Association for Information Systems
AIS Electronic Library (AISeL)

ICIS 2004 Proceedings

International Conference on Information Systems (ICIS)

December 2004

Reconciling Attribute Values from Multiple Data Sources

Zhengrui Jiang
University of Texas at Dallas

Sumit Sarkar
University of Texas at Dallas

Prabuddha De
Purdue University

Debabrata Dey
University of Washington

RECONCILING ATTRIBUTE VALUES FROM MULTIPLE DATA SOURCES

Zhengrui Jiang
School of Management
University of Texas at Dallas
Richardson, TX U.S.A.
zxj011000@utdallas.edu

Sumit Sarkar
School of Management
University of Texas at Dallas
Richardson, TX U.S.A.
sumit@utdallas.edu

Prabuddha De
Krannert School of Management
Purdue University
West Lafayette, IN U.S.A.
pde@purdue.edu

Debrabata Dey
University of Washington
Business School
Seattle, WA U.S.A.
ddey@uwashington.edu

Abstract

Because of the heterogeneous nature of multiple data sources, data integration is often one of the most challenging tasks of today's information systems. While the existing literature has focused on problems such as schema integration and entity identification, our current study attempts to answer a basic question: When an attribute value for a real-world entity is recorded differently in two databases, how should the "best" value be chosen from the set of possible values? We first show how probabilities for attribute values can be derived, and then propose a framework for deciding the cost-minimizing value based on the total cost of type I, type II, and misrepresentation errors.

Keywords: Data integration, heterogeneous databases, probabilistic databases, misclassification errors, misrepresentation errors

Introduction

Business decisions often require data from multiple sources. As has been widely documented, integrating data from several existing independent databases poses a variety of complex problems. Data integration problems arising from heterogeneous data sources can be divided into two broad categories: the schema-level problems and instance level-problems (Rahm and Do 2000). Topics such as schema integration (Batini et al. 1986) and semantic conflict resolution (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 duplication removal (Hernandez and Stolfo 1998) belong to the second. All of these problems have been extensively studied and various solutions have been proposed. However, after schema integration and entity matching, another problem emerges: 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 conflicting attribute values when we merge, for example, alumni data stored separately by a university and by one of its departments and encounter two different work addresses for the same person?

One solution is to store all conflicting values with associated probabilities; probabilistic relational models (Dekhtyar et al. 2001; Dey and Sarkar 1996) have been proposed in that context. However, despite the theoretical progress on the probabilistic database model, it is not commercially available as yet. Even if it becomes readily available in the future, because of the significant

overhead associated with the storing and handling of the probabilistic data, it remains to be seen whether it is cost-justifiable to implement a probabilistic database model. Therefore, storing the most likely value or the "best" value based on some given criteria seems to be a more practical solution at this point.

To choose a single deterministic value, we need to first evaluate the probabilities assigned to each conflicting value. Various pieces of information, such as values of related attributes, time stamps of stored values in different data sources, and data source reliabilities, may be utilized to estimate the probability distributions associated with all possible true attribute values. The approach we propose in this study is as follows: We first estimate the probability for each conflicting attribute value based on source data (attribute) reliability, and then determine the best value to store, based on total expected error costs associated with each candidate value. We examine stochastic attributes with only discrete domains in this study.

The paper is organized as follows. In the next section, we derive the probability associated with each possible attribute value based on source data reliability. We then classify queries based on possible errors that may result from incorrect attribute values. Such classifications are used in computing the total expected cost associated with incorrect values being stored in the database. We demonstrate how the cost minimizing attribute values can be determined for a discrete attribute. We extend the solution to multiple discrete attributes. The last section provides concluding remarks and discusses possible extensions.

Computing Attribute Value Probabilities

We first derive the probabilities for a single discrete attribute. Consider data sources S_1 and S_2. We denote by A_{S1} the value of attribute A for a particular entity instance as observed in S_1, and by A_{S2} the value of attribute A for the same entity instance as observed in S_2. For example, we may find $A = a_1$ in S_1 ($A_{S1} = a_1$) and $A = a_2$ in S_2 ($A_{S2} = a_2$). For any number of reasons, the data in these data sources may be inaccurate (Dey et al. 1998b). We would like to determine the probability that a specific value (which may or may not be the value observed from a data source) is indeed the true value of an attribute. When multiple sources are involved, it would require us to consider the reliability of the different data sources.

In general, the required probability terms can be expressed as $P(A = a_k | A_{S1} = a_i, A_{S2} = a_j)$. We identify the following situations that cover all the possibilities:

Case 1a: $k = i = j$; $P(A = a_i | A_{S1} = a_i, A_{S2} = a_i)$, **Case 1b**: $k \neq i = j$; $P(A = a_k | A_{S1} = a_i, A_{S2} = a_i)$; and
Case 2a: $k = i \neq j$; $P(A = a_i | A_{S1} = a_i, A_{S2} = a_j)$, **Case 2b**: $k \neq i \neq j$; $P(A = a_k | A_{S1} = a_i, A_{S2} = a_j)$.

What is Available?

We can sample S_1 and S_2 to find what proportion of values of attribute A is accurate in S_1, and what proportion of values of attribute A is accurate in S_2, in general. For example, we may sample S_1 and S_2, and find that attribute A is accurate in S_1 80 percent of the time (i.e., 80 percent of the values for attribute A are correct in the sample from S_1), and that it is accurate in S_2 90 percent of the time (Morey 1982). Then,

$P(A = a_i \mid A_{S1} = a_i) = 0.8$, $P(A \neq a_i \mid A_{S1} = a_i) = 0.2$; and
$P(A = a_j \mid A_{S2} = a_j) = 0.9$, $P(A \neq a_j \mid A_{S2} = a_j) = 0.1$.

Assumptions

Our objective is to determine the desired probabilities for the attribute values based on sample estimates from each individual data source. In order to do that, we need to make a few assumptions. These are listed next.

Assumption 1: The value of A recorded in S_1 (i.e., A_{S1}) is not dependent on the value of A recorded in S_2, once we know the true value of A (and vice versa). This implicitly assumes that the causes of errors are independent in the two data sources. Mathematically, this implies

$$P(A_{S1} = a_i \mid A = a_k, A_{S2} = a_j) = P(A_{S1} = a_i \mid A = a_k) \; \forall \; i, j, k.$$

The above assumption would, of course, be violated if the data in one source are derived from the data in the other source.

Assumption 2: Our priors for $P(A = a_i)$, $P(A = a_j)$, etc. are the same if we have no reason *a priori* to believe that one value is more likely to occur than another. This would be true in general when the domain is large or quite unpredictable. Mathematically, this implies

$$P(A = a_k) = 1/ | A | \forall k,$$

where | A | is the number of possible realizations of attribute A. If the domain is restricted to just a few values, and one (or a subset of those values) is known to be predominant, then the appropriate priors should be used. These priors can be incorporated in our estimate (we omit this analysis here for space considerations).

