gov* in source code.
D	(3)	*Herald*'s source code contains icons and links from *WUnderground*.
E	(4)	*AccuWeather* lists *CNN* as a client.
N	(5)	*AccuWeather* lists *WashingtonPost* as a client.
R	(6)	*WeatherBug* lists *FoxNews* as a partner, but they share only
E		11.4% of the non-key data.
M	(7)	*WUnderground* lists *www.nws.noaa.gov* (*Weather.gov*) but they
O		share only 2 non-key attributes and 16.5% non-key data.
V	(8)	*WeatherForYou* lists *www.nws.noaa.gov* (*Weather.gov*) but they
E		share only 32% non-key data dispersed among attributes and objects.
	(9)	*Weather.com* lists *AOL* and *Yahoo!* as affiliates.
S	(10)	*Weather.com* lists *MSN* and *Yahoo!* as affiliates.
I	(11)	*WDT* lists *CNN* and *FoxNews* as customers.
L	(12)	Potential co-copiers and share a lot of data.
V	(13)	Potential co-copiers and share a lot of data.
E	(14)	No explicit claim from *Herald* but sharing a lot of data.
R	(15)	No explicit claim from *FindLocalWeather* but sharing a lot of data.
	(16)	No explicit claim from *Climaton* but sharing a lot of data.
	(17)	No explicit claim from *NYTimes* but sharing a lot of data.

Figure 3: Copying between weather sources. There are 18 sources and 2 hidden sources (in dashed box). A solid arrow represents a copying indicated by the source website (non-crossed ones are "golden" dependencies); a dashed arrow represents a copying associated w. a hidden source; and a thin line represents a "silver" dependence that we derive.

6.2 Results on the weather data

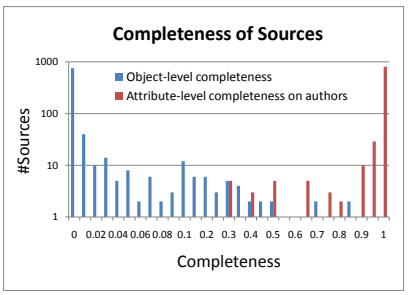


Figure 4: Coverage of AbeBooks sources. 92% of the sources provide less than 5% of the books.

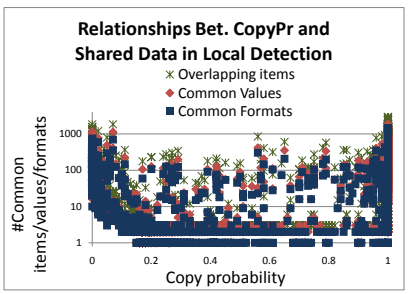


Figure 5: Copy probability vs. shared items, values, formats.

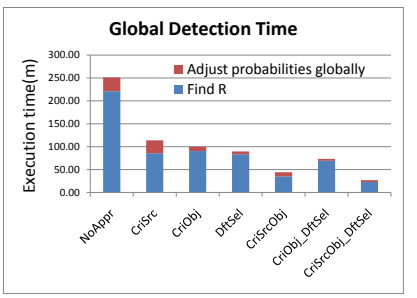


Figure 6: Execution time of various approximations for global detection.

Data: We collected weather data for 30 major USA cities from 18 websites about every 45 minutes. We consider (city, time) as the key. There are in total 33 collections in a day and thus 990 objects. We manually map the attributes and there are 28 distinct attributes. Among them, 10 are provided by at least 10 sources and 11 are provided by only 1 source; on the other hand, a source on average provides 11 attributes, while the max is 15 and the min is 3.

This data set introduces four new challenges. First, there is hardly a true or false notion for weather-related data; thus, we need to consider popularity of provided values and we do so in a similar way as we consider formatting popularity. Second, the weather data are often updated frequently and a copier may not have copied the most up-to-date data at some time of crawling; thus, we need to be able to detect copying even when the copying percentage is not high and so setting proper parameters is critical. Third, most sources have the same object-level completeness and similar completeness for each attribute, and each source has a consistent formatting style for the same attribute (by applying some style sheet); thus, we lack evidence from completeness and formatting for direction decision. Fourth, there are two hidden sources, *WDT*[7] and *Weather.com*, that are co-copied by sources in our data set, but we cannot crawl them because of commercial or technical reasons.

