ABSTRACT
A fundamental problem in data fusion is to determine the veracity of multi-source data in order to resolve conflicts. While previous work in truth discovery has proved to be useful in practice for specific settings, sources’ behavior or data set characteristics, there has been limited systematic comparison of the competing methods in terms of efficiency, usability, and repeatability. We remedy this deficit by providing a comprehensive review of 12 state-of-the-art algorithms for truth discovery. We provide reference implementations and an in-depth evaluation of the methods based on extensive experiments on synthetic and real-world data. We analyze aspects of the problem that have not been explicitly studied before, such as the impact of initialization and parameter setting, convergence, and scalability. We provide an experimental framework for extensively comparing the methods in a wide range of truth discovery scenarios where source coverage, numbers and distributions of conflicts, and true positive claims can be controlled and used to evaluate the quality and performance of the algorithms. Finally, we report comprehensive findings obtained from the experiments and provide new insights for future research.

1. INTRODUCTION
As online user-generated content grows exponentially, the reliance on Web data is inevitably growing in every application domain. However, data can be biased, noisy, outdated, incorrect, and thus, misleading and unreliable. Massive data coming from multiple sources amplifies the difficulty of ascertaining information veracity. The problem of truth discovery is intellectually and technically interesting enough to have attracted a lot of prior study, from the artificial intelligence and the database communities, sometimes investigated under the names of fact-checking [9], information credibility [13], information corroboration [8], data fusion [14] [10], conflicting data integration [5], or knowledge fusion [7]. Truth discovery problem can be formulated as follows. Given a set of assertions claimed by multiple sources, label each claimed value as true or false and compute the reliability of each source. One major line of work extends truth discovery models by incorporating prior knowledge either about the claimed assertions (e.g., SIMPLELCA and GUESSLCA [13]) or about the source reputation via trust assessment (e.g., SourceRank [1]). Another important line of research aims at iteratively computing and updating the trustworthiness of a source as a function of the belief in its claims, and then the belief score of each claim as a function of the trustworthiness of the sources asserting it (e.g., TRUTHFINDER [17]). In this line, several probabilistic models have been proposed to incorporate various aspects beyond source trustworthiness and claim belief, namely: the dependence between sources (e.g., DEPEN and its extensions [3]), the temporal dimension in discovering evolving truth [6], the difficulty of ascertaining the veracity of certain claims (e.g., COSINE, 2- and 3-ESTIMATES [8]), and the management of collections of entities (e.g., LTM [18]) or linked data [9].

There are a number of challenges in truth discovery. The first challenge is a theoretical one since it is difficult to formalize a method general enough to handle various data set characteristics and truth discovery scenarios. We observe that none of the methods constantly outperforms the others in terms of precision and a “one-fits-all” approach does not seem to be achievable. Another challenge is related to the usability of the methods. Assumptions of truth discovery models and complex parameter setting make current approaches still difficult to use and apply to the wide diversity of information available on the Web.

Related Work. Previous comparative studies such as the work of Li et al. [11] and [10] are based on real-world data sets and gold standards because, in practice, the complete ground truth often does not exist or is out-of-reach. Such gold standards are samples of the ground truth (generally less than 10% of the original data set’s size). We claim that they are not statistically significant to be legitimately used for evaluating and comparing existing methods in a systematic way. Moreover, previous comparisons did not study important algorithmic aspects of the methods such as parameter settings, time complexity, repeatability, computational issues, scalability, and convergence of the algorithms. They did not test them extensively for a wide range of truth discovery scenarios systematically generated with the control of the complete ground truth distribution. The experimental framework and data set generator we propose for comparing the methods are novel, practical contributions to the field, so that others can use and extend them for benchmarking, parameter setting and tuning of existing and new truth discovery algorithms. Publicly-available data sets with complete ground truth are notoriously difficult to obtain. The data set generator can serve as a useful proxy for what-if scenarios and reproducibility, to understand, in a systematic way, the data set characteristics that have significant impact on the performance and quality of the algorithms. The goals of our study are:
(1) To provide a clear explanation of each algorithm, and allow comparison of their properties by using common notation, termi-
ology, experimental set-ups, data sets, and test cases.
(2) To provide reference implementations of these algorithms against which future algorithms can be compared, new data sets can be analyzed, and on top of which algorithms for different problems or applications can be built, and finally,
(3) To perform a thorough experimental evaluation of the algorithms over a variety of data sets and report their performance and quality for a wide spectrum of parameter settings.

This paper is structured as follows. In Section 2, we define the problem of truth discovery and describe the algorithms in detail. In Section 3, we present our comparative study based on synthetic data sets systematically generated to demonstrate the quality of the algorithms in various truth discovery scenarios. Then, we study scalability, and finally, we evaluate the methods on five real-world data sets. In Section 4, we recapitulate our findings and conclude the paper.

2. TRUTH DISCOVERY ALGORITHMS

We consider the truth discovery algorithms that take, as input data, a set of claims in the form of quadruplets (claimID, sourceID, dataItemID, value) and infer, as output result, a Boolean truth label for each claim. In addition, the truth discovery algorithms may also return T v, the truthworthiness of each source s, and C v, the confidence of each value v. For example, consider the four sources in the example of Table 1(a) adapted from [3]. They provide claims on affiliation of four researchers such as (AuthorA,AuthorB,AuthorC). Source coverage (Cov) is 1 for S 1, .25 for S 2, and .75 for S 3 and S 4. Only S 1 actually provides a correct value for each data item, from d 1 to d 4 , in conformance with the ground truth (GT). Depending on the number of distinct values per data item (Conf) — e.g., d 1-d 4 have 2 distinct values — some algorithms can make random guessing or wrong decisions if some sources copy claims from another source.

In Table 1(b), source truthworthiness has been computed by DEPEN and TRUTHFINDER algorithms. The precision is computed from the number of true positives in (GT) also returned by the algorithms (.75 and .25, respectively). Truthworthiness of S 1 is .0489 for TRUTHFINDER, whereas it is .0323 for DEPEN. Table 1(c) shows the confidence of each value computed by TRUTHFINDER. The values considered to be true by this algorithm are in bold. As illustrated by this example, truth discovery algorithms may have different precision and output results depending on parameter setting and data set characteristics. In this paper, we study the effect of both on the quality and performance of 12 truth discovery algorithms from the literature. We use the notations presented in Table 2. Each truth discovery algorithm is presented in detail with its pseudocode where Gorrefers to the computation of value confidence C v, and F refers to the computation of source truthworthiness, T S.

We study the impact of various parameter settings on the quality of each algorithm and we analyze time complexity in Table 3. We made several choices for the consistency and fairness of our study. First, we initialized source truthworthiness T S to .8 for all algorithms because it maximizes the precision of most algorithms. Second, we use the Book data set for this preliminary parameterization study. The Book data set has been formatted in different versions so that all algorithms can be compared from the same input data set. Third, we use the same convergence test for all algorithms: the difference of source truthworthiness cosine similarity between two successive iterations to be less than or equal to a given threshold, δ, as we will describe in this section. We will discuss these choices at the end of the section and conclude on this first set of experiments dedicated to parameter setting. Due to the space limitation, we had to limit the presentation of our results but we invite the reader to access the full set of the experimental results and codes in [2].

2.1 TruthFINDER

TRUTHFINDER proposed in 2008 by Yin et al. [17] applies a Bayesian analysis to compute the confidence of a claim.

