PROBCLEAN: A PROBABILISTIC DUPLICATE DETECTION SYSTEM

GAGANDEEP SINGH*

SHUBHRA SAGGAR**

Abstract:

One of the most prominent data quality problems is the existence of duplicate records. Current data cleaning systems usually produce one clean instance (repair) of the input data, by carefully choosing the parameters of the duplicate detection algorithms. We propose ProbClean, a system that treats duplicate detection procedures as data processing tasks with uncertain outcomes. We use a novel uncertainty model that compactly encodes the space of possible repairs corresponding to different parameter settings. ProbClean efficiently supports relational queries and allows new types of queries against a set of possible repairs.

KEYWORDS:

ProbClean; Problistic ETL tool; Tool for data cleaning; Duplicate record detector; Multiple repair.

* PROJECT TRAINEE, MCA 6TH SEM, GURU NANAK INSTITUTE OF MANAGEMENT

** HOD MCA, GURU NANAK INSTITUTE OF MANAGEMENT

I. INTRODUCTION

We introduced a new ETL approach that defers the destructive operations of the transformation process to query time as we show in Figure 1(b). Specifically, in our proposal, the transformation process is divided into an offline expensive procedure that produces multiple possible repairs, and an online procedure that can be efficiently executed at query time. We propose the use of probabilistic databases principles and techniques to model and maintain multiple possible repairs of the unclean data. Moreover, online cleaning specifications can be provided by the user to process a specific clean instance. In this demonstration, we introduce ProbClean, a probabilistic ETL tool that focuses on a specific data transformation task, which is detecting and eliminating duplicate records. Duplicate records are defined as (possibly non-identical) records in the unclean database that refer to the same real world entity. In this case, a repair of an unclean database is a clustering of records where records that refer to the same real world entity are grouped in the same cluster. Figure 2 shows an example input relation representing sample census data that contains duplicate records. Where each cluster is a set of duplicates that are eventually merged into one representative record. The current duplicate detection approaches identify records as either *duplicates or non-duplicates based on the given cleaning* .

