Information Synthesis in Statistical Databases

WEE-KEONG NG CHINYA V. RAVISHANKAR

Department of Electrical Engineering and Computer Science
The University of Michigan, Ann Arbor, MI 48109–2122
URL: http://www.eecs.umich.edu/~wkn
Email: {wkn,ravi}@eecs.umich.edu.

Abstract

Given a statistical database containing a set of summary tables, this paper examines the complexity of retrieving data from the database in order to satisfy a query. In particular, we consider the case when the query cannot be directly satisfied via a single summary table and requires two or more summary tables. We show that a system of linear equations can be constructed from a set of summary tables whose solution(s) satisfy a query in *varying degrees*. We derive a formula for determining the degree of acceptability of the solution as a function of the characteristics of the summary tables which derive the algebraic system. We also show that selecting the optimal set of summary tables from the database that yields the best solution to the query is NP-complete. These findings offer important insights into the retrievability of information from a statistical database when designing a statistical query processor.

1 Introductions

A statistical database is a collection of summary tables, each holding statistical information about some set of data objects [8]. For instance, a census database holds various statistics about people in the population. Summary tables are often the only reasonable means for disseminating information when legal or policy concerns, physical limitations on storage capacity, or security and confidentiality issues restrict the public availability of the original data. These tables provide summary information adequate for purposes such as high-level census data analyses, economic planning, policy analyses and forecasting, and so on.

Queries to a statistical database generally desire an aggregation of some attributes of data stored in the summary tables. Existing query languages allow queries that aggregate data stored in a single summary table. This is often inadequate when no single summary table can be used to satisfy the query, for instance, when the query desires more attributes describing a statistic than are contained in each table. In this case, a set of summary tables is joined and aggregated together. In addition, the scope of data that is accessible can be widened when data from more than one summary tables can be aggregated.

Extending existing query languages to handle multi-table queries presents new difficulties. Two important issues stand out: (1) the mechanics of performing a multi-table statistical join and (2) the selection of summary tables for joining. The nature of a statistical join is completely different from that of a relational algebraic join, and the output of the join is not always unique, as we shall describe in the later sections. The set of *candidate* tables for the join affects the quality of the query output.

In this paper, we focus our attention on queries that are satisfiable by multiple summary tables only. In particular, we examine the complexity of satisfying such queries; we show that the *quality* of the response to such a query depends on the set of summary tables chosen and that the optimal selection of summary tables that yield the *best* response is NP-complete. These results are important because they highlight additional parameters a statistical query processor must take into account when designing a query processing algorithm to handle multiple-table queries.

There is much ongoing research in statistical databases [1, 4, 5, 6, 7, 8, 9, 10, 11, 12, 14, 16, 17]. Of relevance to our work are [7, 8, 9]. In [9, 8], the author used an *intersection hypergraph* to establish a universal scheme in order to model the relationship among the categories of the candidate tables and provided procedures for testing the evaluability of queries and for evaluating the queries. Our primary concern in this paper is practicality. Neither the intersection graph nor the universal scheme approaches appear to have been adopted for practical implementation.

The organization of this paper is as follows: We begin in Section 2 with formalization of entities that we shall be using throughout the paper. These include statistical databases, summary tables and queries, etc. Section 3 demonstrates the process of satisfying a multi-table query and points out how the output table is not always unique. We then derive an relationship expressing the *quality* of the output table to the set of chosen summary tables. This relationship permits us to express the optimal selection of summary tables as an optimization problem in Section 4. We then prove that the problem is NP-complete. The last section concludes the paper.

2 Definitions and Terminology

Attributes in a statistical relation may be classified into category and summary attributes. For example, in relation R_1 of Figure 1, C_1 and C_2 are category domains, and S_1 is

CIKM '95, Baltimore MD USA
© 1995 ACM 0-89791-812-6/95/11..$3.50

a summary domain. Category attributes are generally descriptive (non-numeric) and have discrete values that are known in advance. They are used in queries as access keys for retrieving tuples. Summary attributes tend to be numeric because they are usually the observed or measured values in some experiment or survey. They are used in the computation of statistics for statistical queries. In order to facilitate our discussions in the paper, we shall describe and formulate a model for a statistical database in this section. We also specify the conditions for a set of summary tables for satisfying a multi-table query.

2.1 Statistical Database

A *statistical database* $\boldsymbol{D} = \{R_1, R_2, \ldots, R_q\}$ is a set of q summary tables, each of which is an instance of a particular summary table schema. Let $\boldsymbol{R} = \{\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_q\}$ be the set of all q *summary table schemas* and $\boldsymbol{C} = \{\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_p\}$ and $\boldsymbol{S} = \{\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_q\}$ be the set of all *category* and *summary* domains for all summary tables in the database respectively. We refer to $\boldsymbol{C}$ and $\boldsymbol{S}$ as the *base category* and *base summary* domains of $\boldsymbol{D}$ respectively. We call any subset of $\boldsymbol{C}$ a *category* because it is a set of attributes describing some data objects.

We make two assumptions concerning the database: (1) that the database is *homogeneous* in the sense that all the category and summary attributes pertain to the same type of object. For instance, $\boldsymbol{D}$ may be a census database containing data about people as the sole object of concern, and (2) that the summary domains are *additive*. That is, there is an additive function defined on them. For instance, most summary data are obtained using the aggregation functions COUNT and SUM. Furthermore, most nonadditive summary variables such as AVERAGE and RATE are defined in terms of or are derived from additive summary variables, and therefore need not be stored explicitly in the database.

2.2 Summary Table Schema

A *summary table schema* $\mathcal{R}_u$ is defined by a set of category domains $\boldsymbol{C}_u \subseteq \boldsymbol{C}$ and summary domain $\mathcal{S}_u \in \boldsymbol{S}$. Let $\boldsymbol{C}_u = \{\mathcal{C}_{k_1}, \mathcal{C}_{k_2}, \ldots, \mathcal{C}_{k_n}\}$ where $\mathcal{C}_{k_i} \in \boldsymbol{C}$ for $1 \leqslant k_i \leqslant p$, $1 \leqslant i \leqslant n$. The schema $\mathcal{R}_u$ is the Cartesian product $\mathcal{C}_{k_1} \times \cdots \times \mathcal{C}_{k_n} \times \mathcal{S}_u$ containing the set of all possible summary tuples with respect to the stated domains. We denote $\mathcal{R}_u$ as $\langle \boldsymbol{C}_u, \mathcal{S}_u \rangle$. Although the size of the space is given by $\|\mathcal{R}_u\| = \prod_{i=1}^{n} |\mathcal{C}_{k_i}| \times |\mathcal{S}_u|$, the maximum number of valid tuples is only $\prod_{i=1}^{n} |\mathcal{C}_{k_i}|$ since each category subtuple in $\mathcal{R}_u$ describes exactly one value of $\mathcal{S}_u$.