Algorithm. TRUTHFINDER relies on the honesty of the sources and follows the heuristics that a source providing mostly true claims for many data items will likely provide true claims for other objects. In Algorithm 2.1, the probability of a value being wrong is $1 - T_s$. Thus, if the value is provided by many sources, then its probability of being wrong is $\prod_{s \in S}(1 - T_s)$. Following this general idea, the source truthworthiness in TRUTHFINDER is $T_s = \frac{\sum_{v \in S} C_v |/| V_s|}{\theta}$ and the confidence score of a value is $\sigma_v = \frac{\sum_{s \in S} \ln(1 - T_s)}{\theta}$. Logarithm is used to avoid underflow of the truthworthiness when the quantities are small. TRUTHFINDER adjusts the confidence score of a claim so that it incorporates the influence (or support) that similar claims may have mutually on each other as $\sigma_v = \sigma_v + \rho \sum_{v' \neq v} \sigma_{v'} \cdot \text{sim}(v, v')$. For instance, for a multi-valued data item, a source providing the values (AuthorA, AuthorB) for a book will support another source that provides the values (AuthorA, AuthorB, AuthorC) for the same book (but not inversely). The weight of such support between the values is controlled by the parameter $\rho \in [0, 1]$. The final confidence of a claim is then computed in $\theta$ with a logistic function to be positive. The damping factor $\gamma$ compensates the effect when sources with similar values are actually dependent. Since TRUTHFINDER computes similarity between values, it can be dramatically affected by the number of distinct values to compare which explains relatively lower performance when the number of conflicts is high. Finally, TRUTHFINDER uses the difference of source truthworthiness cosine similarity between two successive iterations to be less than or equal to a given threshold, $\delta$. The value with the highest confidence is then selected as the true value among the other (false) values for a given data item.
Parameter Setting. TruthFinder has three different parameters to be set: \(\rho, \gamma, \) and \(T_s\). We vary every parameter value while fixing the other parameters’ values as reported in the next table.

| Fixed Values \(\rho, \gamma, T_s\) | Variables \(\delta, T_s\) | Change if one of the fixed parameters is changed |
|--------------------------------|-------------------|-----------------------------------------------|
| \(\rho = .5, T_s = .8\) | \(\gamma, \delta\) | No significant change |
| \(\rho = .5, \gamma = .1\) | \(T_s\) | No significant change |
| \(\rho = .5, \gamma = .1, T_s = .8\) | \(\delta\) | Max (9777) for \(\rho = .5\) |

We vary \(\delta\), the convergence threshold from .001 to 1E–5 without any change in precision but increasing the execution time from 435 ms to 526 ms (\(\approx +21\%\)) for the Book data set. Finally, we use \(\delta = .001\) and the values that maximize the precision for the Book data set: \(\rho = .5, \gamma = .1, T_s = .8\) (noted \(\checkmark\) in the table).

2.2 Information Corroboration

Three algorithms have been proposed in 2010 by Galland et al. in [3], namely COSINE, 2-ESTIMATES, and 3-ESTIMATES.

COSINE in Algorithm 2.2 starts by initializing the confidence of each value and the truthworthiness of each source. Then, it iteratively computes source truthworthiness in \(\Theta\) as a linear function of the truthworthiness achieved in the previous iteration. For each claimed value, the value confidence is computed as a function of the current truthworthiness scores of the sources claiming this value minus the truthworthiness scores of disagreeing sources in \(\Theta\).

2-ESTIMATES in Algorithm 2.3 is a probabilistic model for estimating source truthworthiness and value confidence. As in COSINE, 2-ESTIMATES takes into consideration disagreeing sources for every data item while computing the value confidence. It starts by initializing source truthworthiness and iteratively computes the value confidence as a function of the confidence of all values and normalized as a function of \(\Theta\). Then, it computes the source truthworthiness as a function of the value confidence in \(\Theta\) as a function of the confidence of all values for every data item while computing the value confidence. It starts by initializing source truthworthiness and iteratively computes the source truthworthiness as a function of the value confidence in \(\Theta\) as a function of the confidence of all values for every data item while computing the value confidence.

3-ESTIMATES in Algorithm 2.4 uses a third parameter beside \(T_s\) and \(C_v\): the value error factor, \(\epsilon_v\). Then, for each value, the algorithm computes the value confidence in \(\Theta\) as a function of the value error factor and the truthworthiness of each source providing the value, as well as the truthworthiness of the sources claiming other values. The confidence is normalized and used to compute a new value confidence for each of the disagreeing sources claiming different values.

As in COSINE, 2-ESTIMATES takes into consideration disagreeing sources for every data item while computing the value confidence. It starts by initializing source truthworthiness and iteratively computes the value confidence as a function of the confidence of all values for every data item while computing the value confidence.
some cases, we used the same convergence test as TRUTHFINDER with \( \delta = .001 \).

**Parameter Setting.** Information corroboration algorithms include four parameters to be initialized: \( T_{\epsilon}, \eta \) for Cosine, \( \lambda \), and \( \epsilon_s \). We initialize \( T_{\epsilon} = .8 \). For Cosine, we set \( \eta = 0.2 \) since it maximizes the precision. In our parameterization analysis on the Book data set, we found unstable results for 3-Estimates from one execution to another giving different results for precision, accuracy, and recall for certain values of \( \lambda \). As shown in the table, based on 100 runs with \( \lambda = .8 \), the 95% confidence interval of precision varies from .9214 to .9587.

| Parameter | Value | Precision | Stability |
|-----------|-------|-----------|-----------|
| \( T_{\epsilon} \) | From 0 to .99 | Max (.9805) for \( T_{\epsilon} = .8 \) | stable |
| \( T_{\epsilon} \) | From .1 to 1 | Min (.6647) for \( T_{\epsilon} = .7 \) | stable |
| \( \epsilon_s \) | From 1 to .9 | Max (.9924) for \( \epsilon_s = .8 \) | stable |

Finally, for the Book data set, we select \( T_{\epsilon} = .8, \eta = .2 \) for Cosine, \( \lambda = .5 \), and \( \epsilon_s = .4 \) for 3-Estimates.

### 2.3 Latent Truth Model

Latent Truth Model (LTM) proposed in 2012 by Zhao et al. [13] uses Bayesian networks for estimating the truth. LTM has two important assumptions on the format of the data sets it can handle: (1) the data set should contain only one attribute with atomic values and (2) LTM can handle multiple true values for the same data item. For example, in the case of the Book data set where a list of authors provided by a source \( s \) is \((AuthorA, AuthorB)\), LTM actually takes as input two claims from \( s \), each one having an atomic value that can be true such as: \((c1, s, ThisBook:AuthorOf, AuthorA)\) and \((c2, s, ThisBook:AuthorOf, AuthorB)\). LTM considers, for each source, its prior probability of true positive and negative errors, noted \((\alpha_{t1}, \alpha_{t0})\) as source sensitivity, as well as its prior probability of false positive and negative errors, noted \((\alpha_{f1}, \alpha_{f0})\) as source specificity. Finally, values with confidence higher than .5 are considered to be true, thus, for some data item, LTM may not detect any true value.

**Algorithm.** In Algorithm 2.5, LTM maintains four counters for each source, noted \( n_{x,t,s} \), where \( t \) is the Boolean truth label for each value \( v \), and \( \alpha_s \) is whether value \( v \) is actually claimed by the source or not. LTM first initializes the label of each claim randomly and updates the counters of each source. In each iteration, LTM samples each truth label from its distribution conditioned on all other truth labels, and the source counters are updated accordingly. LTM uses a collapsed Gibbs sampling process with \( K \), the number of iterations required to define the sample size as \((K – burnin)/n\). Then, LTM updates the values truth probability in \( \Theta \) by discarding the first set of samples \((burnin)\) parameter and, for every \( n \) samples in the remainder \((thin)\), LTM computes the average to prevent correlation between adjacent samples. Since LTM relies on the random initialization of the truth labels, as well as random sampling, we can not report the precision of one single run. In the original paper, average precision over 10 runs was reported. In our experiment, we reported the average precision over 100 runs because we observed fluctuating results with wide standard deviations over 10 runs. LTM does not compute source truthworthiness which gives an advantage in terms of execution time.

