TD-AC: Efficient Data Partitioning based Truth Discovery

Osias Noël Nicodème Finagnon TOSSOU
African Institute for Mathematical Sciences
Mbour, Senegal
osias.tossou@aims-senegal.org

Mouhamadou Lamine Ba
Université Alioune Diop de Bambey
Bambey, Senegal
mouhamadoulamine.ba@uadb.edu.sn

ABSTRACT

This paper introduces an effective algorithm, called **TD-AC**, for the truth discovery problem in scenarios where data attributes are *correlated* by distinct levels of reliability of the sources. **TD-AC** is built on an abstract representation of the truth in the data to automatically find an optimal partitioning of the input data using the k-means clustering technique and the silhouette measure. Such a data partitioning strategy ensures to maximize the accuracy of any *base* truth discovery process when executed on each partition. The intensive experiments conducted on synthetic and real datasets show that **TD-AC** outperforms baseline approaches with a more reasonable running time. It improves on synthetic datasets the accuracy of standard truth discovery algorithms by 1% at least and by 14% at most and also significantly when the data coverage rate is high for the other types of datasets.

KEYWORDS

Truth discovery, attribute, data partitioning, clustering, attribute truth vector, k-means, silhouette index, performance evaluation

1 INTRODUCTION

Dealing with contradictory claims about the same facts is a real concern in many real-world applications such as Web data integration systems [3], online crowdsourcing platforms, online news Websites, social media, etc. Truth discovery resolves such a issue by predicting which of the values provided by conflicting sources is true with no prior knowledge about the level of reliability of the sources. Many approaches [1, 2, 5, 7, 12] for truth discovery have been proposed based on an estimation of the reliability of sources by corroborating their claims under various settings. As in [2], we investigate in this work the truth discovery with attribute partitioning problem that may occur in cases where the attributes over data are structurally correlated so that sources exhibit different levels of reliability on distinct groups of data attributes, as in the setting given in Table 1. Table 1 shows conflicting claims about facts (or data attributes) on two distinct topics (Table 1b) from three sources as depicted in Table 1a. Given the correct answers inside red ellipses, we note that the sources present different levels of reliability according to distinct subsets of facts. For instance, Source 1 is good on Q1 and Q3 while being bad on Q2. Meanwhile Source 2 is good on Q2 and bad on Q1 and Q3. We say that Q1 and Q3 are about data attributes that are *correlated* according to the sources' reliability levels ; *capturing these unknown groups of correlated attributes may help to avoid having a biased truth discovery process.*

Football (FB)	Q1 - Which country won the 2019 Africa Cup of Nations? Q2 - In which year did Benin reach the quarter-finals for the first time in the Africa Cup of Nations? Q3 - How many players are there per team in a football-game?
Computer science (CS)	Q1- Who created the kernel of the linux system? Q2- In which year did he create it? Q3 - What does this python code display? *print(3+4)*

(a) Several facts about two different topics

Sources	Topic	Q1	Q2	Q3
Source 1	FB	Algeria	2000	12
Source 2	FB	Senegal	2019	11
Source 3	FB	Algeria	1994	12
Source 1	CS	Linux Torvalds	1830	7
Source 2	CS	Bill Gate	1991	8
Source 3	CS	Steve Jobs	1991	10

(b) Source claims about those facts

Table 1: Example with sources having different levels of reliability with respect to distinct groups of data attributes

The approach in [2] finds the set of correlated data attributes for truth discovery as an optimal partitiioning of the set of input data attributes using various weighting functions over sources' reliability levels themselves estimated by the truth discovery algorithm. However, the different exploration strategies introduced in [2] are *time-consuming* and *error-prone*. In addition, its different weighting functions do not give any guarantees about the correctness of the returned optimal partition.

This paper revisits [2] and proposes a new more effective and efficient approach to the problem of truth discovery with attribute partitioning. The presented approach, called **TD-AC**, is based on an abstract representation of the truth in the data using the new concept of *attribute truth vector*. Given the set of attribute truth vectors, we rely on *k-means clustering* technique from machine learning domain to find the optimal partitioning of the data attributes. To determine the optimal number of clusters, we assess the homogeneity of the individuals in a clustering result with the help of the silhouette measure. This methodology guarantees to find an optimal partition or a near-optimal one maximizing the accuracy of any base truth discovery process, without an exploration of all the possible partitions. The results of our intensive experiments on synthetic, semi-synthetic and real datasets show that **TD-AC** outperforms approaches in [2], with a more reasonable time cost. On synthetic data, it improves the accuracy of standard algorithms at least by 1% and at most by 14% and also significantly when the data coverage is high for the other types of datasets.

The remainning of this paper is organized as follow. First, we give some preliminaries and define the studied problem in Section 2. Then, we detail our proposed approach by providing its different building blocks in Section 3. In order to validate our approach, we present in Section 4 the results of our intensive experiments conducted on various types of datasets and a thorough analysis of the obtained results. We briefly review the state-of-the-art truth discovery algorithms in Section 5 before concluding in Section 6 with some research perspectives.

2 CONCEPTS AND STUDIED PROBLEM

This section resumes the key concepts of the truth discovery problem and informally introduces the studied problem.