Golden standard: To find copying between sources, we investigated the websites for explicitly mentioned partnerships, clients, and resources; in addition, we checked source code of the webpages for URLs and citations of other sources. We found 8 copyings between the crawled sources and 8 from the crawled sources to the hidden ones (shown in Fig.3). Accordingly, we created a semi-golden standard as follows. First, we manually examined every pair of sources with investigated copying; we removed 3 of them where we observed very small commonality of data with no particular pattern. The remaining 5 are called *golden dependencies*. Second, for each copier of a hidden source, we tried to find its co-copiers with which it shares a large portion of common data; if such co-copiers exist, we added a dependence (with no particular direction) with the one with the largest overlap. We added 5 dependencies in this way and call them *silver dependencies*. Third, for each source where we cannot find any claim from the website, we manually checked if there exist sources that share a large portion of common data; if so we added a dependence (without direction) with the one with the largest overlap. There are 5 such sources and we added 4 dependencies, also called *silver dependencies*. In total, there are 14 copyings (5 golden and 9 silver) in the semi-golden standard and we list the reasons for including them in Fig.3.

Results: GLOBAL obtained both precision and recall as .79. (1) Among the 5 golden dependencies, GLOBAL finds 4 of them, 2 in the correct direction and 1 with uncertain direction. It misses *Herald* → *WUnderground*: among the 8 common non-key attributes, they highly overlap only on conditions and visibility, and the shared values are fairly common also among other sources (especially for visibility), so the accumulated positive evidence is slightly weaker than the negative evidence, even with a reasonable parameter setting. (2) Among the 9 silver dependencies, GLOBAL finds 7. It misses the dependence between *Yahoo!* and *MSN*: even though local detection detects it, global detection removes it because the common data are covered by those between *Yahoo!* and *USWX*. This decision is reasonable because *Weather.com*, from which *Yahoo!* copies, and *USWX* might derive data from the same source and *Yahoo!* happens to share more with *USWX* at the crawlings. GLOBAL also misses the dependence between *Climaton* and *MSN*: it finds copyings from other sources to *Climaton* in local detection but eliminates them in global detection. If *Climaton* is indeed a copier, then the error is caused by wrong direction decisions in local detection. (3) Finally, GLOBAL has three false positives (including *Yahoo!* → *USWX*). For each of such pairs there is more or less co-copying and the global detection does not eliminate it because of some additional shared data (maybe shared at some crawlings). To summarize, except the false negative on *Herald* → *WUnderground*, errors are mainly caused by wrong direction decisions (partially because of the *lack-of-evidence* issue; this is consistent with our observation that considering completeness and formatting can improve detection of copying directions on the synthetic data) and uneliminated co-copying (partially because of the *updating* issue).

Table 4: Results of various methods on the weather data.

	ACCU	BASIC	LOCAL	GLOBAL
Precision	.5	1	.33	.79
Recall	.43	.14	.86	.79
F-measure	.46	.25	.48	.79

Tbl.4 compares the results of various methods. First, LOCAL obtains a higher recall (it detects in addition the dependence between *Yahoo!* and *MSN*) but a much lower precision. It returns 38 copyings, among which 38-12=26 are false positives, and GLOBAL removes 26-3=23 (88.5%) of them. Second, BASIC, in contrast, finds only 2 correct copyings. It misses a lot of copyings as it ignores evidence from per-object copying (as illustrated in Ex.3.2 and Ex.4.1); in addition, it tends to set s (selectivity) to a high value. Third, ACCU finds 2 golden dependencies (in the correct direction) and 4 silver dependencies. It finds more copyings than BASIC because it considers each shared uncommon value as a wrong value and so accumulates more positive evidence. It has 6 false positives, all co-copyings and transitive copyings. Note, however, that experiments on the synthetic data show improvement of BASIC over ACCU when the true/false notion does apply and there is extra evidence from completeness and formatting of data.

[7] *WDT* collects raw data from *Weather.gov* and applies some aggregation model, then resells the data to online media agencies.

Table 5: Approximation for detecting transitive copying.

	NOAPPR	CRISRC	CRIOBJ	DFTSEL	CRIOBJ_DFTSEL
Sensitivity	.75	.75	.77	.7	.78
Specificity	.81	.76	.89	.74	.84
Time(s)	99.3	81.8	50.2	32.4	14.9

Finally, on average GLOBAL spent 8 seconds for initialization (finding overlapping items, shared values and formats), 2.7 minutes for local detection, and 10 minutes for global detection (5.9 minutes for finding $\mathbf{R}$ and 4.1 minutes for globally adjusting probabilities). Thus, the efficiency of our algorithm is acceptable when the number of sources is small.

6.3 Results on AbeBooks data

Data: The AbeBooks data set was extracted in 2007 from *AbeBooks.com* by searching computer-science books. In the data set there are 877 bookstores, 1263 books, and 24364 listings, each containing attributes ISBN (key), name, and often authors. Unlike the weather data set, the true/false notion does apply here and this data set has much higher variety in completeness (Fig.4) and formatting (Fig.1).