**Parameter Setting.** Nine parameters have to be set in LTM: \((K, burnin, thin)\); the collapsed Gibbs sampling process parameters, \( \alpha = (\alpha_{t1}, \alpha_{t0}, \alpha_{f1}, \alpha_{f0}) \), the prior true/false positive/negative claim counts for the sources, and \( \beta = (\beta_1, \beta_0) \), the prior true and
false counts for the data item-value pairs. We study the values proposed by the authors for all parameters on the Book data set: varying one parameter and fixing the others successively and we observe: (1) No significant changes in the precision of LTM, neither for \((K, burnin, thin) = (50, 10, 1), (500, 100, 9)\) or \((2000, 100, 9)\) nor for \(\beta = (1, .1)\) or \((.5, .5)\). (2) For high \(\alpha_{0.1}\) and \(\alpha_{1.0}\) (.7 to .9) and low \(\alpha_{0.0}\) and \(\alpha_{1.1}\) (.1 to .3), the precision algorithm was low with high standard deviation (±0.32 in average) and minimal precision in the 95\% confidence interval over 100 runs. We did not consider this parameter setting \(s\) because of too high variability of precision. (3) For the remaining permutations of \(\alpha_{1.1}, \alpha_{1.0}, \alpha_{0.1}\), and \(\alpha_{0.0}\), LTM reaches stability in precision for 100 runs with small 95\% confidence intervals (.002 in average) as follows.

| \((K, burnin, thin)\) | \([\alpha_{1.1}, \alpha_{1.0}, \alpha_{0.1}, \alpha_{0.0}]\) | Precision in 95\% CI |
|----------------------|--------------------------------|---------------------|
| (50, 10, 1)          | \((1, .1, .1, .5)\)          | Max [.8556; .8580]  |
|                      | \((1, .1, .9, .5)\)          | Max [.8585; .8585]  |
| (500, 100, 9)        | \((1, .9, .1, .5)\)          | Min [.8581; .7953]  |
|                      | \((1, .9, .9, .5)\)          | Min [.8536; .8712]  |
|                      | \((.5, .5)\)                 | [8576; 8601]        |

Finally, we select \((K, burnin, thin) = (500, 100, 9)\), \(\alpha = (.9, .1, .9, .1)\) and \(\beta = (.1, .1)\) to get maximal precision average over 100 runs on the Book data set.

2.4 Maximum Likelihood Estimation

Maximum Likelihood Estimation (MLE) proposed in 2012 by Wang et al. in [16] is based on the Expectation Maximization (EM) algorithm to quantify the reliability of sources and the correctness of their observations. MLE only deals with Boolean positive observations (e.g., data items such as thisPerson-hasKids with True or False value). Negative observations are ignored. To be able to test MLE on the Book data set, we reformatted every claim such as \((c1, s, ThisBook:AuthorOf, (AuthorA, AuthorB))\) such as two claims: \((c1, s, ThisBook:AuthorOf:AuthorA:True)\) and \((c2, s, ThisBook:AuthorOf:AuthorB:True)\).

Algorithm. In Algorithm 2.6, MLE starts with initializing the sources’ parameters: \(a(s)\), the probability that source \(s\) reports a value to be true when its intended true and \(b(s)\), the probability that \(s\) reports a value to be true when it is in reality false (similar to source sensitivity \(\alpha_{1.1}\) and \(\alpha_{1.0}\) in LTM). In the Expectation step, MLE iteratively computes the conditional probability of a value \(v\) to be true based on its source probabilities \(a(s)\) and \(b(s)\), and on the probabilities of the sources not providing \(v\) \((\forall s \in S)\). Then, it iteratively computes the confidence of each value in \(\Theta\). In the Maximization step, MLE updates the sources’ probabilities \(a(s)\) and \(b(s)\) in \(\Theta\). The Expectation-Maximization steps are repeated until convergence of both \(a(s)\) and \(b(s)\). An important observation of MLE algorithm is when the number of sources tends to be very large, source probabilities tend to zero and \(C\) tends to 0/0. MLE can not be used with a large number of sources (> 5,000).

Parameter Setting. Two parameters are needed in MLE: \(r\) and \(\beta_1\) to compute the initial parameters of the sources, \(a(s)\) and \(b(s)\). \(\beta_1\) is the overall prior truth probability of the claims (similarly to LTM). \(r\) is the probability that a source provides a value for all data items. In its original paper, MLE is tested on a synthetic data set with no indication on how to set these parameters. So, for the Book data set, we successively vary \(r\) and \(\beta_1\) using a uniform constant value for all sources parameters initialization.

| Fixed Values | Variables | Precision, Accuracy, Recall |
|--------------|-----------|----------------------------|
| \(r = .5\) for all sources | \(\beta_1\) from 1.0 to .9 | All equal to 1 for \(\beta_1 = .5\) |
| \(\beta_1 = .5\) | \(r\) from 1.0 to .9 | All equal to 1 for \(r = .5\) |

Finally, we select \(\beta_1 = .5\) and \(r = .5\) uniformly constant for all sources to get precision, accuracy, and recall equal to 1.

2.5 Source Dependence in Truth Discovery

DEPEN proposed in 2009 by Dong et al. [5] and further extended in [6] is the first Bayesian truth detection model that takes into consideration the copying relationships between sources. DEPEN penalizes the vote count of a source if the source is detected to be a copier of another source. DEPEN is presented with 4 extensions in its original paper [5]. Our study focuses on DEPEN, ACCU, ACCUSIM, and ACCU-NODEP: ACCU extends DEPEN model and relaxes the assumption that the sources have the same accuracy and for each data item, all independent sources have no longer the same probability of providing a true value. ACCUSIM extends ACCU to take into account value similarity, and ACCU-NODEP assumes that all sources are independent.

Algorithm. Algorithm 2.7 presents DEPEN and starts by initializing all sources’ truthworthiness to .8. For every data item, it selects the true value by majority voting, and computes the dependence between sources with \(CompDepen(s, s, \alpha, \beta, n)\) function where \(\alpha\) is the a priori probability that \(s\) and \(s_j\) are dependent, and \(n\) is the number of false values per data item. To iteratively compute the value confidence in \(\Theta\), the sources claiming the considered value are first ordered by their dependence probabilities with \(orderDepn\). Then, each source’s \(voteCount\) is computed in a way that minimizes the vote if the source is dependent on other sources in Pre, the list of ranked sources, such as \(voteCount = \prod_{s, c(v)}(1 - c.depen(s, s_j))\), with \(c\) the probability that a value provided by a copier is copied. \(voteCount\) is then weighted by \(n_{score}^v\), the source’s score to compute the value confidence. Source truthworthiness is computed iteratively in \(\Theta\) as a function of the confidence of all values claimed by the sources. True values are expected to be the values with the highest confidence. In ACCU-NODEP, no dependence computation is needed, and \(voteCount\) is always 1. In ACCU and ACCUSIM, the algorithm computes value confidence with \(n_{score}^v = \ln(\alpha T_j / (1 - T_j))\), whereas in DEPEN, \(n_{score}^v = 1\). In ACCUSIM, the value similarity is considered for the confidence computation in each iteration and, \(\rho \sum_{v \in \Theta} C_r(v, \cdot, \cdot, \cdot) = v\) is added to \(C_r\) (similarly to \(TRUTHFINDER\)). It is worth noticing that DEPEN model and its extensions estimate the source \(voteCount\) for a given value based on ordering the sources by decreasing dependence probability. This ordering could be different from one run to the next, because two sources with the same dependence probabilities could appear in different positions. We observed that this dependence-based ordering introduced small fluctuations of the quality metrics for 20 executions of the models with the same parameterization on the Book data set. In particular, we observe DEPEN precision (.9814 ± .0002), ACCU precision (.9741 ± .0061) and ACCUSIM precision (.9413 ± .0051). To mitigate this problem, we decided to use the lexical ordering rather than the dependence probability-based ordering of the sources. This slightly improves the quality of the models by +.02 (DEPEN_lex, precision .9814, ACCU_lex precision .9809, and ACCUSIM_lex precision .9073) for the Book data set and it also improves the stability of the results that remain constant from one run to another.