2.1 DEFINITION OF CONCEPTS

A typical truth discovery process usually assumes a *structured world* where input data consist of a set O of objects corresponding to real world entities. Each object is characterized by a set A of attributes (or properties) with values in V coming from a collection S of data sources. In a *one-truth setting*, every attribute for each object has one true value and several possible false values. Thus, the notion of value confidence C_v is used to assess the level of veracity of every value v. Meanwhile, the level of reliability T_s of a source s (or source accuracy) models its ability to provide true values for given real-world object attributes. In real applications, the confidence scores over provided values and the reliability levels of sources are both often unknown and initialized to default values depending on the setting before being updated during the execution of the truth discovery algorithm. *This work considers groups of attributes over data to be structurally correlated if every source has the same reliability level on these latter.*

2.2 PROBLEM STATEMENT

Given the triplet (S, A, O) in a one-truth setting in which a given source may not cover all the objects or attributes, the truth discovery problem is commonly defined as follows.

Problem 1. *Find, for each object o in O, the true value of every attribute a in A_o amongst its set V_{o-a} of possible values by corroborating claims from sources in S_o where A_o and S_o are the set of attributes of o and the set of sources providinng values for o.*

We informally introduce the truth discovery with attribute partitioning problem as follows.

Problem 2. *Find an optimal partitioning P of A that maximizes the accuracy of any solution for Problem 1 where each partition in P contains correlated data attributes according to sources' reliability levels.*

In next, we propose an efficient clustering based approach to solve Problem 2 when data attributes are structurally correlated.

3 TRUTH DISCOVERY WITH CLUSTERING

This section presents our proposed algorithm, called **TD-AC**, that discovers the truth by data partitioning. **TD-AC**, that stands for *Truth Discovery with Attribute Clustering*, applies *k-means* to find optimal clusters of structurally correlated data attributes based on sources' reliability level by relying on *attribute truth vectors* and the *silhouette index*, as we detail it below.

3.1 DATA ATTRIBUTE TRUTH VECTORS

We define and use the concept of *data attribute truth vectors* as an abstract representation of the precision (or quality) of a given truth discovery algorithm using attributes as dimensions. To build such vectors, we firstly apply a *base truth discovery algorithm* (e.g. majority voting) on input data to obtain a *reference truth*. Then, for each attribute of an object and every source we verify whether or not the value given by the source is true regarding the reference truth; we set the value for each rank of any attribute truth vector according to Equation 1.

$$\forall a \in A, \forall o \in O, \forall s \in S; x(a,o,s) = \begin{cases} 1 & if \rho \text{ } is \text{ } true \\ 0 & otherwise \end{cases} \quad (1)$$

where $\rho = (v(a,o,s)$ exists $\wedge$ $v(a,o,s) = v_F(a,o))$ with $v(a,o,s)$ representing the value given by s about a of o, $v_F(a,o)$ is the true value of a of o predicted by the base algorithm and $x(a,o,s)$ is a binary value of our truth vector. Table 2 sketches the matrix of attribute truth vectors obtained on our running example in Table 1 by applying the procedure described above and Equation 1.

-	FB	CS	FB	CS	FB	CS
Q1	1	0	0	0	1	1
Q2	0	0	1	1	0	1
Q3	1	0	0	0	1	1

Table 2: Matrix of attribute truth vectors with data in Table 1 using TruthFinder as base algorithm

3.2 GROUPING CORRELATED ATTRIBUTES

We find and group correlated data attributes by assessing the similarity distance of their corresponding truth vectors. Given two distinct attributes a_1, a_2 and their truth vectors $(a_1^1, a_1^2, \dots, a_1^l)$ and $(a_2^1, a_2^2, \dots, a_2^l)$, we define the similarity between a_1 and a_2 using the Hamming distance as : $d(a_1,a_2) = \sum_{i=1}^{l} |a_1^i - a_2^i|$ (2).

To automatically devise the *threshold value* for grouping the attributes based on our similarity measure, we rely on *k-means* and its optimization strategy in order to provide a domain-independent clustering process in practical cases. The k-means clustering approach [8] uses a similarity distance metric between data points to group them in k clusters. Given a set of observations $(a_1, a_2, ..., a_n)$, where every observation is an attribute truth vector having l dimensions, we define the partitioning of these attributes using k-means algorithm as the clustering of the n observations in k ($k \leq n$) disjoint sets (or clusters) $C = \{g_1, g_2, ..., g_k\}$ in such a way that the sum of the squares (i.e. the Inertia) within each cluster is minimized. Formally, the goal is to find: $argmin_C \sum_i^k \sum_{a \in g_i} ||a - \mu_i||^2 = argmin_C \sum_i^k |g_i| Inertia(g_i)$ (3) where μ_i is the centroid of the points in g_i. This corresponds to minimize the squared deviations of the points in the same cluster: $argmin_C \sum_i^k \sum_{a_1 \in g_i} \frac{1}{2|g_i|} \sum_{a_1,a_2 \in g_i} ||a_1 - a_2||^2$ (4). K-means requires to specify the value of k in input. We find the optimal k using the *silhouette index* as described next.