2.3 Summary Table

A *summary table* $R_u = \langle \mathcal{C}_u, \mathcal{S}_u \rangle$ from $\mathcal{R}_u$ is a strict subset of summary tuples $R_u \subset \mathcal{R}_u$ where $\mathcal{S}_u \in \boldsymbol{S}$ and $\mathcal{C}_u = \{C_{k_1}, C_{k_2}, \ldots, C_{k_n}\}$ with $C_{k_i} \subseteq \mathcal{C}_{k_i} \in \boldsymbol{C}$ for $1 \leqslant k_i \leqslant p$, $1 \leqslant i \leqslant n$. The *order* of a summary table R_u is the number of category domains in $\mathcal{C}_u$, i.e., $O(R_u) = n$.

If the category subspace of R_u, given by the projection $\pi_{\mathcal{C}_u}(R_u)$, contains all elements of the Cartesian product of the category domains in $\mathcal{C}_u$, we say that R_u is *complete*. If $\|R_u\|$ denotes the number of summary tuples in R_u, then $\|R_u\| = \prod_{i=1}^{n} |C_{k_i}|$.

Example 1: We define a fictitious census database $\boldsymbol{D}$ that will be used for illustration throughout the paper. Let $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$, $\mathcal{C}_4$, $\mathcal{C}_5$ denote the age group, race, marital status, educational level and sex category domains and $\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4$ denote the cardinality (in thousands), income (in thousands), hours worked per week and miles traveled per week summary domains in the database respectively. Let the category domains be exhaustively defined as: $\mathcal{C}_1 = \{$10s, 20s, 30s, 40s, 50s, 60s$\}$, $\mathcal{C}_2 = \{$white, black, hispanic, asian$\}$, $\mathcal{C}_3 = \{$single, married, widowed, divorced, separated$\}$, $\mathcal{C}_4 = \{$diploma, college, masters, Ph.D.$\}$, and $\mathcal{C}_5 = \{$male, female$\}$. The summary domains are subsets of integers or real numbers: $\mathcal{S}_1 \subseteq \mathbb{Z}$, $\mathcal{S}_2 \subseteq \mathbb{R}$, $\mathcal{S}_3 \subseteq \mathbb{R}$, and $\mathcal{S}_4 \subseteq \mathbb{R}$.

Let $\mathcal{R}_1 = \langle \{\mathcal{C}_1, \mathcal{C}_2\}, \mathcal{S}_1 \rangle$, $\mathcal{R}_2 = \langle \{\mathcal{C}_2, \mathcal{C}_4\}, \mathcal{S}_2 \rangle$, $\mathcal{R}_3 = \langle \{\mathcal{C}_2, \mathcal{C}_4\}, \mathcal{S}_3 \rangle$ and $\mathcal{R}_4 = \langle \{\mathcal{C}_3, \mathcal{C}_4, \mathcal{C}_5\}, \mathcal{S}_4 \rangle$ be four summary table schemas. Then the census database shown in Figure 1 can be described as $\boldsymbol{D} = \{R_1, R_2, R_3, R_4\}$, where $R_1 = \langle \{C_1, C_2\}, \mathcal{S}_1 \rangle \subset \mathcal{R}_1$, $R_2 = \langle \{C_2, C_4\}, \mathcal{S}_2 \rangle \subset \mathcal{R}_2$, $R_3 = \langle \{C_2, C_4\}, \mathcal{S}_3 \rangle \subset \mathcal{R}_3$ and $R_4 = \langle \{C_3, C_4, C_5\}, \mathcal{S}_4 \rangle \subset \mathcal{R}_4$. The orders of R_1, R_2, R_3, R_4 are $2, 2, 2, 3$ respectively. ∎

2.4 Statistical Query

A *statistical query* is defined by a 4-tuple $\mathcal{Q} = \langle \mathcal{A}, \psi, \mathcal{S}, \varphi \rangle$ comprising a category $\mathcal{A} = \{C_{g_1}, C_{g_2}, \ldots, C_{g_m}\}$ with $C_{g_i} \subseteq \mathcal{C}_{g_i} \in \boldsymbol{C}$ for $1 \leqslant g_i \leqslant p$, $1 \leqslant i \leqslant m$, a *well-formed propositional formula* ψ on the category domains, a set of summary domains $\mathcal{S} = \{\mathcal{S}_{e_1}, \mathcal{S}_{e_2}, \ldots, \mathcal{S}_{e_\ell}\}$ with $\mathcal{S}_{e_j} \in \boldsymbol{S}$ for $1 \leqslant e_j \leqslant q$, $1 \leqslant j \leqslant \ell$, and a function $\varphi : \mathcal{S} \rightarrow \mathbb{R}$ on the summary domains. This definition captures the following description of a query: A statistical query desires some statistics $\mathcal{S}_\varphi$ that can be computed by applying a function φ on a set $\mathcal{S}$ of base summary domains. $\mathcal{S}$ is described by a category $\mathcal{A}$ that is in turn qualified by a well-formed propositional formula ψ. In relational algebraic terminology, $\mathcal{A}$ is the set of category attributes involved in the query selection criteria ψ. The output to $\mathcal{Q}$ is a $(m+1)$-column summary table comprising the m category columns in $\mathcal{A}$ and a statistical column containing values from the output of φ. We label this domain as $\mathcal{S}_\varphi$. The generality of the definition is illustrated below:

Example 2: Using the database defined in Example 1, let $\mathcal{Q}_1$ be a statistical query that desires a tabulation of total income for all males with masters or Ph.D. qualifications from a population relation. In conventional SQL, this query is expressed as:

```
SELECT      SUM(income)
FROM        population
WHERE       sex = male AND
            (education = masters OR education = Ph.D.)
GROUP BY    sex, education
```

We denote the query as $\langle \{C_4, C_5\}, \psi, \{\mathcal{S}_1, \mathcal{S}_2\}, \varphi_1 \rangle$ where $C_4 \subseteq \mathcal{C}_4$, $C_5 \subseteq \mathcal{C}_5$, $\psi \equiv [C_4 =$male $\wedge$ $(C_5 =$masters $\vee$ $C_5 =$Ph.D.$)]$, and $\varphi_1 : \mathcal{S}_1 \times \mathcal{S}_2 \rightarrow \mathbb{R}$ is defined as $\varphi_1(s_1, s_2) = s_2 \times s_1$ for all $s_2 \in \mathcal{S}_2, s_1 \in \mathcal{S}_2$. Notice that function φ_1 realizes the computation of total income. Suppose the query desires the average income instead of the total income. In this case, only $\mathcal{S}_2$ is involved and $\varphi_1 : \mathcal{S}_2 \rightarrow \mathbb{R}$ is now defined as the identity function: $\varphi_1(s_2) = s_2$. ∎

The *order* of a query $\mathcal{Q}$ is the number of category domains in $\mathcal{A}$, i.e., $O(\mathcal{Q}) = m$. The above example is an order-2 query. As in summary tables, we say that $\mathcal{Q}$ is *complete* if its category $\mathcal{A}$ is such that $\|\mathcal{A}\| = \prod_{i=1}^{m} |C_{g_i}|$.