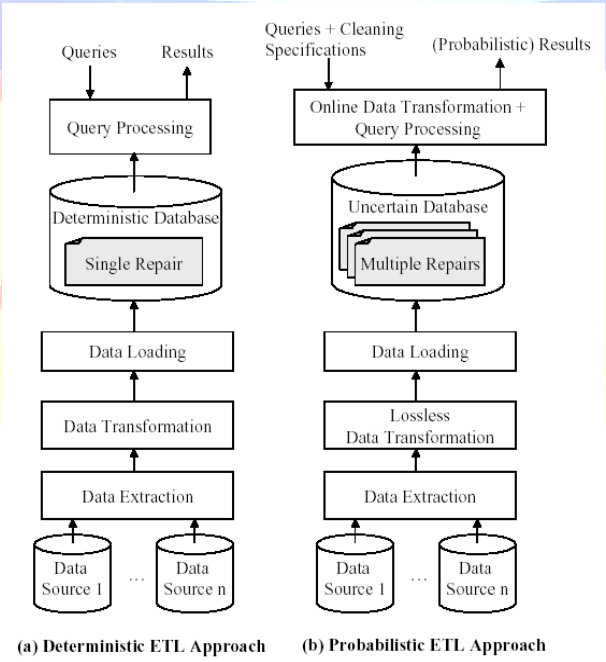

FIG.1. ONE SHORT DATA CLEANING VS PROBABILISTIC DATA CLEANING

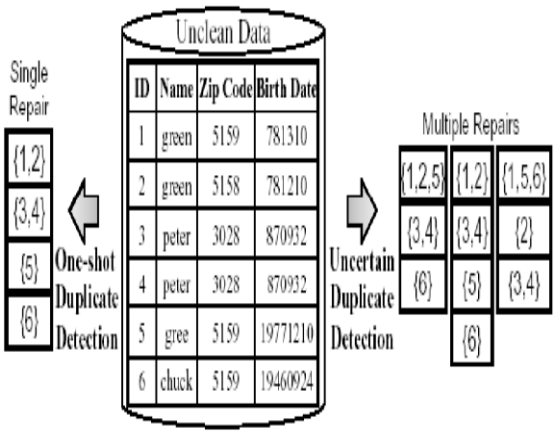

FIG .2. ONE-SHOT VS PROBABILISTIC DUPLICATE DETECTION

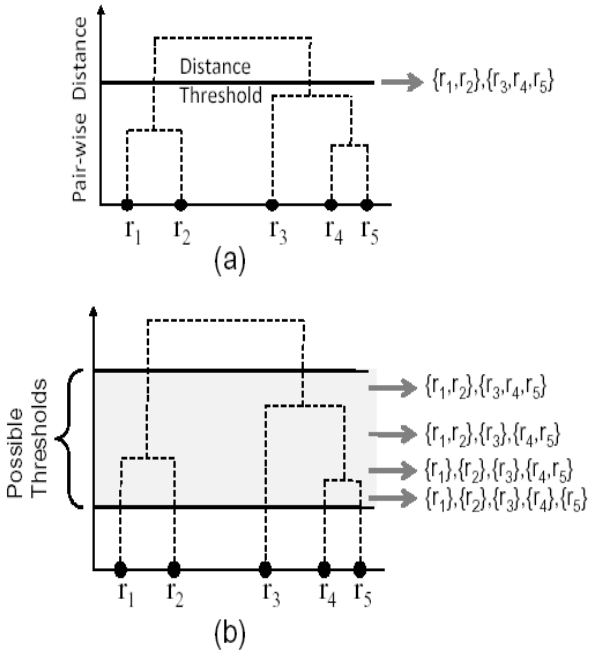

FIG. 3. LINK-BASED HIERARCHICAL CLUSTERING

(A) USING A SINGLE THRESHOLD

(B)USING A RANGE OF POSSIBLE THRESHOLDS

Hence, the result is a single clustering (repair) of the input relation. On the other hand, in the probabilistic duplicate detection approach, this restriction is relaxed to allow forum certainty in deciding on the true duplicates (e.g., based on multiple possible similarity thresholds). The result is a set of possible clusterings (repairs) as shown in Figure 2.

II. UNCERTAIN DUPLICATE DETECTION

We generate all possible repairs corresponding to a set of possible parameter settings of any fixed parameterized clustering algorithm (e.g., single-linkage [5][6]). We focus on a class of clustering algorithms, namely hierarchical clustering algorithms, to perform uncertain clustering due to two features of such a class: (1) the number of generated repairs is linear in the number of records, which allows for efficiently generating, storing and querying the possible repairs, and (2) hierarchical clustering algorithms can be used to generate multiple repairs through simple modifications to the algorithms. The algorithms use specific criteria to determine which clustering to return. For example, Figure 3(a) gives the hierarchy generated by a greedy agglomerative clustering algorithm [5], for the relation R = {r1, . . . , r5}. The algorithm parameter _ represents a threshold on inter-cluster distances, which are represented by the Y -axis. Setting _ to the value depicted by the solid horizontal line in Figure 3(a) results in the clustering {r1, r2}, {r3, r4, r5}.In order to capture multiple possible outcomes of a hierarchical clustering algorithm[8,9], we view the parameter _ as a random variable rather than a fixed value. We denote by f_ the probability density function of _ .f_ is defined over the interval[_ l, _u]. Applying a clustering algorithm A to an unclean relation R using a parameter value t 2 [_ l, _u], generates a possible clustering (i.e., repair) of R, denoted A(R, t). Note that multiple parameter values may lead to the same clustering. The set of possible repairs, denoted X, is defined as{A(R, t) : t 2 [_ l, _u]}. The set X defines a probability space created by drawing random parameter values in [_ l, _u],based on the density function f_ , and using the algorithm A to generate the possible repairs corresponding to these values. The probability of a specific repair X 2 X, denoted Pr(X),is equal to the probability of the parameter settings resulting in the repair X.

III. MODELING POSSIBLE REPAIRS

We define an uncertain clean relation (U-clean relation for short) that encodes the possible repairs X of an unclean relation R, which are generated by a parameterized clustering algorithm A. A U-clean relation, denoted Rc, is a set of crecords where each c-record is a representative record of a cluster of records in R. Attributes of Rc are all attributes of R, in addition to two distinguished attributes: C and P. Attribute C of a c-record is the set of records identifiers in R that are clustered together to form this c-record. The attribute P of a c-record represents the parameter settings of the clustering algorithm A that lead to generating the cluster represented by this c-record. Attribute P is represented as an interval within the range of the parameter _ . Note that it is possible to have overlapping clusters represented by c-records of Rc since Rc may encapsulate multiple clusterings of records in R. Figure 4 shows an example U-clean relation, named Personc that captures possible repairs generated by the algorithm A using a uniformly distributed parameter _ in the range [0, 10]. For brevity, we omit Attributes Name , ZIP and Birth Date from Relation Personc. We assume that the attribute Income of each cluster is the average value of its member records.