Parameter Setting. Fixing \(\rho = .5\) and \(n = 100\), we study various parameterization setting reported in the table.

| Fixed Values | Variables | Precision |
|--------------|-----------|-----------|
| \(\alpha = .2, c = .8\) | \(T_1\) from 0.0 to .99 | DEPEN: Max [.9814] for \(T_1 = .8\) |
| \(\alpha = .2, c = .8\) | \(T_1\) from 0.0 to .95 | DEPEN: Max [.9814] for \(T_1 = .8\) |
| \(\alpha = .2, c = .8\) | \(c\) from .05 to .25 | DEPEN & ACCU-NODEP: Max [.9814] for \(c = .8\), ACCU: Max [.9809] for \(c = .1\), ACCUSIM: Max [.973] for \(c = .05\) |

Finally, we select \(\alpha = .2\), \(T_1 = .8\), \(c = .8\) for DEPEN and ACCU-NODEP and \(c = .1\) for ACCU and \(c = .05\) for ACCUSIM.
2.6 Latent Credibility Analysis

Latent Credibility Analysis (LCA) proposed in 2013 by Pasternack and Roth in [14] is a probabilistic model that also uses the Expectation Maximization algorithm to calculate the probability of a claim being true, by grouping claims related to the same data items into mutual exclusion sets where only one true claim exists. Four LCA variants have been proposed in the original paper. In our study, we focus on: SIMPLELCA and GUESSLCA. Both algorithms require \( W \), a confidence matrix that expresses the confidence of each source \( s \) in its assertions for each data item \( d \) with \( w_{sd} \) elements. Typically, \( w_{sd} = 1 \) if the source \( s \) asserts with full certainty a value for \( d \), or 0 if the source says nothing about \( d \). SIMPLELCA is the simplest and straightforward approach where each source has a probability of being honest and all sources are considered to be independent. In the Expectation step of Algorithm 2.8, SIMPLELCA iteratively computes the confidence of each value in \( \Theta \) with \( \beta_i \), the prior truth probability of the claimed value (similarly to LTM and MLE). Then, SIMPLELCA iteratively computes the source truthworthiness in the Maximization step in \( \Theta \), in the same way as TRUTHFINDER, averaging the confidence of the values that the source provides weighted by the certainty of the source on each of its assertions.

GUESSLCA extends SIMPLELCA with the probability of a source guessing when being honest, noted \( p_w \). GUESSLCA rewards hard claims with correct truth label and penalizes getting easy claims wrong. It also assumes that no source will do worse than guessing, which is a significant advantage over other methods for pessimistic scenarios, as we will see in the next section. \( p_w \) can be uniformly constant or set to the distribution of sources asserting the claims for a given data item. The main assumption is that a guessing source chooses randomly according to the distribution of votes. In Algorithm 2.9, the confidence of value \( v \) is computed in \( \Theta \) as the product of \( \beta_i \) with the probability that the sources assert \( v \) as a true claim knowing the truth and also guessing as \( T_{\delta} + (1 - T_{\delta})p_{ws} \), and the probability of not knowing the truth and guessing as \( (1 - T_{\delta})p_{ws} \) for \( s' \in S \) to the power \( w_{sd} \), the source’s confidence in the value it claims for each data item. Source truthworthiness is computed in \( \Theta \). Convergence test for the LCA models was not explicitly mentioned in the original paper, only the required number of iterations was stated to be 50 iterations. In our experiments, we use the same convergence test as for the other methods: the difference of cosine similarity of both source truthworthiness and value confidence between two iterations, to be less than or equal to \( \delta = .001 \).

Parameter Setting. Similarly to LTM and MLE, LCA models require, as input parameters, the prior truth probability \( \beta_i \) and the honesty of the sources, noted \( T_{\delta} \) in our notation. We tested various parameter settings on the Book data set. We finally select \( \beta_i = .5 \) and \( T_{\delta} = .8 \) for maximizing precision of LCA models.

| Fixed Values | Variables | Precision |
|--------------|----------|-----------|
| \( T_{\delta} = .8 \) | \( \beta_i \) from 1 to 1 | GUESSLCA: Max (.9806) and SIMPLERCA: Max (.9758) for \( \beta_i = .5 \) |

2.7 Conclusions on Parameter Setting

The main conclusions of our parameterization study are mainly related to the modeling assumptions, the usability of the algorithms, and the repeatability of the results.

(1) Modeling Assumptions. First, all methods rely on various assumptions that have direct impact on the quality and applicability of the models: (A1)– a source is supposed to contribute uniformly to all the claims it expresses. In every algorithm, \( T_{\delta} \) and \( \beta_i \) probabilities are uniformly distributed either across all sources or all claims. As a consequence, the models do not explicitly consider both the expertise of certain sources (which can be either general or more specialized on particular topics or claims) and the hardness of certain claims (except 3-ESTIMATES or GUESSLCA). Only LCA models express the degree of certainty some sources may have on their claimed values. (A2)– Concerning the type of the claims as inputs of the algorithms: all claims are assumed to be positive and directly attributed to a source, i.e., cases such as “S claims that A is false”, or “S does not claim A is true”, or “According to S, A is claims that A is true” are not considered in the models we studied. For LTM, claim structure is restricted to single-property assertions and MLE requires Boolean values to be comparable with other algorithms. This requisites may cause information omissions or distortions due to data formatting. Except for LTM, (A3)– all models consider that exactly one of the claims for a given data item has to be true. Thus, multiple views of the truth are not modeled. None of the models penalize the sources claiming multiple values (similar or distinct) for the same data item. Except DEPEN and its recent extensions in [14] , (A4)– sources and claims are assumed to be independent, as well as real-world objects they refer to.

(2) Usability. Our main observation is that all models require complex, ad hoc parameter setting and tuning depending on the considered data set. We observe that the parameter settings we selected to maximize precision for the Book data set do not maximize precision of the algorithms when they are applied to other data sets.