Results LOCAL finds 1553 pairs of sources with copying. Fig.5 plots the copying probabilities for pairs whose Jaccard distance on data items (intersection over union) is at least .1. The plot is fairly semetric from left to right, showing that the numbers of shared items, values, and formats themselves do not decide the copying relationship, and our model consider in addition the popularity. GLOBAL finds only 465 pairs with direct copying, as it eliminates co-copying and transitive copying. Compared with LOCAL, it finds that most sources copy from or being copied by only a few sources (the max is 17 and 9 for GLOBAL respectively, but 44 and 37 for LOCAL).

On average, GLOBAL took 1.6 minutes for initialization, 3.8 minutes for local detection, and 251.1 minutes for global detection. As this data set contains a lot of sources, global detection becomes the bottleneck; especially, between finding $\mathbf{R}$ and globally adjusting probabilities, the former is much more expensive (221.2 min). Fig.6 compares various approximations, including whether to apply Eq.(32) on only "critical" sources (CRISRC), on only "critical" objects (CRIOBJ), and to use s_{obj} instead of $P(S_1 \overset{O.A}{\rightarrow} S)$ (DFTSEL). We observe that CRISRC considers only critical sources (59%) and reduced time in finding $\mathbf{R}$, CRIOBJ considers only critical objects (6%) and reduced time in both steps, and DFTSEL simplifies computation of global probability and also reduced time in both steps. Finally, CRISRCOBJ_DFTSEL took only 26.8 minutes, fairly acceptable given that dependence detection is offline.

6.4 Experiments on synthetic data

6.4.1 Data generation

To test effectiveness of our models, we also experimented on synthetic data. To generate a copier C, we chose an original source S and a copier template T from the AbeBooks data set. Copier C copies from S and independently provides some values or uses some formats, for which we use those provided by T; essentially C is a copier of both S and T, but we discarded T. We assume C provides data in three steps: (1) among books in $\bar{O}(S) \cap \bar{O}(T)$, C copies p_o percent on all attribute values from S, then for each attribute A, modifies m_A percent of the copied values (per-object copying); (2) for the rest of the books in $\bar{O}(S) \cap \bar{O}(T)$, for p_a percent of non-key attributes C copies all values from S, then modifies m'_A percent of the values for each A (per-attribute copying); (3) for other data items in $\bar{O}(T)$, C provides values on its own while copies c_A percent for each A. For p_f percent of the copied items, C keeps the copied format and for the rest it changes the format. We believe real-world copiers copy in a more or less similar fashion, though may skip some steps or change their order. We randomly decided which data to copy and modify, and generated m_A, m'_A and c_A between $[0, 2m]$ by Gaussian distribution with mean m. By default we set $p_o = .8, p_a = 0, p_f = 1$, and $m = .1$.

We considered four cases: (1) *transitive copying:* five copiers C_1-C_5, where C_1 copies from S and C_{i+1} copies from $C_i, i \in [1, 4]$; (2) *co-copying:* five copiers all copying from S; (3) *multi-copying:* one copier copying sequentially from S_1-S_5 (*i.e.*, copying from S_i a random subset of $\bar{O}(S_i) \cap \bar{O}(T) - \cup_{j=1}^{i-1} \bar{O}(S_j), i \in [1, 5]$); (4) *single-source copying:* one copier copying from S. We pre-selected 10 sources $\bar{S}$ Whose object-level completeness ranges from 0.05 to .9 (not necessarily independent). For (1)(2) and (4), for each $S \in \bar{S}$ we ran the experiments 10 times; at each time we randomly chose 5 templates for (1)(2) and 10 templates for (4), and the templates are judged as independent of S by local detection (but not necessarily independent between themselves) and differ in completeness by at least .05. For (3), we ran the experiment 100 times, each time randomly choosing a sequence of 5 sources from $\bar{S}$ and trying to[8] randomly choose 10 templates that are independent of them and differ in completeness by at least .05; we detected copying for each copier separately. For all cases, we reported the average results of varying parameters and considering different evidence.

We set parameters by (1) setting default values (FIXPARA) where $s_{obj} = s = k = .8, s_A^{\rightarrow} = .9, s_A^{\not\rightarrow} = .4$; (2) setting empirical values (EMPPARA); and (3) first setting empirical values and then adjusting them (ADJPARA) (Sec.4.1.1). By default, we applied EMPPARA. We used true values decided from the real-world data, setting $p(v) = 1$ for true values and $p(v) = 0$ for false ones.

