Auditing Sum-Queries to Make a Statistical Database Secure

Francesco M. Malvestuto, Mauro Mezzini and Marina Moscarini

"La Sapienza" University of Rome, Italy

Abstract. In response to queries asked to a statistical database, the query system should avoid releasing summary statistics that could lead to the disclosure of confidential individual data. Attacks to the security of a statistical database may be direct or indirect, and in order to repel them, the query system should audit queries by controlling the amount of information released by their responses. The paper focuses on sum-queries with a response variable of nonnegative real type and proposes a compact representation of answered sum-queries, called an information model in "normal form", which allows the query system to decide whether the value of a new sum-query can or cannot be safely answered. If it cannot, then the query system will issue the range of feasible values of the new sum-query consistent with previously answered sum-queries. Both the management of the information model and the answering procedure require solving linear-programming problems and, since standard linear-programming algorithms are not polynomially bounded (despite their good performances in practice), effective procedures that make a parsimonious use of them are stated for the general case. Moreover, in the special case that the information model is "graphical", then it is shown that the answering procedure can be implemented in polynomial time.

1 Introduction

A statistical database (SDB) [1] is an ordinary database that contains information on individuals (persons, households, companies, organisations etc.), but its users are allowed only to ask for summary statistics over groups of individuals, possibly for on-line analytic processing (OLAP) purposes. For example, consider an SDB containing a relation name `Personnel` with scheme {NAME, SSN, GENDER, AGE, SALARY}. The users of the SDB can ask for summary statistics on the attribute SALARY (using aggregate functions such as **sum**, **average**, **max** and **min**) for groups of employees which at the conceptual level are specified by predicates involving the attributes GENDER, AGE and DEPARTMENT but not NAME and SSN which are private attributes. In this paper, we focus on queries such as

Q: select **sum**(SALARY)
from `Personnel`
where GENDER = M

we call *sum-queries* (*). In the previous example, the attribute SALARY is called the *response variable* of the sum-query *Q* and the "where"-clause specifies the *category* of interest. Given an instance *D* of the SDB containing the relation *R* of name `Personnel`, the category specified by the "where"-clause of *Q* determines a subset of *R*, called the *query-set* of *Q*, and the sum of the values of the attribute SALARY over the query-set of *Q* is the *value* of *Q* on *D*. If SALARY is a confidential attribute, then answering *Q* (and, more in general, answering a statistical query whose response variable is a confidential attribute) raises concerns on the compromise of individual privacy since releasing the value of *Q* could lead to the ("exact" or "approximate") disclosure of the value of SALARY for some element of the query-set of *Q* [8, 9, 26, 27]. Such a sum-query, which risks the confidentiality of the response variable and, hence, the security of the SDB, is called *intrusive* and the category *S* specified by its "where"-clause is said to be *sensitive* in *D*. Typically, for a fixed positive integer *k*, *S* is sensitive if the number of tuples in *R* that fall in *S* are less than *k* (exact disclosure) or there are *k* or fewer tuples in *R* that give a dominant contribution to the value of *Q* (approximate disclosure) [8, 9, 26, 27].

(*) Note that a sum-query containing the "group-by" clause such as

select AGE, **sum**(SALARY)
from `Personnel`
where GENDER = M
group by AGE

is equivalent to a set of our sum-queries.

The confidentiality of a response variable σ can be attacked either (in a directed way) by an intrusive sum-query or (in an indirect way) by a non-intrusive sum-query whose value on *D*, combined with the responses to previously-answered sum-queries on *D*, leads to an accurate estimate of the total of σ for some category that is sensitive in *D*. In the latter case, we call the sum-query *tricky*.

In order to make an SDB secure, when a new instance *D* is created, for each confidential attribute σ the sensitive categories in *D* are identified and each of them is assigned a fixed nonnegative number, called its *protection level* [26, 27]. Such a category *S* will be considered *protected* at a certain time if its protection level comes out to be less than the width of the interval, called the *feasibility range* [26, 27], of the feasible values for the total of σ for *S* that are permitted by the responses to previously answered sum-queries. In our proposal, if the current sum-query *Q* is recognized to be intrusive or tricky, then the query system of the SDB will give a non-informative response to *Q* by issuing the feasibility range for *Q* determined by the responses to all previously answered sum-queries with response variable σ. Now, it is easy to decide whether *Q* is or is not intrusive since it is sufficient to check the presence of the category specified by *Q* in the list of the categories that are sensitive in *D* for σ; but, deciding whether *Q* is or is not tricky requires 'auditing' [5, 6, 16, 18, 21] the previously answered sum-queries on *D* with response variable σ and, for each sensitive category *S*, comparing the protection level assigned to *S* with the width of the feasibility range for the total of σ for *S* determined by the value of *Q* and by the responses to the previously answered sum-queries on σ. If each sensitive category is protected, then we say that *Q* can be *safely* answered and the value of *Q* will be issued. A special case occurs when *Q* is *evaluable* from previously answered sum-queries, that is, when the value of *Q* is uniquely determined by them; then, *Q* is neither intrusive nor tricky and it can be safely answered.

The scenario we are considering can be depicted as a competitive game played by the query system, which has as its opponent a hypothetical statistical user, henceforth referred to as the (*data*) *snooper*, who (with no prior information) attempts to pry an accurate estimate of the total of a response variable σ for some sensitive category out of the responses to all answered sum-queries on *D* with response variable σ. This assumption is not unrealistic if the query system of an SDB (also in order to lighten its workload) allows every user to look into the archive of answered sum-queries on *D*.

In most previous work [1, 5, 6, 16, 18, 21], the technique of auditing was applied under the assumption that the snooper also knows the query-set of each answered sum-query. Thus, for each answered sum-query the snooper can write down an equation, whose unknowns represent the unknown values of the response variable for the tuples in its query-set. As a consequence, the size of the snooper's model comes out to be proportional to the size of the instance *D* of the SDB, which may contain a very large number of tuples [1]. On the other hand, the hypothesis that the snooper knows the query-sets of answered sum-queries is not realistic. In order to make the snooper's model independent of the size of the instance of the SDB, some authors [3, 4, 5, 22] have suggested working with categories instead of query-sets. Accordingly, in order to model the information conveyed by a set of answered sum-queries, we shall introduce an equation system

whose unknowns correspond to the classes of a suitable partition of the union of the categories specified by the answered sum-queries. Thus, typically (i.e., with a very large database) the size of our equation system comes out to be far less than the size of the equation system based on query-sets.

To repel the attacks of the snooper, the query system will make use of its own information model, which essentially is the same as the snooper's model and will be constructed incrementally as the value of a new-sum query is issued. Such an information model may suffer from certain drawbacks (e.g., redundancy [3, 4, 5, 19]) and we will give a procedure for getting a "compact" representation of the information model, we call a *normal form*, that the drawbacks are missing from. Finally, using a normal form of the current information model, we shall address the question whether or not a new sum-query can be safely answered.

Answering this question raises some computational problems (recognizing evaluable sum-queries, updating the information model, computing a feasibility range), whose solutions depend on the data type of the response variable. If it is of real type, then standard algebraic methods can be used to solve all of them efficiently; moreover, if sum-queries contain all the "group-by" clause, that is, if they are table queries (or "cuboids" [24, 25]), then there also exist cardinality-based conditions that are sufficient for them to be inference free [24, 25, 28]. If it is of nonnegative type, then we can resort to linear-programming or integer linear-programming methods depending on the specific data type. In general, the case that the response variable is of nonnegative-integer type is extraordinarily difficult from a computational viewpoint and a general theory has yet to be developed [11].

In this paper, we only consider the case that the response variable is of nonnegative real type. Then, a natural approach consists in resorting to standard linear-programming algorithms (e.g., the simplex method [7, 23]). Unfortunately, none of them is polynomial even if they are polynomial on the average and have good performances in practice [7, 23]; on the other hand, existing polynomial linear-programming algorithms (e.g., the ellipsoid method [7, 23]) have bad performances in practice. Therefore, in order to solve the computational problems raised by the security of the SDB, it is convenient to make a parsimonious use of standard linear-programming algorithms and "there is considerable interest in finding alternative techniques" [11]. Accordingly, we will present a cubic evaluability test based on standard algebraic methods, and show that, in the case that the current information model is "graphical", then also the problem of finding a feasibility range can be solved efficiently.

The paper is organized as follows. After introducing basic definitions in Section 2, in Section 3 we present the semantic part and the equation system of the model used by the snooper to capture the information conveyed by the responses to answered sum-queries. Section 4 contains the query system's answering procedure we propose to avoid the disclosure of sensitive categories. Section 5 introduces the normal form of a model which not only provides a compact representation of the information conveyed by the responses to answered sum-queries but also allows to test evaluability in cubic time without resorting to linear-programming methods. Section 6 deals with the updating of the query system's model when a new sum-query is answered. In Section 7 it is

shown that in a graphical model our answering procedure can be implemented efficiently using network algorithms. Finally, Section 8 contains some closing notes and possible directions for future research.

2 Basic Definitions

Suppose we are given an instance D of an SDB which contains a relation R. In this section we consider sum-queries whose response variable, say σ, belongs to the schema of R. Let a be an attribute in the schema of R that is used by such a sum-query. Typically, if the domain of a is large, then the "where"-clause of the sum-query will contain re-coded values in such a way that the size of the re-coded domain of a is made small [26, 27]. For example, the re-coding of the attribute `AGE` may consist of year classes instead of numbers of years, or the re-coding of a geographic attribute with a hierarchical structure (e.g., country, state, county) may consist in 'chopping off' some digits of its values of precision [26, 27]. To avoid confusion, the attributes in R present in the "where"-clause of a sum-query will be referred to as *categorical variables* [26, 27] to take into account that their domains may have been re-coded. For example, it is likely that the schema of a relation R reporting labour data must contain the four attributes `GENDER`, `AGE`, `CONDITION` and `SECTOR`; then, we may assume that the domains of the categorical variables `GENDER`, `AGE`, `CONDITION` and `SECTOR` be {`M`, `F`}, {`young`, `middle`, `old`}, {`employed`, `unemployed`} and {`Agriculture`, `Industry`, `Services`}, respectively. By V_σ we denote the set of all categorical variables corresponding to the attributes in the schema of R that can occur in the "where"-clauses of sum-queries with response variable σ. We assume that the values of each categorical variable in V_σ are mutually exclusive but we don't assume that its values be globally exhaustive. Thus, the values of `SECTOR` are mutually exclusive but not are globally exhaustive because they do not apply to unemployed people. If the values of a categorical variable are not globally exhaustive, we simply add a *null value* to its domain. Finally, let c be a tuple on V_σ where $c(a)$ may be null for some $a \in V_\sigma$; the tuple c can or cannot be meaningful. For example, if `GENDER` and `DIVISION` are the categorical variables in a hospital database, then the tuple c with