Assumption 3: All possible values of A other than that observed from a particular data source are assumed to be equally likely. Mathematically, this implies

$$P(A = a_k \mid A_{S1} = a_i) = P(A \neq a_i \mid A_{S1} = a_i)/ [|A|-1] = [1- P(A = a_i \mid A_{S1} = a_i)]/ [|A|-1] \; \forall \; k \neq i,$$

where | A | is the number of possible realizations of attribute A. The implications are similar to those of the previous assumption.

Probability Estimates for Attribute Values: Single Attribute Case

Based on the reliability information about an attribute in two data sources and the assumptions discussed above, we derive the probabilities of true values in various situations as follows (detailed derivations are provided in the appendix).

Case 1a: $k = i = j$; $P(A = a_i \mid A_{S1} = a_i, A_{S2} = a_i) =$

$$\frac{P(A = a_i \mid A_{S1} = a_i) \times P(A = a_i \mid A_{S2} = a_i)}{P(A = a_i \mid A_{S1} = a_i) \times P(A = a_i \mid A_{S2} = a_i) + P(A \neq a_i \mid A_{S1} = a_i) \times P(A \neq a_i \mid A_{S2} = a_i) / [| A | -1]}$$

The above expression illustrates that, as the number of possible values of an attribute increases, the likelihood that both sources have incorrectly captured $A = a_i$ goes down.

Case 1b: $k \neq i = j$; $P(A = a_k \mid A_{S1} = a_i, A_{S2} = a_i) =$

$$\frac{P(A \neq a_i \mid A_{S1} = a_i) \times P(A \neq a_i \mid A_{S2} = a_i)/ [| A | -1]^2}{P(A = a_i \mid A_{S1} = a_i) \times P(A = a_i \mid A_{S2} = a_i) + P(A \neq a_i \mid A_{S1} = a_i) \times P(A \neq a_i \mid A_{S2} = a_i) / [| A | -1]}$$

Case 2a: $k = i \neq j$; $P(A = a_i \mid A_{S1} = a_i, A_{S2} = a_j) =$

$$\frac{P(A = a_i \mid A_{S1} = a_i) \times P(A \neq a_j \mid A_{S2} = a_j)}{P(A = a_i \mid A_{S1} = a_i) \times P(A \neq a_j \mid A_{S2} = a_j) + P(A \neq a_i \mid A_{S1} = a_i) \times P(A = a_j \mid A_{S2} = a_j) + P(A \neq a_i \mid A_{S1} = a_i) \times P(A \neq a_j \mid A_{S2} = a_j)[| A | -2] / [| A | -1]}$$

In the situation where A can have only two values a_i and a_j, the above expression becomes

$P(A = a_i \mid A_{S1} = a_i, A_{S2} = a_j) =$

$$\frac{P(A = a_i \mid A_{S1} = a_i) \times P(A \neq a_j \mid A_{S2} = a_j)}{P(A = a_i \mid A_{S1} = a_i) \times P(A \neq a_j \mid A_{S2} = a_j) + P(A \neq a_i \mid A_{S1} = a_i) \times P(A = a_j \mid A_{S2} = a_j)}$$

The expression for $P(A = a_j \mid A_{S1} = a_i, A_{S2} = a_j)$ is analogous.

Case 2b: $k \neq i \neq j$; $P(A = a_k \mid A_{S1} = a_i, A_{S2} = a_j) =$

$$\frac{P(A \neq a_i \mid A_{S1} = a_i) \times P(A \neq a_j \mid A_{S2} = a_j) / [|A| - 1]}{P(A = a_i \mid A_{S1} = a_i) \times P(A \neq a_j \mid A_{S2} = a_j) + P(A \neq a_i \mid A_{S1} = a_i) \times P(A = a_j \mid A_{S2} = a_j) + P(A \neq a_i \mid A_{S1} = a_i) \times P(A \neq a_j \mid A_{S2} = a_j)[|A| - 2] / [|A| - 1]}$$

Probability Estimates for Attribute Values: Multiple Attributes Case

In the above derivation, only one common attribute is considered. The solutions can, however, be easily extended to situations where multiple attributes are common across the databases. If the reliabilities of the attributes are independent of one another, the analysis presented in the previous subsection applies for each attribute individually. When the reliabilities across attributes are dependent, we treat that group of attributes as one *composite attribute* and all possible combinations of the multiple attributes values as the possible realizations of the composite attribute. If two attributes A and B form a composite attribute, then the number of realizations for this composite attribute is $|A| \cdot |B|$. For example, if the values stored for a composite attribute are the same in two locations, then analogous to case 1a, we have

$P(A = a_i, B = b_j \mid A_{S1} = a_i, B_{S1} = b_j, A_{S2} = a_i, B_{S2} = b_j) = \frac{N}{N + D}$, where

$$N = P(A = a_i, B = b_j \mid A_{S1} = a_i, B_{S1} = b_j) \times P(A = a_i, B = b_j \mid A_{S2} = a_i, B_{S2} = b_j),$$

$$D = P(A \neq a_i, B \neq b_j \mid A_{S1} = a_i, B_{S1} = b_j) \times P(A \neq a_i, B \neq b_j \mid A_{S2} = a_i, B_{S2} = b_j) / [|A| \cdot |B| - 1].$$

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

Alumni Database Example

Consider alumni data that are collected independently in both an university database (S_1) and a department database (S_2), and suppose we want to merge them into one database. Some attribute values for the same person could be different in the two sources.

Binary Attribute

Suppose there exists a binary attribute *Self-Employed* (SE for brevity), which can take a value of a_1 = "Yes" or a_2 = "No." Assume that, based on sampling, we have found that this attribute is accurate in S_1 80 percent of the time and that it is accurate in S_2 90 percent of the time, i.e.,

$P(SE = a_i \mid SE_{S1} = a_i) = 0.8$, $P(SE \neq a_i \mid SE_{S1} = a_i) = 0.2$, $\forall$ i = 1, 2; and
$P(SE = a_j \mid SE_{S2} = a_j) = 0.9$, $P(SE \neq a_j \mid SE_{S2} = a_j) = 0.1$, $\forall$ i = 1, 2.

When the stored values of SE in the two sources are different for an alumnus, i.e., $a_i \neq a_j$, we have:

$P(SE = a_i \mid SE_{S1} = a_i, SE_{S2} = a_j) = (0.8 \times 0.1) / (0.8 \times 0.1 + 0.9 \times 0.2) = 8 / 26 = 0.308$,
$P(SE = a_j \mid SE_{S1} = a_i, SE_{S2} = a_j) = (0.9 \times 0.2) / (0.8 \times 0.1 + 0.9 \times 0.2) = 18 / 26 = 0.692$.

In case the stored values in the two sources are the same for a particular person, we have

$P(SE = a_i \mid SE_{S1} = a_i, SE_{S2} = a_i) = (0.8 \times 0.9) / (0.8 \times 0.9 + 0.2 \times 0.1) = 72 / 74 = 0.973$,
$P(SE = a_k \mid SE_{S1} = a_i, SE_{S2} = a_i) = (0.2 \times 0.1) / (0.8 \times 0.9 + 0.2 \times 0.1) = 2 / 74 = 0.027$.

