1 NOT ALL ANSWERS ARE EQUALLY GOOD: ESTIMATING THE QUALITY OF DATABASE ANSWERS

Amihai Motro, Igor Rakov

Department of Information and Software Systems Engineering
George Mason University
Fairfax, VA 22030-4444
{ami, irakov}@gmu.edu

Abstract: With more and more electronic information sources becoming widely available, the issue of the quality of these often-competing sources has become germane. We propose a standard for rating information products with respect to their quality, and we show how to estimate the quality of answers issued by databases from the quality specifications that have been assigned to these databases. The annotation of answers with their quality provides valuable information to users and is an important new kind of cooperative behavior in databases. We report on preliminary simulations that were carried out to test the validity of our methods.

1.1 INTRODUCTION

Traditionally, database researchers and developers have focused most of their efforts on building systems that are robust, efficient, and flexible. Issues concerning the *quality* of the information products stored in these systems have largely been ignored.[†] With more and more electronic information sources becoming widely available, the issue of the quality of these often-competing sources has

[†] A prominent exception is the incorporation of various mechanisms that control the integrity of the data by requiring that the data satisfy a set of predefined constraints. These mechanisms, however, are limited in their ability to assure that the data stored in the database is indeed accurate.

become germane. In this paper we propose a standard for rating information products with respect to their quality. An important consideration is that the quality of information products often varies considerably when specific areas within these products are considered. This implies that the assignment of a single rating of quality to an information product is usually unsatisfactory. Of course, to the user of an information product the overall quality of the product may not be as important as the quality of the specific information that this user is extracting from the product. Therefore, methods must be developed that will derive reliable estimates of the quality of the information provided to users from the quality specifications that have been assigned to the products.

Our work here bears on all these concerns. We describe an approach that uses dual quality measures that gauge the distance of the information in a database from the truth. We then propose to combine manual verification with statistical methods to arrive at useful estimates of the quality of databases. We consider the variance in quality by isolating areas of databases that are homogeneous with respect to quality, and then estimating the quality of each separate area. These composite estimates may be regarded as *quality specification* that will be affixed to the database. Finally, we show how to derive quality estimates for individual queries from such quality specifications.

An important application of information quality measures is in systems that integrate multiple information sources. Such information sources are often mutually inconsistent, providing different answers to the same query. In such cases, quality ratings of the sources could be used (1) to rank the individual answers according to their quality, or, more ambitiously, (2) to produce an integrated answer with a quality specification.

As commonly perceived, a query answering system is *cooperative* if it goes beyond strict interpretation of queries and attempts to infer and address the intentions behind the queries. In doing so, most cooperative query answering systems attempt to emulate some cooperative trait of human behavior. In [Motro, 1996b] we offered a simple classification of various cooperative techniques. One of the categories used was *explanation and annotation*, which includes techniques that annotate answers with useful information. Two important examples are intensional answers and meta-answers. The former kind refers to the derivation of compact statements that describe extensional answers intensionally; for example, a query on the employees who earn over $60,000 might be answered by (in addition to a set of employee identifiers) a statement such as "all the engineers except Tim". The latter kind is more general and involves the derivation of various properties of answers from overall properties of the database. Two important such properties are *soundness* and *completeness*. As an example, the answer to a bibliographic query might be accompanied by statements that guarantee soundness only for items whose year of publication

is 1990 or later, and completeness only for items published in the USA. In many ways, the annotation of answers with their quality, the subject of this paper, is an elaboration of meta-answers, and would be classified in the same category. Whereas a meta-answer is expressed with definitions of views of the answer whose extensions are guaranteed to be sound or complete (or possibly have some other property), here we would be annotating each answer with its *levels* of soundness and completeness.

There is a growing awareness in the database research community [Chignell, Parsaye, 1993, Firth *& al*, 1995] and among database practitioners [Bort, 1995] of the problem of data quality. By now, the need for data quality metrics and for methods for incorporating them in database systems is well understood. Data quality can be metricized in a number of different ways depending on which aspects of information are considered important [Kon *& al*, 1995, Fox *& al*, 1994]. The addition of data quality capabilities to database systems will enhance decision-making processes, improve the quality of information services, and, in general, provide more accurate pictures of reality. On the other hand, these new capabilities of databases should not be demanding in terms of resources, e.g., they must not add too much complexity to query processing or require much more memory than existing databases. The recent advances in the field of data quality concern data at an attribute value level [Kon *& al*, 1995] and at a relation level [Reddy, Wang, 1995]. The comprehensive survey of the state-of-the-art in the field is given in [Firth *& al*, 1995]. The relational algebra extended with data accuracy estimates based on the assumptions of uniform distributions of incorrect values across tuples and attributes was first described in [Reddy, Wang, 1995].