$$c(\texttt{GENDER}) = \texttt{M} \qquad \text{and} \qquad c(\texttt{DIVISION}) = \texttt{Gynaecology}$$

is not meaningful since no male patient can be found in the gynaecological division. A meaningful tuple c on V_σ is *maximal* if there is no meaningful tuple c' on V_σ that agree with c on non-null values and contains (strictly) fewer null values. For example, in the previous labour database, the tuple c with

$$c(\texttt{CONDITION}) = \texttt{Employed},\ c(\texttt{GENDER}) = \texttt{F},\ c(\texttt{SECTOR}) = \textit{null}$$

is not maximal and the two tuples c_1 and c_2 with

c_1(CONDITION) = Employed, c_1(GENDER) = F, c_1(SECTOR) = Industry

c_2(CONDITION) = Unemployed, c_2(GENDER) = M, c_2(SECTOR) = *null*

are both maximal. The maximal meaningful tuples on V_σ will be referred to as *cells* and their set is denoted by *dom*(V_σ); moreover, subsets of *dom*(V_σ) will be referred to as *categories*.
Let K be a category. By the *statistic* of σ *over* K we mean the collection (i.e., the multiset) of the values of σ over the (possibly empty) set of tuples in R that fall into the category K, and by the *total* of σ *for* K we mean the sum of the data in the statistic of σ over K. In order to speed up the evaluation of sum-queries with response variable σ, the query system will make use of a materialized aggregate view on σ, which will be referred to as the *summary table* on σ; it is created initially and reports the total of σ for each cell c. As we said, a user can ask for the total of a statistic of σ by submitting a sum-query Q, where the selection criterion is specified be an arbitrary (consistent) condition P on a (possibly empty) subset of the set V_σ of categorical variables built up with logical connectives. Owing to the assumptions of mutual exclusiveness and global exhaustiveness of cells, P uniquely determines a category K, we call the *target* of Q, such that P is logically equivalent [20] to the formula

$$\vee_{c\in K}\ (\wedge_{a\in V_\sigma}\ a = c(a)).$$

The value q of Q can then be computed by the query system from the summary table on σ (without accessing the database relation R) simply as the total of σ for K.

Example 1. Consider an instance of an SDB containing a relation of name Personnel with schema {NAME, GENDER, AGE, SALARY}. Here, SALARY is assumed to be an attribute of nonnegative real type. A sum-query with response variable SALARY will take its categorical variables from the set V_{SALARY} = {GENDER, AGE}. We assume that the domains of GENDER and AGE are {M, F} and {young, middle, old}, respectively, and that the summary table on SALARY contains the following data.

GENDER	AGE	SALARY
M	young	15.0
M	middle	9.0
M	old	7.5
F	young	6.5
F	middle	1.5
F	old	0.0

Consider the following four sum-queries with response variable SALARY:

Q_1: select **sum**(SALARY)
from Personnel
where GENDER = M and AGE ≠ old

Q_2: select **sum**(SALARY)
from Personnel
where (GENDER = M and AGE ≠ young) or (GENDER = F and AGE = middle)

Q_3: select **sum**(SALARY)
from Personnel
where (GENDER = M and AGE ≠ middle) or (GENDER = F and AGE = young)

Q_4: select **sum**(SALARY)
from Personnel
where GENDER = F and AGE ≠ middle.

Their targets are the following relations with schema {GENDER, AGE}:

K_1	K_2	K_3	K_4
(M, young)	(M, middle)	(M, young)	(F, young)
(M, middle)	(M, old)	(M, old)	(F, old)
	(F, middle)	(F, young)	

Using the summary table on SALARY, the query system is able to compute the values

$q_1 = 24$ $q_2 = 18$ $q_3 = 29$ $q_4 = 6.5$

Q_1, Q_2, Q_3 and Q_4 without accessing the relation of name Personnel. ■

In the next section, we shall see how the snooper can use the information released by the query system in response to sum-queries to get a more-or-less detailed information on the total of σ for a (possibly sensitive) category of interest.

3 The snooper's model

Let $Q_1, \ldots, Q_n$ be answered sum-queries whose response variable σ is of nonnegative real type. The amount of information conveyed by the answers to $Q_1, \ldots, Q_n$ will be defined in such a way to capture all the snooper's knowledge about them (that is, their targets and their values). The snooper's information model consists of a set of variables, each of which is interpreted as the total of σ for a certain category, and of a system of linear constraints.

Let K_i and q_i, $i = 1, \ldots, n$, be the target and the value of Q_i, respectively. By $\Omega \subseteq dom(V_\sigma)$ we denote the set of cells contained in the categories K_i, that is, $\Omega = \cup_{i=1,\ldots,n} K_i$. Let $\mathbf{K} = \{K_1, \ldots,$

$K_n\}$ and let $\mathbf{C} = \{C_1, \ldots, C_m\}$ be a partition of Ω such that each K_i is the union of some of the C_j's. Using the following notation

$$N = \{1, \ldots, n\}$$
$$M = \{1, \ldots, m\}$$
$$J_i = \{j \in M: C_j \subseteq K_i\} \qquad (i \in N)$$

one has that

$$K_i = \cup_{j \in J_i} C_j \qquad (i \in N)$$

so that the snooper's information model can be described by the system of n linear equations

$$\sum_{j \in J_i} x_j = q_i \qquad (i \in N)$$

we re-write in matrix form as

$$\mathbf{A}\,\mathbf{x} = \mathbf{q} \qquad (1)$$

Here, each variable x_j stands for a feasible value of the total of σ for C_j and, hence, is subject to the nonnegativity constraint: $x_j \geq 0$. Since the size of system (1) depends on the number of classes of the partition $\mathbf{C}$, it is convenient to choose $\mathbf{C}$ with the minimum number of classes. It is well-known (e.g., see [14], pages 132-133]) that such a partition of Ω is uniquely determined and is formed by grouping together the cells in Ω that are contained in exactly the same sets in $\mathbf{K}$ (see Figure 1). According to the terminology used in [14], this partition of Ω will be referred to as the *minimal disjoint-set basis* of $\mathbf{K}$ (the *basis* of $\mathbf{K}$, for short).

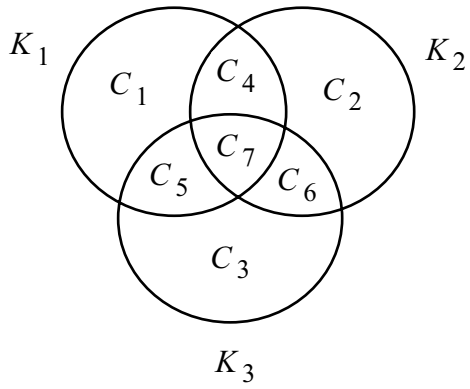

Figure 1. The basis of a set $\mathbf{K}$ of three categories

Remark 1 A category K_i is itself a member of the basis of **K** if and only if, for each $i' \neq i$, either $K_i \cap K_{i'} = \emptyset$ or $K_i \subseteq K_{i'}$.

To sum up, the information model has a semantic component, given by **C**, and an analytical component, given by system (1).

Example 1 (continued). Consider again the four sum-queries Q_1, …, Q_4. Then $\Omega = dom(\{\texttt{AGE}, \texttt{GENDER}\})$ and the basis **C** of **K** is formed by the following six elementary categories (i.e., singletons):

$C_1 = \{(\texttt{M}, \texttt{young})\}$ $C_2 = \{(\texttt{M}, \texttt{middle})\}$ $C_3 = \{(\texttt{M}, \texttt{old})\}$
$C_4 = \{(\texttt{F}, \texttt{young})\}$ $C_5 = \{(\texttt{F}, \texttt{middle})\}$ $C_6 = \{(\texttt{F}, \texttt{old})\}$.

The targets of Q_1, …, Q_4 can be written as

$K_1 = C_1 \cup C_2$ $J_1 = \{1, 2\}$
$K_2 = C_2 \cup C_3 \cup C_5$ $J_2 = \{2, 3, 5\}$
$K_3 = C_1 \cup C_3 \cup C_4$ $J_3 = \{1, 3, 4\}$
$K_4 = C_4 \cup C_6$ $J_4 = \{4, 6\}$

and system (1) reads

$$\begin{cases} x_1 + x_2 = 24 \\ x_2 + x_3 + x_5 = 18 \\ x_1 + x_3 + x_4 = 29 \\ x_4 + x_6 = 6.5 \end{cases} \qquad (1.1) \qquad \blacksquare$$

Of course, system (1) has at least one nonnegative solution. Suppose now that the snooper is interested in the total of σ for a (possibly sensitive) category K of interest. He can obtain the tightest lower bound and the tightest upper bound on the total σ for K as follows. Let (see Figure 2)

$J = \{j \in M: C_j \subseteq K\}$,
$J' = \{j \in M: C_j \cap K \neq \emptyset \text{ and } C_j - K \neq \emptyset\}$.

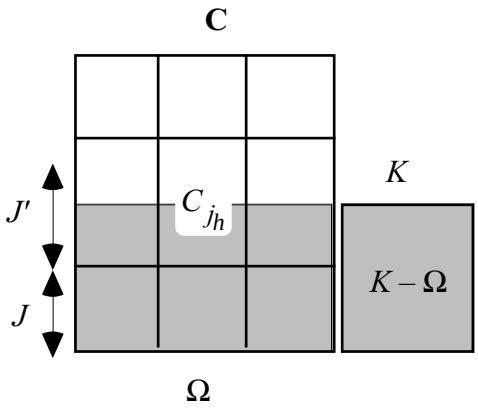

Figure 2. Overlap of category *K* with basis **C**

The tightest lower bound, denoted by *lower*(*K*), is set to zero if $J = \emptyset$; otherwise, it is taken to be

$$lower(K) = \min \{\sum_{j \in J} x_j : \mathbf{A}\,\mathbf{x} = \mathbf{q},\ \mathbf{x} \geq \mathbf{0}\} .$$

The tightest upper bound, denoted by *upper*(*K*), is set to $+\infty$ if *K* is not contained in Ω; otherwise, it is taken to be

$$upper(K) = \max \{\sum_{j \in J \cup J'} x_j : \mathbf{A}\,\mathbf{x} = \mathbf{q},\ \mathbf{x} \geq \mathbf{0}\} .$$

The interval [*lower*(*K*), *upper*(*K*)] is called the *feasibility range* of the total for *K*. If *lower*(*K*) = *upper*(*K*), then the total for *K* is said to be *evaluable* in the information model.