Multi-Valued Attribute

Suppose the original alumni database also stores the current home location of the alumni and the attribute *Home_Location* (HL for brevity) can be any of the 50 states in the United States. We may find that, for instance, the *Home_Location* for an alumnus Robert Black is stored as "TX" in the university database S_1 and as "LA" in the department database S_2. Assuming that the attri-

Table 1. Alumni Data

A_ID	FName	LName	Employer	Title	Home_Location Value	Prob.
10001	Robert	Black	Walmart	Sales Manager	TX	0.286
					LA	0.644
					AOV	0.00146
10002	Timothy	Earnest	GTE	Accountant	NY	0.99943
					AOV*	1.15627×10^{-05}

*AOV – Any Other Value. The probability for AOV reflects the probability that any other specific attribute value except those listed separately is true. For example, from Table 1 we know that the probability that California is the true *Home_Location* for Robert Black is 0.00146.

bute is accurate in S_1 80 percent of the time and accurate in S_2 90 percent of the time, we are able to calculate the distribution of the true home location values for Robert Black as follows:

$|HL| = 50$ (i.e., there are 50 possible state values).
$P(HL = TX \mid HL_{S1} = TX, HL_{S2} = LA) = (0.8 \times 0.1) / (0.8 \times 0.1 + 0.9 \times 0.2 + 0.2 \times 0.1 \times 48/49) = 0.286,$
$P(HL = LA \mid HL_{S1} = TX, HL_{S2} = LA) = (0.9 \times 0.2) / (0.8 \times 0.1 + 0.9 \times 0.2 + 0.2 \times 0.1 \times 48/49) = 0.644,$
$P(HL = HL_k \mid HL_{S1} = TX, HL_{S2} = LA) = (0.2 \times 0.1/49)/(0.8 \times 0.1+0.9 \times 0.2+0.2 \times 0.1 \times 48/49) = 0.00146,$
(for each HL_k other than TX or LA.)

On the other hand, if the *Home_Location* shown for Timothy Earnest in both data sources is "NY," the value distribution is as follows:

$P(HL = NY \mid HL_{S1} = NY, HL_{S2} = NY) = (0.8 \times 0.9) / (0.8 \times 0.9+0.2 \times 0.1/49) = 0.99943,$
$P(HL = HL_k \mid HL_{S1} = NY, HL_{S2} = NY) = (0.2 \times 0.1/2401) / (0.8 \times 0.9+0.2 \times 0.1/49) = 1.15627 \times 10^{-05},$
(for each HL_k other than NY.)

Table 1 summarizes the *Home_Location* value distribution under the two different cases.

Classification of Queries and Errors

As in prior research (Dey et al. 1998a; Mendelson and Saharia, 1986), we assume that all relevant queries have been identified. Based on where the attribute being examined appears in a query, we categorize the relevant queries into three classes. If the stochastic attribute(s) appear only in the selection condition of a query, we call this query a *class C* (Conditioning) *query*. If the attribute(s) appear only in the projection list of a query, we call this query a *class T* (Targeting) *query*. We call a query a *class CT query* if the attribute(s) being examined appear in both the selection condition and the projection list. In the alumni example, if *Home_Location* is a stochastic attribute, then query Q1 is of class C, Q2 is of class T, and Q3 is of class CT.

Q1: *Display ID of those alumni who live in LA.*
Q2: *Display Name and Home_Location of all alumni who work for GTE.*
Q3: *Display ID, Name, and Home_Location of all alumni who live in TX.*

If the stored value of an attribute is not the true value, three types of errors can occur. A *type I error* occurs when an object should have been selected by a query based on the true value of an attribute, but was not selected because the stored value was different from the true value. A *type II error* occurs when an object that should not have been selected based on the true attribute value was selected because of the incorrectly recorded value. A *misrepresentation error* occurs when the value displayed for an attribute in a query output is not the true value.

The following parameters are applicable to all classes of queries:

(1) $f(q)$ Frequency of query q.

Table 2. Cost Matrix for Three Classes of Queries

Error Type / Query class	Type I	Type II	Misrepresentation
(C) Conditioning	$\alpha(q)$	$\beta(q)$	N/A
(T) Targeting	N/A	N/A	$g(q)$
(CT) Conditioning and Targeting	$\alpha(q)$	$\beta(q)$	$\gamma(q)$

(2) $\alpha(q)$ Cost of type I error for query q.

(3) $\beta(q)$ Cost of type II error for query q.

(4) $\gamma(a, q)$ Average cost of one occurrence of misrepresenting attribute a in the query output of q. The parameter a is omitted in the single attribute problem.

(5) $\tau(q)$ Expected percentage of objects in a relation that may be selected by query q. For the simple query, *display all alumni who live in Texas*, $\tau(q)$ equals the expected percentage of employees who live in Texas. For the complex query, *display all alumni who live in Texas AND are at least 50 years old*, $\tau(q)$ equals the product of the expected percentage of employees who live in Texas and the expected percentage of employees who are at least 50 years old, assuming that attributes *Home_Location* and *AGE* are independent of each other.

The three cost parameters of a query are estimated based on the utilization of the query output. Consider a direct marketing firm that runs a query based on some given criteria to identify potential customers. In this case, the expected net profit per potential customer and the average cost of sending a promotion to each potential customer constitute the type I error cost and the type II error cost, respectively. While both the type I error cost and the type II error cost are unique for a particular query, the misrepresentation cost is specific to a query as well as to one of its attributes displayed in the query output. In the direct marketing example, if the potential customer's street address is in the query output, then the cost of misrepresenting the street address of a potential customer equals the expected net profit per potential customer times the probability that the mail would be lost or returned due to the incorrect address information.

The three classes of queries and their relevant error types are summarized in Table 2.

Attribute Reconciliation: Single Stochastic Attribute

We start our analysis with the simplest case where there is only one stochastic attribute in a relation. We use the alumni example shown in Table 1 to illustrate how the cost-minimizing value for the attribute *Home_Location* can be determined for Robert Black, given the value distribution listed in Table 1. We start our analysis using a set of simple queries, follow it by some more complex queries, and finally provide the standardized procedure for the analysis.

An Example with Queries Having a Single Clause in the Condition

For the purpose of illustration, we assume that the three queries discussed in the previous section are the only queries relevant to the *Home_Location* attribute, i.e, only these three queries have the *Home_Location* attribute either in the selection condition or in the projection list. All three queries have a single clause in the condition. We first calculate the expected type I, type II, and misrepresentation error costs when different values are chosen to be stored in the merged table, and then compare the total costs to determine the best value to store.

