Estimating the Quality of Data in Relational Databases *

Amihai Motro and 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 sources with respect to their quality. An important consideration is that the quality of information sources often varies considerably when specific areas within these sources are considered. This implies that the assignment of a single rating of quality to an information source is usually unsatisfactory. Of course, to the user of an information source the overall quality of the source may not be as important as the quality of the specific information that this user is extracting from the source. 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 sources. 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 specifications* that will be affixed to each database. Finally, we show how to derive quality estimates for individual queries from such quality specifications.

1 Introduction

The importance of data quality in the information age cannot be overestimated. People, businesses, and governments rely more and more on information in their everyday operations,

*This work was supported in part by DARPA grants N0014-92-J-4038 and N0060-96-D-3202.

and databases of different kinds are the primary source of this information. Our dependence on databases grows simultaneously with their size, yet most large databases contain errors and inconsistencies. There is a growing awareness in the database research community [13, 19] and among database practitioners [1] 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 aspect of information are considered important [18, 5]. 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 [18] and at a relation level [14]. The comprehensive survey of the state-of-the-art in the field is given in [19]. 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 [14].

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 sources with respect to their quality. An important consideration is that the quality of information sources often varies considerably when specific areas within these sources are considered. This implies that the assignment of a single rating of quality to an information source is usually unsatisfactory. Of course, to the user of an information source the overall quality of the source may not be as important as the quality of the specific information that this user is extracting from the source. 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 sources.

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.

2 Overall Approach

Our overall approach for achieving the goals that were stated in the introduction can be described as a sequence of problems.

We begin, in Section 3, by describing the dual 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.

Our measures 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.

Having obtained accurate information about the quality of the samples, we proceed to partition the given database to a set of components that are homogeneous with respect to our quality measures. We then estimate the quality of these components, using the samples. This implies that when information is extracted from a single component, its quality ratings are inherited from the containing component. These methods, described in Sections 4.2 and 4.3, provide us with quality specifications for the given databases.

Finally, in Section 5, we describe the process of inferring the quality of answers to arbitrary queries from the quality specifications that have been assigned to the database.

Our treatment of the problem is in the context of relational databases, and we assume the standard definitions of the relational model [17]. In particular, the database components mentioned earlier are defined using the mechanism of views. We also 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. The stored information (the database instances) are relatively static, and hence the quality of data does not change frequently.

Because of space limitations, several key issues and solutions are only sketched in this paper, and fuller discussions are provided in [11].

3 Soundness and Completeness as Measures of Data Quality

We define two measures of data quality that are general enough to encompass most existing measures and aspects of data quality[5, 19]. The basic ideas underlying these measures were first stated in [7]. 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 accuracy of 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 [10]. These ideas were further developed in [9] and are summarized below.

Given a database scheme D, we assume the existence of a hypothetical database instance d_0 that captures perfectly that portion of the real world that is modeled by D (the *ideal* or *true* database). In addition, we assume one or more actual instances d_i ($i \geq 1$). The actual instances are considered *approximations* of the ideal instance d_0.

Given a view V, we denote by v_0 its extension in the ideal database d_0 (the *ideal* or *true* extension to V) and we denote by v_i its extension in the actual database d_i. Again, the extensions v_i are *approximations* of the ideal extension v_0.

Consider view V, its ideal 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 ideal extension.

With these definitions, each view extension is either complete or incomplete, and either sound or nonsound. We now refine these definitions by assigning each extension a value that denotes how well it approximates the ideal extension. We shall term this value the *goodness* of the extension. We require that the goodness of each extension be a value between 0 and 1, that the goodness of the ideal extension be 1, and that the goodness of extensions that are entirely disjoint from the ideal extension be 0. Formally, a *goodness measure* is a function g on the set of all possible extensions that satisfies

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

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.[1] Soundness and completeness are similar to *precision* and *recall* in information retrieval [15].