Example 1 (continued). The general solution of system (1.1) is

$$(x_1 = 24 - x_2, x_2, x_3, x_4 = 5 + x_2 - x_3, x_5 = 18 - x_2 - x_3, x_6 = 1.5 - x_2 + x_3).$$

By the nonnegativity constraints, the couple (x_2, x_3) is any point of the region shown in Figure 3.

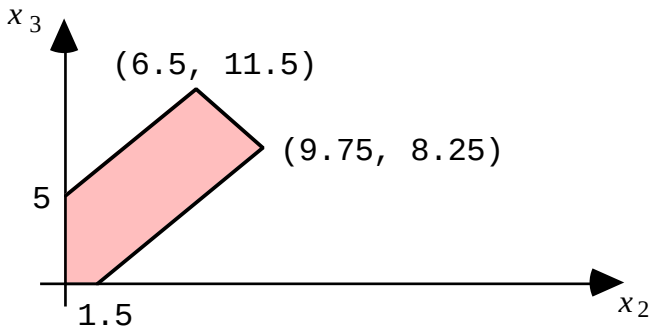

Figure 3. The set of nonnegative solutions of system (1.1)

Thus, if the snooper is interested in the total of SALARY for the category

K
(F, middle)
(F, old)

then, after computing the minimum and the maximum of the function $x_5 + x_6$ using a standard linear-programming method, he can find that the feasibility range for the total of SALARY for K is [0,19.5]. ■

4 How to beat the snooper

To repel the attacks of the snooper to the confidentiality of the response variable, the query system will make use of its own information model, which will be constructed incrementally as the value of a new-sum query is issued. Suppose that, after a certain number of answered sum-queries, every sensitive category is protected in the query system's model and a new sum-query Q is submitted. We call the query system's model the *prior model*. As said in the Introduction, if Q is intrusive or tricky, then Q should be answered by issuing the feasibility range of its value given by the prior model; otherwise, the value of Q should be issued. Since recognizing intrusive sum-queries is a matter of routine, the most demanding task that the query system is called to carry out is how to decide whether or not Q is tricky. Of course, if Q is evaluable in the prior model, then Q is not tricky. Let us assume that Q is not evaluable. Then, the query system should add to the prior model the piece of information conveyed by the value of Q, and in the "augmented" information model, we call the *posterior model*, it should check that each sensitive category is still protected. If this is the case, then (and only then) Q is not tricky. We now present a procedure for constructing the posterior model from the prior model given the target K and the value q of Q.

Let system (1) be the equation system of the prior model, and let $\mathbf{C} = \{C_1, \ldots, C_m\}$ be the basis in use, that is, the partition of Ω that gives the interpretation of the variables $x_1, \ldots, x_m$ of system (1). Let

$$J = \{j \in M: C_j \subseteq K\} \quad J' = \{j \in M: C_j \cap K \neq \emptyset \text{ and } C_j - K \neq \emptyset\}.$$

Let $p = |J'|$ and, if $p > 0$, let $J' = \{j_1, \ldots, j_p\}$. Let $\Omega' = \Omega \cup K$ and let $\mathbf{C}'$ be the partition of Ω' defined as follows (see Figure 4):

If $p > 0$, then set $C_{m+h} := C_{j_h} - K$ and $C_{j_h} := K \cap C_{j_h}$ for $h = 1, \ldots, p$.
If $\Omega' \neq \Omega$, then set $C_{m+p+1} := K - \Omega$

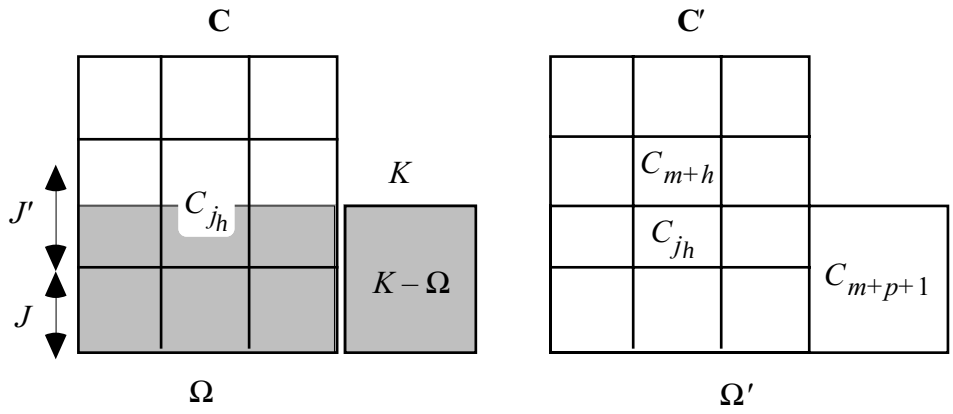

Figure 4. Updating basis **C**

Without loss of generality, henceforth we assume that $J' \neq \emptyset$ and $\Omega' \neq \Omega$. The posterior model consists of $m+p+1$ nonnegative variables $x_1, \ldots, x_m, \ldots, x_{m+p+1}$, which correspond to the categories in $\mathbf{C}' = \{C_1, \ldots, C_m, \ldots, C_{m+p+1}\}$. The equation system of the posterior model is obtained from system (1) by replacing each occurrence of x_{j_h} by $x_{j_h} + x_{m+h}$, and by adding the equation

$$\sum_{j \in J \cup J'} x_j + x_{m+p+1} = q \ .$$

After constructing the posterior model, each sensitive category S contained in Ω' is examined by computing the feasibility range of the total for S in the posterior model which is, then, compared with the protection level assigned to S. If all the sensitive categories contained in Ω' are protected in the posterior model, then (and only then) Q is not tricky.

Example 2. Consider again the database instance of Example 1. Suppose there are only two sensitive categories for SALARY:

S_1	S_2
(M, young)	(M, young)
	(F, old)

both of which have been assigned the protection level 3.0. Assume that the four sum-queries Q_1, …, Q_4 of Example 1 are submitted in this order. We now show that each of them can be safely answered. The following are the four information models resulting from answering the sum-queries Q_1, …, Q_4 respectively, and the feasibility ranges for the sensitive categories. Note that answering Q_1, Q_2 and Q_3 risks S_1 only, but answering Q_4 risks S_2 too.

$$\{x_1 = 24$$

C_1 = {(M, young), (M, middle)}

feasibility range for S_1: [0, 24]

$$\begin{cases} x_1 + x_2 = 24 \\ x_2 + x_3 = 18 \end{cases}$$

C_1 = {(M, young)} C_2 = {(M, middle)}
C_3 = {(M, old), (F, middle)}

feasibility range for S_1: [6, 24]

$$\begin{cases} x_1 + x_2 = 24 \\ x_2 + x_3 + x_5 = 18 \\ x_1 + x_3 + x_4 = 29 \end{cases}$$

C_1 = {(M, young)} C_2 = {(M, middle)} C_3 = {(M, old)}
C_4 = { (F, young)} C_5 = {(F, middle)}

feasibility range for S_1: [6, 24]

$$\begin{cases} x_1 + x_2 = 24 \\ x_2 + x_3 + x_5 = 18 \\ x_1 + x_3 + x_4 = 29 \\ x_4 + x_6 = 6.5 \end{cases}$$

$C_1 = \{(\texttt{M}, \texttt{young})\}$ $C_2 = \{(\texttt{M}, \texttt{middle})\}$ $C_3 = \{(\texttt{M}, \texttt{old})\}$
$C_4 = \{(\texttt{F}, \texttt{young})\}$ $C_5 = \{(\texttt{F}, \texttt{middle})\}$ $C_6 = \{(\texttt{F}, \texttt{old})\}$

feasibility range for S_1: [14.25,24]
feasibility range for S_2: [14.25,30.5]

Suppose now that the following sum-query is submitted after Q_4:

Q_5: select **sum**(SALARY)
from Personnel
where GENDER = F and AGE ≠ young.

The target of Q_5 is

K_5
(F, middle)
(F, old)

and the value of Q_5 is q_5 = 1.5 (obtained from the summary table). Neither the category K_5 is sensitive nor the total for K_5 is evaluable in the prior model. The posterior model consists of the following equation system

$$\begin{cases} x_1 + x_2 = 24 \\ x_2 + x_3 + x_5 = 18 \\ x_1 + x_3 + x_4 = 29 \\ x_4 + x_6 = 6.5 \\ x_5 + x_6 = 1.5 \end{cases} \qquad (1.2)$$

with the same meaning of the variables as in the prior model. The general solution of system (1.2) is

$(x_1 = 15, x_2 = 9, x_3, x_4 = 14 - x_3, x_5 = 9 - x_3, x_6 = -7.5 + x_3)$

By the nonnegativity constraints, x_3 is any number in the interval [7.5,9]. Note that the sensitive category S_1 is exactly the category associated with x_1 and, since the value of x_1 is uniquely determined, S_1 would not be protected if the value of Q_5 were released. Therefore, Q_5 is tricky and, in response to Q_5, the query system will issue the feasibility range [0,19.5] for K_5 given by the prior model (see Example 1). ■

To sum up, a procedure for answering a sum-query Q with target K and value q should work as follows:

(*i*) If K is sensitive, then the answer to Q will be the feasibility range of the total for K in the prior model.

(*ii*) If the total for K is evaluable in the prior model, then the answer to Q will be q.

(*iii*) If neither K is sensitive nor the total for K is evaluable in the prior model, then the posterior model will be constructed, and
for each sensitive category S contained in $\Omega' = \Omega \cup K$, the feasibility range of the total for S in the posterior model will be computed and compared with the protection level associated with S so that, if S is not protected, then the answer to Q will be the feasibility range of the total for K in the prior model.

(*iv*) If each sensitive category S contained in Ω' is protected in the posterior model, then the answer to Q will be q, and the posterior model will serve as the prior model for processing the next sum-query.

From a computational point of view, tasks (*i*), (*ii*) and (*iii*) require computing the feasibility range of the total for K and for each sensitive category contained in Ω'. Of course, a feasibility range can be found using standard linear-programming methods (e.g., the simplex method [7, 23]). Unfortunately, as recalled in the Introduction, standard linear-programming methods are not polynomial algorithms even if they are polynomial on the average and have good performances in practice. So, a parsimonious use of linear-programming methods is recommended [11].
We shall prove that, if a compact representation of the prior model is used, we call a "normal form" (see the next section), then the evaluability of the total for K — task (*ii*) — can be tested in cubic time using standard algebraic methods so that the query system needs not to resort to linear-programming methods for computing the feasibility range of the total for K. Moreover, the cubic algebraic algorithm can also be used to reduce the computational cost of carrying out task (*iii*) if a compact representation of the posterior model is also used: before computing the feasibility range of the total for S in the posterior model, S is tested for evaluability and, if the total of S comes out to be evaluable (this was the case for S_1 in Example 2), then its feasibility range need not be

computed since the query system can soon conclude that S is not protected. Accordingly, we propose the following answering procedure.