Cost of Type I and Type II Errors. We first examine the cost of type I and type II errors when "TX" is the stored *Home_Location* value for Robert Black. Only Q1 and Q3 need to be considered for type I and type II errors. Obviously, Q1 will not select Robert Black if the stored *Home_Location* value for him is "TX." Given that he is not selected, if Robert Black's true

Table 3. Cost of Type I and Type II Errors (If "TX" is Stored)

Query	Retrieval Criterion	Query Result	If True Value is	Type I Error Cost	Type II Error Cost
Q1 (C)	LA	Not Select	Not LA	0	N/A
			LA	$\alpha(Q1)f(Q1)P(LA)$	N/A
Q3 (CT)	TX	Select	TX	N/A	0
			Not TX	N/A	$\beta(Q3)f(Q3)[1-P(TX)]$

Home_Location is indeed LA, then a type I error occurs. The frequency of this occurrence equals the product of the frequency $f(Q1)$ and the probability that the true value is "LA," denoted by P(LA). On the other hand, when Q3 is processed, Robert Black will be selected. Given that Robert Black has been selected based on the stored deterministic value "TX," if the true value is not TX, but LA or any other value, then a type II error occurs. The frequency of this occurrence equals the product of the frequency $f(Q3)$ and the probability that the true value is not "TX," which equals (1-P(TX)). The above analysis is summarized in Table 3. By multiplying the error frequencies with the cost parameters for each query, we obtain the total of type I and type II error costs resulting from choosing "TX" as the stored value:

$$C_{I,II}(TX) = \alpha(Q1)f(Q1)P(LA) + \beta(Q3)f(Q3)[1 - P(TX)]. \quad (1)$$

Similarly, the type I and type II error costs resulting from choosing "LA" as the stored value for Robert Black (shown below) is obtained based on the analysis presented in Table 4.

$$C_{I,II}(LA) = \beta(Q1)f(Q1)[1 - P(LA)] + \alpha(Q3)f(Q3)P(TX). \quad (2)$$

Table 5 shows the type I and type II error costs incurred if any value other than "TX" and "LA" is stored, and equation (3) shows the resulting cost expression:

$$C_{I,II}(AOV) = \alpha(Q1)f(Q1)P(LA) + \alpha(Q3)f(Q3)P(TX). \quad (3)$$

In this example, the cost analysis shown in Table 5 is also valid if "NULL" is chosen. Therefore, we have

$$C_{I,II}(NULL) = \alpha(Q1)f(Q1)P(LA) + \alpha(Q3)f(Q3)P(TX). \quad (4)$$

Table 4. Cost of Type I and Type II Errors (If "LA" is Stored)

Query	Retrieval Criterion	Query Result	If True Value is	Type I Error Cost	Type II Error Cost
Q1 (C)	LA	Select	LA	N/A	0
			Not LA	N/A	$\beta(Q1)f(Q1)[1-P(LA)]$
Q3 (CT)	TX	Not Select	Not TX	0	N/A
			TX	$\alpha(Q3)f(Q3)P(TX)]$	N/A

Table 5. Cost of Type I and Type II Errors (If Any Value Other than "TX and "LA" is Stored)

Query	Retrieval Criterion	Query Result	If True Value is	Type I Error Cost	Type II Error Cost
Q1 (C)	LA	Not Select	Not LA	0	N/A
			LA	$\alpha(Q1)f(Q1)P(LA)$	N/A
Q3 (CT)	TX	Not Select	TX	$\alpha(Q3)f(Q3)P(TX)$	N/A
			Not TX	0	N/A

Cost of Misrepresentation Errors. A misrepresentation error occurs when a value in a query output is not the true attribute value, and this type of error is relevant to only class T and class CT queries. In our example, Q2 is a class T query and Q3 is a CT query. We first assume that "TX" is chosen to be the deterministic value for Robert Black. Therefore, whenever Robert Black is selected by a query and *Home_Location* is in the projection list, the value displayed will be "TX." Given that "TX" is displayed in the query output, if the actual true value is not "TX," a misrepresentation error occurs. The frequency of this occurrence equals the frequency that Robert Black is selected by a class T or class CT query times the probability that "TX" is not the true value. To calculate the frequency that Robert Black is selected, we examine Q2 and Q3 separately. Since Q3 always selects Robert Black if "TX" is chosen to be the deterministic value, the frequency that Robert Black is selected by Q3 equals the frequency of the query $f(Q3)$. For the class T query Q2, we assume that all objects are equally likely to be selected by this query. Therefore, the frequency that Robert Black is selected by Q2 equals $f(Q2)\tau(Q2)$. Multiplying the error frequencies by γ, the unit cost of a misrepresentation error, and summing over all class T and class CT queries, we obtain the total misrepresentation cost when "TX" is chosen to be the stored value as follows:

$$C_m(TX) = [\gamma(Q3)f(Q3) + \gamma(Q2)f(Q2)\tau(Q2)][1 - P(TX)]. \quad (5)$$

If "LA" had been chosen to be the stored value, Robert Black would never appear in the query output of Q3. The total misrepresentation cost, denoted by $C_m(LA)$, thus equals the following:

$$C_m(LA) = \gamma(Q2)f(Q2)\tau(Q2)[1 - P(LA)]. \quad (6)$$

Now consider when a value other than "TX" or "LA" is chosen for Robert Black. We can ignore Q3 since it will not select Robert Black. The misrepresentation cost is straightforward:

$$C_m(AOV) = \gamma(Q2)f(Q2)\tau(Q2)[1 - P(AOV)]. \quad (7)$$

Finally, we examine the misrepresentation cost if "NULL" is stored. For simplicity, we assume that "NULL" is never the true value and the unit misrepresentation cost when "NULL" or any other incorrect value is stored is the same. Then the resulting misrepresentation cost is

$$C_m(NULL) = \gamma(Q2)f(Q2)\tau(Q2). \quad (8)$$

If the cost of misrepresentation is different when "NULL" is stored, then we can estimate another misrepresentation parameter γ' specifically for "NULL" and replace γ in (8) by γ'. All other analyses remain the same.

Minimizing Total Error Cost. The best deterministic *Home_Location* value for Robert Black is chosen by minimizing the total expected error cost, obtained by summing up the cost of type I and type II errors and the cost of misrepresentation errors:

$$TC(TX) = \alpha(Q1)f(Q1)P(LA) + [\beta(Q3)f(Q3) + \gamma(Q3)f(Q3) + \gamma(Q2)f(Q2)\tau(Q2)][1 - P(TX)] \quad (9)$$

$$TC(LA) = \alpha(Q3)f(Q3)P(TX) + [\beta(Q1)f(Q1) + \gamma(Q2)f(Q2)\tau(Q2)][1 - P(LA)] \quad (10)$$

$$TC(AOV) = \alpha(Q1)f(Q1)P(LA) + a(Q3)f(Q3)P(TX) + \gamma(Q2)f(Q2)\tau(Q2)][1 - P(AOV)], \quad (11)$$

$$TC(NULL) = \alpha(Q1)f(Q1)P(LA) + \alpha(Q3)f(Q3)P(TX) + \gamma(Q2)f(Q2)\tau(Q2). \quad (12)$$

The value that minimizes the total cost should be the one stored in the merged table. Depending on the cost parameters, any of the values can be chosen. In normal situations, TX or LA should be the likely best value. However, in cases where the cost of type II errors is significantly higher than cost of type I errors, AOV 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.

Additional Queries with Disjunctive Clauses in the Condition

The example discussed above involves three simple queries with a single clause in the condition. To generalize the solution, we consider three additional queries Q4, Q5, and Q6 that have disjunctive clauses in the selection condition. Among them, Q4 is of class CT and Q5 and Q6 are of class C.