6.4.2 Transitive copying, co-copying, and multi-source copying

Transitive copying: We varied p_o from .1 to 1 and examined copying between sources in $\{S, C_1, \ldots, C_5\}$ (Fig.7). We observe that while GLOBAL slightly reduces sensitivity (by 3%) compared with LOCAL, it significantly improves specificity (avg .88). Also, GLOBAL obtains fairly stable results when p_o varies: when p_o is very small, the sensitivity is slightly lower because a copier can copy very few data and is not detectable; when p_o is very high, the sensitivity and specificity are slightly lower because a copier may transitively copy a lot of data from its transitive ancestor and even share some additional data (local detection found an average copying probability of .39 between templates), which is indistinguishable from direct copying. Finally, ACCU (hereafter we set $s = p_o(1 - m)$) performs worst; it assumes *item-wise independence* and considers only accuracy, so can often make mistakes about copying direction (it finds all direct copying but only 18% in the correct direction).

Table 5 compares various approximation methods and Fig.8 gives more details on specificity w.r.t. *#hops* between sources (#hops between C_i and $C_j, i > j$, is $i - j$ and #hops between C_i and S is i; 1-hop indicates direct copying) when $s_o = .8$. With no surprise, the more hops, the higher specificity; when #hops> 2, CRIOBJ obtains a specificity of above .9. Among the approximations, CRIOBJ spends half of the execution time as NOAPPR but obtains the best results (as it is not biased by effects on non-critical objects). CRIOBJ_DEFSEL further cuts the execution time by 70% but still obtains better results than NOAPPR; however, it reduces

[8]Sometimes this is impossible when there are insufficient number of sources independent of S_1-S_5.

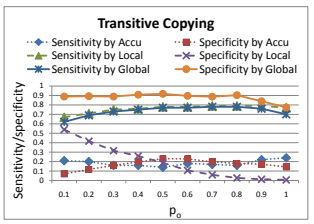


Figure 7: Transitive copying.

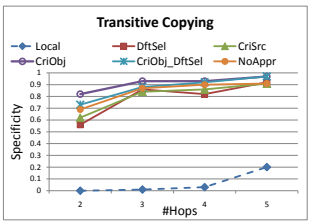


Figure 8: Specificity vs. #hops.

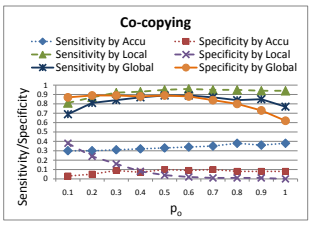


Figure 9: Co-copying.

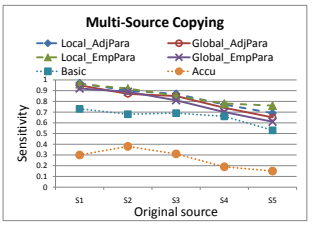


Figure 10: Multi-source copying.

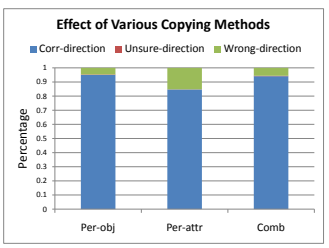


Figure 11: Different copying methods.

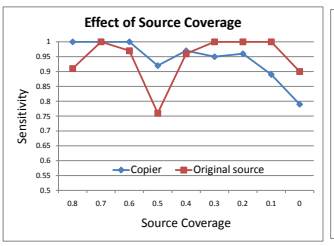


Figure 12: Effect of source coverage.

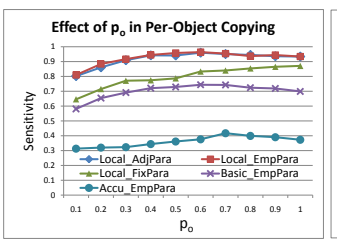


Figure 13: Effect of object selectivity.

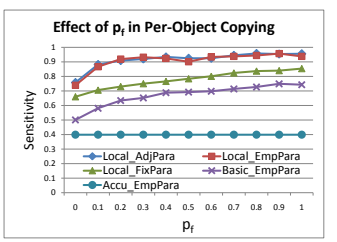


Figure 14: Effect of format keeping rate.

the specificity by 6% compared with CRIOBJ.

Co-copying: Again, we varied p_o and examined copying between $\{S, C_1, \ldots, C_5\}$ (Fig.9). GLOBAL again reduces sensitivity slightly (by 10%) and improves specificity significantly. However, we observe two differences from transitive copying. First, specificity is less stable: when $p_o = 1$, the specificity is only .62. When p_o is high, co-copiers can share a large number of copied values and maybe some additional values, so indistinguishable from direct copying. Second, LOCAL obtains higher sensitivity than in the transitive-copying case (avg .94 vs. .76). Actually, we have detected *all* copying in both cases, but did worse in finding the correct direction in case of transitive copying, as there the randomly chosen "original" sources are likely to have high completeness and lead to less precise direction detection.