3.3 ESTIMATION OF k WITH SILHOUETTE

The silhouette index [11] evaluates the quality of a clustering result with the help of the separation criteria β and the cohesion criteria α. Consider two attributes a_1 and a_2 that belong to clusters $g(1)$ and $g(2)$, respectively. Formally, the silhouette coefficient $CS(a_1)$ of the attribute a_1 is defined as: $CS(a_1) = \frac{\beta(a_1) - \alpha(a_1)}{\max(\alpha(a_1), \beta(a_1))}$ with $\alpha(a_1) = \frac{1}{|g(1)|-1} \sum_{a_j \in g(1); a_j \neq a_1} d(a_1, a_j)$ and $\beta(a_1) = \min_{a_1 \neq a_2} \frac{1}{|g(2)|} \sum_{a_k \in g(2)} d(a_1, a_k)$ (5). If $CS(a_1) <$

0, a_1 is *badly classified.* Conversely, if $CS(a_1) > 0$ a_1 is *well classified.* Finally, if $CS(a_1) = 0$ then a_1 is between two clusters. The silhouette coefficient $CS(g)$ of a cluster g is thus given by: $CS(g) = \frac{1}{|g|} \sum_{a \in g} CS(a)$ (6).

The silhouette value of a partition P is the average of the silhouette coefficients of all its clusters: $CS(P) = \frac{1}{|P|} \sum_{g \in P} CS(g)$ (7). The optimal k is the one associated to the partition having the highest silhouette coefficient amongst all the possible partitions.

3.4 TD-AC TRUTH DISCOVERY APPROACH

As depicted by Algorithm 1, our proposed algorithm **TD-AC** runs as follows: (i) considers a *base truth discovery algorithm* and input data (A, O, S); (ii) computes the matrix of attribute truth vectors from input data using the base algorithm and Equation 1; (iii) efficiently clusters the data attributes by applying k-means combined with the silhouette index ; and (iv) executes the input base truth discovery algorithm on each data partition, and then aggregates the partial results to generate the entire result.

Algorithm 1 TD-AC(F, A, O, S) – Truth discovery with Attribute clustering using k-means and silhouette coefficient

```
Require: Set of observations (A, O, S), Base algorithm F
Ensure: results // Truth predicted by TD-AC
results ← [ ]
truth_vector_matrix ← buildTruthVectors(F, A, O, S)
// Find the optimal partition with k-mean and silhouette
indice_silhouette ← 0
opt_partition ← []
for all k ∈ [2, |A| − 1] do
    partition=kmeansAttClustering(truth_vector_matrix,k)
    silhouette_index_tmp ← CS(partition)
    if k == 2 then
        silhouette_index ← silhouette_index_tmp
        opt_partition ← partition
    else
        if indice_silhouette < indice_silhouette_tmp then
            silhouette_index ← silhouette_index_tmp
            opt_partition ← partition
        end if
    end if
end for
// Truth discover on the optimal partition found
for each g ∈ opt_partition do
    A_p, O_p, S_p ← getData(g)
    partial_result ← F(A_g, O_g, S_g)
    Add partial_result in results
end for
```

4 EXPERIMENTS AND RESULTS

In this section, we demonstrate the efficiency of our approach on various datasets, proving that it outperforms approaches proposed in [2] and standard truth discovery algorithms in the literature in the presence of structurally correlated data attributes. We also show that its execution time is similar to that of standard algorithms unlike partitioning strategies in [2]. We start by presenting the experiment setting up and performed tests.

4.1 EXPERIMENTATION SETTING UP

For the comparison purposes, we have implemented the different analyzed algorithms using *Python* programming language. The following standard truth discovery algorithms have been implemented: **MajorityVote**,**TruthFinder** [14], **DEPEN**, **Accu** and **AccuSim** [4]. We have compared ourselves to these algorithms because they are amongst the best in terms of efficiency and effectiveness for solving the truth discovery problem in various settings. In addition we have also implemented **AccuGenPartition** in [2] along with the different weighting functions to

	DS1	DS2	DS3
m_1	1.0	1.0	1.0
m_2	0.0	0.0	0.2
m_3	1.0	0.8	0.8

Table 3: Average accuracy values for the various configurations of the synthetic datasets

compute the optimal partition. The source codes of the tested algorithms are all available at **https://github.com/osiastossou/ProjetTD-AC.git**.

We have conducted all our experiments on a Intel Core i5 2.6GHz laptop computer with 8GB of RAM, 250GB of hard disk space, and 1.5GB of graphics memory. The implemented algorithms here require all hyper-parameters in input whose values have been fixed for the various tests according to [12]. At last, we have relied on usual metrics such as *precision*, *recall*, *F1-measure*, *accuracy*, and *execution time* to evaluate and compare the performance of our tested algorithms.

4.2 EXPERIMENTS ON SYNTHETIC DATA

We detail here the results of our experiments on synthetic data which simulate conditions where data attributes are structurally correlated.

We have used and re-implemented in Python the synthetic data generator in [2] to produce our synthetic data sets; we defer to [2] for the details. For the evaluation process, we have then generated three synthetic datasets (DS1, DS2 and DS3) of 6 attributes, 1000 objects, 10 sources and $60,000$ observations with three different configurations as depicted in Table 3; DS1 meets the seeting of this work while DS3 relaxes the assumptions to test the robustness of our approach. The partition selected for each configuration is given in Table 5.