In order to facilitate our discussion of the key ideas in this paper, we shall make the following two assumptions: (1) We assume all summary tables and queries concerned are complete. The assumption of query completeness implies that there are no restrictions on the category attribute values in a

$\mathcal{C}_1$	$\mathcal{C}_2$	$\mathcal{S}_1$
20s	white	7
20s	black	7
30s	white	5
30s	black	8

R_1

$\mathcal{C}_2$	$\mathcal{C}_4$	$\mathcal{S}_2$
white	diploma	40
white	college	50
white	masters	70
white	Ph.D.	80
black	diploma	30
black	college	40
black	masters	50
black	Ph.D.	60
hispanic	diploma	30
hispanic	college	40
hispanic	masters	50
hispanic	Ph.D.	60

R_2

$\mathcal{C}_2$	$\mathcal{C}_4$	$\mathcal{S}_3$
white	diploma	40
white	college	40
white	masters	50
white	Ph.D.	60
black	diploma	70
black	college	60
black	masters	50
black	Ph.D.	50
hispanic	diploma	60
hispanic	college	60
hispanic	masters	50
hispanic	Ph.D.	40

R_3

$\mathcal{C}_3$	$\mathcal{C}_4$	$\mathcal{C}_5$	$\mathcal{S}_4$
single	diploma	male	100
single	diploma	female	120
single	college	male	85
single	college	female	94
single	masters	male	72
single	masters	female	79
single	Ph.D.	male	60
single	Ph.D.	female	64
married	diploma	male	98
married	diploma	female	93
married	college	male	70
married	college	female	80
married	masters	male	64
married	masters	female	68
married	Ph.D.	male	52
married	Ph.D.	female	58

R_4

Figure 1: A census database consisting of four summary tables, R_1, R_2, R_3 and R_4.

query's selection criteria, i.e., ψ is *null*. (2) We assume that only one summary domain is involved in the query. Thus, function φ is an *identity* function, i.e., $\varphi(x) = x$. With this assumption, a query is denoted as $\mathcal{Q} = \langle \mathcal{A}, \psi, \mathcal{S}_v, \varphi \rangle$ where $\mathcal{S}_v$ is a single summary domain rather than a set of summary domains as in the query definition above.

2.5 Candidate Tables

A prerequisite condition for a multi-table query $\mathcal{Q}$ to be satisfiable is that there be *candidate* tables in the database.

Definition 1 (Table Candidacy): *Given a complete query $\mathcal{Q} = \langle \mathcal{A}, \psi, \mathcal{S}_v, \varphi \rangle$, any collection of $w > 1$ complete summary tables $R_1 = \langle \mathcal{C}_1, \mathcal{S}_1 \rangle$, $R_2 = \langle \mathcal{C}_2, \mathcal{S}_2 \rangle$, ..., $R_w = \langle \mathcal{C}_w, \mathcal{S}_w \rangle$ is a set of candidate tables if and only if $\mathcal{S}_v = \mathcal{S}_i$ for all $1 \leqslant i \leqslant w$ and $\mathcal{A} = \bigcup_{i=1}^{w} \mathcal{C}_i$.* ∎

The first condition must be true, for otherwise the candidate tables are irrelevant to the query. The second condition states that the union of the table's categories must be equal to the set of category domains desired by $\mathcal{Q}$ otherwise the tables are irrelevant. The categories in candidate tables are assumed to cover $\mathcal{A}$ exactly for the following reason. If a candidate table contains attributes not in $\mathcal{A}$, these extraneous attributes are first eliminated by aggregating across all their values. Since the tables are complete, such aggregation is equivalent to creating new summary tables that contain only attributes in $\mathcal{A}$.

3 Multi-Table Query Processing

From the foregoing sections, we now describe the process of processing a multi-table query. Specifically, we demonstrate how the candidate tables are *statistically joined* by constructing a system of linear equations to compute the desired statistics. The reader will note that this process is entirely orthogonal to a relational algebraic join. In addition, we also show that this process sometimes results in nonunique outputs to a query. We present a formulation of this process and derive a measure for the degree of goodness of a set of candidate tables for producing the output to a query.

3.1 Statistical Query Join

Let $\mathcal{Q} = \langle \mathcal{A}, \psi, \mathcal{S}_v, \varphi \rangle$ with category $\mathcal{A} = \{\mathcal{C}_{g_1}, \mathcal{C}_{g_2}, \ldots, \mathcal{C}_{g_n}\}$ be a complete order-n query that is not directly satisfiable. Let there be $w > 1$ complete candidate tables $R_1 = \langle \mathcal{C}_1, \mathcal{S}_u \rangle$, $R_2 = \langle \mathcal{C}_2, \mathcal{S}_u \rangle$, ..., $R_w = \langle \mathcal{C}_w, \mathcal{S}_u \rangle$ such that $\mathcal{S}_v = \mathcal{S}_u$ and $\mathcal{A} = \bigcup_{i=1}^{w} \mathcal{C}_i$.

The output summary table T for $\mathcal{Q}$ is *defined* except for the $\mathcal{S}_v$ domain: Each tuple in T is of the form $\langle c_{g_1}, c_{g_2}, \ldots, c_{g_n}, x \rangle$ where $c_{g_i} \in \mathcal{C}_{g_i}$, $1 \leqslant g_i \leqslant n$, is known and $x \in \mathcal{S}_v$ is the desired (unknown) summary attribute value. Since $\mathcal{Q}$ is complete, there are $h = \|\mathcal{A}\| = \prod_{i=1}^{n} |\mathcal{C}_{g_i}|$ unknown summary attribute values, denoted by $X = \{x_1, x_2, \ldots, x_h\}$.

If a system of h or more linear equations can be constructed with x_i's as the unknown variables, then $\mathcal{Q}$ may be satisfiable. In the following, we show that an equation may be derived from each tuple of the candidate tables. As there are $\|\mathcal{C}_i\|$ summary tuples from the R_i's, the total number of derived equations is $e = \|\mathcal{C}_1\| + \|\mathcal{C}_2\| + \cdots + \|\mathcal{C}_w\|$.