Answering Procedure

Step 1. If K is sensitive, then

compute *lower*(K) and *upper*(K)] in the prior model,

answer Q by issuing the feasibility range for the total for K, and Exit.

Step 2. If the total for K is evaluable in the prior model, then answer Q by releasing its value, and Exit.

Step 3. Construct the posterior model and find a normal form.

Step 4. For each sensitive category S contained in $K \cup \Omega$,

if the total for S is evaluable in the posterior model, then

compute *lower*(K) and *upper*(K) in the prior model,

answer Q by issuing the feasibility range of the total for K, and Exit;

otherwise

compute the feasibility range [*lower*(S), *upper*(S)] in the posterior model,

compare its width with the protection level associated with S,

if S is not protected, then

compute *lower*(K) and *upper*(K) in the prior model,

answer Q by issuing the feasibility range of the total for K, and Exit.

Step 5. Answer Q by issuing its value.

In the next section, we shall introduce the normal form of an information model and present a cubic test for evaluability.

5 A polynomial test for evaluability

The cost of the storage representation of an information model such as system (1) depends on the *size* of its equation system, by which we mean the sum of the numbers of occurrence of its variables (that is, $\sum_{i=1,\ldots,n} |J_i|$). For example, a storage representation of the information model may be the hypergraph of which the coefficient matrix **A** of system (1) is the (node-hyperedge) incidence matrix, where the i-th node is "weighted" by q_i and the j-th hyperedge is "labeled" by C_j. In this section, we provide a method for reducing the size of system (1) and, hence, the storage cost of the information model. We first give a procedure for reducing the size of system (1) while preserving the number of its variables and, then, a procedure for reducing the number of its variables. Preliminarily, we state some useful definitions.

Let **X** be the set of nonnegative solutions of equation system (1). A variable x_j is *determined* over **X** if $x_j = x'_j$ for every **x** and **x′** in **X**, and an equation in system (1) is *redundant* if the set of nonnegative solutions of the equation system obtained from system (1) by deleting that equation coincides with **X**. Of course, the presence of determined variables and redundant equations increases the size of equation system (1) surreptitiously. For example, if $K_{i*} = C_{j*}$ for some $i*$ and $j*$ (see Remark 1), then the variable x_{j*} is determined over **X** and its value equals the value q_{i*} of Q_{i*} so that, if x_{j*} occurs more than once in system (1), then x_{j*} can be replaced by q_{i*} in each equation of system (1) other than the equation $x_{j*} = q_{i*}$. Moreover, if $K_i = K_{i'}$ for $i \neq i'$, then the i-th equation is redundant and can be deleted.

We say that an equation system is *irreducible* if

— each determined variable x_j occurs in exactly one equation which is of the form $x_j = c_j$ (where c_j is the value of x_j), and

— no equation is redundant,

and that an equation system with variables $x_1, \ldots, x_m$, say

$$\mathbf{A}'\mathbf{x} = \mathbf{q}', \tag{2}$$

is a *reduced form* of system (1) if

— **A′** is a 0-1 matrix,

— system (2) is irreducible, and

— the set of nonnegative solutions of system (2) coincides with **X**.

Sometimes, the size of system (2) can be further reduced as follows. Let us consider its determined variables whose values are zero. They will be referred to as *null variables*. Let $\{x_j: j \in$

$Z\}$ be the set of null variables of system (2). If $Z \neq \emptyset$, then system (2) contains the set of equations

$$x_j = 0 \qquad (j \in Z)$$

which, by the nonnegativity constraints, is equivalent to the single equation

$$\sum_{j \in Z} x_j = 0.$$

Let us introduce a new variable x_0 which stands for the sum $\sum_{j \in Z} x_j$ and is to be interpreted as the total for the category $C_0 = \cup_{j \in Z} C_j$. Then, we can replace the equations $x_j = 0, j \in Z$, in system (2) by the single equation $x_0 = 0$ so that the resulting equation system has $|Z|-1$ variables less and $|Z|-1$ equations less. Let $\{x_j: j \in L\}$ be the set of nonnull determined variables, and $\{x_j: j \in H\}$ the set of undetermined variables. If $Z \neq \emptyset$ and $L \neq \emptyset$, then, the resulting equation system is like

$$\begin{cases} x_0 = 0 \\ \mathbf{x}_L = \mathbf{c} \\ \mathbf{G}\ \ \mathbf{x}_H = \mathbf{w} \end{cases} \qquad (3)$$

System (3) with the interpretation of its variables given by the following partition of Ω

$$\{C_0\} \cup \{C_j: j \in L \cup H\}$$

defines what we call a *normal form* of the information model. Given system (3) and a category K, the feasibility range of the total for K can be obtained as follows (see Figure 5).

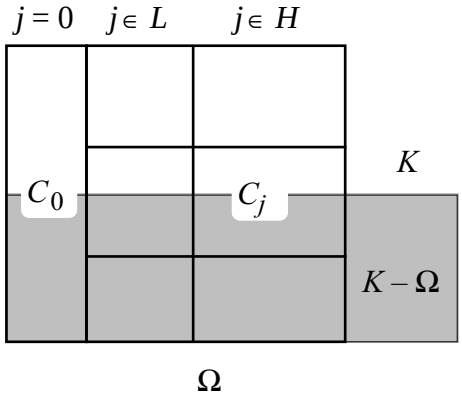

Figure 5. Overlap of the target with the category basis of a normal model

Let

$$J = \{j \in L \cup H\colon C_j \subseteq K\} \qquad J' = \{j \in L \cup H\colon C_j \cap K \neq \emptyset \text{ and } C_j - K \neq \emptyset\}.$$

Since the total of the response variable for the category $C_0 \cap K$ is definitely zero, the tightest lower bound for K is equal to

$$lower(K) = \sum_{j \in J \cap L} c_j + \min \{\sum_{j \in J \cap H} x_j\colon \mathbf{G}\,\mathbf{x}_H = \mathbf{w},\ \mathbf{x}_H \geq \mathbf{0}\} \tag{4}$$

and, if K is contained in Ω, then the tightest upper bound for K is equal to

$$upper(K) = \sum_{j \in (J \cup J') \cap L} c_j + \max \{\sum_{j \in (J \cup J') \cap H} x_j\colon \mathbf{G}\,\mathbf{x}_H = \mathbf{w},\ \mathbf{x}_H \geq \mathbf{0}\}\ . \tag{5}$$

Remark 2 Since $c_j > 0$ for $j \in L$, then one has that

— $\sum_{j \in J \cap L} c_j \geq 0$ where the equality holds if and only if $J \cap L = \emptyset$, and

— $\sum_{j \in J \cap L} c_j \leq \sum_{j \in (J \cup J') \cap L} c_j$ where the equality holds if and only if $J' \cap L = \emptyset$.

The following is a characterization of evaluability.

Theorem 1 Let system (3) be the equation system of an information model in normal form. The total for a category K is evaluable if and only if

(a) K is contained in Ω,

(b) $J' = \emptyset$, and

(c) the function $\sum_{j \in J \cap H} x_j$ is constant over the set of nonnegative solutions of the equation system $\mathbf{G}\,\mathbf{x}_H = \mathbf{w}$.

Proof. The "if"-part trivially follows from equations (4) and (5). On the other hand, if the total for K is evaluable, then condition (a) must hold and from Remark 2 it follows that $J' \cap L$ must be empty. Suppose by contradiction that condition (b) does not hold. Then, $J' \cap H \neq \emptyset$ and

$$\max \{\sum_{j \in J \cap H} x_j\colon \mathbf{G}\,\mathbf{x}_H = \mathbf{w},\ \mathbf{x}_H \geq \mathbf{0}\} < \max \{\sum_{j \in (J \cup J') \cap H} x_j\colon \mathbf{G}\,\mathbf{x}_H = \mathbf{w},\ \mathbf{x}_H \geq \mathbf{0}\}$$

for, otherwise,

$$\max \{\sum_{j \in J' \cap H} x_j\colon \mathbf{G}\,\mathbf{x}_H = \mathbf{w},\ \mathbf{x}_H \geq \mathbf{0}\} = 0$$

and each x_j, $j \in J' \cap H$ would be a null variable (contradiction). So, condition (*b*) must hold. As to condition (*c*), suppose by contradiction that

$$\min \{\sum_{j \in J \cap H} x_j \colon \mathbf{G}\, \mathbf{x}_H = \mathbf{w},\ \mathbf{x}_H \geq \mathbf{0}\} < \max \{\sum_{j \in J \cap H} x_j \colon \mathbf{G}\, \mathbf{x}_H = \mathbf{w},\ \mathbf{x}_H \geq \mathbf{0}\}.$$

Then, the following contradiction would arise

$$\begin{aligned} \mathit{lower}(K) &= \sum_{j \in J \cap L} c_j + \min \{\sum_{j \in J \cap H} x_j \colon \mathbf{G}\, \mathbf{x}_H = \mathbf{w},\ \mathbf{x}_H \geq \mathbf{0}\} < \\ &< \sum_{j \in J \cap L} c_j + \max \{\sum_{j \in J \cap H} x_j \colon \mathbf{G}\, \mathbf{x}_H = \mathbf{w},\ \mathbf{x}_H \geq \mathbf{0}\} = \mathit{upper}(K). \end{aligned}$$ □

From a computational point of view, conditions (*a*) and (*b*) are easy to test and require $O(|\Omega|)$ and $O(1)$ time, respectively. As to condition (*c*), it can be tested by comparing

$$\min \{\sum_{j \in J \cap H} x_j \colon \mathbf{G}\, \mathbf{x}_H = \mathbf{w},\ \mathbf{x}_H \geq \mathbf{0}\}$$

and

$$\max \{\sum_{j \in J \cap H} x_j \colon \mathbf{G}\, \mathbf{x}_H = \mathbf{w},\ \mathbf{x}_H \geq \mathbf{0}\},$$

whose computation requires solving two linear-programming problems. However, we can do better since we don't need to resort to linear-programming methods but can use a merely algebraic method, which rests on a technical lemma (see Lemma 3 below) involving an arbitrary linear equation system whose variables are constrained to be nonnegative real-valued.