### Algorithm 2.8: SIMPLELCA(\( S, D, V, W, \beta_i, \delta \))

\[
\forall v \in S : T_{\delta} \rightarrow 8
\]

repeat

for each \( d \in D \)

Expectation step:

\[
C_{d_{\text{new}}} \leftarrow 0
\]

for each \( v \in V_d \)

\[
C_v \leftarrow \beta_v \prod_{v' \in S_v} t_{v'v}^{w_{vd} - 1} \cdot \prod_{r \in R_v}(1 - T_{\delta})p_{vs}^{w_{vd} - 1} \Theta
\]

\[
C_{d_{\text{new}}} \leftarrow C_{d_{\text{new}}} + C_v
\]

Maximization step:

for each \( s \in S \)

\[
t_s = \sum_{v \in V_s} C_v w_{sd} \sum_{v' \in V_s} w_{vd} \Theta
\]

until Convergence(\( T_{\delta}, \delta \))

for each \( d \in D \):

\[
\text{trueValue}(d) \leftarrow \arg \max_{v \in V_d} (C_v)
\]

### Algorithm 2.9: GUESSLCA(\( S, D, V, W, \beta_i, \delta \))

\[
\forall v : p_w \leftarrow |S|/|S| \leftarrow \bar{S}|S|
\]

\[
\forall v \in S : T_{\delta} \rightarrow 8
\]

repeat

for each \( d \in D \)

Expectation step:

\[
C_{d_{\text{new}}} \leftarrow 0
\]

for each \( v \in V_d \)

\[
C_v \leftarrow \beta_v \prod_{v' \in S_v} t_{v'v}^{w_{vd} - 1} \cdot \prod_{r \in R_v}(1 - T_{\delta})p_{vs}^{w_{vd} - 1} \Theta
\]

\[
C_{d_{\text{new}}} \leftarrow C_{d_{\text{new}}} + C_v
\]

Maximization step:

for each \( s \in S \)

\[
t_s = (\sum_{v \in V_s} C_v + \sum_{v \in V_s} \beta_v/\sum_{v} \beta_v C_v (\sum_{v \in V_s} w_{sd})) \Theta
\]

until Convergence(\( T_{\delta}, \delta \))

for each \( d \in D \):

\[
\text{trueValue}(d) \leftarrow \arg \max_{v \in V_d} (C_v)
\]
The gold standard of the Book data set represents 7.91% of the data set. We argue that it is not representative enough for a systematic, rigorous comparison of the algorithms’ quality. Optimal parameterization of the algorithms based on a real-world data set is jeopardized when the ground truth is partial and reduced to samples of the real-world data set. This problem actually motivated us to develop a framework and a synthetic data set generator to systematically control the complete ground truth distribution, as we will describe in the next section.

3. COMPARATIVE EXPERIMENTS
A first set of experiments has been conducted over synthetic data sets to evaluate the quality (Section 3.1) and scalability of each algorithm (Section 3.2). A second set of experiments has been conducted over five real-world data sets to report the running time, number of iterations, and memory usage in addition to each algorithm’s quality metrics (Section 3.3).

3.1 Experiments on Synthetic Data
First, we generated synthetic data to evaluate the algorithms under a wide range of truth discovery scenarios. Table 4 summarizes the parameters we used to control the characteristics of the synthetic data set generation. In particular, we control the percentage and distribution of data items for which a source claims a value (Cov) and the number and distribution model of distinct values per data item (Conf). We also control the percentage and distribution model of true positive values per source (GT). This actually constitutes the ground truth we used for computing the quality metrics of the algorithms. Finally, we ran our experiments on 9,120 data sets generated with $|S| = 50$ and $|D| = 1,000$: 10 data sets for each of the $(3 \times 8 \times 2 \times 19)$ possible configurations presented in Table 4. Due to space limitation, only 8 configurations are presented in this section and in Fig. 1 – see [2] for more detailed and experimental results. Dependence between sources and value similarity were not the scope of this study since these aspects are considered only by TRUTHFINDER, ACCUSIM, and DEPEN. In the set of experiments on synthetic data, our objective is to identify the data set characteristics that have an impact on the quality of the algorithms, in particular: (1) the number of values claimed by the sources; (2) the number and distribution of distinct values per source; and (3) the type of distribution of true positive claims per data item. Fig. 1 shows the algorithms’ precision average over 10 data sets for each configuration with an increasing number of distinct values per data item (from 2 to 20).

3.1.1 Source Coverage
We compare the quality of the truth discovery models for three types of source coverage: Uniform U25, U75, and Exponential. Uniform source coverage corresponds to the case where all the

| Control Parameter                        | Value                | Description                                                                 |
|------------------------------------------|----------------------|-----------------------------------------------------------------------------|
| Number of Sources (S)                    | 50, 1,000 to 10,000  | The number of sources providing claims. $|S| = 50$ in Section 3.1 and from 1,000 to 10,000 in Section 3.2.            |
| Number of Data Items (D)                 | 1,000, 100 to 10,000 | The number of data items, i.e., pairs of (object, attribute) with claimed values: $|D| = 1,000$ in Section 3.1 and from 100 to 10,000 in Section 3.2. |
| Source Coverage (Cov)                    | U25, U75 (Uniform)   | The number of values provided by the sources is uniformly distributed on 25% and 75% of the data items. |
|                                         | E (Exponential)      | The number of values provided by the sources is exponentially distributed across the data items. |
| Ground Truth Distribution per Source (GT)| K (Random)           | The number of true positive claims per source is random.                    |
|                                         | U25, U75 (Uniform)   | Each source provides the same number of true positive claims.               |
|                                         | FF (Fully Optimistic)| 80% of the sources provide always false claims and 20% of the sources provide always true positive claims. |
|                                         | FO (Fully Pessimistic)| 80% of the sources provide always true positive claims and 20% of the sources provide always false claims. |
|                                         | BPF (80-Pessimistic)| 80% of the sources provide 20% true positive claims.                       |
|                                         | BPO (80-Optimistic)  | 80% of the sources provide 80% true positive claims.                       |
|                                         | E (Exponential)      | The number of true positive claims provided by the sources is exponentially distributed. |
| Distinct Value Distribution per Data Item| U (Uniform)          | All data items have the same number of distinct values claimed by the set of sources. |
|                                         | E (Exponential)      | Each data item has a number of distinct values that is exponentially distributed. |
| Number of Distinct Values                | 2, 20                 | The number of distinct values per data item.                               |

Table 4. Parameters for Synthetic Data Sets Generation for Configuring a Truth Discovery Scenario

The execution time is the total time to compute the truth discovery results, including initialization, convergence, eventual normalization, computation of source truthworthiness and value confidence. We re-implemented all the algorithms in Java 7 under a common implementation framework to test as accurately as possible their relative quality, performance, and behavior. Source codes are available in [2]. We ran experiments on 3 PCs with Intel Core i7-2600 processor (3.40GHz×8, 32GB).

The number of true positive claims per source, $TP$, and the number of true negative claims per source, $TN$, are used to compute the quality metrics

$$Precision = \frac{TP}{TP + FP} \quad \text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Recall = \frac{TP}{TP + FN} \quad \text{Specificity} = \frac{TN}{FP + TN}$$

with

| Ground Truth / Gold Standard | True | False |
|-----------------------------|------|-------|
| True Positive (TP)          | True Positive (TP) | False Negative (FN) |
| False                       | False Positive (FP) | True Negative (TN) |

The number of true positive claims per source, $TP$, and the number of true negative claims per source, $TN$, are used to compute the quality metrics

$$Precision = \frac{TP}{TP + FP} \quad \text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Recall = \frac{TP}{TP + FN} \quad \text{Specificity} = \frac{TN}{FP + TN}$$

with

| Ground Truth / Gold Standard | True | False |
|-----------------------------|------|-------|
| True Positive (TP)          | True Positive (TP) | False Negative (FN) |
| False                       | False Positive (FP) | True Negative (TN) |
sources provide claims for respectively 25% or 75% of the data items. Exponential source coverage corresponds to a more realistic case where few sources provide claims for most of the data items and the majority of the remaining sources only covers very few data items. We observe that increasing the source coverage from $U_{25}$ to $U_{75}$ generally increases the precision of all algorithms and fewer distinct values are needed to reach the same precision, except in two cases: (1) When the distribution of true positive claims is randomly distributed across the sources ($GT=R$), increasing the number of data items per source does not change the precision of any method; algorithms' precision for $Conv=U_{25}$ and $Conv=U_{75}$ are identical and merged in Fig. 1(a) irrespectively of the type of conflict distribution. Precision of all methods does not differ by more than 2% and decreases in both cases, $Conf=U$ and $Conf=E$. (2) When the distribution of true positive claims is exponentially distributed across the sources ($GT=E$), precision of all methods remains constant and close to zero even when increasing the source coverage and the number of conflicts (Fig. 1(f)).

