

A Framework for Reconciling Attribute Values from Multiple Data Sources

Zhengrui Jiang

College of Business, University of North Alabama, Florence, Alabama 35632,
zjiang@una.edu

Sumit Sarkar

School of Management, University of Texas at Dallas, Richardson, Texas 75083,
sumit@utdallas.edu

Prabuddha De

Krannert School of Management, Purdue University, West Lafayette, Indiana 47907,
pde@purdue.edu

Debabrata Dey

Michael G. Foster School of Business, University of Washington, Seattle, Washington 98195,
ddey@u.washington.edu

Because of the heterogeneous nature of different data sources, data integration is often one of the most challenging tasks in managing modern information systems. While the existing literature has focused on problems such as schema integration and entity identification, it has largely overlooked a basic question: When an attribute value for a real-world entity is recorded differently in different databases, how should the “best” value be chosen from the set of possible values? This paper provides an answer to this question. We first show how a probability distribution over a set of possible values can be derived. We then demonstrate how these probabilities can be used to solve a given decision problem by minimizing the total cost of type I, type II, and misrepresentation errors. Finally, we propose a framework for integrating multiple data sources when a single “best” value has to be chosen and stored for every attribute of an entity.

Key words: data integration; heterogeneous databases; probabilistic databases; data quality; type I error; type II error; misrepresentation error

History: Accepted by Ramayya Krishnan, information systems; received February 15, 2005. This paper was with the authors 1 year and 2½ months for 3 revisions.

1. Introduction and Motivation

Business decisions often require data from multiple sources. As has been widely documented, integrating data from several existing independent databases, however, poses a variety of complex problems. Data integration problems arising from heterogeneous data sources can be divided into two broad categories—schema-level problems and instance-level problems. Schema matching and integration (Larson et al. 1989, Ram and Park 2004) belong to the first category, while problems such as entity identification and matching (Dey et al. 1998b) and data cleaning and duplicate removal (Hernandez and Stolfo 1998) belong to the second category. These issues have been extensively studied, and various solutions proposed. However, after schema integration and entity matching, another set of questions may arise: What should be done if, once all schema-level problems have been resolved, all real-world entities optimally matched, and duplicates removed, we still face two

conflicting data values for the same attribute of a real-world entity? How should we deal with the situation when we merge, for example, customer data stored separately by the sales department and the customer-service department and encounter two different mailing addresses for the same person? How should business decisions be made using such conflicting data? If we are forced to store only a single value for an attribute from a list of possible ones, which value should we choose?

In the real world, attribute value conflicts pose significant problems (Bischoff and Alexander 1997, pp. 167–168). Such conflicts are closely associated with poor data quality, which may result in considerable financial loss (Betts 2001, Tremblay 2002). A forecast by Gartner, Inc. contends that one-fourth of critical data in Fortune 1000 companies will be incomplete or inaccurate through 2007, and Data Warehousing Institute estimates that poor data quality costs businesses \$600 billion each year (Ziff Davis Media 2006). In a

survey of information workers in the United States, United Kingdom, France, and Germany, 75% of those surveyed admitted that they had made wrong decisions because of incorrect data and spent up to 30% of their work time in verifying the correctness of the data (Harris Interactive 2006). There are many specific examples of significant loss due to poor data quality in business situations. A study on retail operations observed that 65% of the inventory records in a leading retailer's database were inaccurate, leading to a 10% loss in profit (Raman et al. 2001). Similarly, 50% to 80% of computerized criminal records in the United States were found to be inaccurate, incomplete, or ambiguous, leading to a social and economic cost of billions of dollars (Strong et al. 1997). An audit conducted by the Department of Veteran Affairs found that about 8% of their master records had attribute value conflicts with the corresponding records in the Social Security Administration, affecting annual compensation and pension payments to the tune of \$376.8 million (Department of Veteran Affairs 1998).

Faced with conflicting attribute values from multiple data sources, it is often impractical to verify the true values. This is because ascertaining a large number of true values could be very costly and time consuming. In a direct-marketing campaign, for example, an organization with two mailing lists may have to manually verify a large number of mailing addresses with conflicting values. Instead, the organization could consider one of three possible approaches. In the first approach, which we call *on-demand computation*, original data sources are separately maintained. When a certain decision problem needs data, all sources are searched, probabilities are estimated for all possible attribute values, and decisions are made based on these estimated probabilities as well as the costs associated with incorrect decisions. The second approach is similar to the first, except that the probabilities for all possible attribute values are estimated periodically and stored in a *probabilistic database* (Dey and Sarkar 1996, Dekhtyar et al. 2001). At decision time, the relevant probabilities are retrieved, and decisions are made based on these probabilities and error costs. The third approach is *deterministic data integration*, in which the "best" values are determined based on some given criteria and stored in a reconciled database. Subsequent information requests are satisfied by searching the reconciled database directly.

Related research falls into two categories. The first category focuses on extending the relational database model and the relational algebra to deal with inconsistent attribute values. For example, Raju and Majumdar (1988) generalize the basic relational concepts by using fuzzy relations. Dey and Sarkar

(1996) and Dekhtyar et al. (2001) propose probabilistic relational models that incorporate the probabilities associated with possible data values directly into the relations and extend the relational algebra to handle the additional probability attribute. Lim et al. (1996) extend the relational model and algebra using the Dempster-Shafer theory of evidence to incorporate interval-based uncertainty estimates regarding attribute values. However, despite all the theoretical progress, these extended database models and associated operations are yet to be supported by commercial database systems. Furthermore, even if we have commercial software support in the future, deploying these database systems would require one to reconcile conflicting data sources and estimate the uncertainties associated with attribute values, which is the focus of this paper.

The second category of research focuses on mechanisms to identify systematic causes of attribute-value conflicts (Fan et al. 2001, 2002). The primary goal of these studies is to determine conversion rules based on contextual information available about the heterogeneous data sources (e.g., different units of measurement, different definitions of attributes, etc.). However, such techniques cannot help resolve conflicts generated from nonsystematic causes such as those resulting from recording errors and asynchronous updates.

Given the limitations of prior research, we focus on the on-demand computation and deterministic data integration approaches mentioned earlier to deal with conflicting attribute values from different data sources. These approaches can be easily implemented using existing commercial database systems. Further, we focus on the resolution of nonsystematic causes of conflicts in attribute values. In each approach, the probability assigned to every conflicting value needs to be calculated. Various pieces of information, such as the values of correlated attributes, the time stamps of stored values in different data sources, and the source data reliabilities, may be utilized to estimate the probability distributions. In this research, we estimate the probability for each conflicting attribute value based on the source data reliabilities. We restrict our analyses to attributes with discrete domains.

Our framework provides technology-based solutions for decisions that use data from multiple sources with possibly conflicting attribute values. As mentioned earlier, this framework can be applied to direct marketing, which has emerged as an important part of the U.S. economy. The Direct Marketing Association (2005) estimated the U.S. advertising expenditure in direct marketing to be \$161.3 billion and to have generated \$1.85 trillion in increased sales in 2005. Our framework is, however, quite general and can be used

in many real-world situations besides direct marketing. The following are some examples:

- In a modern health-care system, patients' medical histories are created by reconciling data from different sources (e.g., multiple clinics, pharmacies, and insurance agencies) for clinical outcome analyses, cost management, and easy access to medical information (Hegg 1998).

- To identify potential terrorist threats, the U.S. government has initiated a project to integrate data on known and suspected terrorists from more than a dozen data sources, with the objective of helping law-enforcement officials and private-sector operators of critical infrastructure facilities in selecting suspects for closer monitoring and for screening visa applicants, cross-border visitors, and airline travelers (Verton 2003).

- Government agencies need to merge data related to natural disasters (damages, missing persons, hospital treatments, etc.) collected from multiple and unreliable sources for managing the government's various relief efforts (Bilke et al. 2005).

- Managing a political campaign requires one to make use of data from various sources (e.g., Department of Motor Vehicles, voter registration, and prior campaign contributions) to organize fund-raising events and GOTV (Get Out The Votes), among others.

- To detect and prevent financial reporting errors and fraud, auditors are often faced with determining the extent to which entries in different documents (e.g., purchase orders, receiving reports, and invoices) are in agreement with one another.

One could view our framework as belonging to a three-layer architecture, where layer 1 represents the true state of the world, layer 2 represents the realizations of parts of the true state, and layer 3 deals with decision making by taking into consideration the noise inherent in these realizations. We illustrate this architecture in Table 1 with the help of examples in several of the problem contexts mentioned above. Similar examples can be easily found in other contexts as well.

There are three main contributions of this study. First, we show in §2 how to derive the probabilities associated with possible attribute values based on the observed values from multiple data sources, along with the reliabilities of these data sources. Second, we demonstrate in §3 how a decision problem can be solved based on the derived probabilities. Third, in §4, we propose a framework for deterministic integration, during which the "best" values are chosen and stored in an integrated database for subsequent use in different decision problems. For this purpose, a standardized procedure to simplify the cost calculation and value selection process is developed in §5, and the results of an extensive set of experiments for validating its performance are presented in §6.

Section 7 provides concluding remarks and future research directions.

2. Computing Attribute Value Probabilities

We first consider the case of a single discrete attribute A . Suppose that there are n data sources S_1, S_2, \dots, S_n . For a particular entity instance, the stored value of attribute A in data source S_k ($k = 1, 2, \dots, n$) is denoted by A_{S_k} . For a variety of reasons, the values stored in these sources may be inaccurate (Dey et al. 1998b). We would like to determine the probability that a specific value a_i is indeed the true value of A , i.e.

$$P(A = a_i | A_{S_1}, \dots, A_{S_n}) \quad \forall i \in \{1, 2, \dots, m\},$$

where m represents the size of the domain of A . To obtain these probabilities, we first estimate the reliability of A in each data source S_k , denoted by $R_{S_k}^A$ ($k = 1, 2, \dots, n$). Statistical sampling methods could be used to estimate these reliabilities. If the inconsistent data sources have been manually reconciled before, we will already have the data to estimate what proportions of the values are incorrect for each attribute in these data sources. In the absence of existing reliability data, we have to sample each data source to find the percentage of incorrect attribute values in that source (Morey 1982).¹ Typically, some cost will be incurred for the sampling. However, we note that this is only a one-time cost and, being amortized over a large number of decisions, is relatively small. Furthermore, unlike in areas where various specific, idiosyncratic aspects of an environment need to be considered, for example, in accounting and auditing applications as discussed by Krishnan et al. (2005), sampling requirements in our context are straightforward and less demanding.