1.2 OVERALL APPROACH

Our treatment of the problem is in the context of relational databases, and we assume the standard definitions of the relational model [Ullman, 1988]. In addition, we make the following assumptions:

1. Queries and views use only the projection, selection, and Cartesian product operations, selections use only range conditions, and projections always retain the key attribute(s).

2. Database instances are relatively static, and hence the quality of data does not change frequently.

We adopt the relational model for its simplicity, its solid formal foundations, and its widespread popularity. We emphasize, however, that our solutions can be customized to work with other data models.

We begin, in Section 3, by describing the measures that will be used to gauge the quality of database information. We claim that these measures capture in a most natural way the relationship of the stored information to truth, and are therefore excellent indicators of quality.

A given database is *homogeneous* with respect to a quality measure if any view of this database has the same quality. When a database is homogeneous, it would be sufficient to estimate the overall quality of the database; every answer issued by this database would then inherit this quality estimate. In general, however, such homogeneity cannot be assumed. Our approach is to *partition* the given database to a set of views that are homogeneous with respect to the quality measure. This partition is referred to as the *goodness basis* of the database. The quality of the views of the basis is then measured by human research. This process is described in Section 4.

Every answer issued by this database is partitioned by the goodness basis. Every component of this answer partition is contained in some view of the basis, and since the views of the basis are all homogeneous, these answer components inherit their quality ratings from the corresponding basis views. The quality estimates for the different components of the answer partition can then be put together to create a single quality estimate for the entire answer. This process is described in Section 5.

Our methods for discovering a goodness basis and establishing its quality require the authentication of database information, which is a process that needs to be done by humans. However, we advocate the use of statistical methods (essentially, sampling) to keep the manual work within acceptable limits. This subject is discussed in Section 4.1.

Section 6 describes simulations that were carried out to test the vaidity of our method, and Section 7 states our conclusions and directions for future work. Because of space limitations, several key issues and solutions are only sketched in this paper, and detailed discussions are provided in [Motro, Rakov, 1996].

1.3 SOUNDNESS AND COMPLETENESS AS MEASURES OF DATA QUALITY

We define two measures of data quality that are general enough to encompass many existing measures and aspects of data quality [Fox *& al*, 1994, Firth *& al*, 1995]. The basic ideas underlying these measures were first stated in [Motro, 1989]. In that paper the author suggested that declarations of the portions of the database that are known to be perfect models of the real world (and thereby the portions that are possibly imperfect) be included in the definition of each database. With this information, the database system can *qualify* the answers it issues in response to queries: each answer is accompanied by statements

that define the portions of the answer that are guaranteed to be perfect. This approach uses *views* to specify the portions of the database or the portions of answers that are perfect models of the real world.

More specifically, this approach interprets information quality, which it terms *integrity*, as a combination of *soundness* and *completeness*. A database view is sound if it includes *only* information that occurs in the real world; a database view is complete if it includes *all* the information that occurs in the real world. Hence, a database view has integrity, if it includes the whole truth (completeness) and nothing but the truth (soundness). A prototype database system that is based on these ideas is described in [Motro, 1996a]. These ideas were further developed in [Motro, 1993] and are summarized below.

For every database scheme D, we assume two database instances. One, denoted d, is the information presently stored in the system (the *stored* database). The other, denoted d_0, is a hypothetical database instance that captures perfectly that portion of the real world that is modeled by D (the *true* database). The stored instance d is therefore an *approximation* of the true instance d_0.

Given a view V, we denote by v_0 its extension in the true database d_0 (the *true* extension to V) and we denote by v its extension in the stored database d. Again, the stored extension v is an *approximation* of the true extension v_0.

By assigning the stored extension a value that denotes how well it approximates the true extension, we denote the quality of the stored extension. We shall term this value the *goodness* of the extension. In general, we require that the goodness of each extension be a value between 0 and 1, that the goodness of the true extension be 1, and that the goodness of extensions that are entirely disjoint from the true extension be 0. Formally, a *goodness measure* is a function g on the set of all possible extensions that satisfies

$$\begin{aligned} &\forall v: \quad g(v) \in [0,1] \\ &\forall v: \quad v \cap v_0 = \emptyset \implies g(v) = 0 \\ &g(v_0) = 1 \end{aligned}$$

Consider view V, its true extension v_0, and an approximation v. If $v \supseteq v_0$, then v is a *complete* extension. If $v \subseteq v_0$, then v is a *sound* extension. Obviously, an extension which is sound and complete is the true extension. With these definitions, each view extension is either complete or incomplete, and either sound or unsound.

A simple approach to goodness is to consider the intersection of the extensions; that is, the tuples that appear in both v and v_0. Let $|v|$ denote the number of tuples in v. Then

$$\frac{|v \cap v_0|}{|v|}$$

expresses the proportion of the database extension that appears in the true extension. Hence, it is a measure of the *soundness* of v. Similarly,

$$\frac{|v \cap v_0|}{|v_0|}$$

expresses the proportion of the true extension that appears in the database extension. Hence, it is a measure of the *completeness* of v.