A disadvantage of these measures is that a database tuple is assumed to be sound (and contribute to the soundness measure) only if it *identical* to a tuple of the ideal database (similarly in the case of completeness). Thus, a tuple that is correct in all but one attribute, and a tuple that is incorrect in all its attributes are treated identically. 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.[2] 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 [11]. 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 [8].

The data quality measures that have been mentioned most frequently as essential are accuracy, completeness, currentness, and consistency [5, 18]. It is possible to relate these quality measures to our own goodness measures [11].

4 Rating the Quality of Databases

4.1 Necessary Procedures for Goodness Estimation

The amount of data in practical databases is often large. To compute the exact soundness and completeness of a particular view we would need to (1) authenticate every value pair in the stored view, and (2) determine how many pairs are missing from this view. This method is clearly infeasible in any real system. Thus, we must resort to sampling techniques [16, 4].

Sampling techniques allow us to estimate the mean and variance of a particular parameter

[1] 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.

[2] 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.

of a population by using a sample which 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 to achieve 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, [12] for a good survey).

Note that two different populations must be sampled. To estimate soundness we sample the *given* (stored) view, whereas to estimate completeness, we sample the *ideal* view.

To establish both soundness and completeness it is necessary to have access to the ideal database. For soundness, we need to determine whether a specific value pair of the stored database is in the ideal database. For completeness, it is necessary to determine whether a specific pair from the ideal database is in the stored database. These procedures (verify a pair from a stored database against the ideal database and retrieve an arbitrary pair from the ideal database) must be implemented in an ad-hoc manner [1]. For each concrete database, human expertise will be required. 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.

We assign to an incorrect value pair the value 0 and to a correct pair the value 1. Thus, we can represent an error distribution pattern in a view extension as a two-dimensional matrix of 0s and 1s, in which rows correspond to the tuples and columns correspond to the attributes of the view. A value in a particular cell of this matrix is either 0 or 1 depending on the correctness of the corresponding pair of attribute values. We call this new data structure a *view map* or a *relation map*, as appropriate. Now, the task is to partition this two-dimensional array into areas in which elements are distributed homogeneously with respect to our quality measures.

Note that the correctness of a particular nonkey attribute value can be determined only in reference to the key attribute of that tuple, i.e., in determining whether a specific cell should be 0 or 1 we consider the correctness of the pair: *(key value; nonkey value)* determining the correctness of an attribute value. 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 view map is a nonparametric statistical method called CART (Classification and Regression Trees) [2]. 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 [3] for estimating the selectivity of selection queries. We assume that tuples and attributes of a relation are ordered uniquely.

4.2 Homogeneity Measure

Intuitively, a view is perfectly homogeneous with respect to a given property if every subview of the view contains the same proportion of pairs with this property as the view itself. Moreover, the more homogeneous a view, the closer its distribution of the pairs with the given property is to the distribution in the perfectly homogeneous view. Hence, the difference between the proportion of the pairs with the given property in the view itself and in each of its subviews can be used to measure the *degree* of homogeneity of the given view.

Specifically, let $\bar{v}$ denote an extension of a view of a relation in a stored database, let $v_1, \ldots, v_N$ be the set of all possible projection-selection views of $\bar{v}$, let $s(\bar{v})$ and $s(v_i)$ denote the proportion of pairs in views $\bar{v}$ and v_i ($i = 1, \ldots, N$), respectively, that occur in their corresponding ideal representations (i.e., proportions of correct pairs in these views). Then

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

measures the homogeneity of the view $\bar{v}$ with respect to soundness. The homogeneity with respect to completeness is defined analogously. Similar measures of homogeneity were proposed in [6, 3].

Due to the large number of possible views, computation of these measures is often prohibitively expensive. The *Gini index* [2, 3] was proposed as a simple alternative to these homogeneity measures.