Suppose we find, based either on existing reliability data or sampling, that A is accurate in S_1 80% of the time. Then,

$$R_{S_1}^A = P(A_{S_1} = a_i | A = a_i) = 0.8 \quad \forall i \in \{1, 2, \dots, m\}.$$

The reliability of attribute A in all other data sources can be estimated similarly. We implicitly assume that the reliability of an attribute in each data source is the same for all instances of that attribute. In practice, if additional information such as time stamps or some domain knowledge is available, we may be able to estimate the reliability of an attribute in a data source at a finer level of granularity. However, regardless of what level of granularity is used, the probability expressions we derive in this section can be used with appropriate source reliabilities.

¹ If errors detected during sampling are corrected in the respective databases, the reliability estimates will have to be adjusted accordingly (Moffat 1987, pp. 100–101).

Table 1 Examples Illustrating the Three-Layer Architecture

Context	Layer 1 (True values)	Layer 2 (Realized values)	Layer 3 (Information requirement)
Direct marketing	Name: Rosbert Black Add: 123 McLean St ZIP: 91234 Gender: M Marital status: M DOB: 2/2/1980	Mailing list 1: Bob Black, 123 Maclean Street, 91234, M, M, 2/2/80 Mailing list 2: Robert Back, 123 McLean, 91234, M, S, NULL	Find the names and addresses of all married customers in a region. Application: Promotion of a product designed for a specific group of customers
Health-care management	Name: Patrick Green Add: 234 McLeod St ZIP: 92345 Ins: BCBS Ins Id: QX345 Blood group: O+ DOB: 12/12/1970	Midland clinic: Pat Green, 234 Macleod Street, 92345, Aetna, R457, NULL, 12/12/70 Highland clinic: Patrick Green, 234 McLeod, 92345, BCBS, NULL, O, 12/12/70	Find the list of patients along with their addresses who are insured by a particular insurance company. Application: Identifying payment lead times for different insurance companies
Terrorism prevention	Name: John Doe Add: 123 Main St ZIP: 12345 Police record: Y Immigrant: Y Military training: Y Marital status: S Kids: 0 Member: Al Bin Travel flag: Y	Immigration: John Doe, 123 Main Street, 12345, Y, Y, NULL, NULL, O, Albin, Y FBI: John Doe, 123 Main, 12345, Y, Y, Y, S, O, Al Bin, NULL	Find the list of people who have some military training and have traveled to a specific set of countries in the last six months (Travel Flag = Y). Application: Screening passengers and identifying potential terrorists
Natural disaster recovery	Name: Christoph Burke Add: 777 Broad Ave ZIP: 12344 Blood group: O+ DOB: 2/3/1975 Residence type: Single Marital status: D Kids: 2	Municipal record: Christophe Burke, 777 Broad Avenue, 12344, O, 2/3/75, Single, D, 2 Emergency clinic: Chris Burk, NULL, 12344, O+, 2/3/75, NULL, D, NULL	Find the list of people for a particular blood group near a specific location. Application: Identifying potential blood donors

2.1. Assumptions

We make two assumptions regarding the reliabilities of the data sources:

ASSUMPTION 1. *The errors are independent across different data sources.*

In other words, for any two data sources S_k and S_l ($k \neq l$), the value of attribute A recorded in S_k is not dependent on the value of A recorded in S_l , given the true value of A :

$$P(A_{S_k} = a_j | A = a_i, A_{S_l} = a_h) = P(A_{S_k} = a_j | A = a_i) \quad \forall h, i, j \in \{1, 2, \dots, m\}.$$

This assumption is reasonable as long as the data sources are independently maintained.

ASSUMPTION 2. *The errors are not systematic, and all values other than the true attribute value are equally likely to be recorded in a given data source.*

Mathematically, this is equivalent to

$$P(A_{S_k} = a_j | A = a_i) = \frac{P(A_{S_k} \neq a_i | A = a_i)}{m - 1} = \frac{1 - R_{S_k}^A}{m - 1}, \quad j \neq i \quad \forall k. \quad (1)$$

This is also a reasonable assumption because we may not have sufficient data or resources to precisely estimate the pattern of errors in a data source. In situations where we can obtain the probability of any observed value given a true value, those probabilities can be used instead in our subsequent analyses.

2.2. Obtaining the Probability Distribution

Let $P(A = a_i)$ be the prior probability that attribute A has the value a_i . Then, based on Bayes' formula and Assumption 1, we have

$$P(A = a_i | A_{S_1}, \dots, A_{S_n}) = \frac{P(A_{S_1}, \dots, A_{S_n} | A = a_i)P(A = a_i)}{\sum_{j=1}^m P(A_{S_1}, \dots, A_{S_n} | A = a_j)P(A = a_j)} = \frac{P(A_{S_1} | A = a_i) \times \dots \times P(A_{S_n} | A = a_i)P(A = a_i)}{\sum_{j=1}^m P(A_{S_1} | A = a_j) \times \dots \times P(A_{S_n} | A = a_j)P(A = a_j)}. \quad (2)$$

The values of the terms in (2) can be easily obtained. Some of them are simply the reliabilities of the data sources, and the others can be calculated from (1). If the values of attribute A for a particular entity instance are stored in only a subset of the n data

sources, the conditional probability can be obtained by considering just that subset of data sources and ignoring the rest.

It may appear that, when the size of the domain of an attribute is large, calculating the probabilities associated with all possible values can pose a considerable challenge. Upon further examination, however, we find that the computations need not be repeated for every possible attribute value. Propositions 1 and 2 summarize our analytical findings that lead to computational efficiencies.

PROPOSITION 1. *For a given entity instance, the likelihood ratio for any two values of an attribute, which are not recorded in any of the data sources, is the same as the ratio of their prior probabilities. (The proofs of all propositions are provided in §EC.1 of the online appendix, which is provided in the e-companion.)²*

Mathematically, Proposition 1 can be written as

$$\frac{P(A = a_i | A_{S_1} \neq a_i, A_{S_2} \neq a_i, \dots, A_{S_n} \neq a_i)}{P(A = a_j | A_{S_1} \neq a_j, A_{S_2} \neq a_j, \dots, A_{S_n} \neq a_j)} = \frac{P(A = a_i)}{P(A = a_j)} \quad \forall i, j.$$

In the special case where attribute A has a diffuse prior, the above ratio is one, i.e., all unrecorded values have the same probability. Based on Proposition 1, once we know the probability associated with one of the unrecorded values, obtaining the probability associated with all other unrecorded values is straightforward. As we show in §5.1, in situations where the most likely value in a group of values is always the best, we only need to consider the value with the largest prior probability. This makes our methodology computationally more efficient, especially when the size of the domain is large.

PROPOSITION 2. *For an attribute with an infinitely large domain and a finite ratio of prior probabilities for any two of its values, the value that appears in most (and at least two) data sources for an entity instance is true with probability one, and all other values have a probability of zero. In case there are more than one such most frequent attribute values, the probabilities for these values can be calculated by ignoring all the less frequent ones and their associated data sources.*

Thus, for attributes with an extremely large number of possible values, such as *Email_Address* (E), the number of values to be considered is further reduced. For example, if *Email_Address* e_1 is recorded in two data sources for a particular customer, and all other

sources have different and distinct values, then the probability that e_1 is the true value equals one. On the other hand, if e_1 is recorded in data sources S_1 and S_2 , e_2 is recorded in S_3 and S_4 , and all other data sources have different and distinct values, then

$$\begin{aligned} P(E = e_1 | E_{S_1}, \dots, E_{S_n}) &= \{P(E = e_1)R_{S_1}^E R_{S_2}^E (1 - R_{S_3}^E)(1 - R_{S_4}^E) \\ &\cdot \{P(E = e_1)R_{S_1}^E R_{S_2}^E (1 - R_{S_3}^E)(1 - R_{S_4}^E) \\ &+ P(E = e_2)R_{S_3}^E R_{S_4}^E (1 - R_{S_1}^E)(1 - R_{S_2}^E)\}^{-1} \end{aligned}$$

and

$$P(E = e_2 | E_{S_1}, \dots, E_{S_n}) = 1 - P(E = e_1 | E_{S_1}, \dots, E_{S_n}).$$

2.2.1. Probability Estimates for Two Data Sources.

In the case of two data sources, (2) reduces to

$$P(A = a_i | A_{S_1}, A_{S_2}) = \frac{P(A_{S_1} | A = a_i)P(A_{S_2} | A = a_i)P(A = a_i)}{\sum_{j=1}^m P(A_{S_1} | A = a_j)P(A_{S_2} | A = a_j)P(A = a_j)}. \quad (3)$$

Based on the reliability of an attribute in two data sources and the assumptions discussed earlier, we derive the probabilities of the possible true values in various situations (see §EC.2 of the online appendix for details).

Case 1a. $A_{S_1} = A_{S_2} = a_i$.

$$\begin{aligned} P(A = a_i | A_{S_1} = a_i, A_{S_2} = a_i) &= P(A = a_i)R_{S_1}^A R_{S_2}^A \cdot \{P(A = a_i)R_{S_1}^A R_{S_2}^A + [1 - P(A = a_i)] \\ &\cdot (1 - R_{S_1}^A)(1 - R_{S_2}^A)/(m - 1)^2\}^{-1}. \quad (4) \end{aligned}$$

Case 1b. $A_{S_1} = A_{S_2} = a_j \neq a_i$.

$$\begin{aligned} P(A = a_i | A_{S_1} = a_j, A_{S_2} = a_j) &= \{P(A = a_i)(1 - R_{S_1}^A)(1 - R_{S_2}^A)/(m - 1)^2\} \\ &\cdot \{P(A = a_j)R_{S_1}^A R_{S_2}^A + [1 - P(A = a_j)] \\ &\cdot (1 - R_{S_1}^A)(1 - R_{S_2}^A)/(m - 1)^2\}^{-1}. \quad (5) \end{aligned}$$

Case 2a. $A_{S_1} = a_i, A_{S_2} = a_j \neq a_i$.

$$\begin{aligned} P(A = a_i | A_{S_1} = a_i, A_{S_2} = a_j) &= \{P(A = a_i)R_{S_1}^A (1 - R_{S_2}^A)\} \\ &\cdot \{P(A = a_i)R_{S_1}^A (1 - R_{S_2}^A) + P(A = a_j)(1 - R_{S_1}^A)R_{S_2}^A \\ &+ [1 - P(A = a_i) - P(A = a_j)] \\ &\cdot (1 - R_{S_1}^A)(1 - R_{S_2}^A)/(m - 1)\}^{-1}. \quad (6) \end{aligned}$$

The expression for $P(A = a_j | A_{S_1} = a_i, A_{S_2} = a_j)$ can be derived analogously.