It is easy to verify that soundness and completeness satisfy all the requirements of a goodness measure.† Soundness and completeness are similar to *precision* and *recall* in information retrieval [Salton, McGill, 1983].

The use of $|v \cap v_0|$ in both measures implies that only tuples that are *identical* to true tuples contribute to soundness and completeness. For example, tuples that are correct in all but one attribute, and tuples that are incorrect in all their attributes are treated identically: both do not contribute to the goodness measures. An essential refinement of these measures is to consider the goodness of individual attributes.

Assume a view V has attributes $A_0, A_1, \ldots, A_n$, where A_0 is the key.‡ We decompose V into n key-attribute pairs (A_0, A_i) $(i = 1, \ldots, n)$, and then decompose each extension of V into the corresponding value pairs. We call this the *decomposed extension* of V. Using decomposed extensions in the previously-defined measures improves their usefulness considerably, and we shall assume decomposed extensions throughout.

Soundness and completeness can also be approached by means of probability theory [Motro, Rakov, 1996]. For example, the definition of soundness can be interpreted as the probability of drawing a correct pair from a given extension. Probabilistic interpretations give new insight into the notions of soundness and completeness and also help us to connect this research with a large body of work on uncertainty management in information systems [Motro, Smets, 1996].

The data quality measures that have been mentioned most frequently as essential are accuracy, completeness, consistency, and currentness [Fox *& al*, 1994, Kon *& al*, 1995]. In general, we find that the classification and analysis of quality measures has not been sufficiently rigid. Of these four, the former two correspond to our own soundness and completeness measures, although until now their treatment by information quality researchers has been mostly informal, and their duality has not been recognized .

Consistency is a requirement that different sources of overlapping information do not contradict each other; the sources (which may be within a single database or in different databases) may be two sets of data, two sets of constraints, or a set of data and a set of constraints. Clearly, inconsistency is

†When v is empty, soundness is $0/0$. If v_0 is also empty then soundness is defined to be 1; otherwise it is defined to be 0. Similarly for completeness, when v_0 is empty.

‡ We consider a tuple as a representation of the real world entity identified by a key attribute; the nonkey attributes then capture the properties of this entity. For simplicity, we assume that keys consist of a single attribute.

evidence to lack of soundness. Currentness concerns the temporal aspect of the information; to consider currentness, information must be stamped with its time of validity. Currentness of information may be used to resolve inconsistencies among contradicting sources. Also, by comparing timestamps to the present time, quality estimates could be adjusted automatically (using appropriate assumptions on the rate of degradation).

In our opinion, only soundness and completeness should be used for *rating* the quality of sources. Other aspects, such as consistency or currentness, are useful indicators that can be used to establish more accurate soundness and completeness ratings. The model we present here is based solely on soundness and completeness. The proper incorporation of other aspects of quality into this model requires additional research.

1.4 RATING THE QUALITY OF DATABASES

1.4.1 Necessary Procedures for Goodness Estimation

The amount of data in practical databases is often large. To calculate the exact soundness and completeness of a database we would need to (1) authenticate every value pair in the stored database, and (2) determine how many pairs are missing from this database. This method is clearly infeasible in any real system. Thus, we must resort to sampling techniques [Thompson, 1992, Cochran, 1963].

Sampling allows us to estimate the mean and variance of a particular parameter of a population by using a sample whose size is usually only a fraction of the size of the entire population. The theory of statistics also gives us methods for establishing a sample size needed to achieve a predetermined accuracy of the estimates. It is then possible to supplement our estimates with confidence intervals. For more detailed discussion on sampling from databases the reader is referred to the literature on the topic (see, for example, [Olken, Rotem, 1995] for a good survey).

Note that two different populations must be sampled. To estimate soundness we sample the *stored* database, whereas to estimate completeness we sample the *true* database.

To establish both soundness and completeness it is necessary to have access to the true database. For soundness, we need to determine whether a specific value pair from the stored database is in the true database. For completeness, it is necessary to determine whether a specific pair from the true database is in the stored database. These tasks require two procedures (verify a given pair against the true database and retrieve an arbitrary pair from the true database) that must be implemented in an ad-hoc manner requiring human expertise [Bort, 1995]. The expert will access a variety of available sources to

perform these two procedures. Note that this effort is performed only once and only for a sample, which then helps estimate the overall goodness.