Q4: *Display ID, Name, and Home_Location of those alumni who live in OK or TX.*
Q5: *Display ID of those alumni who live in CA, NY, or TX.*
Q6: *Display ID, Name, and Employer of those alumni who live in IN, MN, or PA.*

We use the example data for Robert Black to illustrate how the costs associated with these new queries can be determined. Assume that "TX" is the stored value. We first derive the misrepresentation cost since it is relatively simple. As discussed earlier, for the misrepresentation cost, we only need to consider the Class T queries and those class CT queries that select Robert Black. Therefore, only Q4 needs to be considered for the misrepresentation cost. Based on the discussion presented above, the misrepresentation cost associated with Q4 is $[1 - P(TX)][\gamma(Q4)\, f(Q4)]$. The costs of type I and type II errors are summarized in Table 6. Based on Table 6 and Table 3, we make the following observations:

Observation 1: Given that the chosen deterministic value is included in the retrieval criterion of a query (such as Q3, Q4, and Q5), the probability of type II error equals the probability that a value other than those included in the retrieval criterion is the true value.

Observation 2: If the chosen deterministic value is not included in the retrieval criterion of a query (e.g., Q1 or Q6), then the probability of type I error equals the probability that one of the values included in the query's retrieval criterion is the true value.

To determine which value is the best choice, the total cost associated with other possible values also needs to be calculated. The value that results in the smallest cost should be stored.

Table 6. Cost of Type I and Type II Errors
(If "TX" is chosen to be the deterministic value for Robert Black)

Query No.	Retrieval Criterion	Query Result	If True Value is	Type I Error cost	Type II Error cost
Q4 (CT)	OK, TX	Select	OK or TX	N/A	0
			Others	N/A	β(q4)f(q4)[1-P(OK)-P(TX)]
Q5 (C)	CA, NY, TX	Select	CA, NY or TX	N/A	0
			Others	N/A	β(q5)f(q5)[1-P(CA)-P(NY)-P(TX)]
Q6 (C)	IN, MN, PA	Not Select	IN, MN, or PA	α(Q6)f(Q6)[P(IN)+P(MN)+P(PA)]	N/A
			Others	0	N/A

Table 7. Query Coverage Bitmap

Criterion / Queries	…	CA	IN	LA	MN	NY	OK	PA	TX	…	PS
Q1 (C)	…	0	0	1	0	0	0	0	0	…	P(LA)
Q3 (CT)	…	0	0	0	0	0	0	0	1	…	P(TX)
Q4 (CT)	…	0	0	0	0	0	1	0	1	…	P(OK) + P(TX)
Q5 (C)	…	1	0	0	0	1	0	0	1	…	P(CA) + P(NY) + P(TX)
Q6 (C)	…	0	1	0	1	0	0	1	0	…	P(IN) + P(MN) + P(PA)

A Standardized Procedure Based on Query Coverage Bitmap

As we can see from the previous analyses, if the number of possible realizations of the attribute or the number of relevant queries is large, the error cost calculation can be a tedious process. To simplify the computation, we construct a *query coverage bitmap* as shown in Table 7. The query coverage bitmap summarizes the values included in the retrieval criterion of each query. For example, for Q5, the columns for CA, NY, and TX are marked as "1" since these three state values are included in the retrieval criterion of Q5. The last column, labeled as "PS," represents the probability sum that any one of the attribute values included in the query's retrieval criterion is the true attribute value.

Input: Probabilistic value vector ***V*** and corresponding probability vector ***P***.

1. Find out the corresponding column index numbers in the Query Coverage Bitmap for all components of ***V*** and keep them in vector ***J***.
2. Let C_{min} = *A very large number*; *BestVal*= V_0.
   For all $j \in \boldsymbol{J}$ (representing all probabilistic values):
   Begin
   $TC = 0; C_I = 0; C_{II} = 0; C_m = (1 - P_j) \sum_{q \in \text{class T quries}} f(q)\tau(q)\gamma(q)$.
   For each query with row index *i*:
   Do
   If QCBitmap[*i*][*j*] equals 1, /*Query *i* selects the object, possible type II errors and misrepresentation errors.*/
   Then
   increase cost of type II error C_{II} by $f(q_i)\beta(q_i)$[1-PS(q_i)]; and
   if q_i is class CT query, then increase misrepresentation cost C_{m} by (1-P_j)$f(q_i)\gamma(q_i)$;
   Else /*Query *i* will not select the object, possible type I errors */
   increase cost of type I error C_{I} by $f(q_i)\alpha(q_i)$PS(q_i).
   Endif
   End
   $TC = C_I + C_{II} + C_m$.
   If ($TC < C_{min}$)
   Then $C_{min} = TC$, $BV = V_{\text{j}}$.
   End

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

Figure 1. Procedure for Determining the Best Value

Based on the bitmap, we can automate the cost calculation and value determination process. Figure 1 shows the algorithm for determining the best deterministic value for an object with a probabilistic value vector $\boldsymbol{V}$ and a corresponding probability distribution $\boldsymbol{P}$. The total cost associated with each chosen value is determined as follows: First, if there are class T queries, the associated misrepresentation cost is calculated. Second, the column in the query coverage bitmap that corresponds to the chosen value is identified. Third, 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 is 1, the cost of type II errors is calculated based on observation 1 discussed in the previous subsection, and the misrepresentation cost associated with this query is determined if the query is of Class CT; if the value is 0, the cost of type I error is calculated based on observation 2 in the previous subsection. Fourth, the total costs associated with the chosen value are determined by summing up all three types of costs. Finally, the best deterministic value is chosen based on the total error costs. If the number of queries is n and the number of probabilistic values is r, then the complexity of this standardized procedure is O(nr).

Discussion

For the above procedure, not all possible values need to be explicitly examined to decide which one is the best. If, in a group of candidate values, all have the same probability of being the true value, and one of the following two conditions holds: (1) none of the candidate values appears in any query or (2) if one value appears in a query, then all other candidate values in the group also appear in the query in exactly the same manner, then the total expected cost associated with each value in the group is always the same. In the given example, the expected cost when either "CA" or "NY" is chosen is the same; the expected cost associated with choosing "IN" or "MN" is the same; and the costs associated with all other values except "TX," "LA," "CA," "NY," "IN," "MN," "OK" and "NULL" are also the same.

Attribute Reconciliation: Multiple Stochastic Attributes

In the previous section, we have shown how the cost-minimizing value can be determined for a single stochastic attribute based on query parameters. In this section, we extend the analysis to multiple stochastic attributes with conflicting values from different data sources. There are two possible cases:

(1) The reliabilities for the stochastic attributes are mutually independent. In this case, the cost-minimizing value for each attribute can be determined individually without considering other attributes. The probability derivation for a single attribute shown in the second section and the value determination process discussed in the previous section can be applied.

(2) The reliabilities are dependent. In that scenario, we have to consider the combinations of attribute values and their joint probabilities.