Tables 4a, 4b and 4c respectively present the performances of each algorithm on DS1, DS2 and DS3. For the tests, we used **Accu** as our base algorithm similarly to the approaches in [2]. We observe that the attribute partitioning truth discovery algorithms perform better than the standard ones on all three synthetic datasets, proving the importance of partitioning when data attributes are structurally correlated. Specifically, **TD-AC** is the only partitionning strategy with a precision comparable to the real world (i.e. an *Oracle*) without a blowup of the running time. Table 5 reports the partitions returned by the different partitioning approaches.

4.3 TESTS ON SEMI-SYNTHETIC DATA

The semi-synthetic datasets have been generated from a real dataset called **Exam**. This real dataset comes from [2] and has been used in that paper to validate the proposed approaches. The **Exam** dataset has been obtained by aggregating the anonymous results of admission examinations. Unfortunately, it cannot be redistributed for privacy reasons. We had access to answers from 248 students (*sources*) to 124 questions (*attributes*) in total, from 9 different domains: **Math 1A, Chemistry 1, Math 1B, Physics, Electrical Engineering, Computer Science, Chemistry 2, Science of life,** and **Math 2**. We also know the correct answer to each question. Math 1A and Physics were only mandatory with the choice of an additional domain between Chemistry 1 and Math 1B. The five remaining domains were completely optional and wrong answers were penalized. As a result, all the attributes were not covered (missing data). For each unanswered

Dataset	Algorithm		Precision	Recall	Accuracy	F1-measure	Time(s)	#Iteration
DS1	**MajorityVote**		0.602	0.667	0.806	0.633	75	1
	TruthFinder		0.568	0.624	0.787	0.595	1261	3
	DEPEN		0.551	0.611	0.778	0.580	1492	3
	Accu		0.667	0.712	0.838	0.689	6495	9
	AccuSim		0.662	0.705	0.836	0.683	5580	11
	AccuGenPartition	**Max**	0.691	0.724	0.849	0.707	757230	-
		Avg	0.682	0.725	0.846	0.703	757230	-
		Oracle	0.997	0.998	0.999	0.998	757230	-
	TD-AC (F=Accu)		0.853	0.870	0.930	0.861	3410	1

(a) Performance measures on DS1

Dataset	Algorithm		Precision	Recall	Accuracy	F1-measure	Time(s)	#Iteration
DS2	**MajorityVote**		0.741	0.834	0.884	0.785	99	1
	TruthFinder		0.736	0.819	0.880	0.775	2276	3
	DEPEN		0.735	0.828	0.881	0.779	1459	3
	Accu		0.659	0.663	0.828	0.661	11263	18
	AccuSim		0.467	0.388	0.734	0.424	9996	20
	AccuGenPartition	**Max**	0.738	0.810	0.879	0.773	861697	-
		Avg	0.867	0.904	0.940	0.885	861697	-
		Oracle	0.985	0.992	0.994	0.989	861697	-
	TD-AC (F=Accu)		0.985	0.992	0.994	0.989	3783	1

(b) Performance measures on DS2

Dataset	Algorithm		Precision	Recall	Accuracy	F1-measure	Time(s)	#Iteration
DS3	**MajorityVote**		0.847	0.891	0.918	0.869	112	1
	TruthFinder		0.838	0.875	0.910	0.856	2762	3
	DEPEN		0.833	0.876	0.909	0.854	1732	3
	Accu		0.873	0.918	0.934	0.895	3478	7
	AccuSim		0.808	0.822	0.886	0.815	7171	15
	AccuGenPartition	**Max**	0.872	0.884	0.925	0.878	675078	-
		Avg	0.938	0.958	0.968	0.948	675078	-
		Oracle	0.965	0.976	0.982	0.970	675078	-
	TD-AC (F=Accu)		0.965	0.976	0.982	0.970	2491	1

(c) Performance measures on DS3

Table 4: Performance of all tested algorithms on the synthetic datatsets DS1, DS2 and DS3

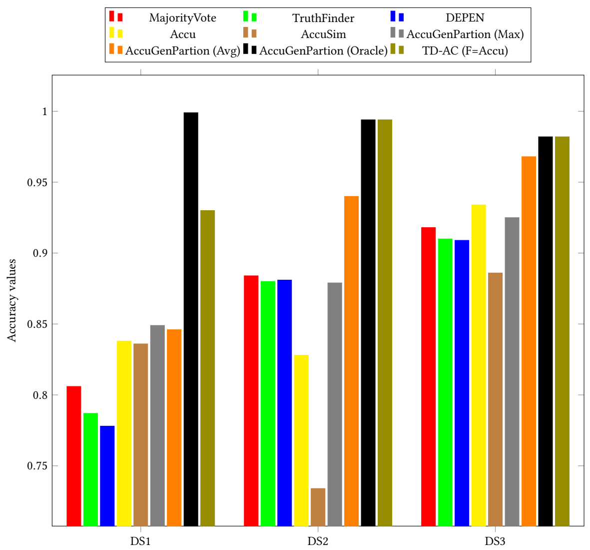

Figure 1: Comparison of the accuracy of all tested algorithms on DS1, DS2 and DS3