Consider a view $\bar{v}$ and a relation map M. We call the part of M that corresponds to $\bar{v}$ a *node*.[3] The Gini index of this node, denoted $G(\bar{v})$, is $2p(1-p)$, where p denotes the proportion of 1s in the node.[4]

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 [2]. Formally, let v be a relation map 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 non-negative.

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

[3] We use the terms node and view interchangeably.

[4] 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.

method of generating soundness and completeness bases is founded on the search for a split that maximizes the gain in homogeneity. This method is discussed next.

4.3 Finding a Goodness Basis

We describe a procedure of building a soundness basis. The procedure of building a completeness basis is similar.

It is important to note that the procedures to be discussed in this section are performed on *samples* of the relations. Therefore, in the discussion that follows, the terms *relation* and *relation map* usually refer to samples of the relations and maps of these samples. Thus, although the best splits are found using only 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.

A soundness basis is a partition of the stored relations, in which each relation is partitioned into views that are homogeneous with respect to soundness. Since the procedure of partitioning is repeated for each relation, it is sufficient to consider this procedure for a single relation. We assume that information on the correctness of a relation instance has already been converted to a corresponding relation map.

Finding a homogeneous partition of a relation can be viewed as 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 call this tree structure a *soundness tree.* 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 in homogeneity. Obviously, the brute-force technique described here is extremely expensive. In practice we apply several, substantiative improvements [11].

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 descendent 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 from elements of the other type in this node. This indicates that this node has a fairly homogeneous distribution of both types of elements.

The stop-splitting rules mentioned earlier are 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. It also means that there might be too few sample elements in this node, which makes the soundness estimate of the node unreliable. Our stop-splitting rule is $\Delta G \cdot n \geq threshold$, where n is the number of elements in the node [11]. An analogous procedure is used for building a completeness tree.

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*. Note that this process is performed only once on every relation, and the goodness basis need not be changed or updated later. The assumption here is that the information is static. 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.

5 Estimating the Quality of Queries

5.1 Projection-Selection Queries

Assume now a query is submitted to this database extension. At this point, we consider only selection-projection queries on a single relation (and in which selections are based on ranges). In this section we discuss the estimation of soundness of such queries. The considerations for estimating completeness are nearly identical. In the next section we discuss queries that involve Cartesian products.

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. Consequently, each component of the answer inherits its soundness from a basis view. As shown in Proposition 1 (see [11] for proof) , the soundness of a view which comprises disjoint components is a weighted sum of the soundness of the individual components. This provides us with an easy way to determine the soundness of the entire answer. As a special case, when the entire answer is contained in a single basis view, the soundness of the answer is simply the soundness of the containing view.

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 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 computed[11].

5.2 Estimating the Goodness of Cartesian Products

To allow more general queries, we consider now queries that include Cartesian products. The following proposition (see [11] for proof) describes how to compute 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*

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

respectively, where k and p are the number of non-key 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:

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

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 [11].

5.3 Estimating the Goodness of General Queries

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 compute soundness and completeness of arbitrary Cartesian product-selection-projection queries it is necessary to show how to compute goodness estimates over sequences of relational algebra operations.

The estimation of each operation in a sequence requires soundness and completeness bases with each view having its associated soundness or completeness estimate. In [11] we extend our methods so that each operation delivers, in addition to a goodness estimate of its result, the necessary bases for future operations. This provides us with the ability to perform sequences of operations.

A legitimate question at this point is whether these estimates depend on the order in which they are computed, i.e., whether the estimates of the goodness of equivalent relational algebra expressions are the same. The answer to this question is that the estimates are independent of the particular expression used [11].

6 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.

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

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

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

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

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

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

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

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

[8] A. Motro. Sources of uncertainty in information systems. In A. Motro and Ph. Smets, editors, *Proceedings of the Workshop on Uncertainty Management in Information Systems: From Needs to Solutions*, 1992.

[9] 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.

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

[11] 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.

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

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

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

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

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

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

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

[19] 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.