Multi-copying: We examined copying between the copier and each original source (sensitivity) (Fig.10). We have five observations. (1) When a copier really copies from multiple sources, GLOBAL reduces the sensitivity only very slightly (GLOBAL_ADJPARA by 3%). (2) The sensitivity reduces from S_1 to S_5, as the copier tends to copy less (if any) data from sources towards the end of the sequence. (3) ADJPARA does not necessarily improve over EMPPARA for LOCAL, but by using it GLOBAL reduces the sensitivity much less (avg 3% vs. 8%) (however, GLOBAL can obtain lower specificity using ADJPARA in case of co-copying and transitive copying; details skipped). (4) BASIC does not consider object copying so obtains much lower sensitivity; (5) ACCU considers only accuracy and uses the same selectivity for all source pairs, so performs the worst.

6.4.3 *Single-source copying*

We now examine effectiveness of our methods in the case of a copier copying from a single source. As there is no multi-copying, co-copying, or transitive copying, GLOBAL obtains the same results as LOCAL.

Varying copying methods: We considered three types of copying: per-object copying ($p_o = .8, p_a = 0$), per-attribute copying ($p_o = 0, p_a = .5$), and combined copying ($p_o = .5, p_a = .5$). Fig.11 shows that in all cases LOCAL found *all* copying relationships, while did better in deciding the direction for per-object copying and combined copying, obtaining a sensitivity of around .95.

Varying parameters: We next examined effect of parameters on copying detection. In all cases all methods detected more than 99.9% of the copying and we next focus on copying direction. We first consider LOCAL_ADJPARA and LOCAL_EMPPARA. As shown in Fig.12, in per-object copying we did slightly better in judging direction when original sources have a small-to-medium coverage and when copiers have a medium-to-high coverage. Note however that this only shows a trend and the performance should also depend on accuracy and formatting styles of the sources. As shown in Fig.13, in per-object copying sensitivity increases as p_o increases and becomes stable when $p_o \geq .4$; this is because with higher p_o the copiers tend to copy more objects and be more detectable. As shown in Fig.14, in per-object copying sensitivity increases as p_f increases and becomes stable when $p_f \geq .2$; this is because with higher p_f the copiers tend to use different formatting patterns for copied data and its own data and be more detectable. As shown in Fig.15, in per-object copying sensitivity is stable even when m_A varies. As shown in Fig.16, in per-attribute copying sensitivity is stable when p_a varies.

Finally, we observe that (1) ACCU [6] obtains the worst results; (2) BASIC performs better than ACCU as it considers format and coverage in addition; (3) FIXPARA obtains better results than BASIC as it considers object copying, even though it uses fixed parameters; (4) ADJPARA and EMPPARA obtain the best results and adjusting parameters does not necessarily bring benefits.

Considering different measures: We considered per-object copying and varied LOCAL in two ways. First, we considered various combinations of measures: only accuracy, accuracy and coverage, accuracy and formatting style, and all. Second, we consider various ways of computing $P(\Phi_{O.A}(S_1))$ and $P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2)$: SRC considers only source-wise measures and uses $P(\Phi_{O.A}(S_1))$ in computing $P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2)$; SRCC considers only source-wise measures but uses $P^c(\Phi_{O.A}(S_1))$ for $P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2)$; BOTHC considers both source-wise and data-item-wise measures and uses $P^c(\Phi_{O.A}(S_1))$ for $P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2)$.

Fig.17 shows that considering only accuracy at best obtains a sensitivity of .92; when we use BOTHC, while considering coverage in addition can reduce the sensitivity, considering formatting in addition improves the sensitivity to .95; considering all evidence obtains the same results as not considering coverage. We also ob-

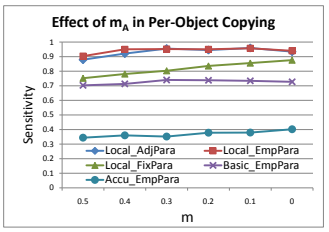


Figure 15: Effect of changing ratio.

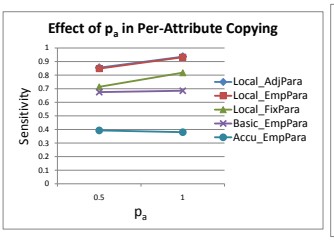


Figure 16: Effect of attribute copying probability.