A critical stage of our solution is to build a set of homogeneous views on a stored database, called a *goodness basis*. The goodness of the views of this basis will be measured and thereafter used in establishing the goodness of answers to arbitrary queries against this database. Since we cannot guarantee a single set of views that will be homogeneous with respect to both quality measures, we construct two separate sets: a soundness basis and a completeness basis. In constructing each basis, we consider each database relation individually. Each relation may be partitioned both horizontally (by a selection) and vertically (by a projection), and the basis comprises the union of all such partitions. Selections are limited to ranges; i.e., the selection criteria is a conjunction of conditions, where each individual condition specifies an attribute and a range of permitted values for this attribute.

To aid the derivation of each goodness basis, we define for a given relation extension a data structure, called *relation map*, that records the distribution of errors in this extension. A relation map is a two-dimensional matrix of 0s and 1s, in which rows correspond to the tuples and columns correspond to the attributes of the relation. A value in the cell at the intersection of row t and column A is 1 if and only if the pair $(t.A_0, t.A)$ (where A_0 is the key attribute of the relation) is correct (with respect to some reference relation); otherwise it is 0. The task, now, is to partition this two-dimensional array into areas in which elements are distributed homogeneously.

Note that the correctness of a particular nonkey attribute value can be determined only in reference to the key attribute of that tuple. The pair is correct if and only if both elements of the pair are correct. This means, in particular, that if a key attribute value is incorrect, then all pairs corresponding to this key attribute value are considered incorrect.

The technique we use for partitioning the relation map is a nonparametric statistical method called CART (Classification and Regression Trees) [Breiman *& al*, 1984]. This method has been widely used for data analysis in biology, social science, environmental research, and pattern recognition. Closer to our area, this method was used in [Chen *& al*, 1990] for estimating the selectivity of selection queries.

1.4.2 Homogeneity Measure

Intuitively, a relation is perfectly homogeneous with respect to a given property if every subview of the relation contains the same proportion of pairs with this property as the relation itself. Moreover, the more homogeneous a relation, the closer its distribution of the pairs with the given property is to the distri-

bution in the perfectly homogeneous relation. Hence, the difference between the proportion of the pairs with the given property in the relation itself and in each of its views can be used to measure the *degree* of homogeneity of the given relation.

Specifically, let v denote an extension of a relation, let $v_1, \ldots, v_N$ be the set of all possible projection-selection views of v, let $p(v)$ and $p(v_i)$ denote the proportion of pairs in v and v_i $(i = 1, \ldots, N)$, respectively, that are correct (with respect to some reference relation). Then

$$\frac{1}{N} \sum_{v_i \subseteq v} (p(v) - p(v_i))^2$$

measures the homogeneity of v. Similar measures of homogeneity were proposed in [Kamel, King, 1993, Chen *& al*, 1990].

Due to the large number of possible views, computation of this measure is often prohibitively expensive. The *Gini index* [Breiman *& al*, 1984, Chen *& al*, 1990] was proposed as a simple alternative to this homogeneity measure.

Let M be a relation map and consider a view v of that relation. We call the part of M that corresponds to v a *node*.† The Gini index of this node, denoted $G(v)$, is $2p(1-p)$, where p now denotes the proportion of 1s in the node.‡

The search for homogeneous nodes involves repeated splitting of nodes. The Gini index guarantees that *any* split improves (or maintains) the homogeneity of descendant nodes [Breiman *& al*, 1984]. Formally, let v be a node which is split into two subnodes v_1 and v_2. Then $G(v) \geq \alpha_1 G(v_1) + \alpha_2 G(v_2)$, where α_i is $|v_i|/|v|$. In other words, the *reduction* of a split, defined as $\Delta G = G(v) - \alpha_1 G(v_1) - \alpha_2 G(v_2)$, is guaranteed to be nonnegative.

Obviously, the best split is a split that maximizes ΔG. We call such a split a *maximal split*. If the number of possible splits is finite, there necessarily exists such a split. The method of generating a homogeneous partition is founded on the search for a split that maximizes the gain in homogeneity. This method is discussed next.

1.4.3 Finding a Goodness Basis

Finding a homogeneous partition of a given relation is a tree-building procedure, where the root node of the tree is the entire relation, its leaf nodes are homogeneous views of this relation, and its intermediate nodes are views produced by the searches for maximal splits. We start by labeling the entire relation map as the root of the tree. We then consider all the possible splits, either horizontal or vertical (but not both), and select the split that gives maximum gain

†We use the terms node and view interchangeably.
‡In general, the Gini index is defined for maps whose elements are of k different types; the index used here is much simpler, because our maps are binary.

in homogeneity. Obviously, the brute-force technique described here is very expensive. In practice we apply several substantiative improvements [Motro, Rakov, 1996].

When the maximal split is found, we break the root node into the two subnodes that achieved the maximal split. Next, we search for a maximal split in each of the two subnodes of the root and divide them in two descendant nodes each. The procedure is repeated on each current leaf node of the tree until a heuristic stop-splitting rule is satisfied on every leaf node: splitting of a node stops when it can provide only marginal improvement in homogeneity. This situation usually arises when a maximal split on a node cannot separate elements of one type (1s) from elements of the other type (0s) in this node. This indicates that this node has a fairly homogeneous distribution of both types of elements.