We consider how an equation is derived from tuple t of summary table R_i, $1 \leqslant i \leqslant w$, involving some *subset* of X. Let $T_i(t) = \{y \in T : \pi_{\mathcal{C}_i}(y) = \pi_{\mathcal{C}_i}(t)\}$ be the set of summary tuples from T whose $\mathcal{C}_i$ attribute values are identical to those of t. Since $\mathcal{A}$ is complete, the size of $T_i(t)$ is the size of the category space $\mathcal{A} - \mathcal{C}_i$, i.e., $b = \prod_{\mathcal{C} \in \{\mathcal{A} - \mathcal{C}_i\}} |\mathcal{C}|$. Let $T_i'(t) = \{\pi_{\mathcal{S}_v}(y) : y \in T_i(t)\}$ be the set of summary attribute values of tuples in $T_i(t)$. Let us re-write $T_i'(t)$ as $\{x_{a_1}, x_{a_2}, \ldots, x_{a_b}\}$ where $x_{a_j} \in X$. Since the summary values are additive, one may construct the following equation: $x_{a_1} + x_{a_2} + \cdots + x_{a_b} = \pi_{\mathcal{S}_v}(t)$. By repeating the procedure for each tuple of the candidate tables, we derive a system of linear equations $\boldsymbol{A}\boldsymbol{X}^t = \boldsymbol{B}^t$ where $\boldsymbol{A}$ is the $(e \times h)$ coefficient matrix, $\boldsymbol{X} = [x_1\ x_2\ \cdots\ x_h]$ is the $(1 \times h)$ matrix containing the unknown summary attribute values, and $\boldsymbol{B}$ is the $(1 \times e)$ matrix containing the constants on the RHS of the system. We say this system of linear equations is *induced* by query $\mathcal{Q}$.

It is a well known result in linear algebra that the system of linear equations $\boldsymbol{A}\boldsymbol{X}^t = \boldsymbol{B}^t$ is solvable if and only if the *rank* of $\boldsymbol{A}$ equals the rank of the augmented matrix $[\boldsymbol{A}|\boldsymbol{B}^t]$ (See Appendix A). Since the solution of $\boldsymbol{X}$ completes the output summary table T, the solvability of the induced system implies the solvability of $\mathcal{Q}$. Furthermore, if the rank of $\boldsymbol{A}$ is h, then the system has a unique solution and $\mathcal{Q}$ is satisfiable. We express this result formally as the following

C_1	C_2	S_1
20s	white	7
20s	black	7
30s	white	5
30s	black	8

R_1

C_2	C_3	S_1
white	married	6
white	widowed	6
black	married	5
black	widowed	10

R_2

C_1	C_2	C_3	S_1
20s	white	married	x_1
20s	white	widowed	x_2
20s	black	married	x_3
20s	black	widowed	x_4
30s	white	married	x_5
30s	white	widowed	x_6
30s	black	married	x_7
30s	black	widowed	x_8

T_2

Figure 2: Evaluating query $\mathcal{Q}_2 = \langle \mathcal{A}, \psi, \mathcal{S}_v, \varphi_2 \rangle$ where $\mathcal{A}$ is equal to the union of category domains of the 2 candidate tables R_1, R_2. T_2 is the output summary table for $\mathcal{Q}_2$.

theorem and provide an example for illustration:

Theorem 1: *Given a complete order-n query $\mathcal{Q} = \langle \mathcal{A}, \psi, \mathcal{S}_v, \varphi \rangle$, if there exist $w > 1$ complete order-m candidate tables $R_i = \langle \mathcal{C}_i, \mathcal{S}_v \rangle$, $1 \leqslant i \leqslant w$, then $\mathcal{Q}$ is*

(i) *solvable if and only if its induced system of linear equations is solvable.*

(ii) *uniquely satisfiable if and only if the induced system is solvable and the rank of the coefficient matrix is equal to $\|\mathcal{A}\|$.*

(ii) *non-uniquely satisfiable if and only if the induced system is solvable and not uniquely satisfiable.*

Example 3: Suppose a demographer wishes to cross tabulate the number of people with respect to age group, race and marital status. We may formulate the query as: $\mathcal{Q}_2 = \langle \{C_1, C_2, C_3\}, \psi, \mathcal{S}_1, \varphi_2 \rangle$. Let the candidate tables for $\mathcal{Q}_2$ be $R_1 = \langle \{C_1, C_2\}, \mathcal{S}_1 \rangle$ and $R_2 = \langle \{C_2, C_3\}, \mathcal{S}_1 \rangle$ as shown in Figure 2, where the category domains are defined as C_1= {20s, 30s}, C_2 = {white, black}, C_3 = {married, widowed}, and $\mathcal{S}_1$ represents frequency. The size of the category space $\mathcal{A}$ is 8. Observe that T_2, the output summary table for $\mathcal{Q}_2$ is defined, except for the unknown summary values $X = \{x_1, x_2, \ldots, x_8\}$. There are 8 summary tuples in all in the candidate tables from which we may form the following system of 8 linear equations in 8 unknowns:

$$\begin{array}{llll} x_1 + x_2 = 7 & x_3 + x_4 = 7 & x_5 + x_6 = 5 & x_7 + x_8 = 8 \\ x_1 + x_5 = 6 & x_2 + x_6 = 6 & x_3 + x_7 = 5 & x_4 + x_8 = 10 \end{array}$$

The first equation $x_1 + x_2 = 7$ comes from the first tuple of R_1, $\langle$20s, white, 7$\rangle$. From T_2, the first two tuples $\langle$20s, white, married, $x_1\rangle$ and $\langle$20s, white, widowed, $x_2\rangle$ have the same C_1, C_2 values. The equation is obtained by adding the unknown variables of the two tuples and eliminating C_3. The other equations are similarly obtained. The system of linear equations thus derived is:

$$\boldsymbol{A}X^t = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} 7 \\ 7 \\ 5 \\ 8 \\ 6 \\ 6 \\ 5 \\ 10 \end{bmatrix} = \boldsymbol{B}^t$$

which in row echelon form is:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix} = \begin{bmatrix} 1 \\ 6 \\ -3 \\ 10 \\ 5 \\ 8 \\ 0 \\ 0 \end{bmatrix}$$

Thus, rank[$\boldsymbol{A}$]=rank[$\boldsymbol{A}|\boldsymbol{B}^t$] and the system of linear equations is solvable. However, since rank[$\boldsymbol{A}$] = $6 < 8$, there is no unique solution. ▮

This example illustrates a case where the system of linear equations does not yield a unique solution. Taking x_6, x_8 as the free variables, the general solution is given by: $\boldsymbol{X} = [1 + x_6, 6 - x_6, x_8 - 3, 10 - x_8, 5 - x_6, x_6, 8 - x_8, x_8]$.

From table T_2 (Figure 2), x_6 is described by attributes 30s, white, widowed. Assuming that each of these attributes is a set, x_6 = |30s| ∩ |white| ∩ |widowed|. Moreover, $x_6 <$ |30s| ∩ |white| = 5 and $x_6 <$ |white| ∩ |widowed| = 6 from tuples 3 and 2 of tables R_1 and R_2 respectively. Thus, we have $0 \leqslant x_6 \leqslant 5$ assuming $x \geqslant 0$. Likewise, we have $0 \leqslant x_8 \leqslant 8$ for the free variable x_8. Therefore, there are 5 × 8 possible instantiations of the above general solution.