As shown earlier, we can estimate the joint probabilities for all feasible value combinations. The cost-minimizing value combination can be determined based on the total expected error cost, which includes, as in the single attribute case, the costs of type I errors, type II errors, and misrepresentation errors.

To illustrate how the cost-minimizing value combination can be determined, consider a modified alumni database example as shown in Table 8. In this example, we assume that the values of both attributes *Employer* and *Home_Location* are different for Robert Black in the two data sources. Further assume that the probabilities are as shown in Table 8 for the combination of *Employer* and *Home_Location*. The number of realizations for *Employer* is assumed to be 200 and the number of possible *Home_Locations* is again assumed to be 50. Thus, the total number of possible realizations of the composite attribute is 200 × 50 = 10,000.

Table 8. Modified Probabilistic Alumni Data

A_ID	FName	LName	Title	Employer	H_L	Prob.
10001	Robert	Black	Sales Manager	(Walmart, (Nortel, AOV	TX) LA)	0.6 0.3 1.02×10^{-05}

Consider the situation when the following are the only queries that have the above two attributes either in their projection lists or in their selection conditions:

Q7: *Display Name, Employer, and Home_Location of those alumni who are managers.*
Q8: *Display ID, Name, and Home_Location of all alumni who work for WalMart.*
Q9: *Display ID and Name of those alumni who work for WalMart OR live in LA.*
Q10 *Display Name and Home_Location of those alumni who work for Nortel AND live in TX.*

Among the four queries, since Q7 has both *Employer* and *Home_Location* in its project list, it is of class T with respect to both *Employer* and *Home_Location*. Similarly, Q8 is of class C with respect to *Employer* and of class T with respect to *Home_Location*; Q9 is of class C with respect to both *Employer* and *Home_Location*; and Q10 is of class C with respect to *Employer* and of class CT with respect to *Home_Location*. Among the four queries, Q9 and Q10 deserve special attention since both attributes appear in the selection condition of the two queries. The difference is that the two parts of the selection condition in Q9 are connected by the OR operator and those in Q10 are connected by the AND operator.

We illustrate how the costs of type I, type II, and misrepresentation errors associated with each query are determined for the example shown in Table 8. We first consider the case when (Walmart, TX) is the stored value for Robert Black in the merged table.

Cost of Misrepresentation Errors. For misrepresentation errors, only those class T and class CT queries with respect to either *Employer* or *Home_Location*, i.e., queries that include either *Employer* or *Home_Location* or both in their project list, need to be considered. In our example, Q7 and Q8 are the only class T queries and Q10 is the only class CT query. The misrepresentation cost associated with each query equals the product of the unit cost of misrepresentation error, the frequency of the query, the probability that the target object is selected by the query, and the marginal probability that the stored value is not the true value. If more than one examined attribute is in the projection list (e.g., Q7), the misrepresentation errors equal the sum of the misrepresentation errors computed separately for each attribute. For example, the misrepresentation errors associated with Q7, Q8, and Q10 are as follows:

$$C_{mQ7}(\text{Walmart, Tx}) = \gamma(\text{Q7, } \textit{Home\_Location})f(\text{Q7})\,\tau(\text{Q7})[1 - P(\text{TX})] + \gamma(\text{Q7, } \textit{Employer})f(\text{Q7})\,\tau(\text{Q7})[1 - P(\text{Walmart})],$$

$$C_{mQ8}(\text{Walmart, Tx}) = \gamma(\text{Q8, } \textit{Home\_Location})f(\text{Q8})\,\tau(\text{Q7})[1 - P(\text{TX})], \text{ and}$$

$$C_{mQ10}(\text{Walmart, Tx}) = 0$$

In the above expressions, the marginal probabilities are P(TX) = 0.602 and P(Walmart) = 0.6005. The expression for $C_{m,Q8}$ does not contain τ(Q8) since this query always selects Robert Black, given that (Walmart, TX) is stored. $C_{m,Q10}$ equals zero because Q10 never selects Robert Black based on the stored values (Walmart, TX). From the above example, we can see that, for misrepresentation error costs, the solution is similar to that for the single attribute case, except that the probability of a single value is replaced by the marginal distribution in the multiple attribute case.

Cost of Type I and Type II Errors. For type I and type II errors, we only need to consider class T or class CT queries with respect to *Employer* or *Home_Location*. Therefore, we can ignore Q7. The costs of type I and type II errors associated with Q8, Q9, and Q10 are summarized in Table 9. We observe that, if only one of the multiple attribute being examined is in the selection condition of a query (e.g., Q8), the same solution that we derive for a single stochastic attribute can be used. For queries with more than one attributes being examined in its selection condition, the same rules still apply: if the object is selected based on the stored attribute values, the cost of type II errors equals the product of the unit type II error cost, the frequency of the query, and the probability that selection condition is violated; if the object is not selected, then cost of type I error equals the product of the unit type I error cost, the frequency of the query, and the probability that the selection condition is satisfied. Although the probability that a selection condition is satisfied or violated is slightly more complex with multiple attributes being considered, as shown in Table 9, it can be derived without much difficulty.

To decide the cost-minimizing value for both *Employer* and *Home_Location* for Robert Black, the total expected cost associated with other value combinations also needs to be examined. Since values that need to be separately examined for *Employer* include "Walmart," "Nortel," "NULL," and any other value, and those for *Home_Location* include "TX," "LA," "NULL," and any other value, there are a total of only 16 cases, instead of 10,000 potential cases, to be examined in order to decide the cost-minimizing value for both attributes.

Table 9. Cost of Type I and Type Error for Two Stochastic Attributes (If (Walmart, TX) is stored)

Query No.	Retreival Criteria		Query Result	If True Vaues are	Type I Error Cost	Type II Error Cost
	Employer	H_L.				
Q8	Walmart		Select	(Walmart, -)	N/A	0
				Others	N/A	β(Q8)f(Q8)[1 – P(Walmart)]
Q9 (OR)	Walmart	LA	Select	(Walmart, -)	N/A	0
				(-, LA)	N/A	0
				Others	N/A	β(Q9)f(Q9)[1 – P(Walmart) – P(LA) + P(Walmart, LA)]
Q10	Nortel	TX	Not Select	(Nortel, TX)	α(Q10)f(Q10)[1 – P(Nortel, TX)]	N/A
				Others	0	N/A

Discussions and Future Research

We have shown how the cost-minimizing values can be determined for discrete attributes based on source reliability information and query information. Based on the proposed procedure, when conflicting data values for a real-world entity are encountered in the data integration process, the cost-minimizing value can be determined and stored in the merged table. Subsequently, queries can be directly executed on the merged table. We call this approach *deterministic integration*. Compared with *probabilistic integration*, i.e., storing all probabilistic values based on the probabilistic database model, the main disadvantage of the deterministic approach is the loss of potentially useful distribution information. The advantages of deterministic integration include the following: First, data storage and subsequent query processing can be efficiently handled by the existing commercial database systems. With probabilistic integration, since the probabilistic algebra is not currently supported by standard database packages, the cost of implementing such a probabilistic model could be prohibitively high. Second, the storage cost is lower with deterministic integration. This is because with the probabilistic relational model (e.g., Dey and Sarkar 1996), a new column has to be added even if only one object in the table has uncertain values for only one attribute. In addition, a row has to be added to the table for every probabilistic value associated with each object. Third, with deterministic integration, the operational performance of the resulting database is better. This is because the computational overhead associated with a probabilistic database is avoided with a deterministic table.