Affine Form of Farkas's Lemma (e.g., see [23], page 93). Let $\mathbf{H}\, \mathbf{x} \leq \mathbf{b}$ be a system of n linear inequalities with m unknowns such that the set $\mathbf{X}$ of its solutions is not empty. Suppose that the linear inequality $\sum_{j=1,\ldots,m} u_j\, x_j \leq d$ holds for each $\mathbf{x}$ in $\mathbf{X}$. If c is the maximum value of the linear function $\sum_{j=1,\ldots,m} u_j\, x_j$ over $\mathbf{X}$, then there exist n real nonnegative numbers $y_1, \ldots, y_n$, such that

(*i*) $\quad u_j = \sum_{i=1,\ldots,n} y_i\, h_{ij} \qquad\qquad (j = 1, \ldots, m)$

(*ii*) $\quad c = \sum_{i=1,\ldots,n} y_i\, b_i.$

The following is a direct consequence of Farkas's Lemma.

Lemma 1 [15] Let $\mathbf{A}\, \mathbf{x} = \mathbf{q}$ be a system of n linear equations with m unknowns, such that the set $\mathbf{X}$ of its nonnegative solutions is not empty. If c is the maximum value of the linear function $\sum_{j=1,\ldots,m} u_j\, x_j$ over $\mathbf{X}$, then there exist n real numbers $\lambda_1, \ldots, \lambda_n$, and m nonnegative real numbers $v_1, \ldots, v_m$ such that

(*i*) $u_j = \sum_{i=1,\ldots,n} \lambda_i \, a_{ij} - v_j$ $\qquad (j = 1, \ldots, m)$

(*ii*) $c = \sum_{i=1,\ldots,n} \lambda_i \, q_i$.

Proof. (See the Appendix). □

Using Lemma 1, we can prove the following.

Lemma 2 Let **A x** = **q** be a system of n linear equations with m unknowns, such that the set **X** of its nonnegative solutions is not empty. If a linear function $\sum_{j=1,\ldots,m} u_j \, x_j$ is constant over **X** with value c, then there exist n real numbers $\lambda_1, \ldots, \lambda_n$, and m nonnegative real numbers $v_1, \ldots, v_m$ such that

(*i*) $u_j = \sum_{i=1,\ldots,n} \lambda_i \, a_{ij} - v_j$ $\qquad (j = 1, \ldots, m)$

(*ii*) $c = \sum_{i=1,\ldots,n} \lambda_i \, q_i$

(*iii*) for each j, if $v_j \neq 0$ then x_j is a null variable.

Proof. (*If*) By Lemma 1, for every **x** in **X**, one has that

$$\begin{aligned} \sum_{j=1,\ldots,m} u_j \, x_j \quad &= \sum_{j=1,\ldots,m} \left(\sum_{j=1,\ldots,m} \lambda_i \, a_{ij} - v_j\right) x_j = \\ &= \sum_{j=1,\ldots,m} \lambda_i \left(\sum_{j=1,\ldots,m} a_{ij} \, x_j\right) - \sum_{j=1,\ldots,m} v_j \, x_j = \\ &= \sum_{i=1,\ldots,n} \lambda_i \, q_i \end{aligned}$$

(*Only if*) Let c be the constant value of the function $\sum_{j=1,\ldots,m} u_j \, x_j$. By Lemma 1, one has

$$u_j = \sum_{i=1,\ldots,n} \lambda_i \, a_{ij} - v_j \qquad (j = 1, \ldots, m)$$

$$c = \sum_{i=1,\ldots,n} \lambda_i \, q_i.$$

Then, for every **x** in **X**, one has that

$$c = \sum_{j=1,\ldots,m} u_j \, x_j = \sum_{i=1,\ldots,n} \lambda_i \, q_i - \sum_{j=1,\ldots,m} v_j \, x_j = c - \sum_{j=1,\ldots,m} v_j \, x_j$$

and, hence,

$$\sum_{i=1,\ldots,m} v_j \, x_j = 0 \, .$$

By the nonnegativity constraints, each term $v_j\, x_j = 0$ for all j, which implies that it is not the case that for some j there exists $\mathbf{x}$ in $\mathbf{X}$ for which $v_j \neq 0$ and $x_j \neq 0$. □

Lemma 3 Let $\mathbf{A}\,\mathbf{x} = \mathbf{q}$ be a system of n linear equations with m unknowns, such that the set $\mathbf{X}$ of its nonnegative solutions is not empty. Let $\{x_j: j \in Z\}$ be the set of null variables. Let $\mathbf{A}'\,\mathbf{x}' = \mathbf{q}$ be the equation system obtained from $\mathbf{A}\,\mathbf{x} = \mathbf{q}$ by deleting each occurrence of x_j, for all $j \in Z$. A linear function $\sum_{j=1,\ldots,m}\, u_j\, x_j$ is constant over $\mathbf{X}$ if and only if the vector $\mathbf{u} = (u_j)_{j \notin Z}$ is a linear combination of rows of $\mathbf{A}'$. Furthermore, if $\mathbf{u} = \sum_{i=1,\ldots,n}\, \lambda_i\, \mathbf{a}'_i$, then the value of the function is given by $\sum_{i=1,\ldots,n}\, \lambda_i\, q_i$.

Proof. Let $\mathbf{X}'$ be the set of nonnegative solutions $\mathbf{A}'\,\mathbf{x}' = \mathbf{q}$. Of course, the function $\sum_{j=1,\ldots,m}\, u_j\, x_j$ is constant over $\mathbf{X}$ if and only if the function $\sum_{j \notin Z}\, u_j\, x'_j$ is constant over $\mathbf{X}'$. By parts (*i*) and (*iii*) of Lemma 2, the function $\sum_{j \notin Z}\, u_j\, x'_j$ is constant over $\mathbf{X}'$ if and only if there exist n real numbers $\lambda_1, \ldots, \lambda_n$, and, for each $j \notin Z$, there exists a nonnegative real number v_j such that

$$u_j = \sum_{i=1,\ldots,n}\, \lambda_i\, a_{ij} - v_j \qquad (j \notin Z)$$

for each $j \notin Z$, if $v_j \neq 0$ then x'_j is a null variable.

Since the equation system $\mathbf{A}'\,\mathbf{x}' = \mathbf{q}$ contains no null variables, one has that each v_j is zero so that the function $\sum_{j \notin Z}\, u_j\, x'_j$ is constant over $\mathbf{X}'$ if and only if there exist n real numbers $\lambda_1, \ldots, \lambda_n$ such that

$$u_j = \sum_{i=1,\ldots,n}\, \lambda_i\, a_{ij} \qquad (j \notin Z)$$

Finally, for every $\mathbf{x}$ in $\mathbf{X}$, one has that

$$\sum_{j=1,\ldots,m}\, u_j\, x_j = \sum_{j \notin Z}\, u_j\, x'_j = \sum_{j \notin Z}\, (\sum_{i=1,\ldots,n}\, \lambda_i\, a_{ij})\, x'_j = \sum_{i=1,\ldots,n}\, \lambda_i\, (\sum_{j \notin Z}\, a_{ij}\, x'_j)$$

$$= \sum_{i=1,\ldots,n}\, \lambda_i\, q_i \ . \qquad \square$$

The following are two straightforward consequences of Lemma 3.

Corollary 1 Let $\mathbf{A}\,\mathbf{x} = \mathbf{q}$ be a system of n linear equations with m unknowns, such that the set $\mathbf{X}$ of its nonnegative solutions is not empty. Let $\{x_j: j \in Z\}$ be the set of null variables. Let $\mathbf{A}'\,\mathbf{x}' = \mathbf{q}$ be the equation system obtained from $\mathbf{A}\,\mathbf{x} = \mathbf{q}$ by deleting each occurrence of x_j, for all $j \in Z$. A variable x_{j^*} is determined over $\mathbf{X}$ if and only if either $j^* \in Z$ or the characteristic vector of the singleton $\{j^*\}$, that is, the $(m-|Z|)$-dimensional binary vector $\mathbf{u}$ with

$$u_j = \begin{cases} 1 & \texttt{if } j = j^* \\ 0 & \texttt{else} \end{cases}$$

is a linear combination of rows of $\mathbf{A}'$. In the latter case, if $\mathbf{u} = \sum_{i=1,\ldots,n} \lambda_i \, \mathbf{a}'_i$ then the value of x_{j^*} is $\sum_{i=1,\ldots,n} \lambda_i \, q_i$.

Corollary 2 Let $\mathbf{A}\,\mathbf{x} = \mathbf{q}$ be a system of n linear equations with m unknowns, such that the set $\mathbf{X}$ of its nonnegative solutions is not empty. Let $\mathbf{A}'\mathbf{x}' = \mathbf{q}$ be the equation system obtained from $\mathbf{A}\,\mathbf{x} = \mathbf{q}$ by deleting all null variables. An equation in $\mathbf{A}\,\mathbf{x} = \mathbf{q}$ is redundant if and only if its corresponding equation in $\mathbf{A}'\mathbf{x}' = \mathbf{q}$ is a linear combination of remaining equations.

We now turn to the problem of testing condition (*c*) of Theorem 1. Since the equation system $\mathbf{G}\,\mathbf{x}_H = \mathbf{w}$ has no null variables, by Lemma 3 the function $\sum_{j \in J \cap H} x_j$ is constant if and only if the characteristic vector $\mathbf{u}$ of $J \cap H$, that is, the vector $\mathbf{u}$ with

$$u_j = \begin{cases} 1 & \texttt{if } j \in J \cap H \\ 0 & \texttt{else} \end{cases}$$

is a linear combination of rows of $\mathbf{G}$. If r is the number of rows of the matrix $\mathbf{G}$, then we have to check the consistency of an equation system with r unknowns, say $\lambda_1, \ldots, \lambda_r$:

$$\mathbf{u} = \sum_{i=1,\ldots,r} \lambda_i \, \mathbf{g}_i$$

If a solution exists, then the function $\sum_{j \in J \cap H} x_j$ is constant with value

$$\sum_{i=1,\ldots,r} \lambda_i \, w_i$$

and the total for K amounts to $\sum_{j \in J \cap L} c_j + \sum_{i=1,\ldots,r} \lambda_i \, w_i$. Finally, since a solution (if any) of an equation system of size s can be found in $O(s^3)$ time [7], the following holds.

Theorem 2 Given an information model in normal form, the evaluability of the total for a category can be tested in cubic time.

Example 3. Consider an instance of an SDB containing a relation of name `Personnel` with schema {`NAME`, `GENDER`, `AGE`, `DEPT`, `SALARY`}, where `SALARY` is assumed to be of nonnegative real type. Let {`A`, `B`, `C`, `D`, …} be the domain of `DEPT`.