The stop-splitting rule mentioned earlier is necessary, because otherwise a tree could grow until all the elements of every leaf are of one type. This could result in a large number of small nodes. Also, since the relations being considered are usually samples, it might mean that the measurements made on the nodes would not be statistically reliable. Our stop-splitting rule is $\Delta G \cdot n \geq threshold$, where n is the number of elements in the node.

So far we have assumed that the given relation has been assigned a map that indicates the correctness of its elements (with respect to some reference relation), and we have shown how to partition this relation to a set of views that are homogeneous with respect to this correctness. When the given relation is a stored database and the reference relation is the true database, then the property of correctness is indeed the soundness of the stored database, and the resulting set of homogeneous views is a *soundness basis*. When the given relation is a the true database and the reference relation is the stored database, then the property of correctness is indeed the completeness of the stored database, and the resulting set of homogeneous views is a *completeness basis*.

Such soundness and completeness trees are constructed for every relation of the database. Each leaf node of every soundness tree contributes one view to the soundness basis and each leaf node of every completeness tree contributes one view to the completeness basis. Together, these soundness and completeness bases form a *goodness basis*. Recall that the assumption here is that the information is static, so this process is performed only once on every relation, and the goodness basis need not be changed or updated later. When a leaf node is converted to a view, in addition to the rows and columns of the node, the view includes the key attribute for these tuples.

It is important to remember that the procedures discussed above are performed on *samples* of the relations. Therefore, the terms *relation* and *relation*

map refer to samples of the relations and maps of these samples. Although the algorithm is applied to the samples, the resulting views are later used as a goodness basis for the entire relation. Care should be taken to ensure that we draw samples whose sizes are sufficient for representing distribution patterns of the original relation. Once samples are drawn (from either the stored or the true database), the correctness of their elements is established and recorded in the corresponding relation maps. Once a goodness basis is obtained, the quality of each basis view (i.e., soundness or completeness, as appropriate) is calculated from the sample and serves to estimate the quality of the view on the entire database. A goodness basis with the associated goodness ratings of each of its views will be referred to as a *measured* goodness basis.

1.5 ESTIMATING THE QUALITY OF ANSWERS

Assume now a query is submitted to a database for which a goodness basis has been obtained. We begin by considering selection-projection queries on a single relation and Cartesian product queries on two relations. We conclude by considering general queries that consist of sequences of operations of these two kinds. Our discussion focuses on the estimation of answer soundness; the considerations for estimating completeness are nearly identical.

Because a basis partitions each relation, an answer to a query intersects with a certain number of basis views. Hence, each of these basis views contains a component of the answer as its subview. The key feature of basis views is their homogeneity with respect to soundness or completeness. Consequently, each component of the answer inherits its soundness or completeness rating from a basis view. As claimed by Proposition 1 (see [Motro, Rakov, 1996] for proof), the soundness of a view that comprises disjoint components is a weighted sum of the soundness of the individual components. This provides us with a simple way to determine the soundness of the entire answer.

Proposition 1 *Let t_1 and t_2 be leaf nodes of a soundness tree with soundness s_1 and s_2 respectively, and let q be an answer to a query Q. Suppose also that $q = (q \cap t_1) \cup (q \cap t_2)$. The soundness of q is*

$$s(q) = s_1 \cdot \frac{|q \cap t_1|}{|q|} + s_2 \cdot \frac{|q \cap t_2|}{|q|}$$

This proposition is easily generalized for n leaf nodes, and the analogous proposition is true for completeness. In practice, we only have estimates $\hat{s_1}$ and $\hat{s_2}$ of s_1 and s_2. Hence, the formula becomes:

$$\hat{s}(q) = \hat{s_1} \cdot \frac{|q \cap t_1|}{|q|} + \hat{s_2} \cdot \frac{|q \cap t_2|}{|q|}$$

The variance of the estimate $\hat{s}(q)$ can be also calculated [Motro, Rakov, 1996].

To allow more general queries, we consider now queries that include Cartesian products. Proposition 2 (see [Motro, Rakov, 1996] for proof) describes how to calculate the soundness and completeness of the Cartesian product given the soundness and completeness of its operands.

Proposition 2 *Let r_1 and r_2 be relations with soundness and completeness s_1, c_1 and s_2, c_2 respectively. The soundness and completeness of the $r_1 \times r_2$ are*

$$s(r_1 \times r_2) = \frac{k \cdot s_1 + p \cdot s_2}{k + p}, \quad c(r_1 \times r_2) = \frac{k \cdot c_1 + p \cdot c_2}{k + p}$$

respectively, where k and p are the number of nonkey attributes in the relations r_1 and r_2 respectively.