IV. IMPLEMENTATION IN RDBMS

In [3], we show how to construct U-clean relations efficiently and how to store U-clean relations in a relational database system. We provide an overview of the system implementation as follows.

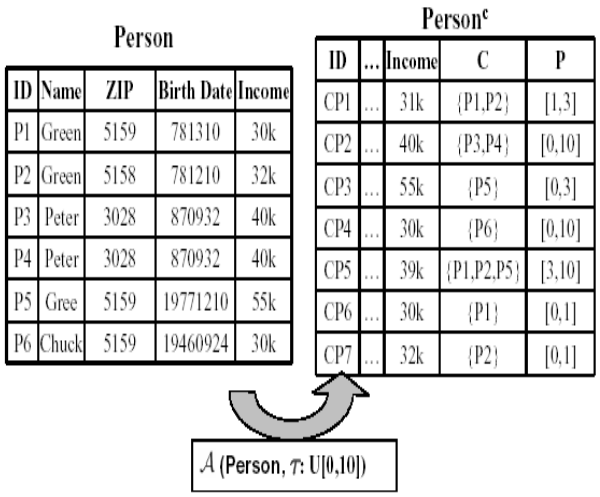

FIG. 4. AN EXAMPLE OF A U-CLEAN RELATION

A. Implementing U-clean Relations

We implement Attributes C and P in a relational database as abstract data types (ADTs). We implement a set of user defined functions (UDFs) over the attributes C and P to manipulate their contents during query processing. More specifically, we provide the following functions:

• ConjC(C1, . . . ,Cn) and ConjP(P1, . . . , Pn) compute the conjunctions of multiple clusters C1, . . . ,Cn and multiple parameter settings P1, . . . , Pn, respectively.

• DisjC(C1, . . . ,Cn) and DisjP(P1, . . . , Pn) compute the disjunctions of multiple clusters C1, . . . ,Cn and multiple parameter settings P1, . . . , Pn, respectively.

• Contains(P, x) returns True iff the parameter settingsP contain a given parameter setting x.

• Contains Base Records(C, S) returns True iff a set of base records identifiers S is contained in a cluster C.

• Prob(P, f_1 , . . . , f_d) computes the probability of parameter settings P, given the probability distribution functions(PDFs) f_1 , . . . , f_d of the parameters _1, . . . , _d that appear in P. PDFs are represented as histograms.

• Most Prob Param(UR, f_1 , . . . , f_d) computes a parameter setting corresponding to the most probable repair of a U-clean relation UR, given the parameters PDFsf_1 , . . . , f_d .
Algorithmic details of these functions are described in [3].

B. Supported Query Types

Our system allows evaluation of selection, projection and join queries over U-clean relations through the UDFs defined in Section IV-A. For example, in order to report the distinct incomes of persons according the possible repairs encoded in Personc (shown in Figure 4), we submit the following SQL query:

Income	C	P
31k	{P1,P2}	[1,3]
40k	{P3,P4}	[0,10]
55k	{P5}	[0,3]
30k	{P6}∨{P1}	[0,10]∨[0,1]
39k	{P1,P2,P5}	[3,10]
32k	{P2}	[0,1]

FIG. 5. THE U-CLEAN RELATION RESULTING FROM THE PROJECTION OF PERSONc ON
THE ATTRIBUTE INCOME

```
SELECT Income, DisjC(C),DisjP(P)
FROM Personc
GROUP BY Price
```

which returns a U-clean relation that is shown in Figure 5.Another example query type is to extract the clean instance corresponding to a given parameter setting. For example, the repair of Relation Person (Figure 4) corresponding to the parameter value _ = 4.1 can be retrieved using the following SQL query:

SELECT ID, Name, ZIP, BirthDate, Income

FROM Personc

WHERE Contains(P, 4.1)

which results in the repair represented by the c-records {CP2,CP4, CP5}.In [3,4], we introduce other query types such as reporting themost probable repair, obtaining the probability of clutteringspecific records, and retrieving c-records with probabilitiesabove a given threshold.