²An electronic companion to this paper is available as part of the online version that can be found at <http://manscijournal.informs.org/>.

Case 2b. $A_{S_1} = a_j \neq a_i, A_{S_2} = a_k \neq a_i, a_j \neq a_k$.

$$\begin{aligned} P(A = a_i | A_{S_1} = a_j, A_{S_2} = a_k) \\ = \{P(A = a_i)(1 - R_{S_1}^A)(1 - R_{S_2}^A)/(m - 1)\} \\ \cdot \{P(A = a_j)R_{S_1}^A(1 - R_{S_2}^A) + P(A = a_k)(1 - R_{S_1}^A)R_{S_2}^A \\ + [1 - P(A = a_j) - P(A = a_k)](1 - R_{S_1}^A) \\ \cdot (1 - R_{S_2}^A)/(m - 1)\}^{-1}. \end{aligned} \quad (7)$$

2.2.2. Probability Estimates for the Case with Multiple Attributes. We now consider situations where multiple attributes are common across the databases. If the reliabilities of the attributes are independent of one another, the analysis presented above can be applied to each attribute individually. When the reliabilities across a group of attributes are dependent, we can treat the entire group as one *composite attribute*. For example, if two attributes B and C form a composite attribute, then the reliability of this composite attribute in data source S_k (denoted by $R_{S_k}^{B,C}$) equals the proportion of records in S_k that store the true values of both B and C , and the size of the domain of this composite attribute is $m_B \times m_C$, where m_B and m_C are the sizes of the domains of B and C , respectively. For example, if the values stored for this composite attribute are the same in two data sources, then analogous to Case 1a, we have

$$\begin{aligned} P(B = b_i, C = c_j | B_{S_1} = b_i, C_{S_1} = c_j, B_{S_2} = b_i, C_{S_2} = c_j) \\ = \{P(B = b_i, C = c_j)R_{S_1}^{B,C}R_{S_2}^{B,C}\} \\ \cdot \{P(B = b_i, C = c_j)R_{S_1}^{B,C}R_{S_2}^{B,C} \\ + [1 - P(B = b_i, C = c_j)](1 - R_{S_1}^{B,C}) \\ \cdot (1 - R_{S_2}^{B,C})/(m_B m_C - 1)\}^{-1}. \end{aligned}$$

The desired probabilities for the other cases can be obtained in a similar manner.

2.3. Customer Database Example

Consider customer data that are collected independently in a sales database (S_1) as well as in a customer service database (S_2). Some attribute values for the same customer could be different in these two sources. We now illustrate how the probability distribution can be obtained.

2.3.1. Binary Attribute. Consider a binary attribute *Marital_Status* (MS for brevity), which can take a value of either “M” or “N.” Assume that the percentage of married people in the entire population is 50%, and, based on sampling, we find that this attribute is accurate 80% of the time in S_1 and 90% in S_2 , i.e., $R_{S_1}^{MS} = 0.8$ and $R_{S_2}^{MS} = 0.9$. Let the stored values of *Marital_Status* be “M” and “N” in S_1 and S_2 ,

respectively, for a customer, say, Robert Black. Then, we have from (6):

$$\begin{aligned} P(MS = \text{“M”} | MS_{S_1} = \text{“M”}, MS_{S_2} = \text{“N”}) \\ = (0.5 \times 0.8 \times 0.1)/(0.5 \times 0.8 \times 0.1 + 0.5 \times 0.9 \times 0.2) \\ = 0.308 \quad \text{and} \\ P(MS = \text{“N”} | MS_{S_1} = \text{“M”}, MS_{S_2} = \text{“N”}) \\ = 1 - 0.308 = 0.692. \end{aligned}$$

If, on the other hand, the stored values are both “M” for Robert Black, we have from (4) and (5):

$$\begin{aligned} P(MS = \text{“M”} | MS_{S_1} = \text{“M”}, MS_{S_2} = \text{“M”}) \\ = (0.5 \times 0.8 \times 0.9)/(0.5 \times 0.8 \times 0.9 + 0.5 \times 0.2 \times 0.1) \\ = 0.973 \quad \text{and} \\ P(MS = \text{“N”} | MS_{S_1} = \text{“M”}, MS_{S_2} = \text{“M”}) \\ = 1 - 0.973 = 0.027. \end{aligned}$$

2.3.2. Multivalued Attribute. Suppose that the original databases also store the *Profession* (PF for brevity) of the customers, and this attribute can take 50 possible values (e.g., “accountant,” “programmer,” etc.) with diffuse priors (i.e., $P(PF = \text{“X”}) = 0.02$ for all X). We may find that, for example, the *Profession* for Robert Black is stored as “accountant” in S_1 and as “programmer” in S_2 . Again, assuming that this attribute is accurate 80% of the time in S_1 and 90% in S_2 , we can calculate the probability that “accountant” is Robert Black’s true *Profession* as follows:

$$\begin{aligned} P(PF = \text{“accountant”} | PF_{S_1} = \text{“accountant”}, \\ PF_{S_2} = \text{“programmer”}) \\ = (0.02 \times 0.8 \times 0.1)/(0.02 \times 0.8 \times 0.1 + 0.02 \times 0.9 \\ \times 0.2 + 0.96 \times 0.2 \times 0.1/49) = 0.286. \end{aligned}$$

Similarly,

$$\begin{aligned} P(PF = \text{“programmer”} | PF_{S_1} = \text{“accountant”}, \\ PF_{S_2} = \text{“programmer”}) = 0.644 \quad \text{and} \\ P(PF = \text{AOV} | PF_{S_1} = \text{“accountant”}, \\ PF_{S_2} = \text{“programmer”}) = 0.00146, \end{aligned}$$

where AOV stands for any other PF value except “accountant” or “programmer.” On the other hand, if the stored *Profession* for Robert Black is “attorney” in both data sources, then

$$\begin{aligned} P(PF = \text{“attorney”} | PF_{S_1} = \text{“attorney”}, \\ PF_{S_2} = \text{“attorney”}) = 0.99943 \quad \text{and} \\ P(PF = \text{AOV} | PF_{S_1} = \text{“attorney”}, \\ PF_{S_2} = \text{“attorney”}) = 1.16 \times 10^{-05}. \end{aligned}$$

In this case, AOV is any other PF value except “attorney.”

Table 2 Customer Data

A_ID	Name	Address		Marital_Status	
		Value (real value omitted for brevity)	Prob.	Value	Prob.
10001	Robert Black	ad_1	0.342	M	0.308
		ad_2	0.572	N	0.692

3. Making Decisions with Data from Multiple Sources: A Direct-Marketing Example

Consider a firm that has decided to launch a direct-marketing campaign to promote a product that is only suitable for married customers. To identify such potential customers, the firm searches the two separately maintained databases described in §2 (where *Marital_Status* is accurate 80% of the time in S_1 and 90% in S_2). Based on §2.3.1, if a potential customer Robert Black’s *Marital_Status* is “M” in S_1 and “N” in S_2 , then the probability associated with Robert Black’s actual *Marital_Status* is as shown in the last column of Table 2.

In addition to *Marital_Status*, another attribute needed for the direct-marketing campaign could be *Address*: a composite attribute consisting of *Street_Address*, *City*, *State*, and *Zip_Code*. Assume that the reliabilities of attribute *Address* in data sources S_1 and S_2 are 80% and 87%, respectively, and that the number of possible values for *Address* is infinitely large. Based on Proposition 2, if the values from the two sources are the same, then the probability that the recorded *Address* is correct is one. Similarly, we find from (6) that, if the values from the two sources are different, the probability that the *Address* from S_1 is correct equals 0.342, and that for the one from S_2 is 0.572.

Given the uncertain information on *Marital_Status* and *Address* as shown in Table 2, should the firm select Robert Black as a target customer? To answer this question, we need to analyze the consequences (or costs) associated with these two possible decisions.

3.1. Errors and Error Costs

A *type I error* occurs when an entity instance, which should have been selected for a decision problem based on the true value of an attribute, is not selected. A *type II error* occurs when an entity instance, which should not have been selected based on the true attribute value, is selected. Type I and type II errors are also referred to as *false negatives* and *false positives*, respectively. For our direct-marketing example, if the true *Marital_Status* for Robert Black is “M,” but the firm fails to select him as a target for direct marketing, then a type I error occurs. On the other hand, if “N” is Robert Black’s true *Marital_Status*, but he is selected, then a type II error occurs.

Type I and type II errors do not, however, cover all possible types of errors that may occur with this decision problem. For example, if the *Marital_Status* for Robert Black is “M” and he is selected, then neither a type I nor a type II error occurs. However, an incorrect mailing address may be used for Robert Black. This is a *misrepresentation error*, which occurs when the value of an attribute used in the action on a selected entity instance is incorrectly represented (Mendelson and Saharia 1986, Dey et al. 1998a). Note that, in the direct-marketing case, if Robert Black is not selected, then whether his *Address* is correct or not has no impact on the decision problem. Therefore, misrepresentation error is relevant only if a particular entity instance is selected.