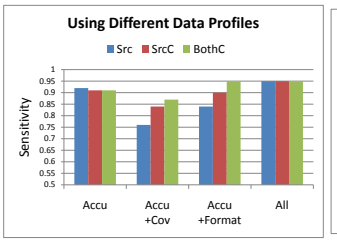


Figure 17: Using different data profiles.

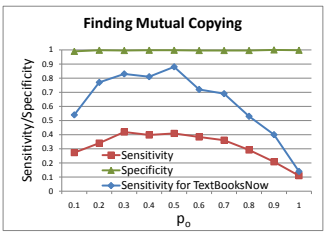


Figure 18: Detecting mutual copying.

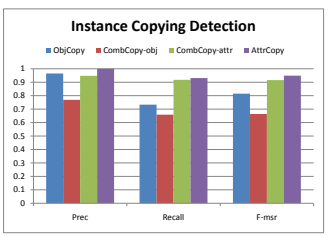


Figure 19: Detection of instance copying for single-source copying.

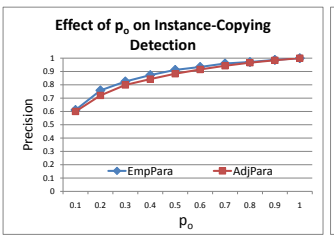


Figure 20: Effect of object selectivity on precision.

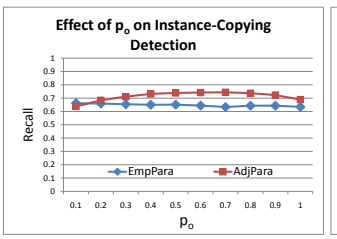


Figure 21: Effect of object selectivity on recall.

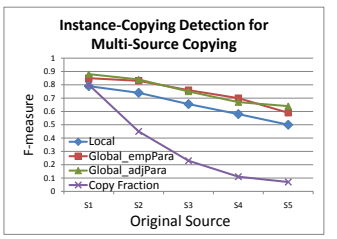


Figure 22: Detection of instance copying for multi-source copying.

serve that when we consider only accuracy, SRCC and BOTHC do not necessarily improve over SRC; but when we consider in addition coverage or formatting, SRCC significant improves over SRC and BOTHC significantly improves over SRCC; when we consider all measures, possibly SRC already obtains the best results we can achieve, so SRCC and BOTHC do not bring further improvement.

Recall from Fig.13-16 that without considering object copying, BASIC (BOTHC w. all evidence) significantly improves over ACCU (SRC w. accuracy). When we consider object copying, even SRC with accuracy already performs well (.92), but there is still room for improvement and LOCAL has achieved 37.5% of it.

Finding mutual copying: Finally, we considered mutual copying and set $\delta = .2$. We first reported specificity on data we generated as before. Then, to test sensitivity, we generated mutually-copying source pairs: given sources S and T, we generated C_S and C_T, such that (1) for each $O \in \bar{O}(S) \cap \bar{O}(T)$, C_S and C_T copy on O with probability p_o, and each copying direction has the same probability, and (2) C_S provides the rest of items in $\bar{O}(S)$ and C_T provides the rest of items in $\bar{O}(T)$. As Figure 18 shows, while we obtain a specificity of nearly 1, we obtain a low sensitivity (avg .32). The sensitivity goes down when p_o increases, because the more copied objects, the harder it is to distinguish $P^c(\Phi_{O.A}(S_1))$ and $P(\Phi_{O.A}(S_1))$ and to find mutual copying. From the results, we see that detecting mutual copying in general is hard. However, for source *TextBooksNow*, which uses a rare format on authors (only providing the last name of the first author), we obtain a sensitivity of .88 when $p_o = .5$, showing that our method is more effective when the pair of sources have quite different formatting styles.

6.4.4 *Instance-copying detection*

We next report precision, recall, and F-measure of instance-level copying detection. In particular, *precision* measures among objects (resp. attributes) that we decide as being copied in per-object (resp. per-attribute) copying, how many are really copied; *recall* measures among objects (resp. attributes) that are really copied in per-object (resp. per-attribute) copying, how many are detected.

Fig.19 shows results for per-object copying, per-attribute copying, and combined copying under default setting when we adjust parameters (ADJPARA). For per-object copying, we obtained a high precision (.97) but a low recall (.73) in deciding if an object is copied. This is because when the shared objects, values, or formats are also popular among other sources, our algorithm is conservative and concludes with non-copying; in addition, recall that some copied values are revised and we may not detect object-copying in this case. On the other hand, for per-attribute copying, we obtained both high precision (1.0) and high recall (.93) on deciding if an attribute is copied, because there are more values on an attribute so we have more evidence. Finally, for combined copying, we still did well for detecting copied attributes (F-measure .92), but obtained a lower precision for detecting copied objects (.77); such errors are typically made to objects where unpopular attribute values (or formats) are copied in per-attribute copying, and the key values share the same format.