	DS1	DS2	DS3
Synthetic data generator	[(1, 2), (4, 6), (3), (5)]	[(2, 5), (1, 4), (3, 6)]	[(1, 6, 3), (2, 4, 5)]
AccuGenPartition (Max)	[(3, 4), (5), (1, 2, 6)]	[(2), (1, 4, 3, 5, 6)]	[(1), (5, 2, 4, 3, 6)]
AccuGenPartition (Avg)	[(3, 6), (1, 2, 5, 6)]	[(2), (5), (1, 4, 3, 6)]	[(1, 5), (2, 4, 3, 6)]
AccuGenPartition (Oracle)	[(1), (2), (3), (4, 6), (5)]	[(2, 5), (1), (4), (3, 6)]	[(1, 5), (2, 4), (3, 6)]
TD-AC (F=Accu)	[(1, 2), (4, 6), (3, 5)]	[(2, 5), (1, 4), (3, 6)]	[(1, 5), (2, 4), (3, 6)]

Table 5: Partitions chosen by the generator and returned by the different partitioning algorithms

question we have synthetically chosen a false answer, randomly in a range of false values of size equal to 25, 50, 100 or 1000.

Tables 6 and 7 respectively present the different results of these tests on the semi-synthetic data of 62 and 124 attributes, each with configurations on ranges of false values of size 25, 50, 100, and 1000. The tests compare the performances of **Accu** and **TD-AC+Accu** on the one hand and on the other hand **TruthFinder** and **TD-AC+TruthFinder**. In general, we note that combining a base algorithm with **TD-AC** does not highly deteriorate the performance of the standard algorithm whatever the configuration considered, and even improves it in some cases, for example for the dataset with 124 attributes (see Figures 2 and 3).

Dataset	Algorithm	Precision	Recall	Accuracy	F1-measure	Time(s)	#Iteration
Range 25	**Accu**	0.929	0.896	0.938	0.912	4386	8
	TD-AC (F=Accu)	0.920	0.883	0.931	0.901	10256	1
	TruthFinder	0.894	0.917	0.931	0.905	85	6
	TD-AC (F=TruthFinder)	0.897	0.920	0.933	0.908	62	1

(a) Range 25

Dataset	Algorithm	Precision	Recall	Accuracy	F1-measure	Time(s)	#Iteration
Range 50	**Accu**	0.946	0.912	0.951	0.928	4615	**8**
	TD-AC (F=Accu)	0.963	0.970	0.976	0.966	18233	1
	TruthFinder	0.915	0.934	0.946	0.924	80	4
	TD-AC (F=TruthFinder)	0.915	0.934	0.946	0.924	81	1

(b) Range 50

Dataset	Algorithm	Precision	Recall	Accuracy	F1-measure	Time(s)	#Iteration
Range 100	**Accu**	0.988	0.983	0.990	0.985	4017	7
	TD-AC (F=Accu)	0.972	0.982	0.984	0.977	7684	1
	TruthFinder	0.924	0.943	0.954	0.933	134	3
	TD-AC (F=TruthFinder)	0.925	0.944	0.955	0.935	121	1

(c) Range 100

Dataset	Algorithm	Precision	Recall	Accuracy	F1-measure	Time(s)	#Iteration
Range 1000	**Accu**	0.989	0.984	0.991	0.986	4186	7
	TD-AC (F=Accu)	0.972	0.982	0.984	0.977	8467	1
	TruthFinder	0.927	0.946	0.956	0.936	258	4
	TD-AC (F=TruthFinder)	0.927	0.946	0.956	0.936	241	1

(d) Range 1000

Table 6: Performance of Accu, TruthFinder, TD-AC(F=Accu), and TD-AC(F=TruthFinder) on semi-synthetic datasets with 62 attributes

Dataset	Algorithm	Precision	Recall	Accuracy	F1-measure	Time(s)	#Iteration
Range 25	**Accu**	0.847	0.739	0.904	0.789	7805	9
	TD-AC (F=Accu)	0.852	0.744	0.906	0.794	12432	1
	TruthFinder	0.894	0.919	0.954	0.906	104	3
	TD-AC (F=TruthFinder)	0.894	0.919	0.954	0.906	157	1

(a) Range 25

Dataset	Algorithm	Precision	Recall	Accuracy	F1-measure	Time(s)	#Iteration
Range 50	**Accu**	0.885	0.806	0.931	0.844	10680	11
	TD-AC (F=Accu)	0.928	0.916	0.964	0.922	10456	1
	TruthFinder	0.906	0.931	0.962	0.918	278	4
	TD-AC (F=TruthFinder)	0.904	0.929	0.961	0.916	276	1

(b) Range 50

Dataset	Algorithm	Precision	Recall	Accuracy	F1-measure	Time(s)	#Iteration
Range 100	**Accu**	0.905	0.822	0.943	0.862	8516	**10**
	TD-AC (F=Accu)	0.953	0.955	0.980	0.954	10196	1
	TruthFinder	0.905	0.918	0.961	0.911	597	5
	TD-AC (F=TruthFinder)	0.909	0.934	0.965	0.921	460	1

(c) Range 100

Dataset	Algorithm	Precision	Recall	Accuracy	F1-measure	Time(s)	#Iteration
Range 1000	**Accu**	0.930	0.913	0.966	0.921	11951	12
	TD-AC (F=Accu)	**0.934**	0.927	**0.970**	**0.931**	9222	1
	TruthFinder	0.921	0.941	0.970	0.931	1626	4
	TD-AC (F=TruthFinder)	0.909	0.933	0.965	0.921	1401	1