### 3.1.2 Conflict Distribution

In the case of exponentially distributed conflicts over the data items (Fig. 1(a) for $Conf=E$ line), many data items have very few conflicts, whereas few data items have lots of conflicts. Exponential conflict distribution is interesting and realistic since some data items may be more controversial and have more conflicts than others. The majority of the claims in agreement generally help all the models to reach a precision greater than .50 in the worst cases, e.g., when the sources randomly tell the truth among lots of conflicts. In that case, for $Conf=E$ and $GT=R$, we observe precision decreasing from .75 to .525 for all methods. For $Conf=U$ and $GT=R$ in Fig. 1(a), all algorithms behave identically with decreasing precision below .50 (i.e., worse than random guessing). Comparing Fig. 1(c) and (g), we observe two effects when the conflict distribution

\[ \forall i = 0, \ldots, |S| - 1, \text{Conv}_i = 1 + (|D| - 1) \frac{N_{i-1}^{(i)} - 1}{|D| - 1} \]

\[ \forall i \in 1, \ldots, |D|, N_{i} = \max(N_{i-1} + 1) \]

type changes from uniform to exponential: (1) precision is lifted up above .50 irrespectively of the source coverage and (2) precision range becomes more compact within .2 precision interval.

### 3.1.3 Ground Truth Distribution

Finally, we control the distribution of true positive claims among the set of claims provided by each source and we generate synthetic data sets corresponding to 7 scenarios in addition to random ($GT=R$) such as: uniform ($U_{25}$, $U_{75}$), fully pessimistic (FP), 80-pessimistic (80P), fully optimistic (FO), 80-optimistic (80O), and exponential (E) as defined in Table 4.

#### Random Ground Truth Distribution

As mentioned earlier, when true positive claims are randomly distributed across the sources, we observe that (1) none of the methods can be reliable when conflicts are uniformly distributed (decreasing precision below .50 in Fig 1(a) for $Conf=U$), and (2) increasing the source coverage or changing the distribution of conflicts per source (from uniform to exponential) does not improve the precision of any method, (3) algorithms' precision does not differ by 2% and decreases when the number of conflicts increases.

#### Uniform Ground Truth Distribution

For exponential source coverage and cases where the sources are equally saying the truth for 25% of the values they claim ($GT=U_{25}$) in Fig. 1(b), the precision of the methods increases with the number of conflicts. This trend is even more significant when the source coverage increases from uniform $U_{25}$ to $U_{75}$ since increasing the source coverage reduces the number of conflicts needed for comparable precision. 3-ESTIMATES has unstable results due to the instability of $\lambda$ parameter setting. In Fig. 1(b), for $GT=U_{25}$ with exponential source coverage and exponential conflict distribution, all methods behave identically and reach .75 in the best case of 20 distinct values exponentially distributed across the data items. In Fig. 1(e), when the sources are almost always, equally saying the truth ($GT=U_{75}$), precision of all methods does not differ more than 2% (except 3-ESTIMATES) and reaches 1 irrespectively of the distribution type or number of conflicts.

#### Pessimistic Scenarios

In $GT=FP$ scenarios of Fig. 1(e),(g), and (h), 80% of the sources always provide false claims and 20%

![Figure 1: Precision Average for Various Truth Discovery Scenarios with $|S| = 50$ and $|D| = 1,000$](image-url)
always provide true claims. We observe that, for few conflicts – i.e., less than 8 distinct values per data item for \( \text{Cov}=\text{E} \) in Fig. 1(c), or 4 for \( \text{Cov}=\text{U25} \) in Fig. 1(h) – most of the methods perform worse than random guessing with precision from .1 to .4 except \text{Cosine} which reaches a precision peak of .7 in Fig.1(c) for 4 distinct values for \( \text{Cov}=\text{E} \) and .95 precision in Fig. 1(h) for 3 distinct values for \( \text{Cov}=\text{U25} \). In these two cases of source coverage, \text{S}\text{IM}PLE\text{LCA} outperforms all methods from 4 (for \( \text{Cov}=\text{E} \)) or 8 (for \( \text{Cov}=\text{U25} \)) distinct values. In Fig. 1(g), for exponential source coverage and exponential conflict distribution, precision range of the methods increases with the number of conflicts from .5 to .72. And the compacting & lifting up effect of exponential conflict distribution on the precision of the methods is confirmed in \text{GT}=\text{FP} \) scenario of Fig. 1(g). In \text{FP} and \text{80-P} scenarios, the ordering of the methods based on precision remains constant: \text{S}\text{IM}PLE\text{LCA} > \text{A}\text{CC}\text{U}\text{NODEP} > \text{G}\text{UESSLCA} > \text{TRUTHFINDER} > \text{2-ESTIMATES} > \text{VOTING} > \text{COSINE}. Precision of \( 3\)-\text{ESTIMATES} oscillates around or below .50 in \text{FP} and \text{80-P} scenarios in Fig. 1(c) and (d). \text{Cosine}, \text{Voting}, \text{2-ESTIMATES}, \text{TRUTHFINDER}, and \text{GUESSLCA} behave similarly with close precision values. \text{DEPEN} and its variants (except \text{ACC}U\text{NO}DEP) are deeply affected by random source dependence and have very low precision although increasing with the number of conflicts in Fig. 1(c) and (h). For fully pessimistic scenarios with few conflicts – either less than 4 distinct values uniformly distributed or less than 8 distinct values exponentially distributed– none of the methods has precision significantly better than random guessing. In the \text{80-Pessimistic} scenario with exponential source coverage in Fig. 1(d), \text{S}\text{IM}PLE\text{LCA} maintains precision greater than .55 from 4 distinct values, whereas the other methods need at least 7 distinct values to reach .50 precision.

**Optimistic Scenarios.** In \text{GT}=\text{UF75} scenario of Fig. 1(e) with exponential source coverage and exponential conflict distribution, we observe that all methods have very similar, high precision close to 1 (except \text{3-ESTIMATES}). We observe the same behavior with quasi-identical curves for \text{GT}=\text{FO} and \text{GT}=\text{800} (see Fig. 2 for detail). In the case of optimistic scenarios with exponential source coverage and exponential conflict distribution, all methods do not differ in precision by 1% and excel with precision close to 1 except \text{3-ESTIMATES} which oscillates from 9 to 1.

**Exponential Ground Truth Distribution.** This case represents the situation where one source always lies and one source always tells the truth for all the data items it covers and the remaining sources range from 1% to 99% of claims they provide being true.

In this case represented Fig. 1(f), none of the methods is reliable even when the source coverage increases from \( \text{U25} \) to \( \text{U75} \). None of the existing methods can cope with a wide, continuous spectrum of source truthworthiness irrespectively of the source coverage and conflict distribution, which is somehow a bad news because we can expect, in practice, that the variety of online sources may correspond to a wide, potentially continuous range of source truthworthiness and exponential distribution of the true positive claims per source.