Fig.20 and 21 show results for per-object copying when we vary object selectivity (p_o). We observed that whereas the recall is stable, the precision drops when p_o is small, because we are more likely to make mistakes on highly overlapped objects. We also observe that ADJPARA obtained higher recall (and F-measure) than EMPPARA, showing the benefit of adjusting parameters in instance-level copying detection.

Finally, Fig.22 shows results for multi-source copying. We observe that the fraction of objects being copied drops dramatically from S_1 to S_5, but the F-measure of instance-level copying detection drops only mildly. GLOBAL assumes that a source typically copies a particular item from a single source (applying Eq.(35)) and so obtained both higher precision and higher recall than LOCAL, which decides the copied objects only between each pair of sources (Eq.(20)). Finally, we observe that EMPPARA and ADJPARA obtain similar F-measure in this experiment.

Summary: We summarize our experimental results as follows.

- GLOBAL improves over LOCAL significantly on identifying complex copying relationships.
- Among various approximations for global detection, CRIOBJ can cut the execution time by half or more but still obtain the best results. CRIOBJ_DFTSEL further reduces the execution time without sacrificing the results too much.
- In local detection, BASIC improves over ACCU significantly on copying-direction judgment by considering completeness

and formatting in addition to accuracy, using both source-wise and data-item-wise measures, and using $P^c(\Phi_{O.A}(S_1))$ in computation of $P(\Phi_{O.A}(S_1)|S_1 \rightarrow S_2)$. LOCAL improves over BASIC further by considering object copying.

- Instance-level copying detection obtains high accuracy, especially for deciding whether an attribute is copied in per-attribute copying. When a copier copies from multiple sources, assuming the copier copying each item from a single source (GLOBAL) leads to higher precision and recall.
- Setting parameters using EMPPARA beats FIXPARA significantly and can obtain quite stable results. ADJPARA does not show further benefits in either local or global detection for source-level copying detection, but does better for instance-level copying detection.

7. RELATED WORK AND CONCLUSIONS

This paper studied copying detection between a set of sources. We first improved previous techniques for pairwise detection by proposing a framework where we can plug in different types of copying evidence, and consider correlations between copying. We then described techniques for global detection where we eliminate co-copying and transitive copying. Experimental results show high effectiveness and efficiency of our algorithms. Interesting directions for future work include visualization of the copying relationships, and categorization and summarization of the copied instances.

Existing work on copying detection includes detecting copying between texts or programs [12, 2], between videos [11], and between structured data sources [6, 7]. The work most related to ours is [6], with which we have compared in detail in Sec.3 and in experiments. Recently there has been increasing interest in provenance of data [3]; such works assume knowledge of provenance while our models can be used to discover data provenance.

8. REFERENCES

[1] L. Berti-Equille, A. D. Sarma, X. L. Dong, A. Marian, and D. Srivastava. Sailing the information ocean with awareness of currents: Discovery and application of source dependence. In *CIDR*, 2009.

[2] P. Buneman. The recovery of trees from measures of dissimilarity. *Mathematics the Archeological and Historical Sciences*, pages 387–395, 1971.

[3] P. Buneman, J. Cheney, W.-C. Tan, and S. Vansummeren. Curated databases. In *Proc. of PODS*, 2008.

[4] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley & Sons, Inc., New York, 1991.

[5] X. L. Dong, L. Berti-Equille, Y. Hu, and D. Srivastava. Global detection of complex copying relationships between sources. http://www2.research.att.com/∼lunadong/publication/global_techreport.pdf.

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

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

[8] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. Duplicate record detection: A survey. *IEEE Trans. Knowl. Data Eng.*, 19(1):1–16, 2007.

[9] E. Gansner, Y. Hu, and S. Kobourov. GMap: Drawing graphs and clusters as map. In *IEEE Pacific Visualization Symposium*, 2010.

[10] N. Koudas, S. Sarawagi, and D. Srivastava. Record linkage: similarity measures and algorithms. In *SIGMOD*, 2006.

[11] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, and F. Stentiford. Video copy detection: a comparative study. In *CIVR*, pages 371–378, 2007.

[12] S. Schleimer, D. S. Wilkerson, and A. Aiken. Winnowing: Local algorithms for document fingerprinting. In *Proc. of SIGMOD*, 2003.

[13] X. Yin, J. Han, and P. S. Yu. Truth discovery with multiple conflicting information providers on the web. In *Proc. of SIGKDD*, 2007.