Consider fourteen sum-queries Q_1, …, Q_{14} with response variable SALARY whose targets (K_i) are as follows:

K_1	K_2	K_3	K_4
(M, young, D)	(M, middle, B)	(M, middle, D)	(F, young, A)
K_5	K_6	K_7	K_8
(F, young, D)	(F, middle, A)	(F, middle, B)	(F, middle, C)

K_9	K_{10}	K_{11}	K_{12}
(M, young, A)	(M, middle, A)	(F, young, A)	(M, young, A)
(M, young, B)	(M, middle, B)	(F, young, B)	(M, middle, A)
(M, young, C)	(M, middle, C)	(F, young, C)	(F, young, A)
(M, young, D)	(M, middle, D)	(F, young, D)	(F, middle, A)

K_{13}	K_{14}
(M, young, B)	(M, young, C)
(M, middle, B)	(M, middle, C)
(F, young, B)	(F, young, C)
(F, middle, B)	(F, middle, C)

Let us assume that the values of Q_1, …, Q_{14} are as follows:

$q_1 = 0$	$q_2 = 5$	$q_3 = 10$	$q_4 = 10$
$q_5 = 10$	$q_6 = 15$	$q_7 = 20$	$q_8 = 10$
$q_9 = 30$	$q_{10} = 25$	$q_{11} = 25$	$q_{12} = 30$
$q_{13} = 60$	$q_{14} = 15$		

Note that q_1, …, q_{14} can be viewed as being the entries in an incomplete two-dimensional table (see Figure 6), where q_1, …, q_8 are the internal entries, q_9, …, q_{11} are the row totals and q_{12}, …, q_{14} are the column totals.

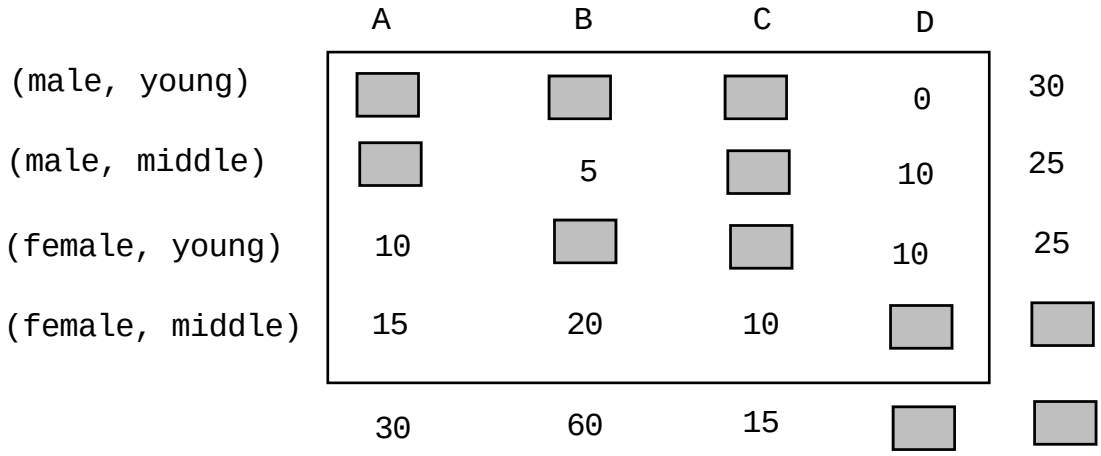

Figure 6. An incomplete two-dimensional table

The basis of $\{K_1, \ldots, K_{14}\}$ is formed by the following fifteen elementary categories:

$C_1 = \{(\mathtt{M}, \mathtt{young}, \mathtt{A})\}$ $C_2 = \{(\mathtt{M}, \mathtt{young}, \mathtt{B})\}$
$C_3 = \{(\mathtt{M}, \mathtt{young}, \mathtt{C})\}$ $C_4 = \{(\mathtt{M}, \mathtt{young}, \mathtt{D})\}$
$C_5 = \{(\mathtt{M}, \mathtt{middle}, \mathtt{A})\}$ $C_6 = \{(\mathtt{M}, \mathtt{middle}, \mathtt{B})\}$
$C_7 = \{(\mathtt{M}, \mathtt{middle}, \mathtt{C})\}$ $C_8 = \{(\mathtt{M}, \mathtt{middle}, \mathtt{D})\}$
$C_9 = \{(\mathtt{F}, \mathtt{young}, \mathtt{A})\}$ $C_{10} = \{(\mathtt{F}, \mathtt{young}, \mathtt{B})\}$
$C_{11} = \{(\mathtt{F}, \mathtt{young}, \mathtt{C})\}$ $C_{12} = \{(\mathtt{F}, \mathtt{young}, \mathtt{D})\}$
$C_{13} = \{(\mathtt{F}, \mathtt{middle}, \mathtt{A})\}$ $C_{14} = \{(\mathtt{F}, \mathtt{middle}, \mathtt{B})\}$
$C_{15} = \{(\mathtt{F}, \mathtt{middle}, \mathtt{C})\}$

and system (1) reads

$$\begin{cases} x_4 = 0, x_6 = 5, x_8 = 10, x_9 = 10, x_{12} = 10, x_{13} = 15, x_{14} = 20, x_{15} = 10 \\ x_1 + x_2 + x_3 + x_4 = 30 \\ x_5 + x_6 + x_7 + x_8 = 25 \\ x_9 + x_{10} + x_{11} + x_{12} = 25 \\ x_1 + x_5 + x_9 + x_{13} = 30 \\ x_2 + x_6 + x_{10} + x_{14} = 60 \\ x_3 + x_7 + x_{11} + x_{15} = 15 \end{cases} \qquad (1.2)$$

Using standard linear-programming methods, we find that each variable x_j is determined:

$x_1 = 0$	$x_2 = 30$	$x_3 = 0$	$x_4 = 0$
$x_5 = 5$	$x_6 = 5$	$x_7 = 5$	$x_8 = 10$
$x_9 = 10$	$x_{10} = 5$	$x_{11} = 0$	$x_{12} = 10$
$x_{13} = 15$	$x_{14} = 20$	$x_{15} = 10$	

The set of these fifteen equalities is the reduced form of system (1.2), and its normal form is

$x_0 = 0$	$x_2 = 30$	$x_5 = 5$	$x_6 = 5$	$x_7 = 5$	$x_8 = 10$
$x_9 = 10$	$x_{10} = 5$	$x_{12} = 10$	$x_{13} = 15$	$x_{14} = 20$	$x_{15} = 10$

where x_0 stands for the total of SALARY for the category

C_0
(M, young, A)
(M, young, C)
(M, young, D)
(F, young, C)

It follows that the total for a category K is evaluable if and only if $K - C_0$ is the union of zero or more categories from $\{C_2, C_5, C_6, C_7, C_8, C_9, C_{10}, C_{12}, C_{13}, C_{14}, C_{15}\}$. ■

6 Management of the normal form

In our Answering Procedure of Section 4, at Step 3, when the posterior model is constructed, it should be reduced in normal form because it becomes the next prior model if Step 5 is executed. This also allows to test the evaluability of each sensitive category (Step 4) in cubic time. In this section, we present a procedure for getting the posterior model in normal form (Step 3). We start with the prior model, which is assumed to be in normal form and to consist of system (3) with the interpretation of its variables given by the partition $\mathbf{C} = \{C_0, C_1, \ldots, C_m\}$ of Ω where the categories $C_1, \ldots, C_m$ are partitioned into the two groups $\{C_j: j \in L\}$ and $\{C_j: j \in H\}$ as shown in Figure 5. Let

$$J = \{j \in L \cup H: C_j \subseteq K\} \qquad J' = \{j \in L \cup H: C_j \cap K \neq \emptyset \text{ and } C_j - K \neq \emptyset\}$$

Let $p = |J'|$ and, if $p > 0$, let $J' = \{j_1, \ldots, j_p\}$. The procedure first constructs the semantic part of the posterior model using the partition $\mathbf{C}' = \{C_0, C_1, \ldots, C_m, \ldots, C_{m+p+1}\}$ of $\Omega' = \Omega \cup K$ defined as follows (see Figure 7):

If $p > 0$, then set $C_{m+h} := C_{j_h} - K$ and $C_{j_h} := K \cap C_{j_h}$ for $h = 1, \ldots, p$.
If $\Omega' \neq \Omega$, then set $C_{m+p+1} := K - \Omega$

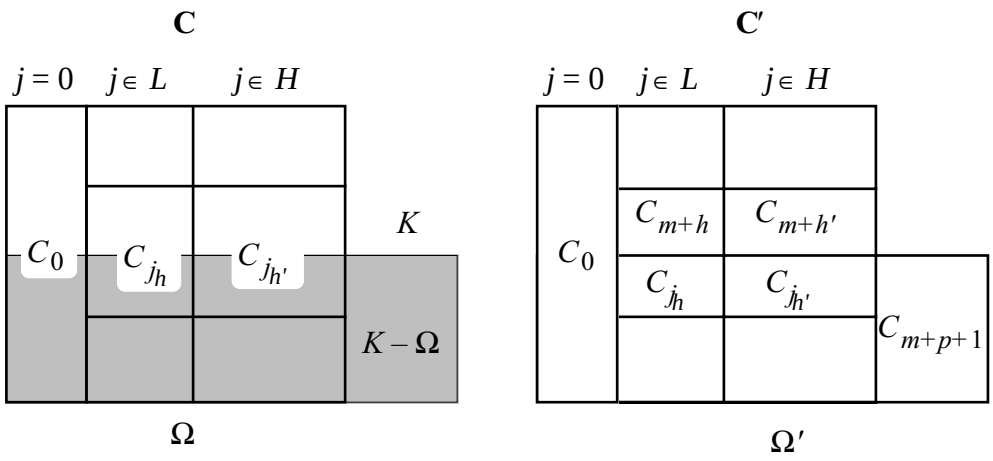

Figure 7. Updating the basis of a normal model

Without loss of generality, again we assume that $J' \neq \varnothing$ and $\Omega' \neq \Omega$. For $h = 1, \ldots, p$, associate with the variable x_{j_h} the category C_{j_h} from $\mathbf{C}'$, and introduce the variable x_{m+h} and associate with x_{m+h} the category C_{m+h} from $\mathbf{C}'$. Next, introduce the variable x_{m+p+1} and associate with x_{m+p+1} the category C_{m+p+1} from $\mathbf{C}'$. The equation system of the posterior model is then obtained from system (3) as follows. Re-write system (3) as

$$\begin{cases} x_0 = 0 \\ x_j = c_j \qquad (j \in L) \\ \sum_{j \in H} g_{ij} x_j = w_i \quad (i = 1, \ldots, n) \end{cases}$$

For $h = 1, \ldots, p$, replace each occurrence of the variable x_{j_h} by the expression $x_{j_h} + x_{m+h}$. Finally, since the total for the category $C_0 \cap K$ is always zero, add the equation

$$\sum_{j \in J \cap L} c_j + \sum_{j \in J \cap H} x_j + \sum_{h=1,\ldots,p} x_{j_h} + x_{m+p+1} = q$$

which we re-write as

$$\sum_{j \in J \cap H} x_j + \sum_{h=1,\ldots,p} x_{j_h} + x_{m+p+1} = q'$$

where $q' = q - \sum_{j \in J \cap L} c_j$.