The unit costs of the three types of errors associated with a decision problem are denoted as follows: γ_I is the cost of a type I error, γ_{II} the cost of a type II error, and $\gamma_m(A)$ the average cost of every misrepresented value of attribute A . While the type I and type II error costs are unique for a particular decision problem, the misrepresentation cost is specific not only to a decision problem but also to the displayed attribute(s) of the selected entity instances. If there is more than one such attribute, the costs of misrepresenting different attributes need not be equal and should be estimated separately.

The cost parameters discussed above are estimated based on the decision problem at hand. Given a decision context, it is often reasonably straightforward to identify the potential loss or opportunity cost associated with each of the three types of errors. Let us illustrate this with the direct-marketing decision problem. Let X be the cost to reach a customer and Y be the expected payoff per married customer reached by this campaign. Then, the cost of type I error (γ_I) equals $(Y - X)$. If we assume that the chance of purchase by an unmarried customer is zero, then the cost of type II error (γ_{II}) equals X . Further, let the probability that the mail would be lost or returned because of an incorrect *Address* be P_l , $0 < P_l < 1$ (implying that if the *Address* is only partially incorrect, there is still some chance that the customer may receive the mail). Therefore, the unit cost of a misrepresentation error for *Address* is $\gamma_m(AD) = Y \cdot P_l$. Although misrepresentation errors may also occur with other attributes such as *Name*, we assume, for simplicity, that the costs of misrepresentation errors for such attributes are negligible.

3.2. Probability Threshold for Decision Making

Based on the probabilities and error cost parameters, we derive the total cost associated with selecting Robert Black as well as that for not selecting him. The costs of the various errors are shown in Table 3. The costs of type I and type II errors are easily obtained.

Table 3 Costs Associated with Selecting and Not Selecting Robert Black

Decision	If true marital status is	Type I error cost	Type II error cost	Misrepresentation error cost*
Select	M	N/A	0	$\gamma_m(AD) \times P(\text{married}) \times [1 - P(ad_2)]$
	N	N/A	$\gamma_{II} \times [1 - P(\text{married})]$	0
Do not select	M	$\gamma_I \times P(\text{married})$	N/A	N/A
	N	0	N/A	N/A

*The misrepresentation error cost is minimized when the most likely value (ad_2) of AD is displayed.

Note that a misrepresentation error occurs only if a target customer is truly married and the *Address* used for this customer is incorrect; the expression in Table 3 reflects their probabilities.

In Table 3, $P(\text{married}) = P(MS = \text{“M”} \mid MS_{S_1}, MS_{S_2})$ and $P(ad_2) = P(AD = ad_2 \mid AD_{S_1}, AD_{S_2})$, where AD denotes *Address*. Clearly, the expected cost of selecting Robert Black equals

$$TC(\text{Select}) = \gamma_{II}[1 - P(\text{married})] + \gamma_m(AD)P(\text{married})[1 - P(ad_2)]. \quad (8)$$

On the other hand, the total cost associated with not selecting Robert Black is

$$TC(\text{Do not select}) = \gamma_I P(\text{married}). \quad (9)$$

Hence, the firm should target Robert Black as a customer if $TC(\text{Select}) < TC(\text{Do not select})$, i.e., if

$$P(\text{married}) > \frac{\gamma_{II}}{\gamma_I + \gamma_{II} - \gamma_m(AD)[1 - P(ad_2)]}. \quad (10)$$

Assume that $\gamma_I = 2$, $\gamma_{II} = 1$, and $\gamma_m(AD) = 0.6$. For Robert Black, $P(ad_2) = 0.572$. Based on these parameters, the right-hand side (RHS) of (10) equals 0.364. Because $P(\text{married}) = 0.308$ for Robert Black, based on condition (10), he should not be selected for direct marketing. Additional implications of this threshold rule are illustrated in §EC.3 of the online appendix.

4. Reconciliation and Storage of Conflicting Attribute Values

In many business situations, data from multiple sources could be used in several applications or decision problems. In such a situation, an organization could repeat the procedure described in the previous section for every decision problem. Of course, in that case, all the original data sources must be maintained and the necessary computations repeated for every problem. In some situations, however, this approach is not practical because of storage and processing time constraints as well as the overhead associated with maintaining and updating multiple copies of data. In

those cases, an organization needs to determine which attribute value should be kept for each record. The concept of such a “master” record is gaining considerable popularity in practice (Lager 2005, Wadehra 2006). We present an approach to create a master record that involves identifying and storing, for every entity instance, the “best” value from a set of possible attribute values.

4.1. Queries to Support Decision Problems

For every decision problem that needs information on a selected subset of entity instances, we can imagine that there exist: (i) a corresponding query that specifies what is needed by a decision problem, and (ii) a set of cost parameters γ_I , γ_{II} , and $\gamma_m(A)$, representing the costs of type I error, type II error, and of misrepresenting an attribute A , respectively. These cost parameters need to be estimated for each decision problem in a manner similar to the parameter estimation for the direct-marketing problem discussed in §3.1.³ For example, the direct-marketing problem discussed in the previous section may be supported by the following query:

Q1. *Display the Names and Addresses of those customers whose Marital_Status is “M.”*

If the decision is to target a customer for direct marketing, that customer should also be selected by Q1. This implies that the best value for *Marital_Status* of that particular customer should be “M” in the reconciled customer table. On the other hand, if we decide not to target that customer, then the best *Marital_Status* value is “N.” Interestingly, this example shows that the most likely value for an attribute may not always be the best value to store. For example, if the RHS of (10) equals 0.3 and the observed *Marital_Status* for a customer is “M” in S_1 and “N” in S_2 , then “M” should be the chosen *Marital_Status* value for this customer, even though “N” has a higher probability of being the true value.

When multiple decision problems are considered, in addition to the cost parameters discussed earlier, the frequency of each query, denoted by $f(\cdot)$, is also needed to determine the cost of errors in the long run. This is because the output of a query may be used at different frequencies for different decision problems. For example, a company may send out similar brochures for a product series to the same group of customers once every month, while sending out samples for another product series only once a year.

³ If the number of decision problems to be considered is large, cost estimation would require some amount of effort. However, because the number of decision problems is usually several orders of magnitude smaller than the number of records to be consolidated, the cost associated with the approach we propose in this paper is generally much lower than the cost of ascertaining the true attribute values for all records.

For expositional convenience, we use the query representations instead of the decision problems themselves in the following discussion. The cost and frequency parameters associated with a query q are denoted by $\gamma_I(q)$, $\gamma_{II}(q)$, $\gamma_m(q, A)$, and $f(q)$. If more than one decision problem uses the same query, the multiple problems can easily be consolidated into a single problem with weighted averages of the respective cost parameters.

4.2. Classification of Queries

In mapping decision problems to queries, the attribute(s) that are used in selecting the appropriate subset of entity instances (e.g., *Marital_Status* in the direct-marketing case) appear in the selection condition of the query. The attributes that are used in the subsequent actions on the selected group of entity instances (e.g., *Name* and *Address* in the direct-marketing case) should appear in the projection list of the query. Although in the direct-marketing case, *Marital_Status* is needed for selection and not for mailing, there are situations where the attributes used in the selection condition may also be needed for the subsequent actions on the selected instances.

Consider an attribute that is being reconciled from multiple sources. If that attribute appears only in the selection condition of a query, an error in the attribute value can cause either a type I or a type II error; we call such a query a *Class C* (*C*onditioning) *query* relative to that attribute. If the attribute appears in just the projection list of a query, an error in that attribute can only cause a misrepresentation error; such a query is called a *Class T* (*T*argeting) *query*. If the attribute appears in both the selection condition and the projection list, any of the three types of errors can occur; we call such a query a *Class CT* *query*. Consider the following examples:

Q1. Display the Names and Addresses of those customers whose *Marital_Status* is “M.”

Q2. Display the Names and Addresses of those customers whose *Profession* is “accountant.”

Q3. Display the Names and Professions of those customers whose *Employer* is “GTE.”

Q4. Display the IDs, Names, and Professions of the customers whose *Profession* is “programmer.”

Among these queries, Q1 is of Class C for *Marital_Status*. Similarly, for *Employer*, Q3 is of Class C.

On the other hand, with respect to *Profession*, Q2 is of Class C, Q3 is of Class T, and Q4 is of Class CT.

Note that any two of the three types of errors cannot occur at the same time with the same entity instance for the following reasons: First, a type I error can occur only if the given entity instance is not selected, and type II and misrepresentation errors can only occur if the entity instance is selected. Second, the correctness of the targeted attribute values of an included entity instance is important only if the instance is selected correctly, that is, without any type II error. If, on the other hand, a type II error is already committed, whether or not the target attribute values are correct is not relevant.

4.3. Determining the Best Value for Queries with a Single Clause in the Selection Condition

We now demonstrate how an attribute value can be determined for an entity instance from a set of possible values by minimizing the total cost of errors for multiple decision problems. We start our analysis by using the set of queries Q1–Q4 along with their corresponding cost and frequency parameters. All four queries have a single clause in their selection conditions. The attributes needed by Q1–Q4 are *ID*, *Name*, *Address*, *Employer*, *Profession*, and *Marital_Status*. Among these six attributes, *ID* is the key. Because entity matching is assumed to have been done already, there is no uncertainty about *ID* values. The possible values of the remaining five attributes for the customer with *ID* “10001” and the associated probabilities are shown in Table 4. All the probabilities in this table are derived based on the procedure discussed in §2. The *Name* value for the customer with *ID* “10001” is assumed to be the same in the two data sources. Because the size of its domain is extremely large, the value “Robert Black” has a probability of one. The recorded values for *Address*, *Employer*, *Profession*, and *Marital_Status*, on the other hand, are assumed to be different in the two data sources; therefore, the probabilities associated with the most likely values are all less than one. We now demonstrate how the cost-minimizing attribute values can be determined for this customer (Robert Black) from the probability distributions in Table 4. For simplicity, the reliabilities of the five attributes are all assumed to be independent.