In practice, we have only estimates of the soundness and completeness, and the formulas from the proposition become:

$$\hat{s}(r_1 \times r_2) = \frac{k \cdot \hat{s}_1 + p \cdot \hat{s}_2}{k + p}, \quad \hat{c}(r_1 \times r_2) = \frac{k \cdot \hat{c}_1 + p \cdot \hat{c}_2}{k + p}$$

where $\hat{s}_1$, $\hat{s}_2$, $\hat{c}_1$, $\hat{c}_2$ are estimates for soundness and completeness of the corresponding relations. For derivation of the variance of the estimates see [Motro, Rakov, 1996].

So far we have shown how to estimate the soundness and completeness of selection-projection queries on a single relation, and of Cartesian products of two relations. To calculate soundness and completeness of arbitrary Cartesian product-selection-projection queries it is necessary to show how to calculate goodness estimates over sequences of relational algebra operations.

Because every individual operation assumes that each of its input relations has an associated measured goodness basis (i.e., a soundness basis with soundness estimates for each of its views and a completeness basis with completeness estimates for each of its views) to perform a sequence of such operations it is necessary that every operation also delivers a measured goodness basis for its result.

Indeed, this amounts to a *generalization* of the relational algebra. Conventionally, the input of each relational algebra operation is a set of relations (possibly just one relation), and the output is a relation. Our generalization extends this so that each relational algebra operation receives as input a set of relations with their measured goodness bases and delivers as output a relation with its measured goodness basis. In other words, the elements of the algebra are generalized from relations to relations with quality information (measured goodness bases), and all operations are generalized to receive and deliver these

generalized elements. A correct definition of the operations requires that when two equivalent relational algebra expressions are attempted, the final goodness estimates would be the same. In [Motro, Rakov, 1996] we show that this indeed is the case.

The output of the final operation is a relation and its measured goodness basis. The overall goodness ratings of the entire answer may then be calculated using weighted sums. The information about the soundness and completeness of individual portions of the result may be presented to users who require additional information, or when the quality of the result is particularly nonhomogeneous.

1.6 EXPERIMENTATION

We conducted a series of preliminary experiments to test our approach to the measurement of information quality. The purpose of the experiments was to verify the performance of the approach as well as analyze the sensitivity of this method to various parameters, such as distributions of incorrect data elements, types of queries, and threshold values.

The design of the experiment, which tested only soundness of selection queries, was to take a relation with a perfectly known distribution of incorrect data elements, draw a sample from it, and build a soundness basis from this sample. After that we issue a set of selection queries against the relation and compare the estimates of the soundness of the queries as calculated by our methods with the actual soundness of the queries. Note that the experiment for estimating completeness would be similar. We would assume that the true relation is available to us along with the distribution of data elements missing from the stored relation. We would draw a sample from the true relation, build a completeness basis, and proceed in the same way as with the estimation of soundness.

We used in the implementation the Oracle 7 relational database running in a Unix environment. The algorithms were written in C-Embedded SQL (Oracle Pro*C). For the experiments we constructed a relation *(tid, tvalue1, tvalue2)* with 1000 tuples. The first attribute (*tid*) is the key; the other two attributes (*tvalue1* and *tvalue2*) hold arbitrary values from the domain of integers between 0 and 999. This relation was extended with two binary attributes (*tvalid1* and *tvalid2*) that specified whether the corresponding values of *tvalue1* and *tvalue2* are correct or incorrect. These auxiliary attributes are used only for calculation of the soundness estimates.

Part of our experiment was aimed at testing whether our methods are affected by the distribution of the errors in the given relation. In this case, we repeated the experiment with three different error distributions. Each er-

ror distribution reflects a different collection of regions with different quality. Table 1.1 shows each distribution as a collection of regions. As an example, Distribution 1 consists of four regions: the pairs with *tid* 0–250 and 501–750 and attribute *tvalue1* (500 pairs in all) make up one homogeneous region (whose soundness is 1.0); the pairs with *tid* 251–500 and 751-999 and attribute *tvalue1* (500 pairs in all) make up another homogeneous region (with soundness 0.5); the pairs with *tid* 0–250 and 501–750 and attribute *tvalue2* (500 pairs in all) make up a third homogeneous region (whose soundness is 0.5); and the pairs with *tid* 251–500 and 751–999 and attribute *tvalue2* (500 pairs in all) make up the fourth homogeneous region (with soundness 1.0).

The size of the sample drawn from the relation was determined by standard statistical formulas [Cochran, 1963]. In particular, the sample size was selected such that the error of the soundness estimate would not be larger than 5% (with probability 0.95). We then built a soundness basis of this relation by applying the algorithm discussed in Section 4.3 to this sample. This algorithm applies a threshold that controls the sensitivity of the stop-splitting rule, and we repeated this procedure with different threshold values.