Thus, the equation system of the posterior model reads

$$\begin{cases} x_0 = 0 & \\ x_j = c_j & (j \in L - J') \\ x_{j_h} + x_{m+h} = c_{j_h} & (j_h \in J' \cap L) \\ \sum_{j \in H - J'} g_{ij} x_j + \sum_{j_h \in J' \cap H} g_{ij_h} (x_{j_h} + x_{m+h}) = w_i & (i = 1, \ldots, n) \\ \sum_{j \in J \cap H} x_j + \sum_{h=1,\ldots,p} x_{j_h} + x_{m+p+1} = q' & \end{cases} \tag{6}$$

At this point, in order to get a normal form of the posterior model, we need a reduced form of system (6), that is, to find its determined variables and its redundant equations. Preliminarily, note that x_0 is a null variable of system (6) and that each $x_j, j \in L - J'$, is a nonnull determined variable of system (6). Moreover, for each $j_h \in J' \cap L$, the variables x_{j_h} and x_{m+h} appear only in the equation $x_{j_h} + x_{m+h} = c_{j_h}$ and are undetermined since the feasibility range for both is $[0, c_{j_h}]$. Let us consider the remaining variables and their equation system

$$\begin{cases} \sum_{j \in H - J'} g_{ij} x_j + \sum_{j_h \in J' \cap H} g_{ij_h} (x_{j_h} + x_{m+h}) = w_i & (i = 1, \ldots, n) \\ \sum_{j \in J \cap H} x_j + \sum_{h=1,\ldots,p} x_{j_h} + x_{m+p+1} = q' & \end{cases} \tag{7}$$

Finding a determined variable or a redundant equation in system (7) can be done using linear-programming methods. We can do better by exploiting Corollaries 1 and 2. Thus, if we first find the set of null variables of system (7) using a standard linear-programming algorithm, then we can decide in cubic time if a nonnull variable in equation system (7) is determined or if an equation in equation system (7) is redundant. Indeed, the search for null variables can be restricted to the variables corresponding to categories for which the summary table gives a zero total; analogously, the search for nonnull determined variables can be restricted to the variables corresponding to categories for which the summary table gives a nonzero total. Suppose that we have already found a reduced form of system (7), say

$$\begin{cases} x_j = 0 & (j \in Z) \\ x_j = c_j & (j \in L') \\ \mathbf{G}' \ \mathbf{x}' = \mathbf{w}' & \end{cases} \tag{8}$$

Finally, we are in a position to get a normal form of the posterior model. Its equation system is obtained by combining system (6) and system (8):

$$\begin{cases} x_0 = 0 \\ x_j = c_j & (j \in (L - J') \cup L') \\ x_{j_h} + x_{m+h} = c_{j_h} & (j_h \in J' \cap L) \\ \mathbf{G}'\ \mathbf{x}' = \mathbf{w}' \end{cases}$$

where the variable x_0 has now associated the category

$$(\cup_{j \in Z} C_j) \cup C_0 \ .$$

Example 2 (continued). Consider again the posterior model constructed when the sum-query Q_5 is processed.

$$\begin{cases} x_1 + x_2 = 24 \\ x_2 + x_3 + x_5 = 18 \\ x_1 + x_3 + x_4 = 29 \\ x_4 + x_6 = 6.5 \\ x_5 + x_6 = 1.5 \end{cases}$$

C_1 = {(M, young)} C_2 = {(M, middle)} C_3 = {(M, old)}
C_4 = { (F, young)} C_5 = {(F, middle)} C_6 = {(F, old)}

Since C_6 is the only category for which the summary table on SALARY gives a zero total, the only candidate for a null variable is the variable x_6. Using the simplex method, we find that the maximum value of x_6 is 1.5 so that x_6 is not a null variable. At this point, the nonnull determined variables are found by exploiting Corollary 1. The variables that are candidates for nonnull determined variables are x_1, …, x_5, and one finds that the only determined variables are x_1 and x_2 with values 15 and 9, respectively. For example, the characteristic vector

[1 0 0 0 0 0]

of the singleton {1} can be written as a linear combination of the rows of the coefficient matrix of the equation system above

$$\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix}$$

with coefficients equal to $\frac{1}{2}$ for the odd rows and $-\frac{1}{2}$ for the even rows. So, x_1 is determined with value $\frac{1}{2}(24 - 18 + 29 - 6.5 + 1.5) = 15$.

After deleting the two determined variables and adding the two equations $x_1 = 15$ and $x_2 = 9$, we obtain the following equation system

$$\begin{cases} x_1 = 15, x_2 = 9 \\ x_3 + x_5 = 9 \\ x_3 + x_4 = 14 \\ x_4 + x_6 = 6.5 \\ x_5 + x_6 = 1.5 \end{cases}$$

At this point, the redundant equations are found by exploiting Corollary 2. If the equation $x_3 + x_5 = 9$ is first examined, then it comes out to be redundant and none of the remaining equations is redundant. Explicitly, for the left-hand side and the right-hand side of the equation $x_3 + x_5 = 9$ one has

$$\begin{aligned} x_3 + x_5 &= (x_3 + x_4) - (x_4 + x_6) + (x_5 + x_6) \\ 9 &= (14) - (6.5) + (1.5) \end{aligned}$$

Thus, we have obtained the following reduced form of the equation system of the posterior model:

$$\begin{cases} x_1 = 15, x_2 = 9 \\ x_3 + x_4 = 14 \\ x_4 + x_6 = 6.5 \\ x_5 + x_6 = 1.5 \end{cases}$$

Since there are no null variables, this is also the equation system of the normal form of the posterior model. ■

7 Graphical model

We saw that, in general, the Answering Procedure given in Section 4 needs to resort to linear-programming methods only to solve the following two problems:

(P1) given the equation system of an arbitrary information model (e.g., the posterior model), find the set of its null variables;

(P2) given the equation system of an information model in normal form (e.g., the prior model), find the feasibility range for a given sum of variables.

However, some special instances of either problem can be very efficiently solved. For example, if the information model represents a "decomposable" set of marginals of an unknown multidimensional table, then the feasibility range for a single variable (corresponding to a cell entry) can be computed using closed formulas [10]. In this section, we show that both problems P1 and P2 can be efficiently solved for a *graphical* information model, that is, for an information model with equation system

$$\mathbf{G}\,\mathbf{x} = \mathbf{w}\,, \tag{9}$$

where the coefficient matrix **G** can be viewed as the incidence matrix of a graph. Note that this is the case whenever the information model represents the contents of internal and marginal cells of a (possibly incomplete) two-dimensional table (see Example 3).
In what follows, by $G = (V, E)$ with $V = \{1, \ldots, n\}$ and $E = \{e_1, \ldots, e_m\}$, we denote the graph of which **G** is the incidence matrix. Moreover, the edges of G are arranged in such a way that, if l is the number of loops, then the loops of G are denoted by $e_{m-l+1}, \ldots, e_m$. Finally, without loss of generality, G is assumed to be connected.

Problem (P1). If G is bipartite, then Gusfield [12] proved that, given a nonnegative solution of system (9), the set of null variables can be found in strongly linear time [12]. Two of the authors [17] proved that the same holds for an arbitrary graph. The point is how to get a nonnegative solution of system (9). Of course, the query system can get the true solution from the summary table, which requires $O(|\Omega|)$ time. However, a nonnegative solution of system (9) can also be obtained without passing through the summary table. For example, if G is bipartite, then Gusfield [12] proved that the nonnegative solutions of system (9) naturally correspond to maximum flows on a suitable bipartite flow network $\mathcal{N}(G, \mathbf{w})$ so that a nonnegative solution can be obtained in time cubic in the number of nodes of G [2]. If G is not bipartite, then two of the authors [17] proved that in linear time one can construct an equation system

$$\mathbf{H}\,\mathbf{y} = \mathbf{v}\,, \tag{10}$$

where **H** is the incidence matrix of a bipartite graph H with $2n$ nodes and $2m–l$ edges and **v** has $v_i := v_{n+i} := w_i$ ($i = 1, \ldots, n$), such that:

— given a solution **y** of system (10), then a solution of system (9) can be obtained by setting

$$x_j = \begin{cases} \frac{1}{2}(y_j + y_{m+j}) & 1 \le j \le m - l \\ y_j & m - l + 1 \le j \le m \end{cases}$$

we call the solution of system (9) associated with **y**;

— given a solution **x** of system (9), then a solution of system (10) can be obtained by setting

$$y_j = \begin{cases} x_j & 1 \le j \le m \\ x_{j-m} & m + 1 \le j \le 2m - l \end{cases}$$

we call the solution of system (10) associated with **x**.

So, since a nonnegative solution of system (10) can be obtained from the flow network $\mathcal{N}(H, \mathbf{v})$ in cubic time, problem (P1) can be solved in $O(n^3)$.

Problem (P2). It has been solved for a single variable by Gusfield [12] if G is bipartite, and by two of the authors [18] if G is not bipartite, using a maximum-flow algorithm in both cases. We shall show that, more in general, in a graphical information model the problem of finding the feasibility range for an arbitrary sum of variables can be solved using a strongly polynomial algorithm.
Suppose that, given a subset J of $\{1, \ldots, m\}$, one wants to compute the tightest lower bound or the tightest upper bound on the sum of variables $\sum_{j \in J} x_j$ over the set of nonnegative solutions of system (9). They can be obtained by solving the linear-programming problem

$$\begin{array}{ll} \text{minimize} & \sum_{j=1,\ldots,m} c_j x_j \qquad (11) \\ \text{subject to} & \mathbf{G}\,\mathbf{x} = \mathbf{w},\ \mathbf{x} \ge \mathbf{0} \end{array}$$

where each c_j is set to the j-th component of the characteristic vector of J (for the tightest lower bound) or to its opposite (for the tightest upper bound). Now, if the graph G is bipartite, then problem (11) can be naturally viewed as a *bipartite transportation problem* [2] and can be efficiently solved using the *network simplex method*, which is a strongly polynomial algorithm [2]. If G is not bipartite, then we can translate problem (11) into the following bipartite transportation problem

$$\text{minimize} \qquad \frac{1}{2} \sum_{j=1,\ldots,m-l} c_j \, (y_j + y_{m+j}) + \sum_{j=m-l+1,\ldots,m} c_j \, y_j \qquad (12)$$

$$\text{subject to} \qquad \mathbf{H}\,\mathbf{y} = \mathbf{v},\ \mathbf{y} \geq \mathbf{0}$$

It is easily seen that, if **x** is a nonnegative solutions of problem (11) and **y** is the nonnegative solution of problem (12) associated with **x**, then

$$\sum_{j=1,\ldots,m} c_j \, x_j = \frac{1}{2} \sum_{j=1,\ldots,m-l} c_j \, (y_j + y_{m+j}) + \sum_{j=m-l+1,\ldots,m} c_j \, y_j$$

and *vice versa*. So, every optimal solution of problem (11) corresponds to an optimal solution of problem (12), and *vice versa*, and the minima of problems (11) and (12) do coincide.