(d) Range 1000

Table 7: Performance of Accu, TruthFinder, TD-AC(F=Accu), and TD-AC(F=TruthFinder) on semi-synthetic datasets with 124 attributes

4.4 EXPERIMENTS ON REAL DATA

To end our performance evaluation, we present in this section the results of the experimentation of our approach and the existing algorithms on real data. The evaluation on real data sets enables to validates our approach against practical applications. For this purpose, we have considered and used the real datasets **Exam** [2], **Stocks** and **Flights** [9]. Real data contain missing values that

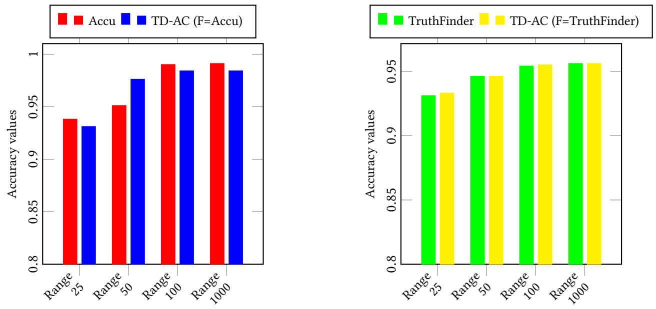

Figure 2: Study of the impact of TD-AC on Accu and TruthFinder by pairwise comparison of the accuracy values on semi-synthetic datasets with 62 attributes

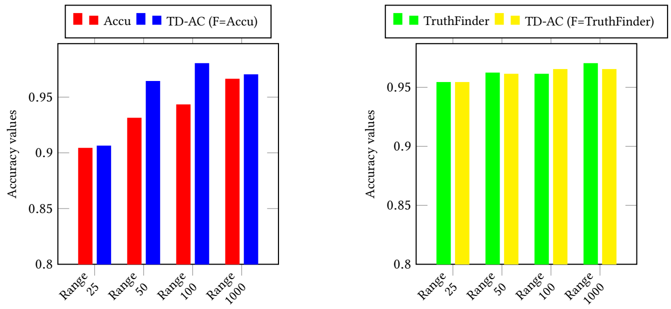

Figure 3: Study of the impact of TD-AC on Accu and TruthFinder by pairwise comparison of the accuracy values on semi-synthetic datasets with 124 attributes

may impact the performance of truth discovery algorithms. We assess the *Data Coverage Rate* (DCR) of each dataset with: $DCR = \left(1 - \frac{\sum_{o\in O}(|S_o|\times|A_o| - \sum_{s\in S_o}|A_{o-s}|)}{\sum_{o\in O}(|S_o|\times|A_o|)}\right)\times 100$ (7). Table 8 presents the details of the three real data sets after pre-processing; for Exam we have considered three configurations.

	Stocks	Exam 32	Exam 62	Exam 124	Flights
Number of sources	55	248	248	248	38
Number of objects	100	1	1	1	100
Number of attributes	15	32	62	124	6
Number of observations	56992	6451	8585	11305	8644
Data Coverage Rate (%)	75	81	55	36	66

Table 8: Statistics about the different real datasets

Table 9 presents the performance measures of **Accu**, **TD-AC+Accu**, **TruthFinder**, and **TD-AC+TruthFinder**. We have also reported in Figures 4 and 5 the comparative study of the accuracy values of **Accu** and **TD-AC+Accu** on the one hand and **TruthFinder** and **TD-AC+TruthFinder** on the other hand on real datasets with data coverage greater than 66% and less than 55% respectively. We observe that **Accu** and **TruthFinder** outperform when used with our **TD-AC** approach, especially when the data coverage rate is greater than 66%. We also remark that the execution time of **TD-AC** is very close to that of standard algorithms on real data, unlike **AccuGenPartition**.

4.5 Analysis of the results and discussion

The analysis of the presented intensive performance evaluation carried out on several datasets yields to three main observations.

Dataset	Algorithm	Precision	Recall	Accuracy	F1-measure	Time(s)	#Iteration
Exam 32	**Accu**	0.607	0.837	0.658	0.704	4059	11
	TD-AC (F=Accu)	0.614	0.912	0.679	0.734	4075	1
	TruthFinder	0.540	0.772	0.570	0.636	6.66	5
	TD-AC (F=TruthFinder)	0.533	0.733	0.558	0.617	13.7	1

(a) Exam with 32 attributes

Dataset	Algorithm	Precision	Recall	Accuracy	F1-measure	Time(s)	#Iteration
Exam 62	**Accu**	0.955	0.962	0.944	0.959	4877	10
	TD-AC (F=Accu)	0.926	0.944	0.911	0.935	2789	1
	TruthFinder	0.937	0.955	0.926	0.945	16.2	5
	TD-AC (F=TruthFinder)	0.898	0.885	0.854	0.891	24.3	1

(b) Exam with 62 attributes

Dataset	Algorithm	Precision	Recall	Accuracy	F1-measure	Time(s)	#Iteration
Exam 124	**Accu**	0.951	0.969	0.947	0.960	3662	9
	TD-AC (F=Accu)	0.917	0.938	0.904	0.927	3733	1
	TruthFinder	0.924	0.949	0.916	0.936	23.5	5
	TD-AC (F=TruthFinder)	0.907	0.906	0.878	0.907	79	1