In the absence of other information, the user will have to make a choice among a subset (possibly one) of the set of possible solutions. Thus, the smaller the size of the solution set, the fewer the choices a user must make. As the number of free variables is equal to the total number of variables less the rank of the augmented matrix, one would prefer a set of candidate tables that yield fewer (preferably no) free variables. That is, we can make use of the rank of the augmented matrix as a measure of the degree of goodness of a set of candidate tables. For this measure to be practical, we need to be able to compute the rank given any arbitrary set of tables without solving the system of linear equations it induces.

3.2 A Goodness Measure for Candidate Tables

In this section, we derive an expression relating the rank of the augmented matrix to the set of candidate tables inducing the algebraic system. Specifically, we show that the ranks of both the coefficient matrix $\boldsymbol{A}$ and the augmented matrix $[\boldsymbol{A}|\boldsymbol{B}^t]$ are bounded above by $e - w + 1$, where e is the number of linear equations and w is the number of candidate tables. Note that e is a function of the number of tuples in each candidate table. In order to derive this inequality, we first investigate some properties of the coefficient and augmented matrices induced by the candidate tables.

Let summary table R_i contribute $e_i = \|\mathcal{C}_i\|$ rows to $\boldsymbol{A}$. We may *partition* $\boldsymbol{A}$ into w submatrices $\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_w$ corresponding to the w summary tables, so that $\boldsymbol{A}_i$, $1 \leqslant i \leqslant w$, consists of rows derived from R_i only. We say that summary table R_i *induces* the submatrix $\boldsymbol{A}_i$ of size ($e_i \times h$). Let $\boldsymbol{A}_i(j,k)$ be the k-th element in the j-th row in $\boldsymbol{A}_i$.

Let $X_{i,j}$ be the set of summary variables in the equation derived from the j-th tuple of R_i, $1 \leqslant j \leqslant e_i$. We show that $X_{i,j}$ partition X.

Lemma 1: *For $1 \leqslant i \leqslant w$, $1 \leqslant j, k \leqslant e_i$, $j \neq k$,*

(i) $|X_{i,j}| > 0$,

(ii) $|X_{i,j}| = |X_{i,k}|$,

(iii) $X_{i,j} \cap X_{i,k} = \varnothing$, *and*

(iv) $\bigcup_{i,j} X_{i,j} = X$.

Proof: (i) By the definition of $T_i(t_j)$, $y \in T_i(t_j)$ if $\pi_{\mathcal{C}_i}(y) = \pi_{\mathcal{C}_i}(t_j)$ for $t_j \in R_i$, $y \in T$. Thus, $|T_i(t_j)|$ is just the number ($n-m$)-ary tuples in $\mathcal{A} - \mathcal{C}_i$, i.e., $|T_i(t_j)| = \prod_{C \in \{\mathcal{A} - \mathcal{C}_i\}} |C|$ for $1 \leqslant j \leqslant e_i$, $1 \leqslant i \leqslant w$. By Condition (ii) of Definition 1, $|\mathcal{C}_i| < |\mathcal{A}|$, $1 \leqslant i \leqslant w$, since $w > 1$, i.e., $\mathcal{A} - \mathcal{C}_i \neq \varnothing$. Therefore, $|T_i(t_j)| > 0$. Since each y introduces one unknown variable

$\pi_{\mathcal{S}_v}(y) \in X$, the number of unknown variables in the equation derived from t_j is $|X_{i,j}| = |T_i(t_j)| > 0$.

(ii) Since all tuples of R_i have the same category domains, $|T_i(t_j)| = |T_i(t_k)|$ for $1 \leqslant j, k \leqslant e_i$, $j \neq k$. Thus, $|X_{i,j}| = |T_i(t_j)| = |T_i(t_k)| = |X_{i,k}|$.

(iii) Given any $t_j \in R_i$, $t_k \in R_k$ and for all $y_j \in T_i(t_j)$, $y_k \in T_i(t_k)$, $\pi_{C_i}(y_j) \neq \pi_{C_i}(y_k)$. Thus, $T_i(t_j) \cap T_i(t_k) = \varnothing$. Since each $y \in T_i(t_j)$ introduces an unknown variable in the equation derived from t_j, $X_{i,j} \cap X_{i,k} = \varnothing$ for $1 \leqslant j, k \leqslant e_i$, $j \neq k$.

(iv) Since R_i is complete, the number of tuples in R_i is $\prod_{C \in \mathcal{C}_i} |C|$, i.e., all combinations of attribute values from the category domains in $\mathcal{C}_i$ appear in the category subtuples in R_i. Since $|T_i(t_j)| = \prod_{C \in \{\mathcal{A} - \mathcal{C}_i\}} |C|$ for $1 \leqslant j \leqslant e_i$, the total number of unknown variables in all the equations derived from R_i is $\prod_{C \in \mathcal{A}} |C| = |T|$, since T is complete. Together with the preceding three results, we conclude that $\bigcup_{\substack{1 \leqslant i \leqslant w \\ 1 \leqslant j \leqslant e_i}} X_{i,j} = X$. ∎

We next show that the *vector sum* of all the rows in any induced submatrix is equal to the $(1 \times h)$ vector $[1\ 1\ \cdots\ 1]$. Consider the coefficient matrix in Example 3, which consists of two induced submatrices:

$$A_1 = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} \quad A_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The vector sum of the 4 rows in any of the two submatrices is equal to $[1\ 1\ 1\ 1\ 1\ 1\ 1\ 1]$. We shall state the fact formally as the following corollary:

Corollary 1: *For* $1 \leqslant i \leqslant w$, $1 \leqslant k \leqslant h$, $\sum_{j=1}^{e_i} A_i(j,k) = 1$

Proof: The mapping from a system of linear equations to the matrices A and B^t follows the rule:

$$A_i(j,k) \quad = \quad \begin{cases} 1 & \text{if } x_k \in X_{i,j} \\ 0 & \text{otherwise} \end{cases} \qquad (1)$$

$$[A|B^t]_i(j, h+1) \quad = \quad \pi_{\mathcal{S}_v}(t_j \in R_i) \qquad (2)$$

with $1 \leqslant i \leqslant w$, $1 \leqslant j \leqslant e_i$ and $1 \leqslant k \leqslant h$. This coupled with Lemma 1 proves the corollary. ∎

Finally, we show that the sum of the elements in the last column of any induced submatrix of the augmented matrix is equal to the sum of all the unknown summary values of the output table. Consider submatrix $[A|B^t]_1$ in Example 3, which is equivalent to the following system of linear equations: $x_1 + x_2 = 7$, $x_3 + x_4 = 7$, $x_5 + x_6 = 5$, $x_7 + x_8 = 8$. Observe that the RHS's form the last column of $[A|B^t]_1$. Since the LHS's contain the entire set of unknown summary values of T, the last column sum of $[A|B^t]_1$ is equal to $\sum_{i=1}^{8} x_i$. We shall state the fact formally as the lemma below:

