12-1-1999

Disclosure Detection in Multivariate Categorical Databases: Auditing Confidentiality Protection Through Two New Matrix Operators

Sumit Dutta Chowdhury

George T. Duncan
Carnegie Mellon University

Ramayya Krishnan
Carnegie Mellon University

Stephen F. Roehrig
Carnegie Mellon University

Sumitra Mukherjee
Nova Southeastern University, sumitra@nova.edu

NSUWorks Citation

Dutta Chowdhury, Sumit; Duncan, George T.; Krishnan, Ramayya; Roehrig, Stephen F.; and Mukherjee, Sumitra, "Disclosure Detection in Multivariate Categorical Databases: Auditing Confidentiality Protection Through Two New Matrix Operators" (1999). *CEC Faculty Articles*. 13.
https://nsuworks.nova.edu/gscis_facarticles/13

Management Science

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Disclosure Detection in Multivariate Categorical Databases: Auditing Confidentiality Protection Through Two New Matrix Operators

Sumit Dutta Chowdhury, George T. Duncan, Ramayya Krishnan, Stephen F. Roehrig, Sumitra Mukherjee,

To cite this article:
Sumit Dutta Chowdhury, George T. Duncan, Ramayya Krishnan, Stephen F. Roehrig, Sumitra Mukherjee, (1999) Disclosure Detection in Multivariate Categorical Databases: Auditing Confidentiality Protection Through Two New Matrix Operators. Management Science 45(12):1710-1723. http://dx.doi.org/10.1287/mnsc.45.12.1710

Disclosure Detection in Multivariate Categorical Databases: Auditing Confidentiality Protection Through Two New Matrix Operators

Sumit Dutta Chowdhury • George T. Duncan • Ramayya Krishnan • Stephen F. Roehrig • Sumitra Mukherjee

605 West View Terrace, Alexandria, Virginia 22301
The H. John Heinz III School of Public Policy and Management, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213
The H. John Heinz III School of Public Policy and Management, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213
The H. John Heinz III School of Public Policy and Management, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213
School of Computer and Information Systems, Nova University, Fort Lauderdale, Florida 33315
schowdhury@kpmg.com • gd17@andrew.cmu.edu • rk2x@andrew.cmu.edu • roehrig@andrew.cmu.edu • sumitra@scis.acast.nova.edu

As databases grow more prevalent and comprehensive, database administrators seek to limit disclosure of confidential information while still providing access to data. Practical databases accommodate users with heterogeneous needs for access. Each class of data user is accorded access to only certain views. Other views are considered confidential, and hence to be protected. Using illustrations from health care and education, this article addresses inferential disclosure of confidential views in multidimensional categorical databases. It demonstrates that any structural, so data-value-independent method for detecting disclosure can fail. Consistent with previous work for two-way tables, it presents a data-value-dependent method to obtain tight lower and upper bounds for confidential data values. For two-dimensional projections of categorical databases, it exploits the network structure of a linear programming (LP) formulation to develop two transportation flow algorithms that are both computationally efficient and insightful. These algorithms can be easily implemented through two new matrix operators, *cell-maxima* and *cell-minima*. Collectively, this method is called matrix comparative assignment (MCA). Finally, it extends both the LP and MCA approaches to inferential disclosure when accessible views have been masked.

(*Confidentiality*; *Data Access*; *Linear Programming*; *Matrix Methods*; *Disclosure Risk*; *Network Models*; *Disclosure Limitation*)

Management Science © 1999 INFORMS
Vol. 45, No. 12, December 1999 pp. 1710–1723

0025-1909/99/4512/1710$05.00
1526-5501 electronic ISSN

1. Introduction

Database administrators implement policies and technologies to limit disclosure of confidential information while providing access to legitimate information (Schlörer 1975, Duncan and Lambert 1986, Adam and Wortman 1989). A "data snooper" must not obtain, directly or through inference, knowledge of the confidential data. Direct disclosure occurs with unauthorized access, as through password breaking or communication eavesdropping. Methods such as multilevel authorization control, password protection, and encryption help prevent direct disclosure and are not our concern here. Rather we seek to protect against *inferential disclosure* (Denning 1980), thereby providing disclosure protection against data snoopers who have access to the database, but lack authorization to every aspect of it.