V. DEMONSTRATION OUTLINE

Our system is mainly composed of two components: (1) an application that implements the uncertain cleaning algorithm sand provides a GUI for submitting user queries, and (2) a modified version of the Postgre SQL database system [2] that includes the ADTs and UDFs defined in Section IV-A to allow storing U-clean relations and executing user queries. In our demonstration, we illustrate the process of uncertain duplicate detection, and querying the U-clean relations.

A. Constructing U-clean Relations

We demonstrate the process of cleaning an input relation using an uncertain duplicate detection algorithm. The user provides specifications for the cleaning process which are used for generating an output U-clean relation. We describe our demonstration scenario as follows.

1) The user selects the source of the unclean data, and the name and location of the output U-clean relation.

2) The user specifies the method that computes the similarity between pairs of records. A distance function is defined for each attribute in the unclean relation. The overall distance between two records is computed by aggregating the individual attributes distances [10](e.g. using the functions sum, max or a linear combination of attributes distances). Once the similarity graph is constructed, a visualization of the resulting graph is provided to the user to allow

viewing and editing of the graph edges (Figure 6). Additionally, a histogram of the pair-wise distances is reported to the user.

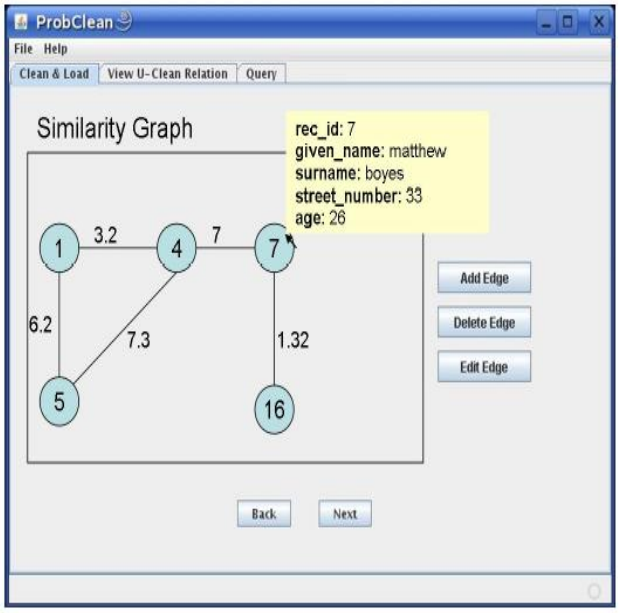

FIG. 6. EDITING SIMILARITY GRAPH

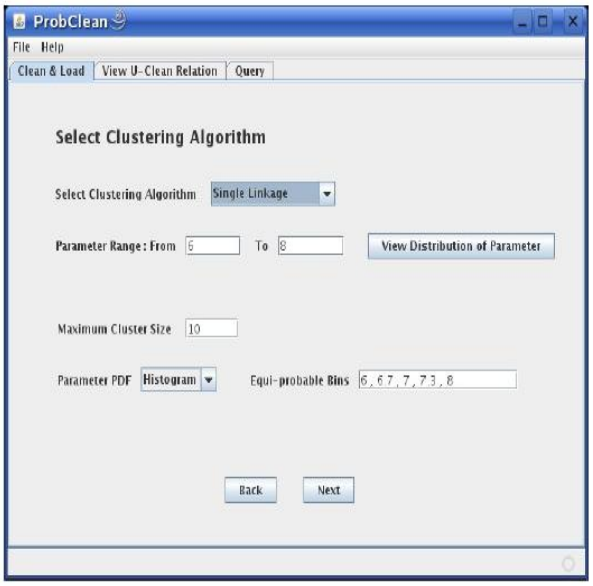

FIG. 7. DEFINING CLEANING SPECIFICATION

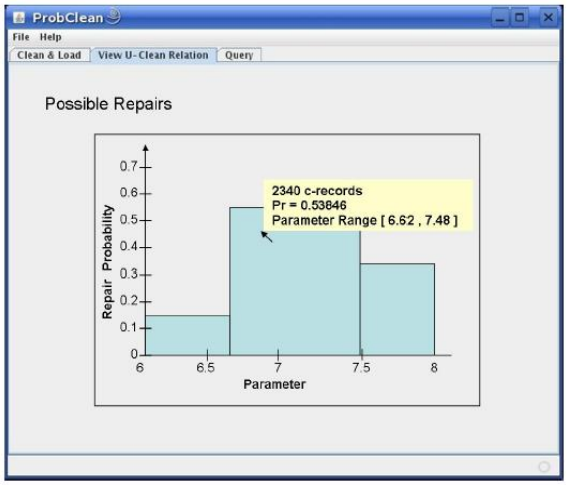

FIG. 8. VIEWING GENERATED REPAIRS

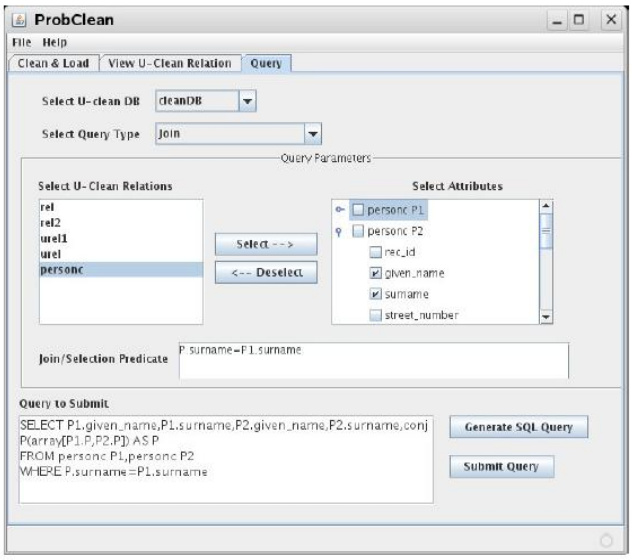

FIG. 9. QUERYING U-CLEAN RELATIONS

3) The user adjusts multiple knobs of the uncertain clustering process such as the used clustering algorithm, the parameter range of the algorithm, and the probability distribution of the parameter. Another optional input isthe maximum cluster size as shown in Figure 7.

4) The user defines how attributes of a cluster are derived from attributes of the member records. After cleaning the input relation, the resulting U-clean relation is presented to the user to view the contained crecords and their corresponding records in the unclean relation. Moreover, the encoded repairs are reported to the user as shown in Figure 8. Statistics about each repair (e.g.,