The procedures we propose here are for discrete attributes only. An extension to this study is to examine stochastic attributes with continuous domains and a mixture of discrete attributes and continuous attributes. Computationally, in the multiple attributes scenario, if the number of realizations of each attribute or the number of queries being considered is large, the computational overhead could increase significantly. We are trying to identify rules or patterns that can help reduce the computational effort.

References

Batini, C., Lenzerini, M., and Navathe, S. B. "A Comparative Analysis of Methodologies for Database Schema Integration," *ACM Computing Surveys* (18:4), December 1986, pp. 323-364.

Dekhtyar, A., Ross, R., and Subrahmanian, V. S. "Probabilistic Temporal Databases, I: Algebra," *ACM Transactions on Database Systems* (26:1), March 2001, pp. 41-95.

Dey, D., Barron, T. M., and Saharia, A. N. "A Decision Model for Choosing the Optimal Level of Storage in Temporal Databases," *IEEE Transactions on Knowledge and Data Engineering* (10:2), February 1998a, pp. 297-309.

Dey, D., and Sarkar, S. "A Probabilistic Relational Model and Algebra," *ACM Transactions on Database Systems (TODS)* (21:3), September 1996, pp. 339-369.

Dey, D., Sarkar, S., and De, P. "A Probabilistic Decision Model for Entity Matching in Heterogeneous Databases," *Management Science* (44:10), October 1998b, pp. 1379-1395.

Hernandez, M. A., and Stolfo, S. J. "Real-World Data is Dirty: Data Cleaning and the Merge/Purge Problem," *Data Mining and Knowledge Discovery* (2:1), January 1998, pp. 9-37.

Mendelson, H., and Saharia, A. N. "Incomplete Information Costs and Database Design," *ACM Transactions on Database Systems (TODS)* (11:2), June 1986, pp.159-185.

Morey, R. C. "Estimating and Improving the Quality of Information in a MIS," *Communication of the ACM* (25:5), May 1982, pp. 337-342.

Rahm, E., and Do, H. H. "Data Cleaning: Problems and Current Approaches," *IEEE Bulletin of the Technical Committee on Data Engineering* (23:4), December 2000, pp. 3-13.

Ram, S., and Park, J. "Semantic Conflict Resolution Ontology (SCROL): An Ontology for Detecting and Resolving Data and Schema Level Conflicts," *IEEE Transactions on Knowledge and Data Engineering* (16:2), February 2004, pp. 189-202.

Appendix. Probability Derivations for One Attribute, Two Data Sources

We first show some simplifications for the general expression that apply to all of the cases.

I. $P(A_{S1} = a_i, A_{S2} = a_j) = \Sigma_m P(A_{S1} = a_i, A_{S2} = a_j, A = a_m)$
$= \Sigma_m P(A_{S1} = a_i, A_{S2} = a_j \mid A = a_m) \times P(A = a_m) = \Sigma_m P(A_{S1} = a_i \mid A = a_m) \times P(A_{S2} = a_j \mid A = a_m) \times P(A = a_m)$
$= \Sigma_m [P(A = a_m \mid A_{S1} = a_i) \times P(A_{S1} = a_i)/ P(A = a_m)] \times [P(A = a_m|A_{S2} = a_j) \times P(A_{S2} = a_j)/P(A = a_m)] \times P(A = a_m)$
$= \Sigma_m P(A = a_m \mid A_{S1} = a_i) \times P(A = a_m|A_{S2} = a_j) \times P(A_{S1} = a_i) \times P(A_{S2} = a_j) / P(A = a_m)$
$= P(A_{S1} = a_i) \times P(A_{S2} = a_j) \times \Sigma_m P(A = a_m \mid A_{S1} = a_i) \times P(A = a_m|A_{S2} = a_j) / P(A = a_m).$

II. $P(A = a_k \mid A_{S1} = a_i, A_{S2} = a_j) = P(A_{S1} = a_i, A_{S2} = a_j \mid A = a_k) \times P(A = a_k) / P(A_{S1} = a_i, A_{S2} = a_j)$

Analogous to I, we can show: $P(A_{S1} = a_i, A_{S2} = a_j \mid A = a_k) \times P(A = a_k) = P(A = a_k \mid A_{S1} = a_i) \times P(A = a_k|A_{S2} = a_j) \times P(A_{S1} = a_i) \times P(A_{S2} = a_j) / P(A = a_k)$.

Therefore, $P(A = a_k \mid A_{S1} = a_i, A_{S2} = a_j) = P(A = a_k \mid A_{S1} = a_i) \times P(A = a_k|A_{S2} = a_j) \times P(A_{S1} = a_i) \times P(A_{S2} = a_j) / [P(A = a_k) \times P(A_{S1} = a_i, A_{S2} = a_j)]$.

Substituting for $P(A_{S1} = a_i, A_{S2} = a_j)$ from I, we get: $P(A = a_k \mid A_{S1} = a_i, A_{S2} = a_j) = [P(A = a_k|A_{S1} = a_i) \times P(A = a_k|A_{S2} = a_j)/P(A = a_k)] / [\Sigma_m P(A = a_m \mid A_{S1} = a_i) \times P(A = a_m|A_{S2} = a_j) / P(A = a_m)]$.

From our second assumption, $P(A = a_m) = P(A = a_k)$, and $P(A = a_m) = 1/|A|$ for all m.

Therefore, $P(A = a_k|A_{S1} = a_i,A_{S2} = a_j) = [P(A = a_k|A_{S1} = a_i) \times P(A = a_k|A_{S2} = a_j)]/[\Sigma_m P(A = a_m \mid A_{S1} = a_i) \times P(A = a_m|A_{S2} = a_j)]$.

We now show how the desired probability estimates may be obtained for each case.

Case 1a: $k = i = j$; $P(A = a_i \mid A_{S1} = a_i, A_{S2} = a_i)$

$P(A = a_i \mid A_{S1} = a_i, A_{S2} = a_i) = [P(A = a_i|A_{S1} = a_i) \times P(A = a_i|A_{S2} = a_i)] / [\Sigma_m P(A = a_m \mid A_{S1} = a_i) \times P(A = a_m|A_{S2} = a_i)]$.

For $m \neq i$, and assumption three, we have $P(A = a_m \mid A_{S1} = a_i) = P(A \neq a_i \mid A_{S1} = a_i) / [|A|-1] \ \forall m \neq i$.

Similarly, $P(A = a_m \mid A_{S2} = a_i) = P(A \neq a_i \mid A_{S2} = a_i) / [|A|-1] \ \forall m \neq i$.