Lemma 2: *For* $1 \leqslant i, k \leqslant w$,

$$\sum_{j=1}^{e_i} [A|B^t]_i(j, h+1) = \sum_{t \in T} \pi_{\mathcal{S}_v}(t)$$

Proof: From the proof of Lemma 1, we noted that a summary tuple $t_j \in R_i$ introduces a set of unknown variables $X_{i,j}$. The sum of $\pi_{\mathcal{S}_v}(t_j)$ for all $t_j \in R_i$ yields $\sum_{j=1}^{e_i} [A|B^t]_i(j, h+1)$. By property (iii) of Lemma 1, the sum of all unknown variables from all $X_{i,j}$'s is equal to $\sum_{x \in X} x = \sum_{t \in T} \pi_{\mathcal{S}_v}(t)$. Hence, the lemma follows. ∎

As the last column sum of any induced submatrix is equal to the sum of all the unknown summary values of the output table, the sums of the elements in the last columns of any two induced submatrices of the augmented matrix are the same. That is,

Corollary 2: *For* $1 \leqslant i, k \leqslant w$,

$$\sum_{j=1}^{e_i} [A|B^t]_i(j, h+1) = \sum_{j=1}^{e_k} [A|B^t]_k(j, h+1)$$

We can now show that the ranks of both coefficient matrix A and the augmented matrix $[A|B^t]$ are no more than $e - w + 1$, where e is the number of linear equations and w is the number of candidate tables. That is,

Theorem 2: $\text{rank}[A] \leqslant e - w + 1$, $\text{rank}[A|B^t] \leqslant e - w + 1$

Proof: From Corollary 1, each induced submatrix A_i may be row-reduced so that it contains one row of 1's. Thus, since A contains w induced submatrices, it may be row reduced to a matrix that exhibits w rows of all 1's using $\sum_{i=1}^{w} e_i - 1$ row operations. Clearly, $w - 1$ of these rows of 1's can be reduced to rows of 0's by $w - 1$ row operations. Thus, $\text{rank}[A]$ is no more than $e - w + 1$. From Corollary 2, $[A|B^t]$ may likewise be reduced to a matrix whose rank is no more than $e - w + 1$. ∎

We conclude this section with an example to illustrate the foregoing concepts.

Example 4: Suppose a demographer would like to know the distribution of the number of whites and blacks who are in their 20's or 30's and who are either single or married. This query may be formulated as $\mathcal{Q}_1 = \langle \mathcal{A}, \psi, \mathcal{S}_1, \varphi \rangle$ where $\mathcal{A} = \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3, \mathscr{C}_5\}$ and $\mathcal{C}_1 = \{\text{20s}, \text{30s}\} \subset \mathscr{C}_1$, $\mathcal{C}_2 = \{\text{white}, \text{black}\} \subset \mathscr{C}_2$, $\mathcal{C}_3 = \{\text{single}, \text{married}\} \subset \mathscr{C}_3$ and $\mathscr{C}_5 = \{\text{male}, \text{female}\}$. The number of unknown variables in T_1 is $\|\mathcal{A}\| = 2 \times 2 \times 2 \times 2 = 16$.

Suppose two sets of candidate tables $E_1 = \{\mathcal{R}_1, \mathcal{R}_2, \mathcal{R}_3\}$, $E_2 = \{\mathcal{R}_4, \mathcal{R}_5\}$ are available with $\mathcal{R}_1 = \langle \{\mathcal{C}_2, \mathcal{C}_3\}, \mathcal{S}_1 \rangle$, $\mathcal{R}_2 = \langle \{\mathcal{C}_1, \mathcal{C}_3\}, \mathcal{S}_1 \rangle$, $\mathcal{R}_3 = \langle \{\mathcal{C}_3, \mathscr{C}_5\}, \mathcal{S}_1 \rangle$, $\mathcal{R}_4 = \langle \{\mathcal{C}_1, \mathcal{C}_2, \mathscr{C}_5\}, \mathcal{S}_1 \rangle$ and $\mathcal{R}_5 = \langle \{\mathcal{C}_1, \mathcal{C}_2, \mathcal{C}_3\}, \mathcal{S}_1 \rangle$. The sizes of the algebraic systems induced by E_1, E_2 are: $\sum_{i=1}^{3} |\mathcal{R}_i| = (2 \times 2) + (2 \times 2) + (2 \times 2) = 12$ and $\sum_{i=4}^{5} |\mathcal{R}_i| = (2 \times 2 \times 2) + (2 \times 2 \times 2) = 16$ respectively. By Theorem 2, the ranks of the augmented matrices in the algebraic systems induced by E_1, E_2 are bounded by $12 - 3 + 1 = 10$ and $16 - 2 + 1 = 15$ respectively. Thus, E_1 results in more free variables than E_2 making it less attractive as the set of candidate tables for satisfying $\mathcal{Q}_1$. ∎

4 Optimal Selection of Candidate Tables

Given the rank of the augmented matrix as a measure of the goodness of a set of candidate tables, we formulate the selection of an *optimal* set of candidate tables as an optimization problem and prove that it is NP-complete.

4.1 Problem Formulation

From Theorem 2, $\sum_{R \in M} |R| - |M| + 1$ is an upper bound on the ranks of the coefficient and augmented matrices. Since $\|\mathcal{A}\|$ is the number of unknown variables, $\|\mathcal{A}\| - \sum_{R \in M} |R| + |M| - 1$ is a lower bound on the number of free variables. Therefore, we can define the Candidate Table Selection problem (CTS) as:

Definition 2: *Given a complete order-n query* $Q = \{\mathcal{A}, \psi, \mathcal{S}_j, \varphi\}$ *and a database* $D = \{R_1, R_2, \ldots, R_q\}$ *of complete summary tables, find an optimal subset* $M \subseteq D$ *of candidate tables such that* $\|\mathcal{A}\| - \sum_{R \in M} |R| + |M| - 1 \geqslant 0$ *is minimized.* ▮

This definition can be simplified as follows: We note that the summary tables in M must satisfy the two candidacy criteria in Definition 1. However, restricting the database to summary tables whose summary domains are all identical to that desired by the query will not make the problem harder. Thus, we can omit the first condition. In addition, by condition (ii) of Definition 1, we can discard those R_i's that have category domains that are not in $\mathcal{A}$.