the repair probability, the number of records and the corresponding range of parameter) are also reported. We also show the running time of each step of the cleaning process to the user.

B. Queries over U-clean Relations

ProbClean supports several query types against U-cleanrelations as shown in Figure 9. Queries are created using aGUI, and then translated into SQL statements that are executedby the modified PostgreSQL DBMS. The user is given theopportunity to edit the generated SQL statements as well asproviding her own SQL statements.

VI. Conclusion

Well at last I can only say that prob clean is one of the most finest tool used for data cleaning, Since it focuses on a specific data transformation task, which is detecting and eliminating duplicate records. Duplicate records are defined as (possibly non-identical) records in theunclean database that refer to the same real world entity. Inthis case, a repair of an unclean database is a clustering of records where records that refer to the same real world entity
are grouped in the same cluster. Moreover this tool can be more effiecient than other available tools since it overcomes the drawbacks of previous tools of data cleaning. In future we can research on this tool more to find how we can make this tool better.

REFERENCES

[1] Oracle data integrator, http://www.oracle.com/technology/products/oracledata-integrator.

[2] PostgreSQL database system, http://www.postgresql.org.

[3] G. Beskales, M. A. Soliman, I. F. Ilyas, and S. Ben-David. Modeling andquerying possible repairs in duplicate detection. In VLDB, 2009.

[4] N. N. Dalvi and D. Suciu.Efficient query evaluation on probabilisticdatabases.VLDB J., 2007.

[5] A. K. Jain and R. C. Dubes.Algorithms for Clustering Data.PrenticeHall College Div, 1988.

[6]Kim,W., Choi, B.-J., Hong, E.-K., Kim, S.-K., & Lee, D. A taxonomy of dirty data, Data Mining and Knowledge Discovery 2003; 7(1):81-99.

[7] Kimball, R. Dealing with Dirty Data, DBMS 1996; 9(10):55-60.

[8] Knorr, E. M. & Ng, R. T. Algorithms for Mining Distance-Based Outliers in Large Datasets. Proceedings of 24th International Conference on Very Large Data Bases; 1998 New York.392-403.

[9] Knorr, E. M., Ng, R. T., &Tucakov, V. Distance-based outliers: algorithms and applications, The International Journal on Very Large Data Bases 2000; 8(3-4):237-253.

[10] Korn, F., Labrinidis, A., Yannis, K., &Faloustsos, C. Ratio Rules: A New Paradigm for Fast, Quantifiable Data Mining. Proceedings of 24th VLDB Conference; 1998 New York.582–593.