For every distribution and every soundness basis we submitted 300 selection queries as follows. 100 values of *tid* were selected at random from the domain $[0, 999]$ and 3 range queries were constructed around each of these values, with ranges containing 100, 200, and 400 values. We compared the soundness estimates calculated using the soundness basis against the actual soundness of the answers to the queries which was calculated directly from the relation using the auxiliary attributes. The results of the experiments are presented in Table 1.2. This table groups the experiments according to the basis and the type of query. The average relative error measures the success of our methods: it reflects the error in our estimation of soundness when compared with the actual soundness rate. For example, the average relative error of the queries that ranged over 100 items, submitted against the relation with the first distribution of errors, and using the soundness basis with threshold 0.5, was 11.11%.

In general, we observe that the accuracy of the estimates for larger ranges is higher than that for smaller ones. This is due to the fact that the larger range includes more data elements (sampling points) thereby producing more accurate estimates. See [Hou, Ozsoyoglu, 1991] for a more theoretical discussion on this subject.

Care should be taken in choosing the threshold value. The experiments show that if this value is too small or too large, the accuracy of the estimates calculated from that basis decreases. If the value of the threshold is too large, the basis-building process will stop early, producing a basis too crude to reflect the actual distribution pattern in the relation. If the value of the threshold is too small, the resulting basis will consist of too many small nodes. The small

nodes will contain too few sampling points and therefore could not predict the actual soundness reliably. Finding a good threshold value requires some experimentation with the distribution at hand.

Clearly, the size of the sample plays a significant role in the performance of the soundness basis. Preliminary results show that increases in the size of the sample tend to improve the accuracy of the results. This is especially true for highly nonhomogeneous distributions of the correct and incorrect data elements. In a further experiment we used the third distribution with the threshold value 0.5. Samples of different sizes were drawn from the relation and in each case a soundness basis was built. After that the same set of 300 queries was submitted, and accuracy of the soundness estimates for each soundness basis was measured. The results are summarized in Table 1.3. As expected, the conclusion is that estimates tend to improve with sample size.

We note that our simulation differs from a field experiment in that we used a synthetic database for which the distribution of "errors" was predetermined. This had two advantages. First, manual authentication for the samples was not required, and, second, it was possible to calculate the actual measures of soundness and hence to estimate the success of our methods. We note, however, that field experiments are still important, because they will demonstrate whether our methodology for establishing quality specifications of databases (essentially, the part that requires the authentication of the data in the sample) is feasible.

1.7 CONCLUSIONS AND FUTURE RESEARCH

We introduced a new model for data quality in relational databases, which is based on the dual measures of soundness and completeness. The purpose of this model is to provide answers to arbitrary queries with an estimation of their quality. We achieved this by adopting the concept of a basis, which is a partition of the database into views that are homogeneous with respect to the goodness measures. These bases are constructed using database samples, whose goodness is established manually. Once the bases and their goodness estimates are in place, the goodness of answers to arbitrary queries is inferred rather simply.

We plan to develop the complete set of procedures for calculating soundness and completeness of the answers to other relational algebra operations; i.e., add procedures for union, difference, and intersection of views. One of our major goals is to use these methods to estimate the goodness of answers to queries against multidatabases, where the same query could be answered differently by different databases, and goodness information can help resolve such inconsistencies.

Distribution 1

tid	soundness1	soundness2
0-250	1.00	0.5
251-500	0.50	1.0
501-750	1.00	0.5
751-999	0.50	1.00

Distribution 2

tid	soundness1	soundness2
0-250	1.00	0.25
251-500	0.75	1.0
501-750	1.00	0.5
751-999	0.25	0.75

Distribution 3

tid	soundness1	soundness2
0-100	1.00	0.75
101-200	0.50	1.00
201-300	1.00	0.75
301-400	0.50	0.50
401-500	1.00	1.00
501-600	0.75	0.75
601-700	1.00	0.50
701-800	0.50	1.00
801-900	1.00	1.00
901-999	0.25	1.00

Table 1.1 The distributions of correct data elements in the relation.