Therefore, $\Sigma_m P(A = a_m \mid A_{S1} = a_i) \times P(A = a_m|A_{S2} = a_j) = P(A = a_i|A_{S1} = a_i) \times P(A = a_i|A_{S2} = a_i) + \Sigma_{m \neq i} P(A = a_m \mid A_{S1} = a_i) \times P(A = a_m|A_{S2} = a_j) = P(A = a_i|A_{S1} = a_i) \times P(A = a_i|A_{S2} = a_i) + \Sigma_{m \neq i} P(A \neq a_i \mid A_{S1} = a_i) \times P(A \neq a_i \mid A_{S2} = a_i) / [|A|-1]^2 = P(A = a_i|A_{S1} = a_i) \times P(A = a_i|A_{S2} = a_i) + P(A \neq a_i \mid A_{S1} = a_i) \times P(A \neq a_i \mid A_{S2} = a_i) / [|A|-1]$.

Hence, $P(A = a_i \mid A_{S1} = a_i, A_{S2} = a_i) =$

$$\frac{P(A = a_i \mid A_{S1} = a_i) * P(A = a_i \mid A_{S2} = a_i)}{P(A = a_i \mid A_{S1} = a_i) * P(A = a_i \mid A_{S2} = a_i) + P(A \neq a_i \mid A_{S1} = a_i) * P(A \neq a_i \mid A_{S2} = a_i) / [|A| - 1]}$$

Case 1b: $k \neq i = j$; $P(A = a_k \mid A_{S1} = a_i, A_{S2} = a_i)$

$P(A = a_k \mid A_{S1} = a_i, A_{S2} = a_i) = [P(A = a_k|A_{S1} = a_i) \times P(A = a_k|A_{S2} = a_i)] / [\Sigma_m P(A = a_m \mid A_{S1} = a_i) \times P(A = a_m|A_{S2} = a_i)]$.

Since $k \neq i$, from assumption 3, we have $P(A = a_k \mid A_{S1} = a_i) = P(A \neq a_i \mid A_{S1} = a_i)/ [|A|-1]$, and $P(A = a_k \mid A_{S2} = a_i) = P(A \neq a_i \mid A_{S2} = a_i)/ [|A|-1]$.

Therefore, $P(A = a_k|A_{S1} = a_i) \times P(A = a_k|A_{S2} = a_i) = P(A \neq a_i \mid A_{S1} = a_i) \times P(A \neq a_i \mid A_{S2} = a_i) / [|A|-1]^2$.

Hence, $P(A = a_k \mid A_{S1} = a_i, A_{S2} = a_i) =$

$$\frac{P(A \neq a_i \mid A_{S1} = a_i) * P(A \neq a_i \mid A_{S2} = a_i) / [|A| - 1]^2}{P(A = a_i \mid A_{S1} = a_i) * P(A = a_i \mid A_{S2} = a_i) + P(A \neq a_i \mid A_{S1} = a_i) * P(A \neq a_i \mid A_{S2} = a_i) / [|A| - 1]}$$

Case 2a: $k = i \neq j$; $P(A = a_i \mid A_{S1} = a_i, A_{S2} = a_j)$

$P(A = a_i \mid A_{S1} = a_i, A_{S2} = a_j) = [P(A = a_i|A_{S1} = a_i) \times P(A = a_i|A_{S2} = a_j)] / [\Sigma_m P(A = a_m \mid A_{S1} = a_i) \times P(A = a_m|A_{S2} = a_j)]$.

Here, we have, from assumption 3, $P(A = a_i|A_{S2} = a_j) = P(A \neq a_j \mid A_{S2} = a_j)/ [|A|-1]$, and $P(A = a_j|A_{S1} = a_i) = P(A \neq a_i \mid A_{S1} = a_i)/ [|A|-1]$.

Now, $\Sigma_m P(A = a_m \mid A_{S1} = a_i) \times P(A = a_m|A_{S2} = a_j) = P(A = a_i \mid A_{S1} = a_i) \times P(A = a_i|A_{S2} = a_j) + P(A = a_j \mid A_{S1} = a_i) \times P(A = a_j|A_{S2} = a_j) + \Sigma_{m \neq i,j} P(A = a_m \mid A_{S1} = a_i) \times P(A = a_m|A_{S2} = a_j) = P(A = a_i \mid A_{S1} = a_i) \times P(A \neq a_j \mid A_{S2} = a_j)/ [|A|-1] + P(A \neq a_i \mid A_{S1} = a_i)/ [|A|-1] \times P(A = a_j|A_{S2} = a_j) + \Sigma_{m \neq i,j} P(A \neq a_i \mid A_{S1} = a_i) \times P(A \neq a_j \mid A_{S2} = a_j) / [|A|-1]^2 = P(A = a_i \mid A_{S1} = a_i) \times P(A \neq a_j \mid A_{S2} = a_j)/ [|A|-1] + P(A \neq a_i \mid A_{S1} = a_i)/ [|A|-1] \times P(A = a_j|A_{S2} = a_j) + P(A \neq a_i \mid A_{S1} = a_i) \times P(A \neq a_j \mid A_{S2} = a_j) \times [|A|-2] / [|A|-1]^2$.

Hence, $P(A = a_i \mid A_{S1} = a_i, A_{S2} = a_j) =$

$$\frac{P(A = a_i \mid A_{S1} = a_i) * P(A \neq a_j \mid A_{S2} = a_j)}{P(A = a_i \mid A_{S1} = a_i) * P(A \neq a_j \mid A_{S2} = a_j) + P(A \neq a_i \mid A_{S1} = a_i) * P(A = a_j \mid A_{S2} = a_j) + P(A \neq a_i \mid A_{S1} = a_i) * P(A \neq a_j \mid A_{S2} = a_j)[|A| - 2] / [|A| - 1]}.$$

Case 2b: $k \neq i \neq j$; $P(A = a_k \mid A_{S1} = a_i, A_{S2} = a_j)$

$P(A = a_k \mid A_{S1} = a_i, A_{S2} = a_j) = [P(A = a_k|A_{S1} = a_i) \times P(A = a_k|A_{S2} = a_j)] / [\Sigma_m P(A = a_m \mid A_{S1} = a_i) \times P(A = a_m|A_{S2} = a_j)]$.

The numerator is $P(A = a_k|A_{S1} = a_i) \times P(A = a_k|A_{S2} = a_j) = P(A \neq a_i \mid A_{S1} = a_i) \times P(A \neq a_j \mid A_{S2} = a_j) / [|A|-1]^2$

The denominator is as shown in case 2a.

Hence, $P(A = a_i \mid A_{S1} = a_i, A_{S2} = a_j) =$

$$\frac{P(A \neq a_i \mid A_{S1} = a_i) * P(A \neq a_j \mid A_{S2} = a_j) / [|A| - 1]}{P(A = a_i \mid A_{S1} = a_i) * P(A \neq a_j \mid A_{S2} = a_j) + P(A \neq a_i \mid A_{S1} = a_i) * P(A = a_j \mid A_{S2} = a_j) + P(A \neq a_i \mid A_{S1} = a_i) * P(A \neq a_j \mid A_{S2} = a_j)[|A| - 2] / [|A| - 1]}.$$