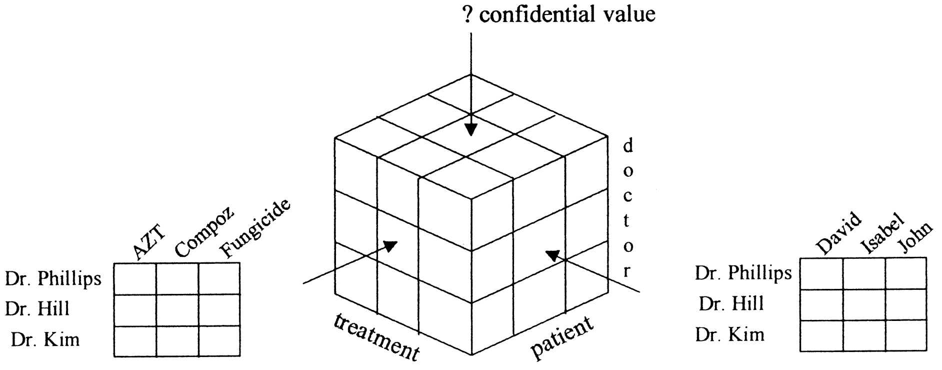
Inferential disclosure is harder to control than direct disclosure. In inferential disclosure the data snooper uses legitimately accessible information to infer confidential information. To guard against inferential disclosure, the database administrator must assess the vulnerability of a database. Disclosure detection techniques can be applied both to a database in its original form or to a database that has been transformed to limit disclosure. Thus disclosure detection—"to distinguish safe from unsafe data" (Willenborg and de Waal 1996, p. vii)—is an essential component of any strategy for database security.

For research and statistical purposes, the most common products disseminated from databases are tables. Previous works have addressed the case of disclosure detection and protection in two-dimensional tables (Cox 1980, Carvalho et al. 1994, Muralidhar et al. 1995). We focus on multidimensional categorical databases, which are N-dimensional tables with each dimension categorical. Cell entries may be counts or other numerical values. Motivated by U.S. Census Bureau surveys such as the Census of Wholesale Trade and the Census of Construction Industries, which release three-dimensional tables, Cox (1992) argues for the need to examine disclosure issues in higher-dimensional tables. In the commercial sector, a variety of tools provide multidimensional views of relational data in data warehousing (Barquin and Edelstein 1997, p. 174). With the increasing use of data warehousing, security concerns in multidimensional tables have become significant.

Our concern is disclosure detection for *linked tables* (i.e., tables that share a common attribute), as discussed in Willenborg and de Waal (1996, pp. 108–111, 130–134) and De Vries (1993). We show that disclosure detection by a purely structural approach, i.e., based on database design alone, can give a false sense of security for linked tables. As proof, we give a method, based on linear programming, to infer bounds on the values of confidential views. Inference is based solely on the actual contents of the accessible views of the database. Beyond showing the inadequacies of a structural approach, this technique provides a systematic method that the database administrator can use for disclosure detection. Linear programming methods have been used in various aspects of confidentiality research, for instance by Kelly et al. (1990), Sande (1984), Cox (1987), and Zayatz (1992).

LP methods are flexible in their application to disclosure detection problems, since additional, possibly external, information can be accommodated in the form of constraints. They do, however, have two general shortcomings. First, LP methods are computationally intensive. This is evidently a problem for a large database. It is also a problem for a dynamic database. With increasingly sophisticated real-time data capture methods, more and more databases are dynamic. Disclosure detection methods must depend on the actual contents of the database, and so disclosure detection must be implemented each time the database changes. As a related point, Gusfield (1988) noted that in actual systems concerned with statistical security, a disclosure detection algorithm is typically part of the inner loop of a larger program that repeatedly modifies tables in attempts to eliminate disclosure. Thus any inefficiencies in the detection stage are magnified many times. The outer loop typically works through imposition of a variety of heuristic procedures. Their efficacy is then checked through the disclosure detection inner loop. The second shortcoming of an LP algorithm is that it yields little insight into the causes of a potential disclosure, and hence does not suggest how to transform a database to limit disclosure.

Figure 1 **Information Structure of the Health Care Example**



To address these two shortcomings of the LP method, we derive a simple and efficient matrix method that gives identical results in an important special case. This method, which we call *matrix comparative assignment* (MCA), exploits the network structure of the LP model when the accessible tables share a common attribute. MCA is fast enough to be used with large and dynamic databases (it can become the fast inner loop of a comprehensive statistical data security system), and its structure suggests efficient disclosure limitation transformations.