(c) Exam with 124 attributes

Dataset	Algorithm	Precision	Recall	Accuracy	F1-measure	Time(s)	#Iteration
Stocks	**Accu**	0.847	0.877	0.809	0.862	2753	4
	TD-AC (F=Accu)	0.886	0.956	0.887	0.920	4169	1
	TruthFinder	0.860	0.700	0.718	0.772	629	5
	TD-AC (F=TruthFinder)	0.887	0.862	0.832	0.875	446	1

(d) Stocks

Dataset	Algorithm	Precision	Recall	Accuracy	F1-measure	Time(s)	#Iteration
Flights	**Accu**	0.958	0.968	0.957	0.963	390	7
	TD-AC (F=Accu)	0.969	0.987	0.974	0.978	452	1
	TruthFinder	0.859	0.900	0.857	0.879	22.3	3
	TD-AC (F=TruthFinder)	0.848	0.885	0.842	0.866	33	1

(e) Flights

Table 9: Performance of Accu, TruthFinder, TD-AC+Accu, and TD-AC+TruthFinder on real datasets

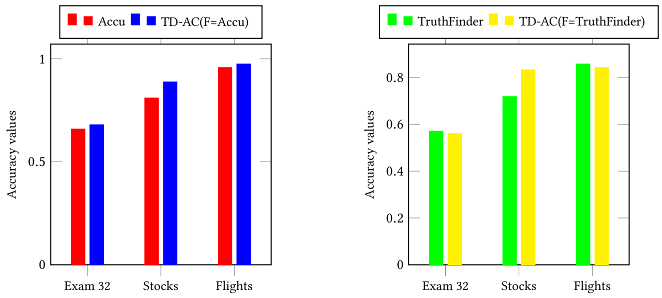

Figure 4: Study of the impact of TD-AC on Accu and TruthFinder by pairwise comparison of the accuracy values on real datasets Exam with 32 attributes, Stocks and Flights (DCR ≥ 66)

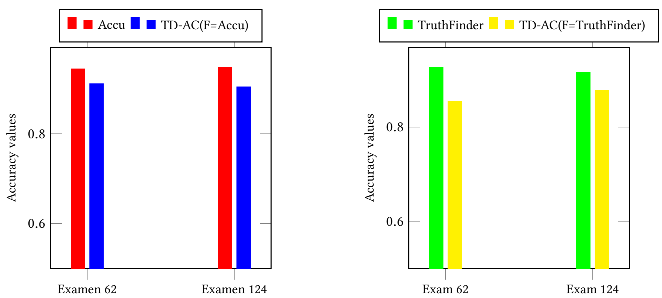

Figure 5: Study of the impact of TD-AC on Accu and TruthFinder by pairwise comparison of the accuracy values on the real datasets Exam with 62 and 124 attributes (DCR ≤ 55)

TD-AC outperforms baseline partitioning approaches. TD-AC highly improves the accuracy of **AccuGenPartition** by 1% at least and by 14% at most (see Figure 1) with a time complexity around 200 less significant (see Tables 4a, 4b and 4c). **AccuGen-Partition** is our baseline *brute force* approach proposed in [2] for the truth discovery with attribute partitioning problem with two weighting functions: **Max** and **Avg**. To discover the optimal partition, *k-means* combined with the*silhouette index* has been shown in Table 5 to be better than **Max** and **Avg** because: (i) k-means is a robust partitioning technique with a well-defined optimization strategy; and (ii) silhouette returns the most structurally homogeneous existing clusters. This explains the effectiveness of **TD-AC**. The drastic reduction of the running time with **TD-AC** is because it only requires one iteration to last without exploring all the possible partitions.

TD-AC improves the accuracy of base algorithms. When data attributes are structurally correlated, **TD-AC** significantly enhances the accuracy (from 5 to 35%) of standard algorithms (see Tables 4a, 4b, 4c, 9 and Figure 3). Standard algorithms alone do not capture the structural correlations between attributes leading to biased results. In the cases where the conditions do not match our working setting, **TD-AC** does not degrade the performances of the standard algorithms (see Tables 6 and 7). The impact of **TD-AC** is more important for **Accu** than **TruthFinder** because the former captures better the different levels of reliability of the sources. Such a impact introduces, however, a surplus in terms of execution time which is fortunately reasonable.

Correlation between coverage and TD-AC accuracy. The main observation is that **TD-AC** is more efficient when the data coverage is very high, i.e. DCR $\geq$ 66% (see Figure 4) because more one has in terms of information the better is the clustering with k-means. Lot of missing values, i.e. very sparse truth vectors affect both the quality of the clustering and the truth discovery process (see Figure 5).

5 RELATED WORK

A significant effort has been made in truth discovery area over the past years which has led to several approaches [12, 15]. The simplest approach is the majority vote which considers the truth said by the majority of sources. More elaborated approaches try to model the different levels of reliability of the sources and domain-specific aspects of the truth. For instance, **TruthFinder**[14], one of the first proposed standard algorithms, is a probabilistic model based on Bayesian analysis with similar values supporting each others in vote counts. Methods such as **DEPEN**, **Accu** and **AccuSim**[4] take into consideration copy relationships that may exist between sources by penalizing the vote of a source if it is detected as a copy of another source. **DART** (Domain-AwaRe Truth Discovery) [10] is both a probabilistic and a bayesian model which integrates the domain expertise level. Very recent methods such as [6, 15] capture the correlations between objects in the domain of **Mobile Crowd Sensing**.