Next, it is possible to abstract Definition 2 to contain only essential parameters. Specifically, only $\mathcal{A}$ in the query definition and the categories in the R_i's are essential to the definition. In addition, as the number of tuples in each table is required in the expression $\|\mathcal{A}\| - \sum_{R \in M} |R| + |M| - 1$, we may incorporate a weight function μ' for each category domain C in $\mathcal{C}$, where $\mathcal{C}$ is the union of all the categories in all the summary tables in D, so that the expression is expressed as a function of $\mu'(R)$. That is, $\|\mathcal{A}\| - \sum_{R \in M} |R| + |M| - 1 = \|\mathcal{A}\| - 1 - \left(\sum_{R \in M} \prod_{C \in \mathcal{C}_R} \mu'(C) - |M|\right) = \|\mathcal{A}\| - 1 - \sum_{R \in M} \prod_{C \in \mathcal{C}_R} \mu(C)$ where $\mathcal{C}_R$ is the category of R and $\mu(R) = \mu'(R) - 1$. Generally, the weight function counts the size of the category domain. With these simplifications, we derive the following definition of CTS:

Definition 3 (CTS): *Given a finite set* $\mathcal{A}$, *a collection* $\mathcal{D} = \{R_1, R_2, \ldots, R_q\}$ *of subsets of* $\mathcal{A}$, *a positive integer weight* $\mu(C)$ *for each element of* $\mathcal{A}$, *and a positive integer* λ, *does there exist a subset* $M \subseteq \mathcal{D}$ *such that* $\mathcal{A} = \bigcup_{R \in M} R$ *and* $|\mathcal{A}| > \sum_{R \in M} \prod_{C \in R} \mu(C) \geqslant \lambda$*?* ▮

Note that $\mathcal{A}$ is now simply a set of elements (category domains) and R is a subset of $\mathcal{A}$ because the essential constituents of the problem are the category domains. This definition expresses CTS as an optimization problem.

4.2 NP-Completeness Proof

In order to show that CTS is NP-complete, we first define a restriction CTS′ of CTS and show that it is NP-complete. We then reduce CTS′ to CTS so that CTS is also NP-complete. The restricted problem CTS′ is defined as:

Definition 4 (CTS′): *Given a finite set* $\mathcal{A}$, *a collection* $\mathcal{D} = \{R_1, R_2, \ldots, R_q\}$ *of subsets of* $\mathcal{A}$, *a positive integer weight* $\mu(C)$ *for each element of* $\mathcal{A}$, *and a positive integer* λ, *does there exist a subset* $M \subseteq \mathcal{D}$ *such that* $\mathcal{A} = \bigcup_{R \in M} R$ *and* $\sum_{R \in M} \prod_{C \in R} \mu(C) = \lambda$*?* ▮

In order to prove that CTS′ is NP-complete, we reduce the Minimum Cover (MC) problem to it in the following theorem. Let U be a collection of subsets of a finite set S and K be a positive integer. The MC problem asks if there exist a subset $U' \subseteq U$ with $|U'| \leqslant K$ such that $S = \bigcup_{R \in U'} R$. The MC problem is known to be NP-complete [3]. Note that the condition $|U'| \leqslant K$ can be changed to $|U'| = K$ without changing the complexity of the problem (see Appendix B). This will be the version of MC we use below:

Theorem 3: *CTS′ is NP-complete.*

Proof: It is easy to verify that CTS′ is in NP since a nondeterministic algorithm need only guess a subset M of $\mathcal{D}$ and check in polynomial time that M is a solution for CTS′.

We will transform MC to CTS′. Let an arbitrary instance of MC be defined by U, a collection of subsets of a finite set S, and K, a positive integer. Let the derived instance of CTS′ be defined by $\mathcal{A} = S$, $\mathcal{D} = U$, $\lambda = K$ and $\mu(C) = 1$ for all $C \in \mathcal{A}$. Since all the weights are 1, the product of the elements of any subset R of $\mathcal{A}$ will be 1. Thus, $\sum_{R \in M} \prod_{C \in R} \mu(C)$ will simply be the number of subsets in M.

If the MC instance has a solution $U' \subseteq U$ where $|U'| = K$ and $S = \bigcup_{R \in U'} R$, then U' is also a solution for the CTS′ instance since $U' \subseteq \mathcal{D}$, $\mathcal{A} = \bigcup_{R \in U'} R$ and $\sum_{R \in U'} \prod_{C \in R} \mu(C) = |U'| = K = \lambda$. Likewise, if the CTS′ instance has a solution, it will also work for the MC instance. ▮

We can now reduce CTS′ to CTS and show that CTS is indeed NP-complete:

Theorem 4: *CTS is NP-complete.*

Proof: Clearly, CTS is also in NP. Now consider an arbitrary instance of CTS′ with parameters $\mathcal{A}$, $\mathcal{D}$, μ and λ as defined in Definition 4. We must now construct an instance of CTS with parameters $\mathcal{A}'$, $\mathcal{D}'$, μ' and λ' as defined in Definition 3.

Suppose $\lambda \geqslant |\mathcal{A}| = n$. Then let $\mathcal{A}' = \mathcal{A} \cup \{C_1', C_2', \ldots, C_{\lambda+1-n}'\}$ and $\mathcal{D}' = \{R_1', R_2', \ldots, R_q'\}$ where $R_i' = R_i \cup \{C_1', C_2', \ldots, C_{\lambda+1-n}'\}$, $1 \leqslant i \leqslant q$. Let $\mu'(C) = 1$ for all $C \in \mathcal{A}'$ and $\lambda' = \lambda - n$.

If the CTS′ instance has a solution $M \subseteq \mathcal{D}$, then let $M' = \{R' : R \in M\}$. Since $\mathcal{A}' = \bigcup_{R' \in M'} R'$ and $|\mathcal{A}'| = \lambda + 1 > \sum_{R' \in M'} \prod_{C \in R'} \mu'(C) = |M'| = \lambda \geqslant \lambda - n = \lambda'$, M' is a solution for CTS. Likewise, we can construct a solution $M = \{R : R' \in M'\}$ for the CTS′ instance if CTS has solution M'.

For $\lambda < n$, let $\mathcal{A}' = \mathcal{A}$, $\mathcal{D}' = \mathcal{D}$, $\lambda' = \lambda$ and $\mu'(C) = 1$ for all $C \in \mathcal{A}'$. Clearly M is a solution of the CTS′ instance if and only if M is a solution for the CTS instance since $\mathcal{A}' = \bigcup_{R \in M} R$ and $|\mathcal{A}'| > \sum_{R \in M} \prod_{C \in R} \mu'(C) = |M| = \lambda = \lambda'$. ▮

4.3 Practical Implications

While we have shown that the problem of optimal table selection is difficult in principle, we do not anticipate that it will be too hard to handle in practice. For one thing, the number of query category domains in $\mathcal{A}$ is likely to small, perhaps less than 5 or 10. Thus, an exhaustive enumeration is really quite practical.

In addition, optimal table selection is quite similar to the Minimal Set Covering problem, for which there are good approximation algorithms. For example, it is known [2] that a greedy approximation algorithm for the MC problem has a ratio bound of $(\ln |S| + 1)$ where S is the finite set of elements to be covered (corresponding to $\mathcal{A}$ in CTS). Since $\mathcal{A}$ is likely to be small, this is really a very satisfactory ratio bound.

We therefore anticipate that the CTS problem may be reasonably solved in practice.

5 Conclusions

In this paper, we addressed the issue of information synthesis in statistical databases. We first illustrated the differences between conventional joins and statistical joins and showed that the output of a statistical join may not be unique. We then examined the problem of satisfying a query using several summary tables, and derived a goodness measure for selecting a set of candidate tables for satisfying a query. We also showed that optimal selection of these tables is NP-complete.