An Example From Health Care Management. The rapid introduction of information technology into health care management has raised sensitivities to confidentiality issues (Duncan 1997). Figure 1, a prototypical three-dimensional categorical database of medical data shows the thrust of our work. The table records the number of patients visiting physicians to receive treatments. The Patient-Doctor and Doctor-Treatment tables, which can be obtained by additive projection, are not sensitive and are publicly accessible. The Patient-Treatment table is sensitive and, so, confidential. Disclosure detection addresses to what extent a data snooper can infer cell entry values for the Patient-Treatment table.

2. A Structural Approach for Disclosure Detection

In a structural approach, a database administrator attempts to detect disclosure potential at the time of database design. Any structural method must be valid irrespective of the specific numerical values in the database. Fellegi (1972) gave a general method for determining whether a published table (or set of tables) will admit inferential disclosure. He derived a set of equations that, if solvable, yield the value of a confidential datum in terms of the data contained in the released table(s). The following results follow directly from Fellegi's work and the fact that the nonconfidential projections give linear equations that underconstrain the cell values in the underlying table.

Proposition 1. *Let $T = X \times Y \times Z$ be a table with three dimensions X, Y, and Z. It is not possible in general to determine the cell entries in the projection $X \times Z$ given only entries in the projections $X \times Y$ and $Y \times Z$.*

Corollary 1. *It is not possible, in general, to uniquely identify an N-dimensional table given any collection of projections of dimension $N - 1$ or lower.*

This proposition and its corollary show that no algorithm exists that is guaranteed to yield exact values of confidential data from related projections. This might seem reassuring to the database administrator. We next provide methods, however, by which a data snooper can get exact or bounded information about restricted views given the *contents* of accessible views of the database. The bounds depend on the cell values of the accessible tables. In many situations the bounds obtained are such that sensitive information would be disclosed. Thus the structural test would not alert a database administrator to the possibility of inferential disclosure.

3. Content-Based Approaches for Disclosure Detection

In contrast to structural approaches, content-based approaches for disclosure detection depend on the actual data values of the accessible tables. We model how inferential disclosure can take place when data are combined from a set of accessible tables. To do this we find the maximum and minimum values that can be assumed by each individual cell of the restricted view.

The well-known method of Fréchet bounds (Fréchet 1940) may be used to find weak minimum and maximum values for sensitive cells. These bounds depend only on the marginal totals, and thus do not take advantage of the information available to the data snooper. Some refinements of this approach are given by Fienberg (1998). As we show next, tight bounds are achieved when the full two-dimensional tables $X \times Y$ and $Y \times Z$ are used. This improvement also extends to the case where the underlying database is an N-dimensional categorical database.

Figure 2 An Example of Nonsensitive Tables

	D1	D2	D3	
P1	14	1	8	23
P2	2	7	1	10
P3	5	2	4	11
	21	10	13	44

	T1	T2	T3	
D1	8	12	1	21
D2	0	9	1	10
D3	4	7	2	13
	12	28	4	44

Figure 3 Results of LP Analysis

$PT(j, k)$	T1	T2	T3
P1	(1, 12)	(7, 20)	(0, 4)
P2	(0, 3)	(6, 10)	(0, 3)
P3	(0, 9)	(1, 11)	(0, 4)

Linear Programming Formulation

Given the contents of the accessible tables, linear programming can be used to calculate the maximum and minimum values possible for the entries in the confidential table. Mathematical optimization methods are used to assess upper and lower possible values in confidential two-way tables by Cox (1987) and Sande (1984). The upper and lower bounds for a sensitive cell are called the *ambiguity width* by Robertson (1994) and *feasibility interval* by Willenborg and de Waal (1996, p. 101). Some problems in three-dimensional tables are addressed by linear programming methodology in Lougee-Heimer (1989). We extend this approach to an N-dimensional problem. We view each cell value in each accessible table as the right-hand side of a linear programming equality constraint. As suggested by Figure 1, each projection cell is a sum of cell values in the underlying, full-dimensional table. Thus, the left-hand side of the corresponding constraint is just that sum. Similarly, a cell in the confidential table is a sum of cell values in the full-dimensional table. We take this sum for the confidential cell to be an objective function. The idea is to both maximize and minimize this objective function subject to the constraints generated by the values of the accessible tables. This two-sided optimization procedure produces bounds for each cell in the confidential table. By the optimality of LP solution procedures, these bounds are tight and so cannot be improved without additional information.

If the maximum and minimum values of some entry in the confidential table are equal, then that table entry is uniquely identified. Such a result evidences a disclosure. Even with different upper and lower bounds, the bounds themselves might suggest unacceptable disclosure. For example, the nonzero lower bounds in our medical care example may constitute disclosure.