The research works that are connexe to our studied problem are [2] and [13]. The proposal in [2] is a brute force approach that explores all the possible partitions of a given set of attributes in order to discover the one maximizing the precision of a standard truth discovery algorithm. The goodness of a partition in this case is based on a weighting function over sources' reliability levels. The work in [13] focuses on object partitioning based on domain knowledge and some additional constraints.

6 CONCLUSION AND PERSPECTIVES

In this work, we have studied the truth discovery problem in a setting where the attributes of the data are structurally correlated. As a solution, we have proposed a new approach, called **TD-AC**, built on an abstract representation of the truth in the data, the k-means clustering technique and the silhouette measure to automatically find an optimal partitioning of the input data (or a near-optimal) maximizing the accuracy of any base truth discovery process. Through an intensive experimental evaluation over various types of datasets, we have then shown the effectiveness and efficiency of **TD-AC** compared to existing partitioning strategies and its positive impact to the accuracy of any standard truth discovery process.

Despites of that, we have noticed that when the dataset contains lot of missing values, the impact of our approach is less significant. This can be explained by the use of sparse truth vectors in the clustering step, making the finding of the optimal partition hard. Moreover, even if the running time of our approach and standard algorithms is reasonable in the presence of small size datasets, it become important when the number of attributes, objects and sources is very large. As research perspectives, we plan to (i) improve our approach to better account for data with lot of missing values on the one hand; and (ii) on the other hand, to propose an optimization of the running time of our approach, in particular the optimal partition computation, by using parallel computation. We also plan to compare ourselves to a larger set of standard truth discovery algorithms and the partitioning approach in [13].

REFERENCES

[1] Mouhamadou Lamine Ba, Laure Berti-Équille, Kushal Shah, and Hossam M. Hammady. 2016. VERA: A Platform for Veracity Estimation over Web Data. In *Proc. International Conference on World Wide Web*. ACM, Montreal, Canada, 159–162.

[2] Mouhamadou Lamine Ba, Roxana Horincar, Pierre Senellart, and Huayu Wu. 2015. Truth Finding with Attribute Partitioning. In *Proc. International Workshop on Web and Databases*. Association for Computing Machinery, New York, USA, 27–33.

[3] Mouhamadou Lamine Ba, Sébastien Montenez, Talel Abdessalem, and Pierre Senellart. 2014. Monitoring moving objects using uncertain web data. In *Proc. International Conference on Advances in Geographic Information Systems*. ACM, Dallas, USA, 565–568.

[4] Xin Dong, Laure Berti-Equille, and Divesh Srivastava. 2009. Integrating Conflicting Data: The Role of Source Dependence. *PVLDB* 2 (08 2009), 550–561.

[5] Xin Luna Dong, Laure Berti-Equille, Yifan Hu, and Divesh Srivastava. 2010. Global detection of complex copying relationships between sources. *Proc. VLDB Endowment* 3, 1-2 (2010), 1358–1369.

[6] Yang Du, Yu-E Sun, He Huang, Liusheng Huang, Hongli Xu, Yu Bao, and Hansong Guo. 2019. Bayesian co-clustering truth discovery for mobile crowd sensing systems. *IEEE Transactions on Industrial Informatics* 16, 2 (2019), 1045–1057.

[7] Alban Galland, Serge Abiteboul, Amélie Marian, and Pierre Senellart. 2010. Corroborating Information from Disagreeing Views. In *Proc. ACM International Conference on Web Search and Data Mining*. Association for Computing Machinery, New York, USA, 131–140.

[8] Eric Gaussier. Accessed Mai 2020.. Partitionement de documents. Clustering, http://ama.imag.fr/~gaussier/Courses/ATD/Clustering.pdf.

[9] Xian Li, Xin Luna Dong, Kenneth Lyons, Weiyi Meng, and Divesh Srivastava. 2012. Truth Finding on the Deep Web: Is the Problem Solved? *Proc. VLDB Endow.* 6, 2 (Dec. 2012), 97–108.

[10] Xueling Lin and Lei Chen. 2018. Domain-aware multi-truth discovery from conflicting sources. *Proc. VLDB Endowment* 11, 5 (2018), 635–647.

[11] Giovanna Menardi. 2011. Density-based Silhouette diagnostics for clustering methods. *Statistics and Computing* 21, 3 (2011), 295–308.

[12] Dalia Attia Waguih and Laure Berti-Equille. 2014. Truth Discovery Algorithms: An Experimental Evaluation. arXiv:cs.DB/1409.6428

[13] Yi Yang, Quan Bai, and Qing Liu. 2019. A probabilistic model for truth discovery with object correlations. *Knowledge-Based Systems* 165 (2019), 360–373.

[14] Xiaoxin Yin, Jiawei Han, and Philip Yu. 2008. Truth Discovery with Multiple Conflicting Information Providers on the Web. *Knowledge and Data Engineering, IEEE Transactions on* 20 (Juil 2008), 796 – 808.

[15] Ming Zhao and Jia Jiao. 2020. Police: An Effective Truth Discovery Method in Intelligent Crowd Sensing. In *Proc. Artificial Intelligence and Security*, Vol. 12239. Springer, Hohhot, China, 384–398.