We are currently designing a statistical query processing algorithm at the conceptual level for a statistical database holding 1990 census data from the U.S. Census Bureau. This algorithm extends conventional statistical query processors by looking for multiple candidate tables when no single table can be found to satisfy a query. The results in this paper define the general principles governing the design of this algorithm, though we have made some simplifying assumptions such as the completeness of queries and summary tables in our discussion of this paper.

A Augmented Matrix

Given an $m \times n$ matrix $\boldsymbol{A}$ and an $m \times p$ matrix $\boldsymbol{B}$, we may *join* the i-th row of $\boldsymbol{A}$ with the corresponding row of $\boldsymbol{B}$ to form a larger $m \times (n+p)$ matrix denoted as $[\boldsymbol{A}|\boldsymbol{B}]$. We refer to $[\boldsymbol{A}|\boldsymbol{B}]$ as the *augmented matrix*. The following example illustrates the *augmentation*:

Example 5: Given two matrices:

$$\boldsymbol{A} = \begin{bmatrix} 1&1&0&0&0&0&0&0 \\ 0&0&1&1&0&0&0&0 \\ 0&0&0&0&1&1&0&0 \\ 0&0&0&0&0&0&1&1 \end{bmatrix} \quad \boldsymbol{B} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

The augmented matrix of $\boldsymbol{A}$ and $\boldsymbol{B}$ is:

$$[\boldsymbol{A}|\boldsymbol{B}] = \left[\begin{array}{cccccccc|c} 1&1&0&0&0&0&0&0&1 \\ 0&0&1&1&0&0&0&0&2 \\ 0&0&0&0&1&1&0&0&3 \\ 0&0&0&0&0&0&1&1&4 \end{array}\right] = \begin{bmatrix} 1&1&0&0&0&0&0&0&1 \\ 0&0&1&1&0&0&0&0&2 \\ 0&0&0&0&1&1&0&0&3 \\ 0&0&0&0&0&0&1&1&4 \end{bmatrix}$$

∎

B Reformulation of Minimum Cover

Given a collection U of subsets of a finite set S and a positive integer $K \leqslant |U|$, the MC problem asks if there exist a subset $U' \subseteq U$ with $|U'| \leqslant K$ such that $S = \bigcup_{R \in U'} R$. We define a reformulation of MC which has the same parameters but asks if there exist a subset $U' \subseteq U$ with $|U'| = K$ such that $S = \bigcup_{R \in U'} R$. For clarity, we refer to these problems as $\mathrm{MC}_{\leqslant}$ and $\mathrm{MC}_{=}$.

We show that $\mathrm{MC}_{=}$ is NP-complete by reducing $\mathrm{MC}_{\leqslant}$ to $\mathrm{MC}_{=}$. Let an arbitrary instance of $\mathrm{MC}_{\leqslant}$ be defined by the parameters U, S and K as above. We construct an instance of $\mathrm{MC}_{=}$ defined by $U_{=}$, $S_{=}$ and $K_{=}$. Let $U = U_{=}$, $S = S_{=}$ and $K = K_{=}$.

If a solution to $\mathrm{MC}_{\leqslant}$ exists, then we have a covering subset $\bar{U}$ of U with $|\bar{U}| = g$ for some $g \leqslant K$. Since g is less than K (and hence $K_{=}$), we can always add $K_{=} - g$ additional sets from U to the cover $\bar{U}$ to get a covering of size exactly $K_{=}$. Therefore, $\bar{U}$ is also a covering for $\mathrm{MC}_{=}$.

Suppose a solution does not exist to $\mathrm{MC}_{\leqslant}$. Then there is no class of covering subsets $\bar{U}$ such that $|\bar{U}| \leqslant K$. Since $K = K_{=}$, there is no covering of size exactly K either.

Acknowledgements

The authors wish to acknowledge the financial support of the Consortium for International Earth Science Information Networking (CIESIN). CIESIN is a non-profit corporation founded in 1989 to facilitate access to, use and understanding of global change information worldwide. More information about CIESIN can be obtained at URL: "http://www.ciesin.org/".

References

[1] M. C. Chen, L. P. McNamee. On the Data Model and Access Method of Summary Data Management. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 1, No. 4, pp. 519–528, Dec. 1989.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest. *Introduction to Algorithms*. MIT Press, Cambridge, Massachusetts, McGraw-Hill, New York, 1990.

[3] M. R. Garey, D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco, 1979.

[4] S. P. Ghosh. Statistical Relational Tables for Statistical Database Management. *IEEE Transactions on Software Engineering*, Vol. 12, No. 12, pp. 1106–1116, Dec. 1986. Also published as *IBM Research Report RJ4394*.

[5] S. P. Ghosh. Statistical Relational Model. Chapter 10 in *Statistical and Scientific Databases*, Z. Michalewicz (Ed.), Ellis Horwood, New York, 1991.

[6] G. Hebrail. A Model of Summaries for Very Large Database. *Proceedings of the 3rd International Workshop on Statistical Databases*, 1986.

[7] F. M. Malvestuto. The Derivation Problem for Summary Data. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 82–89, June 1988.

[8] F. M. Malvestuto. A Universal-Scheme Approach to Statistical Databases Containing Homogeneous Summary Tables. *ACM Transactions on Database Systems*, Vol. 18, No. 4, pp. 678–708, December 1993.

[9] F. M. Malvestuto, M. Moscarini. Query Evaluability in Statistical Databases. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 2, No. 4, pp. 425–430, December 1990.

[10] Z. Michalewicz. *Statistical and Scientific Databases*, Z. Michalewicz (Ed.), Ellis Horwood, New York, 1991.

[11] W. K. Ng, C. V. Ravishankar. A Physical Storage Model for Efficient Statistical Query Processing. *Proceedings of the 7th International Working Conference on Statistical and Scientific Databases*, pp. 97–106, Charlottesville, Virginia, September 28–30, 1994.

[12] W. K. Ng, C. V. Ravishankar. A Tuple Model for Summary Data Management. *Proceedings of the 6th International Conference on Management of Data*, Bangalore, India, December 19–21, 1994.

[13] W. K. Ng, C. V. Ravishankar. Evaluation Strategies for Statistical Query Processing. *Submitted for publication*, 1995.

[14] G. Ozsoyoglu, Z. M. Ozsoyoglu, F. Mata. A Language and a Physical Organization Technique for Summary Tables. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 3–16, 1985.

[15] H. Sato. Handling Summary Information in a Database: Derivability. *Proceedings of the ACM SIGMOD International Conference on Management of Data*, 1981.

[16] H. Sato. Statistical Data Models: From a Statistical Table to a Conceptual Approach. Chapter 7 in *Statistical and Scientific Databases*, Z. Michalewicz (Ed.), Ellis Horwood, New York, 1991.

[17] A. Shoshani. Statistical Databases: Characteristics, Problems, and Some Solutions. *Proceedings of the 8th International Conference on Very Large Data Bases*, pp. 208–222, September 1982.