Table 4 Complete Customer Data

ID	Name		Employer		Address		Profession		Marital_Status	
	Value	Prob.	Value	Prob.	Value	Prob.	Value	Prob.	Value	Prob.
10001	Robert Black	1.0	GTE	0.740	ad_1	0.342	Accountant	0.286	M	0.308
	AOV	0.0	Walmart	0.221	ad_2	0.572	Programmer	0.644	N	0.692
			AOV	1.96×10^{-4}	AOV	8.55×10^{-11}	AOV	0.00146		

Table 5 Cost of Type I and Type II Errors

Stored value	Query	Query result	If true value is	Type I error cost	Type II error cost
Programmer	Q2	Do not select	Not accountant	0	N/A
			Accountant	$\gamma_I(Q2)f(Q2)P(\text{accountant})$	N/A
	Q4	Select	Programmer	N/A	0
			Not programmer	N/A	$\gamma_{II}(Q4)f(Q4)(1 - P(\text{programmer}))$
Accountant	Q2	Select	Accountant	N/A	0
			Not accountant	N/A	$\gamma_{II}(Q2)f(Q2)(1 - P(\text{accountant}))$
	Q4	Do not select	Not programmer	0	N/A
			Programmer	$\gamma_I(Q4)f(Q4)P(\text{programmer})$	N/A
AOV	Q2	Do not select	Not accountant	0	N/A
			Accountant	$\gamma_I(Q2)f(Q2)P(\text{accountant})$	N/A
	Q4	Do not select	Not programmer	0	N/A
			Programmer	$\gamma_I(Q4)f(Q4)P(\text{programmer})$	N/A

Consider the attribute *Profession*. Because the decision problem represented by Q1 does not need this attribute, Q1 can be ignored. Among the remaining three queries, Q2 is of Class C, Q3 is of Class T, and Q4 is of Class CT, all with respect to *Profession*.

4.3.1. Costs of Type I and Type II Errors. We first derive the expected costs of type I and type II errors when “programmer” is the stored *Profession* for Robert Black. Only Q2 and Q4 need to be considered for type I and type II errors. Obviously, Q2 would not select Robert Black. Given that he is not selected, if Robert Black’s true *Profession* is indeed “accountant,” then a type I error occurs. The frequency of this occurrence equals the product of the frequency $f(Q2)$ and the probability that the true value is “accountant,” denoted by $P(\text{accountant})$. On the other hand, when Q4 is executed, Robert Black would be selected and, if the true value is not “programmer,” a type II error occurs. The frequency of this occurrence equals the product of the frequency $f(Q4)$ and the probability that the true value is not “programmer,” which is $(1 - P(\text{programmer}))$. The above analysis is summarized in Table 5; see the rows corresponding to the stored value “programmer.” The total cost of type I and type II errors resulting from storing “programmer” is

$$\begin{aligned}
 C_{I,II}(\text{programmer}) &= \gamma_I(Q2)f(Q2)P(\text{accountant}) \\
 &+ \gamma_{II}(Q4)f(Q4)(1 - P(\text{programmer})). \quad (11)
 \end{aligned}$$

Similarly, the total cost of type I and type II errors resulting from storing “accountant” for Robert Black is obtained based on the analysis shown in Table 5:

$$\begin{aligned}
 C_{I,II}(\text{accountant}) &= \gamma_{II}(Q2)f(Q2)(1 - P(\text{accountant})) \\
 &+ \gamma_I(Q4)f(Q4)P(\text{programmer}). \quad (12)
 \end{aligned}$$

Table 5 also summarizes the type I and type II error costs incurred if any other value (AOV) except “programmer” or “accountant” is stored:

$$\begin{aligned}
 C_{I,II}(\text{AOV}) &= \gamma_I(Q2)f(Q2)P(\text{accountant}) \\
 &+ \gamma_I(Q4)f(Q4)P(\text{programmer}). \quad (13)
 \end{aligned}$$

In this example, the cost analysis for AOV is also valid if NULL is chosen.

4.3.2. Cost of Misrepresentation Errors. As mentioned in §4.2, a misrepresentation error is relevant only when there is no type I or type II error. It implies that, for single-clause selection conditions, one needs to consider only Class T queries. In our example, therefore, the only relevant query is Q3. As before, we assume that “programmer” is the chosen value for Robert Black. Thus, whenever Robert Black is selected by a query and *Profession* is in the projection list, the value displayed will be “programmer.” If the true *Profession* is not “programmer,” a misrepresentation error occurs. If the chosen *Employer* value for Robert Black is not “GTE,” then Robert Black will never be selected by Q3, and the misrepresentation error will not occur. If the chosen *Employer* value is “GTE,” then Robert Black will be selected by Q3. In that case, the cost of the misrepresentation error equals the product of the unit misrepresentation cost, the frequency of Q3, the probability that Robert Black’s true *Profession* is not “programmer,” and the probability that “GTE” is the true *Employer* for Robert Black, denoted by $P(\text{GTE})$. Similar arguments can be made for other chosen values of *Profession* as well. Hence, the misrepresentation cost associated with selecting *Profession* = “X” is given by

$$C_m(X) = \begin{cases} \gamma_m(Q3, \text{Employer})f(Q3)P(\text{GTE})(1 - P(X)) & \text{if “GTE” is the chosen Employer value,} \\ 0, & \text{otherwise,} \end{cases} \quad (14)$$

where X could be “programmer,” “accountant,” or AOV. Finally, we examine the misrepresentation cost if NULL is stored. We assume that NULL is never the true value and that the unit misrepresentation cost when NULL or any other incorrect value is stored is the same. The resulting misrepresentation cost is

$$C_m(\text{NULL}) = \begin{cases} \gamma_m(Q3, \text{Employer})f(Q3)P(\text{GTE}) & \text{if “GTE” is the chosen Employer value,} \\ 0, & \text{otherwise.} \end{cases} \quad (15)$$

If the cost of misrepresentation is different when NULL is stored, then we can use another misrepresentation parameter γ'_m specifically for NULL and replace γ_m in (15) by γ'_m . The rest of the analysis remains the same.

4.3.3. Minimizing the Total Error Cost. To decide which value to store for Robert Black’s *Profession*, we add the costs of type I, type II, and misrepresentation errors associated with each possible value. For the value “programmer,” for example, the following expression is evaluated:

$$TC(\text{programmer}) = C_{I,II}(\text{programmer}) + C_m(\text{programmer}). \quad (16)$$

Similar expressions can be evaluated for “accountant,” AOV, and NULL.

Obviously, the value with the smallest error cost should be stored in the integrated database. Depending on the cost parameters, that could be any of the values. In most situations, “programmer” or “accountant” would likely be the best value. However, when the cost of type II errors is significantly higher than those of type I and misrepresentation errors, AOV or even NULL could be the cost-minimizing option. This is due to the fact that if AOV or NULL is stored, Robert Black will not be selected by Q1 and Q3, and hence type II errors will never occur.

4.3.4. Interrelated Decisions and a Graph-Based Solution. In the above example, the exact costs associated with different *Profession* choices cannot be determined before choosing the *Employer* value. This is because the misrepresentation costs depend on whether “GTE” is the chosen *Employer* value or not. Therefore, it seems that we should first find out the cost-minimizing *Employer* value before deciding the best value to store for *Profession*. Upon further examination, we realize that Q3 is similar to Q1. Therefore, similar to the direct-marketing case where we need to choose the *Address* value before making a decision regarding Robert Black’s *Marital_Status*, we need to know the chosen *Profession* for Robert Black before we

can decide the best *Employer* value for him. Therefore, we are faced with an interrelated decision problem: To decide on a value for *Profession*, we need to know the best value for *Employer*, and vice versa.

One solution to this problem is to consider all attributes as one composite attribute and examine all feasible value combinations separately. For each combination, we could calculate the sum of the error costs associated with all queries, and choose the combination with the lowest cost for a particular entity instance. Computationally, however, it could be quite costly, especially if the number of such value combinations is large. Therefore, we propose a graph-based procedure that enables us to easily identify the sequence in which the attribute values can be resolved efficiently. The procedure starts by constructing an *attribute interconnection graph*, followed by steps to resolve the values of the attributes.

4.3.4.1. Construction of the Attribute Interconnection Graph. In the attribute interconnection graph, every attribute that appears only in the projection list(s) of the queries is represented by a circle, every attribute that appears only in the selection condition(s) by a square, and every attribute that appears in both the selection condition(s) and the projection list(s) by a circle within a square. Two attributes are connected if they appear in the same query, one in the projection list and the other in the selection condition.

4.3.4.2. Resolving Attribute Values. After the attribute interconnection graph is constructed, the attributes are resolved by the following steps:

Step 1. Resolve those attributes that have a value with probability 1. Remove all links that are connected to those attributes. No error occurs if the chosen value of an attribute has a probability one of being the true value. Therefore, we can resolve such an attribute without further examination.

Step 2. Resolve all attributes in circles only, by selecting their most likely values. Remove all links connected to those attributes. An attribute that appears only in the projection lists can cause just misrepresentation errors. Regardless of the values chosen for all other attributes, selecting the most likely value for this attribute always leads to the least expected misrepresentation cost.

Step 3. For the remaining attributes, resolve each interconnected group of attributes separately (a group may be of size one) by selecting the value combination with the minimum error cost. This is because the best set of values for a group of attributes does not depend on the attributes from other groups.

Figure 1 illustrates the graph constructed for our customer database example with Q1–Q4. In Step 1, *Name* is resolved to be “Robert Black” because the probability associated with this value is one (see

Figure 1 Attribute Interconnection Graph

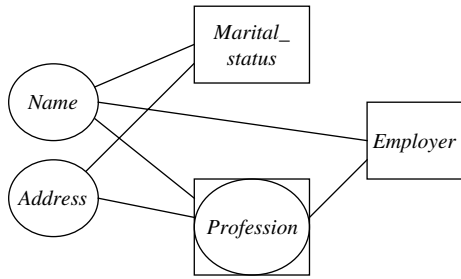


Table 4). In Step 2, *Address* is resolved. The graph then reduces to the one shown in Figure 2, with dashed circles used to represent the resolved attributes. In Step 3, the remaining three attributes are divided into two groups and solved separately. The first group has only one attribute *Marital_Status*, and the second group contains two interrelated attributes *Profession* and *Employer*. We first decide the best *Marital_Status* value for Robert Black. Only Q1 needs to be considered in this case. If the cost parameters associated with Q1 are $\gamma_I(Q1) = 2$, $\gamma_{II}(Q1) = 0.7$, and $\gamma_m(Q1, AD) = 0.6$, then the expected cost of storing Robert Black’s *Marital_Status* as “M,” obtained from (8), equals 0.56, and that of storing “N,” obtained from (9), equals 0.62. Clearly, the best value to store is “M.” The procedure to resolve the attributes *Employer* and *Profession* jointly are discussed next.