### 3.2 Scalability Experiments

To characterize the different algorithms’ behavior in terms of scalability, we evaluate them using large synthetic data sets. Each reported time is the average of 10 executions over 10 different data sets of the same size and same configuration as \( \text{Cov}=\text{U25}=\text{Conf}=\text{U}=\text{GT}=\text{R} \) for which all methods obtain the same precision. We increased the number of data items from 100 to 10,000 and the number of sources from 1,000 to 10,000. The experiment with 10,000 sources and 10,000 data items (i.e., 100 millions claims) exceeded our main memory capacity and is not reported.

Let \( \text{Scat} \) be the case with \( |S|=1,000 \) sources and \( |D|=10,000 \) data items. Let \( \text{Scat} \) be the case with \( |S|=10,000 \) sources and \( |D|=1,000 \) data items. Fig. 2 shows two types of runtime behavior. Fig. 2(a) presents the models including source dependence computation (< 6,000 seconds). Fig. 2(b) presents the other algorithms (< 16 seconds). \text{LTM} lies between these two types of behavior with 256 seconds for \( \text{Scat} \) and twice more (496 seconds) for \( \text{Scat} \) and it is plotted in Fig 2(a). For a large number of sources (\( |S|>5,000 \)), the time for \text{MLE} and \text{LCA} models could not be reported since the algorithms obtained 0/0 undetermined form (\text{NaN}) for the value confidence and source truthworthiness computation. In Fig. 2(a), \text{DEPEN}, \text{ACC}U, and \text{ACC}US\text{IM} exhibit similar performance of linear scaling on the number of data items for 1,000 sources (\( \text{Scat} \) in solid lines), but quadratic scaling on the number of sources (\( \text{Scat} \) in dashed lines): from 5,492 seconds for \text{DEPEN} to 5,788 seconds for \text{ACCUSIM}. Fig. 2(b) shows the fastest algorithms with runtime below 12 seconds for \( \text{Scat} \) and below 16 seconds for \( \text{Scat} \). For \( \text{Scat} \), \text{MLE} performs faster than \text{COSINE} and \text{LCA} models. 2- and 3-\text{ESTIMATES} are the slowest but they maintain almost the same execution times in the two cases, slightly lower for \( \text{Scat} \). For \( \text{Scat} \), \text{ACC}U\text{NO}DEP is the slowest algorithm after 2- and 3-\text{ESTIMATES}.

These results corroborate the time complexity analysis given in Table 3. Finally, Fig. 2(b) demonstrates the efficiency of \text{MAJORITYVOTING} and \text{TRUTHFINDER} in both cases: 438 and 528 milliseconds for \text{MAJORITYVOTING} for \( \text{Scat} \) and \( \text{Scat} \) respectively and 1.912 seconds for \text{TRUTHFINDER} in both cases.
From our scalability experiments, we can conclude that **MAJORITY VOTING** and **TRUTHFINDER** perform best for truth discovery on our synthetic data sets. This concerns both the scaling on the number of data items and claims, as well as the scaling on the memory of the number of sources.

### 3.3 Experiments on Real-World Data

In this set of experiments, our goal is to compare quality metrics and performance of the algorithms on five real-world data sets. Table 5 shows the characteristics of these data sets and provides the quality metrics, number of iterations, execution time, and memory.
usage (with EL when exceeding the time or memory capacity limits of the experiments and NA when value confidence calculation underflows to zero and source truthworthiness computation produces NaN result). In Table 5, red color indicates the best quality metrics, yellow highlight the winner based on maximal precision, green indicates the fastest execution and lowest memory consumption, whereas blue indicates the worst quality metrics, the slowest execution and the highest memory consumption. Results of MAJORITYVOTING as the baseline are in black bold. Fig. 3 shows the distributions of claims per source (black line), true positive claims per source (red line), and distinct values per data items (green line).

**Book.** The Book data set from [4] consists of 33,235 claims on the author names of 1,263 books by 877 book seller sources. The gold standard consists of 100 randomly sampled books for which the book covers were manually verified by the authors of [4] representing 100/1263 = 7.91% of the complete ground truth. Distributions are illustrated in Fig. 3(a). A version of the Book data set has been formatted so that MLE could be compared. MLE reaches precision 1, accuracy 1, recall 1, and null specificity in 2 iterations and 661 milliseconds. It outperforms all methods including MAJORITYVOTING when we compare the gain in precision versus the loss in execution time. 3-ESTIMATES is ranked in the second position for precision but first for specificity: this can be explained by the optimal tuning of its parameters for the Book data set. DEPEN models have the third position in terms of precision but expose prohibitive runtime due to source dependence computation. Even after 500 iterations, LTM has the lowest precision.

**Flight.** The Flight data set from [11] consists of 2,864,985 claims from 38 sources on 34,652 flights for 6 attributes with distributions illustrated in Fig. 3(b). The gold standard contained 16,134 true values which represents 7.76% of the complete ground truth. ACCU outperforms all methods for all quality metrics with the highest memory consumption but a reasonable runtime for 3 iterations compared to the average and the worse case of GUESSLCA in terms of time and number of iterations. However, ACCU is about 120 times slower than MAJORITYVOTING for only +0.0906 precision increase.

**Weather.** The Weather data set from [4] consists of 426,360 claims from 18 sources on the Web for 5 attributes on hourly weather predictions for 49 US cities between January and February 2010 (Fig. 3(c)). As gold standard, we used 30,170 claims from AccuWeather Web site which can cover 74.4% of the complete ground truth. ACCUSIM and ACCU are penalized mainly because weather data are very similar by nature and the weight on similarity is misleading: they did not perform better than random guessing when sources make lots of false claims. However, TRUTHFINDER is the winner reaching .6443 precision after 1,238 milliseconds and 2 iterations, only 13 times slower than MAJORITYVOTING with +.0138 precision increase (Table 4). Again GUESSLCA is the slowest almost doubling the average time in 11 iterations.

**Population.** The Population data set from [12] consists of 49,955 claims extracted from Wikipedia edits from 4,264 sources (Fig. 3(d)). The gold standard used by the authors was 301 true values on the population from US Census representing .702% of the complete ground truth. ACCUNODEP outperforms all methods in 411 milliseconds and 4 iterations, 9 times slower than MAJORITYVOTING for +.0332 precision increase. ACCU is the slowest and ACCUSIM has maximal memory consumption due to similarity computation.

**Biography.** We extended the Biography data set extracted from Wikipedia in [12] with 10,862,648 claims over 19,606 people and 9 attributes from 771,132 online sources (Fig. 3(e)). The gold standard consists in 2,626 true values from authoritative sources representing .069% of the complete ground truth. Computing source dependence expose a prohibitive runtime (EL) and confidence computation by LCA models was not feasible (NA). Finally, 2-ESTIMATES has the best quality metrics in only 2 iterations but almost 10 times slower than MAJORITYVOTING for 5 times more memory usage and +.0023 precision increase.

From Fig. 2 and Table 5, we observe that all real-world data sets have exponential source coverage (Cov=E) and exponential distri...
bution of their distinct values (Conf=E). To confront our findings from the experiments on synthetic data, we generate data sets mimicking the characteristics of the Book and Weather data sets in Fig. 4 with the advantage to generate the complete ground truth.