Distribution 1

threshold value	query range	avg. relative error (%)
0.5	100	11.11
	200	6.94
	400	3.83
0.8	100	12.82
	200	9.02
	400	5.02
1.0	100	9.11
	200	8.78
	400	5.79

Distribution 2

threshold value	query range	avg. relative error (%)
0.3	100	13.26
	200	7.23
	400	3.84
0.5	100	7.52
	200	6.54
	400	4.76
0.8	100	7.96
	200	6.54
	400	4.76
1.0	100	8.36
	200	6.46
	400	4.97
1.5	100	8.03
	200	6.79
	400	5.16

Distribution 3

threshold value	query range	avg. relative error (%)
0.1	100	11.48
	200	6.65
	400	5.54
0.3	100	9.32
	200	5.82
	400	5.06
0.5	100	10.08
	200	5.36
	400	4.54
0.8	100	13.56
	200	7.39
	400	6.67
1.0	100	15.74
	200	7.97
	400	7.33
1.2	100	14.20
	200	10.07
	400	7.99

Table 1.2 The results of the experiment for the three distributions.

sampling rate (%)	query range	avg. relative error (%)
5	100	11.74
	200	9.65
	400	5.85
10	100	10.81
	200	7.31
	400	5.46
20	100	7.87
	200	4.70
	400	2.87
30	100	2.85
	200	1.97
	400	1.41

Table 1.3 The results of the experiment for different sampling rates.

We have already discussed the advantage of considering the correctness of individual attributes over the correctness of entire tuples. Still, an individual value is either correct or incorrect, and, when incorrect, we do not consider the proximity of a stored value to the true value. This direction, which is closely related to several uncertainty modeling techniques, merits further investigation.

Because of the cost of establishing goodness estimations, we have noted that our methods are most suitable for static information. When the information is dynamic, it would be advisable to timestamp the estimations at the time that they were obtained and attach these timestamps to all quality inferences. One may also consider the automatic attenuation of quality estimations as time progresses. This direction is still outside our immediate objectives.

References

[Bort, 1995] J. Bort. Scrubbing dirty data. *InfoWorld*, 17(51), December 1995.

[Breiman *& al*, 1984] L. Breiman, J. Friedman, R. Olshen, and Ch. Stone. *Classification and Regression Trees.* Wadsworth International Group, 1984.

[Chen *& al*, 1990] M. C. Chen, L. McNamee, and N. Matloff. Selectivity estimation using homogeneity measurement. In *Proceeding of the International Conference on Data Engineering*, 1990.

[Cochran, 1963] W. Cochran. *Sampling Techniques.* John Wiley & Sons, 1963.

[Fox *& al*, 1994] C. Fox, A. Levitin, and T. Redman. The notion of data and its quality dimensions. *Information processing and management*, 30(1), 1994.

[Hou, Ozsoyoglu, 1991] W. C.Hou and G. Ozsoyoglu. Statistical Estimators for aggregate relational algebra queries. *ACM Transactions on Database Systems*, 16(4):600–654, 1991.

[Kamel, King, 1993] N. Kamel and R. King. Exploiting data distribution patterns in modeling tuple selectivities in a database. *Information Sciences*, 69(1-2), 1993.

[Motro, 1989] A. Motro. Integrity = validity + completeness. *ACM Transactions on Database Systems*, 14(4):480–502, December 1989.

[Motro, 1993] A. Motro. A formal framework for integrating inconsistent answers from multiple information sources. Technical Report ISSE-TR-93-106, Dept. Information and Software Systems Engineering, George Mason University, 1993.

[Motro, 1996a] A. Motro. Panorama: A database system that annotates its answers to queries with their properties. *Journal of Intelligent Information Systems*, 7(1), 1996.

[Motro, 1996b] A. Motro. Cooperative Database Systems. *International Journal of Intelligent Systems*, 11(10):717–732, October 1996.

[Motro, Smets, 1996] A. Motro and Ph. Smets, editors. *Uncertainty Management in Information Systems: From Needs to Solutions*. Kluwer Academic Publishers, 1996.

[Motro, Rakov, 1996] A. Motro and I. Rakov. On the specification, measurement, and inference of the quality of data. Technical report, Dept. Information and Software Systems Engineering, George Mason University, 1996.

[Olken, Rotem, 1995] F. Olken and D. Rotem. Random sampling from databases—a survey. *Statistics and Computing*, 5(1), 1995.

[Chignell, Parsaye, 1993] K. Parsaye and M. Chignell. *Intelligent Database Tools and Applications*. John Wiley & Sons, 1993.

[Reddy, Wang, 1995] M. P. Reddy and R. Wang. Estimating data accuracy in a federated database environment. In *Proceedings of CISMOD*, 1995.

[Salton, McGill, 1983] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, New York, 1983.

[Thompson, 1992] S. Thompson. *Sampling*. John Wiley & Sons, 1992.

[Ullman, 1988] J. D. Ullman. *Database and Knowledge-Base Systems, Volume I*. Computer Science Press, Rockville, Maryland, 1988.

[Kon *& al*, 1995] R. Wang, M. Reddy, and H. Kon. Toward quality data: An attribute-based approach. *Decision Support Systems*, 13(3-4), 1995.

[Firth *& al*, 1995] R. Wang, V. Storey, and Ch. Firth. A framework for analysis of data quality research. *IEEE Transactions on Knowledge and Data Engineering*, 7(4), August 1995.