Before closing this section, it is worth consider the special case that the components of the vector **w** in system (9) are all nonnegative integers. If G is bipartite then, by the integrality theorem and by the total unimodularity [2, 7, 23] of the incidence matrix of G, problem (11) has an integral optimal solution. If G is not bipartite then, since problem (12) has an integral optimal solution, problem (11) has an optimal solution whose components are either integers or half-integers.

8 Conclusions

In order to protect the confidentiality of individual data, the query system of a statistical database should be sure that no confidential piece of information runs the risk of being disclosed in an exact or approximate way from responses to sum-queries. To achieve this, the query system should audit sum-queries and issue non-informative answers to sum-queries that directly or indirectly would lead to the disclosure of confidential data. We proposed a compact representation of the information conveyed by the responses to answered sum-queries, we called an information model in *normal form*, and proposed an answering procedure, which makes a parsimonious use of standard linear-programming methods. We also showed that, if the information model is graphical, then standard linear-programming methods can be avoided at all and the answering procedure can be implemented in polynomial time using algebraic and network algorithms. This suggests a further security measure which is a sort of query-overlap restriction [1], which loosely speaking refuses releasing the exact value of a sum-query Q if Q is not evaluable from previously answered sum-queries and the target of Q overlaps "too much" with the targets of answered sum-queries. If this is the case, then Q will be answered as being an intrusive sum-query, that is, the response to Q will be the range of its feasible values of Q.
Possible directions of future research are:

auditing sum-queries with a response variable that is of a general additive type (e.g., an Abelian group [3, 4, 19]), or of a specific type (e.g., a nonnegative integer type or a binary type [13]);

auditing max- or min-queries, by relaxing some restrictive assumptions such as the individual values of the response variable are all distinct [5] and there is a single tuple falling in every sensitive category [5, 13].

Finally, note that auditing count-queries requires solving the same integer linear-programming problems as auditing sum-queries with a response variable of a nonnegative integer type.

ACKNOWLEDGEMENTS

The authors thank the anonymous referees whose valuable comments helped improve the presentation of the work.

The work was funded by the MIUR under the project PRIN 2003 "WEB-based management and representation of spatial and geographic data".

APPENDIX

Lemma 1 [15] Let $\mathbf{A}\,\mathbf{x} = \mathbf{q}$ be a system of n linear equations with m unknowns, such that the set $\mathbf{X}$ of its nonnegative solutions is not empty. If c is the maximum value of the linear function $\sum_{j=1,\ldots,m} u_j x_j$ over $\mathbf{X}$, then there exist n real numbers $\lambda_1, \ldots, \lambda_n$, and m nonnegative real numbers $v_1, \ldots, v_m$ such that

(*i*) $u_j = \sum_{i=1,\ldots,n} \lambda_i a_{ij} - v_j \qquad (j = 1, \ldots, m)$

(*ii*) $c = \sum_{i=1,\ldots,n} \lambda_i q_i$.

Proof. First of all, we re-write the constrained equation system $\mathbf{A}\,\mathbf{x} = \mathbf{q}$, $\mathbf{x} \geq \mathbf{0}$ as a system of linear inequalities: $\mathbf{A}\,\mathbf{x} \leq \mathbf{q}$, $-\mathbf{A}\,\mathbf{x} \leq -\mathbf{q}$, $-\mathbf{x} \leq \mathbf{0}$, whose coefficient matrix $\mathbf{H}$ and constant vector $\mathbf{b}$ are

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} \\ -\mathbf{A} \\ -\mathbf{0} \end{bmatrix} \qquad \mathbf{b} = \begin{bmatrix} \mathbf{q} \\ -\mathbf{q} \\ -\mathbf{0} \end{bmatrix}.$$

By the Affine Form of Farkas's Lemma, there exist $2n+m$ real nonnegative numbers $y_1, \ldots, y_{2n+m}$ such that

(*i*) $u_j = \sum_{i=1,\ldots,2n+m} y_i h_{ij} = \sum_{i=1,\ldots,n} y_i a_{ij} - \sum_{i=1,\ldots,n} y_{n+i} a_{ij} - y_{2n+j}$

$= \sum_{i=1,\ldots,n} (y_i - y_{n+i}) a_{ij} - y_{2n+j} \qquad (j = 1, \ldots, m)$

(*ii*) $c = \sum_{i=1,\ldots,2n+m} y_i b_i = \sum_{i=1,\ldots,n} y_i q_i - \sum_{i=1,\ldots,n} y_{n+i} q_i = \sum_{i=1,\ldots,n} (y_i - y_{n+i}) q_i$

The statement follows from setting $\lambda_i = y_i - y_{n+i}$, $1 \leq i \leq n$, and $v_j = y_{2n+j}$, $1 \leq j \leq m$. □

References

1. Adam, N.R., Wortmann, J.C.: Security control methods for statistical databases: a comparative study. *ACM Computing Surveys* **21** (1989) 515-556.

2. Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: *Network Flows*. Prentice Hall, Englewood Cliffs, 1993.

3. Chang Chen, M., L. McNamee, L.: A model of summary data and its applications to statistical databases, Proc. IV Int. Conf. on "Statistical & Scientific Database Management" 1988, (G. Goos and J. Hatmanis, eds.), *Lecture Notes in Computer Sciences* **339** (1989) 354-372.

4. Chang Chen, M., L. McNamee, L., Melkanoff, M.: On the data model and access method of summary data management. *IEEE Trans. on Knowledge and Data Engineering* **1** (1989) 519-529.

5. Chin, F.: Security problems on inference control for SUM, MAX, and MIN queries. *J. ACM* **33** (1986) 451-464.

6. Chin, F.Y., Ozsoyoglu, G.: Auditing and inference control in statistical databases. *IEEE Trans. on Software Engineering* **8** (1982) 574-582.

7. Chvátal, V.: *Linear Programming*. Freeman, New York, 1983.

8. Cox, L.H.: Suppression methodology and statistical disclosure control. *J. American Statistical Association* **75** (1980) 377-385.

9. Cox, L.H., Zayatz, L.V.: An agenda for research on statistical disclosure limitation. *J. Official Statistics* **11** (1995) 205-220.

10. Dobra, A., Fienberg, S.E.: Bounds for cell entries in contingency tables given the marginal totals and decomposable graphs, *Proceedings of the National Academy of Sciences of the United States of America* **97** (2000) 11885-11892.

11. Duncan, G.T., Fienberg, S.E., Krishnan, R., Padman, R., Roehrig, S.F.: Disclosure limitation methods and information loss for tabular data, in *Confidentiality*, *Disclosure and Data Access* (Doyle, P., Lane, J., Theeuwes, J., Zayatz, L., eds.), Elsevier (2001), 135-166.

12. Gusfield, D.: A graph-theoretic approach to statistical data security. *SIAM J. on Computing* **17** (1988) 552-571.

13. Kleinberg, J.M., Papadimitriou, C.H., Raghavan, P.: Auditing Boolean attributes. Proc. XIX ACM Symp. on "Principles of Database Systems" (2000) 86-91.

14. Maier, D.: *The Theory of Relational Databases*. Computer Science Press, Rockville, 1983.

15. Malvestuto, F.M.: A universal-schema approach to statistical databases containing homogeneous summary tables. *ACM Trans. on Database Systems* **18** (1993) 678-708.

16. Malvestuto, F.M., Mezzini, M.: On the hardness of protecting sensitive information in a statistical database. Proc. World Multiconference on "Systemics, Cybernetics and Informatics", vol. XIV (2001) 504-509.

17. Malvestuto, F.M., Mezzini, M.: A linear algorithm for finding the invariant edges of an edge- weighted graph. *SIAM J. on Computing* **31** (2002) 1438-1455.

18. Malvestuto, F.M., Mezzini, M.: Auditing sum-queries. Proc. International Conference on "Database Theory" (2003) 504-509, Lecture Notes in Computer Sciences.

19. Malvestuto, F.M., Mezzini, M.: Privacy preserving and data mining in an on-line statistical database of additive type. Proc. International Conference on "Privacy in Statistical Databases" (2004), Barcelona.

20. Malvestuto, F.M., Moscarini, M.: Query evaluability in statistical databases. *IEEE Transactions on Knowledge and Data Engineering* **2** (1990) 425-430.

21. Malvestuto, F.M., Moscarini, M.: An audit expert for large statistical databases. In Statistical Data Protection. EUROSTAT (1999) 29-43.

22. Malvestuto, F.M., Moscarini, M.: Privacy in multidimensional databases, in *Multidimensional Databases* (Rafanelli, M., editor), Idea Group Pub. (2003), Hershey, USA, 310-360.

23. Schrijver, A.: *Theory of Linear and Integer Programming*. Wiley, New York, 1986.

24. Wang, L., Wijekera, D., Jajodia, S.: Cardinality-based inference control in sum-only data cubes. Proc. European Symposium on "Computer Security" (ESORICS 2002). Lecture Notes in Computer Science, Vol. 2502. Springer-Verlag, New York (2002), 55-71.

25. Wang, L., Wijekera, D., Jajodia, S.: Cardinality-based inference control in datacubes. *J. of Computer Security* **12** (2004) 655-692.

26. Willenborg, L., de Waal, T.: *Statistical Disclosure Control in Practice*. Lecture Notes in Statistics, Vol. 111. Springer-Verlag, New York (1996).

27. Willenborg, L., de Waal, T.: *Elements of Statistical Disclosure*. Lecture Notes in Statistics **155** (2000). Springer-Verlag, New York.

28. Zhang, N., Zhao, W., Chen, J.: Cardinality-based inference control in OLAP systems: an information theoretic approach. Proc. ACM Int. Workshop on "Data Warehousing and OLAP" (DOLAP 2004), 59-64.