4.3.5. Resolving an Interrelated Group of Attributes. To determine the best combination of values for a group of attributes, we need to obtain the total expected error costs associated with different combinations of values. Consider the two attributes *Employer* and *Profession*. The costs associated with all possible values of *Profession* have already been shown in Equations (11) through (16). Because the value for *Employer* is included in each combination of values, the exact misrepresentation cost associated with the value for *Profession* can be determined using Equations (14) and (15). The cost associated with the attribute *Employer* can be obtained by considering Q3 only. Because Q3 is of Class C with respect to the attribute *Employer*, only type I and type II errors can

Figure 2 Graph After Steps 1 and 2

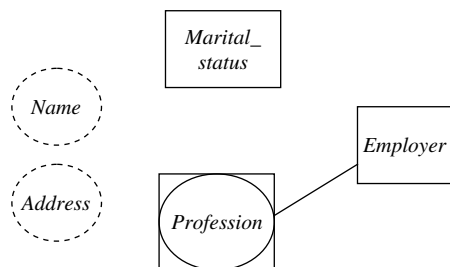


Table 6 Costs Associated with Storing Different *Employer* Values with Respect to Q3

Value to store	Query result	If true value is	Type I error cost	Type II error cost
GTE	Select	GTE	N/A	0
		Not GTE	N/A	$\gamma_{II}(Q3)f(Q3)(1 - P(GTE))$
AOV	Do not select	Not GTE	0	N/A
		GTE	$\gamma_I(Q3)f(Q3)P(GTE)$	N/A

occur. The cost associated with storing “GTE” or any other value (AOV) is shown in Table 6.

Now, we are ready to compute the total cost associated with any combination of values for *Employer* and *Profession*. For example, the total cost associated with (programmer, GTE) equals

$$\begin{aligned}
 TC(\text{programmer, GTE}) &= \gamma_I(Q2)f(Q2)P(\text{accountant}) \\
 &+ \gamma_{II}(Q4)f(Q4)(1 - P(\text{programmer})) \\
 &+ \gamma_m(Q3, \text{Employer})f(Q3)P(GTE) \\
 &\cdot (1 - P(\text{programmer})) \\
 &+ \gamma_{II}(Q3)f(Q3)(1 - P(GTE)),
 \end{aligned}$$

where the first three terms come from (11) and (14), and the last from Table 6. The costs associated with other combinations of values can be obtained analogously, and the combination with the lowest cost should be chosen. Assuming $\gamma_I(Q2) = 5$, $\gamma_{II}(Q2) = 2$, $f(Q2) = 1$, $\gamma_I(Q3) = 2$, $\gamma_{II}(Q3) = 2$, $\gamma_m(Q3, \text{Employer}) = 0.5$, $f(Q3) = 1$, $\gamma_I(Q4) = 2$, $\gamma_{II}(Q4) = 5$, and $f(Q4) = 1$, the combination (accountant, GTE) has the lowest expected cost. Based on this procedure, we find the cost-minimizing values for Robert Black as follows:

$$\begin{aligned}
 \text{Employer} &= \text{“GTE,”} & \text{Address} &= ad_2, \\
 \text{Profession} &= \text{“accountant,”} & & \text{and} \\
 \text{Marital_Status} &= \text{“M.”}
 \end{aligned}$$

Comparing with Table 4, we find that the chosen values for *Employer* and *Address* are their respective most likely values, while those for *Profession* and *Marital_Status* are not so.

Note that in the above example, because the *Name* “Robert Black” has a probability of one, we need not consider the misrepresentation costs of more than two attributes associated with the same query. For other entity instances, it is possible that the misrepresentation costs may have to be counted for multiple attributes from the same projection list. If this is the case, depending on the context of the underlying decision problem, the total misrepresentation cost may take different functional forms.

Table 7 Cost of Type I and Type II Errors if “programmer” Is Stored

Query No.	Selection condition	Query result	If true value is	Type I error cost	Type II error cost
Q5 (CT)	Programmer, physician	Select	Physician or programmer	N/A	0
			Others	N/A	$\gamma_{II}(Q5)f(Q5) \cdot (1 - P(\text{programmer}) - P(\text{physician}))$
Q6 (C)	Accountant, farmer, broker	Do not select	Accountant, farmer, or broker	$\gamma_I(Q6)f(Q6) \cdot (P(\text{accountant}) + P(\text{farmer}) + P(\text{broker}))$	N/A
			Others	0	N/A

4.4. Costs of Errors Associated with Queries with Disjunctive Clauses

The example discussed in §4.3 involves queries with only a single clause in the selection condition. In this subsection, we examine queries with disjunctive clauses in their selection condition. The following are two examples of this type of queries:

Q5. *Display the IDs, Names, and Professions of those customers whose Profession is “programmer” or “physician.”*

Q6. *Display the IDs, Names, and Employers of those customers whose Profession is “accountant,” “farmer,” or “broker.”*

With respect to the target attribute *Profession*, Q5 is of Class CT and Q6 is of Class C. We still use the example data for Robert Black to illustrate how the costs associated with these new queries can be determined. Assume once again that “programmer” is the stored value. The costs of type I and type II errors are summarized in Table 7. Based on Tables 5 and 7, we make the following observations:

OBSERVATION 1. If the chosen value is included in the selection condition of a query (e.g., “programmer” is in Q5), the probability of type II error equals the probability that a value other than those included in the selection condition is the true value.

OBSERVATION 2. If the chosen value is not included in the selection condition of a query (e.g., “programmer” is not in Q6), then the probability of type I error equals the probability that one of the values included in the query’s selection condition is the true value.

We next examine the cost of misrepresentation errors. For selection conditions with disjunctive clauses (unlike single-clause selection conditions), an error in the stored value of the target attribute does not necessarily result in a type II error. Therefore, in addition to Class T queries, here we need to consider Class CT queries as well. For example, with respect to Q5, when the true value is “physician” and the stored value is “programmer,” the entity instance would still be selected without a type II error, but a misrepresentation error would ensue. The cost of this misrepresentation error is $P(\text{physician})\gamma_m(Q5, \text{Profession})f(Q5)$.

So far, we have only considered queries involving a single attribute. Extending this analysis to more complex queries involving disjunctive and conjunctive clauses on multiple attributes is conceptually straightforward. The details are provided in §EC.4 of the online appendix.

4.5. When Reliabilities of Multiple Attributes are Dependent

We have assumed that the reliabilities of the attributes in the customer database are all independent. If the error generation processes of two or more of them are dependent, as discussed in §2.2.2, we can treat them as one composite attribute and obtain their joint value probabilities. While calculating the error costs associated with a specific query, we can ignore the subset of attributes that are not needed by the query and work with the marginal distributions of the ones used.

5. A Standardized Procedure Based on Query-Coverage Bitmaps

If the number of possible values of an attribute or the number of relevant queries is large, the error cost calculation can be a tedious process. In this section, we develop a standardized procedure to automate the process of selecting the cost-minimizing value. We first illustrate how the procedure works when attributes can be resolved one at a time, followed by a discussion for a group of interrelated attributes.

5.1. Procedure for a Single Attribute

We first illustrate, using *Profession* as an example, how the procedure works if an attribute can be resolved in isolation. To make it eligible for this procedure, we assume that *Employer* has been resolved previously and that queries Q1–Q6 support all the decision problems that require the customer data. We construct a *query-coverage bitmap*, as shown in Table 8, for queries that include *Profession* in their selection condition. The query-coverage bitmap summarizes the values included in the selection condition of each query. For example, with respect to Q5, the columns for

Table 8 Query-Coverage Bitmap for Profession

Queries	Condition							Prob_Sum*
	...	acc	bro	far	phy	prg	...	
Q2 (C)	...	1	0	0	0	0	...	P(acc)
Q4 (CT)	...	0	0	0	0	1	...	P(prg)
Q5 (CT)	...	0	0	0	1	1	...	P(phy) + P(prg)
Q6 (C)	...	0	1	1	0	1	...	P(bro) + P(far) + P(prg)

*acc, accountant; bro, broker; far, farmer; phy, physician; prg, programmer.

“physician” and “programmer” are marked as “1” because these two *Profession* values are included in the selection condition of Q5. The last column, labeled as “Prob_Sum,” represents the total probability that any one of the attribute values included in the query’s selection condition is the true attribute value.

Based on the bitmap, we can automate the cost calculation and value determination process. Figure 3 shows the algorithm for determining the best value for an entity instance with a candidate value vector **V** and a probability distribution **P** over **V**. The total cost associated with each value is determined as follows:

Step 1. If there are Class T queries, the associated misrepresentation cost is calculated.

Step 2. The column in the query-coverage bitmap that corresponds to the chosen value is identified.

Step 3. For each query in the bitmap, the value in the cell that corresponds to the row of the query and the column of the chosen value is checked. If the value

Figure 3 Procedure for Determining the Best Value of a Single Attribute

Input: A candidate value vector **V** and a probability distribution **P** over **V**.

Output: The cost-minimizing value *BestVal* and the associated total cost.

1. Find out the corresponding column index numbers in the Query-Coverage Bitmap for all components of **V** and keep them in a vector **J**.
2. Let $C_{\min} = A$ very large number; $BestVal = V_0$.
 For every $j \in J$ (representing all candidate values):
 Begin
 $TC = 0$; $C_I = C_{II} = 0$; $C_m = (1 - P_j)$
 $\cdot \sum_{q \in \text{Class T}} f(q)\gamma_m(q)P(\text{selection condition of } q \text{ is true}).$
 For each query with row index i :
 Do
 If $Bitmap[i][j]$ equals 1, then
 (i) $C_{II} = C_{II} + f(q_i)\gamma_{II}(q_i) \cdot (1 - Prob_Sum(q_i)).$
 (ii) if q_i is of Class CT, then
 $C_m = C_m + (1 - P_j)f(q_i)\gamma_m(q_i) \cdot (Prob_Sum(q_i) - P_j).$
 Else $C_I = C_I + f(q_i)\gamma_I(q_i)Prob_Sum(q_i).$
 End If
 End Do
 $TC = C_I + C_{II} + C_m.$
 If $TC < C_{\min}$, then $C_{\min} = TC$, $BestVal = V_j$.
 End

is 1, the cost of type II error is calculated based on Observation 1 discussed in §4.4; if the query is of Class CT, the misrepresentation cost associated with this query is also determined. If the value is 0, the cost of type I error is calculated based on Observation 2 in §4.4.