**Optimistic scenarios.** The sources of the Book data set generally have no interest in providing wrong information about their products and we can assume that their underlying ground truth distribution can either be **80-Optimistic** or **Fully Optimistic** in the best case. Fig. 4(a) presents the precision of the algorithms for the Book data set with its original gold standard, as well as the averaged precision over 10 synthetic data sets generated with similar characteristics in terms of numbers of sources and data items for **800 and FO** scenarios with maximum 20 conflicts exponentially distributed. All methods have very high precision for the optimistic scenarios with precision in the following 95% confidence intervals: [.9897; .9934] for **GT=FO** and [.9732; 1] for **GT=800** over the total number of true positive claims we generated. In that case, we can conclude that the results obtained from the synthetic data with complete ground truth corroborate the ones obtained from the gold standard of the Book data set. This gold standard has been carefully selected and we observe that it can be considered as a representative sample of the complete ground truth.

**Pessimistic scenarios.** In Table 5, precision average for the Weather data set is (.6134 ± .0489), computed from a gold standard that was considered as an authoritative source. We generated many data sets with similar characteristics in terms of number of sources and data items, numbers and distributions of claims per source and distinct values per data items for a wide range of pessimistic scenarios. Fig. 4(b) represents the closest precision we could get for **GT=80**. We can observe that precision obtained for the gold standard with 74.4% of the original data set size actually corresponds to the precision we can obtain with synthetic data sets generated for a scenario where 35% of the total number of claims provided by the sources uniformly are true positive claims. This leads us to put into perspective the authoritative nature of AccuWeather source as a gold standard, despite its coverage.

Finally, we observe that none of the considered algorithms has clear benefits over **MAJORITYVOTING** when we compare the gain in precision of the best method (+.0319 ± .0696) versus the loss in runtime (+17.97 ± 58.67) seconds in average for the five real-world data sets. Moreover, experiments on real-world data sets confirm our observations: the algorithms of our study have been originally designed to excel in **optimistic** scenarios with lots of conflicts (from maximum 11 to 60) exponentially distributed across all data items. For data sets where most of the sources provide false claims still with lots of conflicts, the methods precision is relatively low (from .6134 to .7072 in average). The experimental results obtained from real-world data sets corroborate the results we obtained from the experiments on the synthetic data sets and demonstrate that our framework and data set generator can help in cross-checking data set gold standards.

4. CONCLUSIONS

Reimplementing and extensively comparing 12 algorithms for truth discovery from multi-source, conflicting data was a challenging task, mainly due to the problems we faced to set up the experimental framework to compare all methods in a unified and fair way. Even so, we had to omit other existing algorithms related to source trust assessment [17]. Web link analysis [12], and recent work on correlated data [14] and conflict resolution [10]. Our main conclusions are the following: (1) Stability and repeatability of the results are significant issues for LTM and 3-ESTIMATES. Fluctuations of their results are due to randomization in LTM and normalization in 3-ESTIMATES. Multiple executions of these algorithms are required to compute meaningful averages of the quality metrics. We also observed that parameter setting can dramatically impact the quality of these algorithms. (2) When the number of sources exceeds 5,000: LCA and MLE computation is not feasible (0/0) or exceeds the memory capacity limit for DEPEN, ACCU, and ACCUSIM models. (3) All methods do not perform significantly better than random guessing when the data set has few conflicts per data item and a large number of non reliable sources (pessimistic scenarios). (4) Although **MAJORITYVOTING** can be misleading when sources are dependent, it remains the most efficient and scalable for a minor degradation in precision compared to the other methods that are from 9 (TRUTHFINDER) to 120 times (ACCUSIM) slower.

Future work consists of extending this work in a number of fronts. Firstly, we hope that our synthetic data set generation framework can be used and extended for parameter setting, testing and in-depth evaluation of other existing or new algorithms in a variety of truth discovery scenarios (e.g., with controlling source dependence and value similarity). The main advantage of our framework is to control a complete ground truth (usually hard to get with real-world data sets) and mimic real-world truth discovery scenarios. Secondly, we can see many challenging research avenues for the next generation of truth discovery methods: (1) To improve scalability on the number of sources to be applicable to data from social networks and social media, (2) To improve the algorithms’ precision for pessimistic scenarios when most of sources are not reliable and have few conflicting values, (3) To improve the usability and repeatability of the algorithms, either by simplifying the parameterization or combining multiple methods to find optimal parameter setting.

5. REFERENCES

[1] R. Balakrishnan and S. Kambhampati. SourceRank: Relevance and Trust Assessment for Deep Web Sources based on Inter-source Agreement. In WWW, pages 227–236, 2011.

[2] L. Berti-Equille and D. A. Waguih. Truth Discovery Algorithms: An Experimental Evaluation. QCRI Technical Report, May 2014, 2014.

[3] X. Dong, L. Berti-Equille, Y. Hu, and D. Srivastava. SOLOMON: Seeking the Truth Via Copying Detection. PVLDB, 3(2):1617–1620, 2010.

[4] X. L. Dong, L. Berti-Equille, Y. Hu, and D. Srivastava. Global Detection of Complex Copying Relationships Between Sources. Proc. VLDB Endow., 3(3-4):1358–1369, 2010.

[5] X. L. Dong, L. Berti-Equille, and D. Srivastava. Integrating conflicting data: The role of source dependence. PVLDB, 2(1):541–561, 2009.

[6] X. L. Dong, L. Berti-Equille, and D. Srivastava. Truth Discovery and Copying Detection in a Dynamic World. PVLDB, 2(1):562–573, 2009.

[7] X. L. Dong, E. Gabriolovich, G. Heitz, W. Horn, K. Murphy, S. Sun, and W. Zhang. From Data Fusion to Knowledge Fusion. In VLDB, 2014.

[8] A. Galland, S. Abiteboul, A. Marian, and P. Senellart. Corroborating Information from Disagreeing Views. In WSDM, pages 131–140, 2010.

[9] F. Gneosdone, K. Karanasos, Y. Katsis, J. Leblay, I. Manolescu, and S. Zampetakis. Fact Checking and Analyzing the Web. In SIGMOD, pages 997–1000, 2013.

[10] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han. Resolving Conflicts in Heterogeneous Data by Truth Discovery and Source Reliability Estimation. In SIGMOD, 2014.

[11] X. Li, X. L. Dong, K. Lyons, W. Meng, and D. Srivastava. Truth Finding on the Deep Web: Is the Problem Solved? PVLDB, 6(2):97–108, 2012.

[12] J. Pasternack and D. Roth. Knowing what to believe (when you already know something). In COLING ‘10, pages 877–885, 2010.

[13] J. Pasternack and D. Roth. Latent Credibility Analysis. In WWW, pages 1009–1020, 2013.

[14] R. Pochampally, A. D. Sarma, X. L. Dong, A. Meliou, and D. Srivastava. Fusing Data with Correlations. In SIGMOD, 2014.

[15] V. G. V. Ydyuswaram, C. Zhai, and D. Roth. Content-driven truth propagation framework. In KDD, pages 974–982. ACM, 2011.

[16] D. Wang, L. M. Kaplan, H. K. Le, and T. F. Abdelzaher. On Truth Discovery in Social Sensing: a Maximum Likelihood Estimation Approach. In IPSN, pages 233–244, 2012.

[17] X. Yin, J. Han, and P. S. Yu. Truth Discovery with Multiple Conflicting Information Providers on the Web. TKDE, 20(6):796–808, 2008.
B. Zhao, B. I. P. Rubinstein, J. Gemmell, and J. Han. A Bayesian Approach to
Discovering Truth from Conflicting Sources for Data Integration. *PVLDB*, 5(6):550–561, 2012.