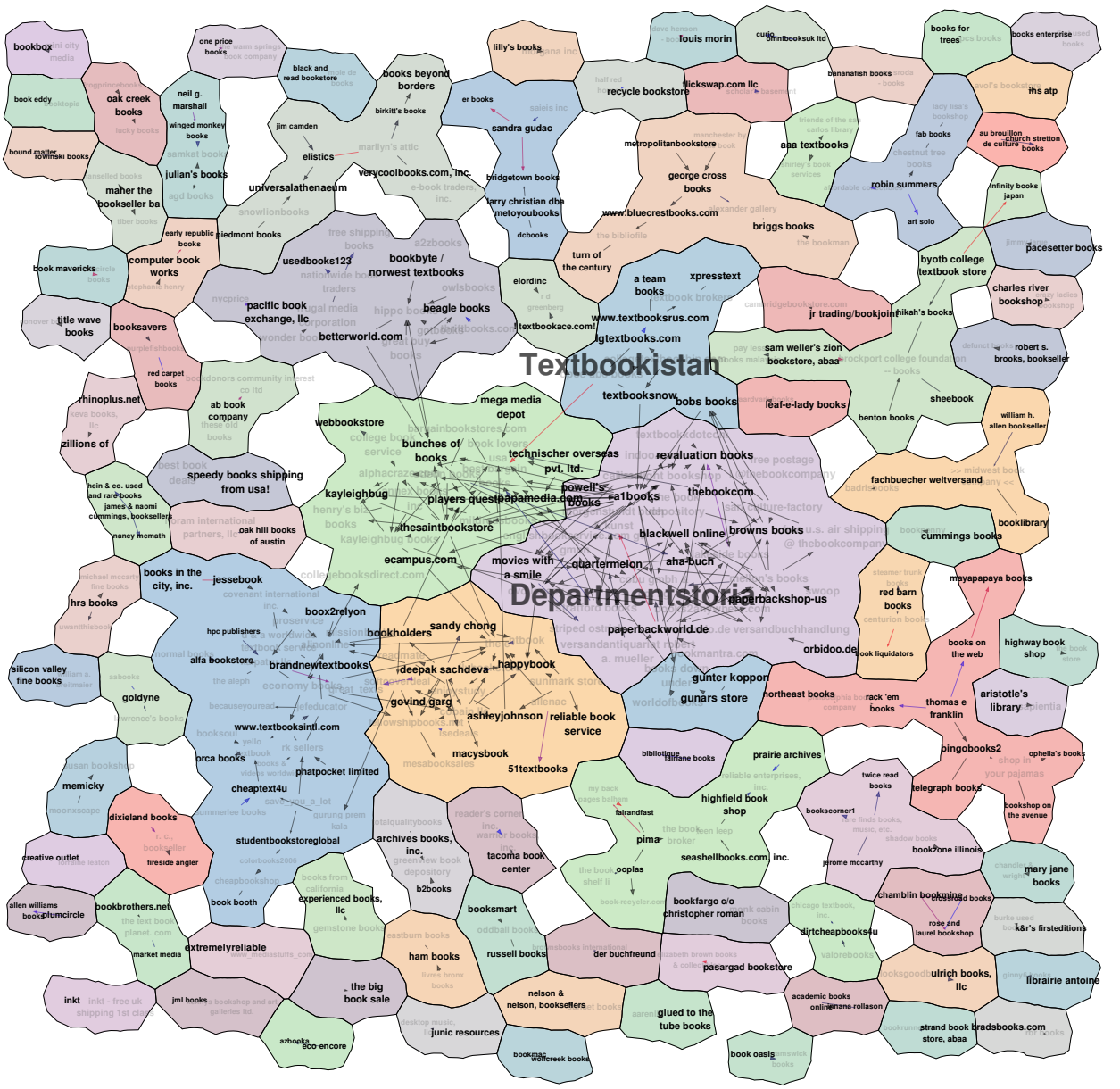


Figure 23: **Map of AbeBooks data sources. Each "node" represents a data source and the size of the font corresponds to the number of provided books; to avoid cluttering, we show overlapping ones half transparently. An edge $S_1 \to S_2$ indicates that S_1 copies from S_2; the size of the arrow indicates our confidence of the copying direction; the color indicates the probability of copying (black for 1, blue for .75, and red for .5, and other probabilities are represented by a blend of these colors; *e.g.*, purple for .5-.75.) Each "country" represents a cluster of sources, clustered by modularity clustering (see "Clauset, A., Newman, M.E.J., Moore, C.: Finding community structure in very large networks. *Physical Review E 70, 066111* (2004)") based on their copying relationships.**