Step 4. The total cost associated with the chosen value is determined by adding all three types of costs.

After the total cost associated with all candidate values are obtained, the procedure determines the value with the minimum error cost. If the number of queries is n and the size of the domain is m , then the worst-case complexity of the above standardized procedure is $O(mn)$. However, as the following proposition indicates, all possible attribute values need not be examined explicitly to determine the best one:

PROPOSITION 3. *In a group of attribute values satisfying either of the following conditions: (i) none of the values in the group appears in any query, and (ii) if one value appears in a query, then all other values in the group also appear in the query in exactly the same manner—the total expected error cost associated with the most likely value in the group is always the lowest.*

For example, we see from Table 8 that “broker” and “farmer” belong to one group, and all other values except “accountant,” “broker,” “farmer,” “programmer,” “physician,” and NULL belong to another. Once such groups are identified, we only need to consider the most likely value in each group and ignore the rest. Because the values that do not appear in any query always form a group, the worst-case computational complexity of the standardized procedure is reduced to $O((r+1)n)$, where r is the number of unique values included in the selection conditions of the relevant queries.

5.2. Procedure for a Group of Interrelated Attributes

To facilitate the computation, a query-coverage bitmap should be constructed for every attribute that appears in at least one selection condition, and each bitmap should include all queries that have the corresponding attribute in their selection condition. When a group of interrelated attributes are examined, we need to search multiple bitmaps to compute the expected error cost associated with a value combination. For example, given a value combination (GTE, programmer), we should check both the “GTE” column in the bitmap for *Employer* and the “programmer” column in the bitmap for *Profession*. Based on the digits stored in the two columns, we determine whether the entity instance will be selected by the relevant queries, and calculate the corresponding type I, type II, and misrepresentation error costs.

Proposition 3 extends to interrelated attributes as well. If a group of value combinations satisfies the

condition specified in Proposition 3 for every inter-related attribute, then we only need to consider the value combination that includes the most likely value for every attribute. For example, when queries Q1–Q6 are considered, the two value combinations (GTE, broker) and (GTE, farmer) form a group because both “broker” and “farmer” appear only in Q6. Similarly, all combinations that consist of an *Employer* value and a *Profession* value, neither of which appears in Q1–Q6, also form a group.

6. Experiments for Performance Validation

We compare the performance of our proposed framework (denoted by *Proposed*) against that of four other possible approaches, namely, *Most Likely Value* (MLV), *Most Reliable Source* (MRS), *Simple Majority Voting* (SMV), and *Weighted Majority Voting* (WMV). The MLV approach chooses the value with the highest probability of being true. Under MRS, the value from the most reliable data source is chosen. With SMV, the value recorded in the most number of data sources for a given entity instance is selected. WMV is similar to SMV, except that the frequencies are weighted by the reliabilities of the data sources. Ties are broken randomly for all approaches.

There is a fundamental difference between our proposed framework and these alternative approaches—the alternative approaches make their selection decision based only on the stored data and/or the reliabilities of different data sources, while our proposed framework, in addition, takes into consideration the decision problems that depend on the data sources. By construction, our framework always selects the value with the minimum expected error cost, whereas the other approaches ignore error costs completely. For this reason, unless one of the other four approaches *always* produces the same result as the one given by our framework (in which case that approach is as good as ours), our framework will outperform the other four approaches in minimizing the total expected error cost. As illustrated in SEC.5 of the online appendix, none of the other four approaches guarantees the same decision as the one provided by our framework for all possible observations. Therefore, our framework should outperform the other approaches in minimizing the total expected error cost.

We conducted a series of experiments to better understand the impact of the various factors on the performance difference between our framework and the other approaches. The experiments were based on a simulation of the customer database example discussed in §§3 and 4. It should be pointed out that such experiments can only be conducted with simulated data sources. The reason is obvious: When two

or more data sources with overlapping attributes are compared, to validate the performance, one would need to ascertain their true values, which is usually impossible in practice.

We first generated a “true” database that recorded the correct attribute values for 100,000 customers. The six attributes included in this database were *ID*, *Name*, *Employer*, *Address*, *Profession*, and *Marital_Status* (as shown in Table 4). We then generated three noisy data sources with 80,000, 90,000, and 85,000 customer records. A particular customer record could appear in none, one, two, or all three of the data sources. The *IDs* of the customers in these data sources were recorded accurately, while the values of other attributes were randomly generated based on a specified reliability of each attribute in every data source. These reliabilities were randomly chosen. Because our methodology uses the estimates of reliabilities and prior distributions from sample data to derive the posterior probabilities, we randomly selected a sample of 800 customers from the noisy data sources to estimate these parameters. The prior distributions of only the attributes *Employer*, *Profession*, and *Marital_Status* needed to be estimated as *Name* and *Address* were assumed to have diffuse priors. We finally ran all five different reconciliation approaches, and recorded the total error cost associated with each of them.

To study the impact of attribute reliabilities, we repeated the experiments 10 times, each with a different set of reliabilities across attributes (because the results are quite consistent, we report the findings of only two experiments in Figures 4 and 5). Further, to study the impacts of error costs, for each reliability level, we varied the ratio of type I and type II error costs as well as the ratio of type II and misrepresentation error costs. We considered a multiple-decision problem consisting of queries Q1–Q4 from §4.2. For the sake of simplicity, the parameter values were kept the same in all these decision problems, and a single attribute was considered for misrepresentation cost in each decision problem.

Figure 4 shows the percentage cost savings obtained using our proposed framework, relative to the other four approaches, for a wide range of the cost of type I errors (the costs of type II and misrepresentation errors for all decision problems were fixed at 1.0 and 0.6, respectively). Figure 4(a) shows the results under a lower level of noise (average reliability of 0.91 across all attributes), and Figure 4(b) shows the same under a higher noise level (average reliability of 0.67). The relative performance shows that our framework consistently resulted in a cost lower than that of any other approach. These performance gains increased markedly at lower reliability levels. For example, with $\gamma_1 = 1/32$ and average reliability at 0.67, the savings

Figure 4 Relative Performance by Varying Cost of Misrepresentation Errors ($\gamma_{II} = 1.0, \gamma_m = 0.6$)

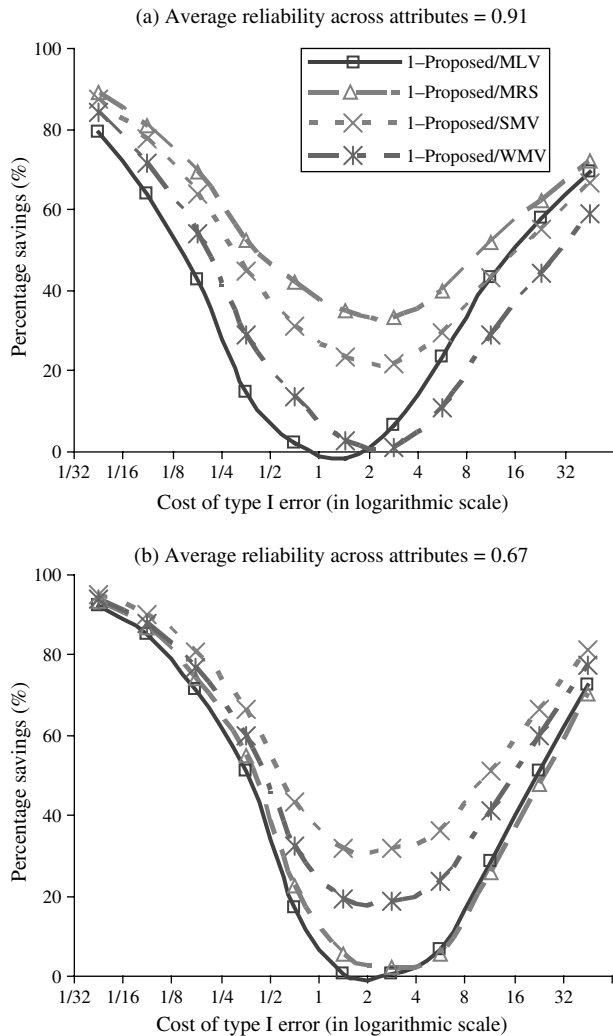
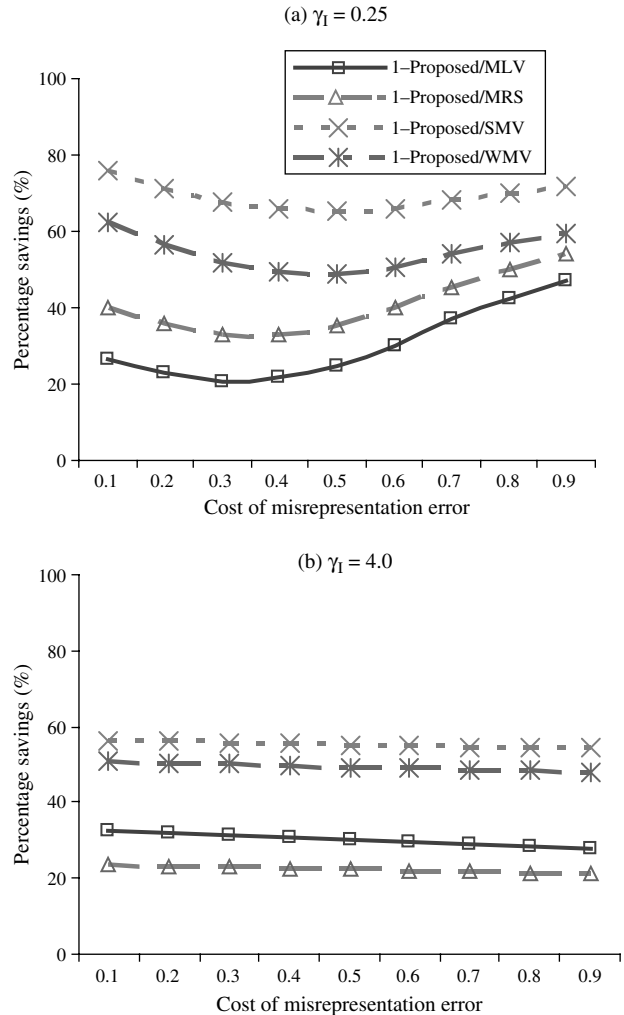


Figure 5 Relative Performance by Varying Cost of Misrepresentation Errors ($\gamma_{II} = 1.0$)



associated with the proposed framework was approximately 92%–95% relative to the four approaches. It is also evident from the figures that the performances of the other approaches are closest to that of the proposed framework when the costs of type I and type II errors are close, and the performance gap widens as the ratio of the costs of type I and type II errors deviates from one. This is because, unlike the other approaches, the proposed framework can adjust its actions based on the relative values of the cost parameters. For example, when the cost of type I error is much higher than that of type II error for a particular decision, attribute values are likely to be chosen in a way that results in the entity instance being selected by the query. The other four approaches, on the other hand, will take the same action regardless of the values of the cost parameters.

We conducted another set of experiments, where we varied the cost of misrepresentation errors from 0.1 to 0.9 while keeping the costs of type I and type II

errors constant. Figure 5 shows the cost savings for two different values of the type I error cost. From this figure, we see that our proposed framework significantly outperforms all other approaches for all values of the cost of misrepresentation errors. For example, with $\gamma_I = 0.25$ and $\gamma_m = 0.1$, the cost savings relative to SMV are approximately 76%.

Additional experiments were conducted to test the robustness of the proposed framework with respect to errors in the estimation of cost parameters and data source reliabilities. We first purposely distorted the value of each error cost parameter by multiplying a distortion factor randomly generated from $[2/3, 3/2]$ and ran the experiments for a wide range of true cost parameter values and noise levels. The proposed framework resulted in a lower cost than the other approaches in 98% of the cases. To further test the robustness of the framework with respect to reliability estimation errors, we repeated the robustness testing, this time assuming that, on top of the cost parameter

estimation errors, there was a 20% chance that the true attribute values had been incorrectly identified during sampling. The results show that our framework still outperforms the other four approaches in 93% of the comparisons. Similar robustness tests were conducted with distortion factors with larger ranges. The results were all consistent.

In summary, our proposed framework works very well under a wide range of cost parameters and noise levels. Computationally, our approach is very efficient, as are the other approaches. We ran the experiments on a desktop PC with a 3.00 GHz Pentium IV processor. The total time it took for all five approaches to process 100,000 customer records was approximately one second.

7. Discussion and Future Research

Attribute-value conflicts are commonly encountered when data from different sources are merged. Such conflicts occur for a variety of reasons such as recording errors, asynchronous updates, and processing errors, and often lead to data quality problems in organizations. This is indeed a serious issue with significant financial implications.

We propose a methodology to estimate distributions of feasible values of an attribute by using the quality metrics associated with the source databases. We then show how optimal business decisions can be made using such data, based on the total cost of type I, type II, and misrepresentation errors. The “best” attribute values can be calculated each time they are needed by a decision problem, or be precomputed and stored for later use. As demonstrated for a direct-marketing application, the value that is best for the decision at hand can be selected instead of arbitrarily picking one of several alternative values. Further, as we show, the most likely attribute value may not always be the best one. Based on experiments conducted over a wide range of parameter values, we find that our approach is consistently superior to other possible approaches, with the performance gains increasing with increasing asymmetry between type I and type II error costs, as well as with the increasing level of noise in the data sources. The proposed methodology will enable firms to reconcile data sources in an automated and efficient manner, thereby enhancing the overall quality of data. By incorporating the appropriate costs and benefits, this approach performs significantly better than the existing (ad hoc) approaches. Furthermore, the ability of this approach to select a single value to store from a set of conflicting values in a way that maximizes the net benefit will improve the overall quality of the reconciled information.

Our work raises the question of when a firm should directly use the source data for business decisions and

when it should perform data integration. The first approach we have discussed incurs higher storage, maintenance, and computation costs, but the decisions made are optimal for each decision. The second approach reduces the overhead of data maintenance and query processing, but the values chosen may not be optimal for each individual decision problem. The conditions under which one approach is preferred to the other is a topic for future research, which requires one to trade off the error costs against the maintenance and processing overheads. Another possible extension is to examine how one can deal with decision problems that involve aggregate queries. Finally, it is worth studying how our methodology can be used by autonomous agents that base their decisions on data from multiple websites.

8. Electronic Companion

An electronic companion to this paper is available as part of the online version that can be found at <http://mansci.journal.informs.org/>.

Acknowledgments

A preliminary and abridged version of this work was presented at the International Conference on Information Systems (ICIS), Washington, D.C., December 2004.

References

- Betts, M. 2001. Data quality should be a boardroom issue. *Computerworld*. (December 17).
- Bilke, A., J. Bleiholder, C. Böhm, K. Draba, F. Naumann, M. Weis. 2005. Automatic data fusion with HumMer. *Proc. 31st Internat. Conf. Very Large Data Bases*, Trondheim, Norway, 1251–1254.
- Bischoff, J., T. Alexander. 1997. *Data Warehouse: Practical Advice from the Experts*. Prentice-Hall, Upper Saddle River, NJ.
- Dekhtyar, A., R. Ross, V. S. Subrahmanian. 2001. Probabilistic temporal databases, I: Algebra. *ACM Trans. Database Systems* 26(1) 41–95.
- Department of Veteran Affairs. 1998. Audit of veterans benefits administration SSA/VA death match procedures. Report 8R4-B01-069, Office of Inspector General, Washington, D.C.
- Dey, D., S. Sarkar. 1996. A probabilistic relational model and algebra. *ACM Trans. Database Systems* 21(3) 339–369.
- Dey, D., T. M. Barron, A. N. Saharia. 1998a. A decision model for choosing the optimal level of storage in temporal databases. *IEEE Trans. Knowledge Data Engrg.* 10(2) 297–309.
- Dey, D., S. Sarkar, P. De. 1998b. A probabilistic decision model for entity matching in heterogeneous databases. *Management Sci.* 44(10) 1379–1395.
- Direct Marketing Association. 2005. Chapter II: Executive summary for 2005. *U.S. Direct Marketing Today: Economic Impact 2005*. <http://www.the-dma.org/research/economicimpact2005/ExecSummary.pdf>.
- Fan, W., H. Lu, S. E. Madnick, D. W. Cheung. 2001. Discovering and reconciling data value conflicts for numerical data integration. *Inform. Systems* 26(8) 635–656.
- Fan, W., H. Lu, S. E. Madnick, D. W. Cheung. 2002. DIRECT: A system for mining data value conversion rules from disparate data sources. *Decision Support Systems* 34(1) 19–39.

- Harris Interactive. 2006. Survey shows majority of information workers have made decisions based on bad data. <http://www.businessobjects.com/products/dataquality/survey.asp>.
- Hegg, C. J. 1998. Data integration technology and applications: An overview. *Drug Benefit Trends* 10(1) 32–34.
- Hernandez, M. A., S. J. Stolfo. 1998. Real-world data is dirty: Data cleaning and the merge/purge problem. *Data Mining and Knowledge Discovery* 2(1) 9–37.
- Krishnan, R., J. Peters, R. Padman, D. Kaplan. 2005. On data reliability assessment in accounting information systems. *Inform. Systems Res.* 16(3) 307–326.
- Lager, M. 2005. Breaking down the silos. *CRM Magazine* 9(7) 48–52.
- Larson, J. A., S. B. Navathe, R. Elmasri. 1989. A theory of attribute equivalence in databases with application to schema integration. *IEEE Trans. Software Engrg.* 15(4) 449–463.
- Lim, E. P., J. Srivastava, S. Shekhar. 1996. An evidential reasoning approach to attribute value conflict resolution in database integration. *IEEE Trans. Knowledge Data Engrg.* 8(5) 707–723.
- Mendelson, H., A. N. Saharia. 1986. Incomplete information costs and database design. *ACM Trans. Database Systems* 11(2) 159–185.
- Moffat, D. W. 1987. *Handbook of Manufacturing and Production Management Formulas, Charts and Tables*. Prentice Hall, Upper Saddle River, NJ.
- Morey, R. C. 1982. Estimating and improving the quality of information in a MIS. *Comm. ACM* 25(5) 337–342.
- Raju, K. V. S. V. N., A. K. Majumdar. 1988. Fuzzy functional dependencies and lossless join decomposition of fuzzy relational database systems. *ACM Trans. Database Systems* 13(2) 129–166.
- Ram, S., J. Park. 2004. Semantic conflict resolution ontology (SCROL): An ontology for detecting and resolving data and schema level conflicts. *IEEE Trans. Knowledge Data Engrg.* 16(2) 189–202.
- Raman, A., N. DeHoratius, Z. Ton. 2001. Execution: The missing link in retail operations. *California Management Rev.* 43(3) 136–152.
- Strong, D. M., Y. W. Lee, R. Y. Wang. 1997. Data quality in context. *Comm. ACM* 40(5) 103–110.
- Trembly, A. C. 2002. Poor data quality: A \$600 billion issue. *National Underwriter Life & Health/Financial Services Edition* 106(11) 48.
- Verton, D. 2003. Bush orders integration of U.S. terrorist watch lists. *Computerworld*. (September 22).
- Wadehra, A. 2006. Poor data governance: Why customer data integration (CDI) projects fail, Part III. *DM Direct Special Report*. (May 18), <http://www.iw.com/node/364>.
- Ziff Davis Media. 2006. Data management dynamics: The ROI from data quality. http://www.oracle.com/data_hub/roi-from-data-quality-white-paper.pdf.