Suppose a Patient-Doctor table and a Doctor-Treatment table are available as in Figure 2. Using the LP approach we obtain the (lower, upper) bounds shown in Figure 3. In the Patient-Treatment (PT) table, a disclosure has taken place for four out of the nine entries, since the minimum is greater than 0. Thus a snooper might get considerable information from these accessible tables. Note that the solutions obtained by the model are integer. This is not happenstance, as we demonstrate in §4.

4. Two-Dimensional Views: Networks and Matrices

In the common case where two-dimensional views of a three-dimensional table are made accessible, the linear programming formulation has a network structure that can be exploited to develop a simple and efficient procedure. Cox (1992, 1995) has exploited a network structure for a two-dimensional problem. Ernst (1989) demonstrated some general problems with network formulations when $N \geq 3$. (See also Cox 1980, 1987, Cox et al. 1986, Gusfield 1988, Sullivan and Zayatz 1991, Rowe 1991.) Cox (1992) identifies the case of $N \geq 3$ as an important research problem. We use the health care management example introduced in §1 with $N = 3$ to illustrate the network structure of the problem where two-dimensional views are made accessible.

Network Formulation of the Health Care Example. The health care example can be recast as a collection of smaller problems, each with a special network structure. Using this insight, we develop simple solution procedures that are compactly expressed as matrix operators on the two-dimensional, accessible views.

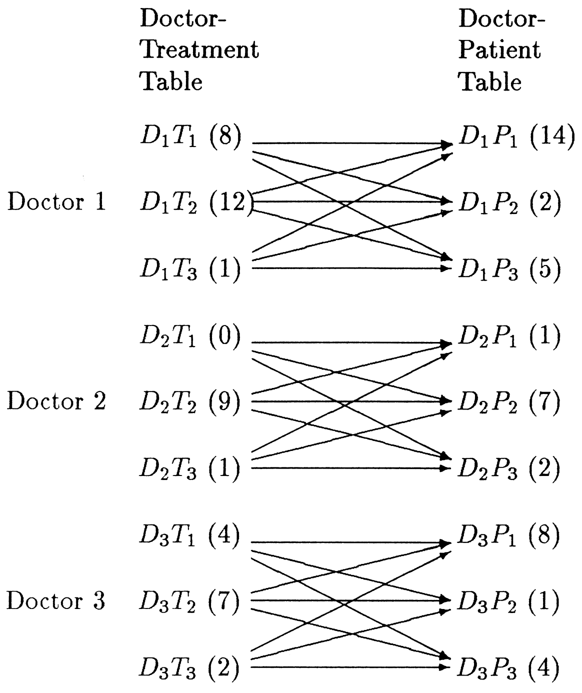
Denote the underlying three-dimensional table by T_{ijk} where $i = 1, \ldots, I$, $j = 1, \ldots, J$, $k = 1, \ldots, K$. The public Patient-Doctor table is the projection T_{ij+} and the public Doctor-Treatment table is the projection T_{+jk}, while the confidential Patient-Treatment table is the projection T_{i+k}. Each of the J constraint pairs of the first and second projections is a node-arc incidence matrix, and all variables are integer-valued. Hence the optimization problem may be decomposed into J independent transportation problems (Chvátal 1983), where for each j,

1. there are K source nodes with supply T_{+jk};
2. there are I destination nodes with demand T_{ij+}; and
3. each variable T_{ijk} represents an arc from source node k to destination node i.

Figure 4 depicts the resulting network for the 3 × 3 × 3 health care example. It consists of three (J) separate, bipartite, fully connected subnetworks. Each of these subnetworks is associated with a different doctor (j), and can be thought of as representing the "flows" of treatments from a particular doctor to each of the patients (i). Each subnetwork has three (K) source nodes and three (I) destination nodes. These correspond to the three rows of the Doctor-Treatment table and the three columns of the Patient-Doctor table as shown in §3. Finally, in each subnetwork the sum of the "supplies" available at each source node is equal to the sum of the "demands" at each destination node. This follows from the fact that the row sums of the Doctor-Treatment table equal the column sums of the Patient-Doctor table. The following definitions and observations generalize this insight.

Definition 1. A network is *densely connected* (DC) iff (a) it is a directed, bipartite graph; (b) the source nodes are in one partition and the destination nodes are in the other; (c) each source node is connected to every destination node; and (d) the sum of the supplies in the source partition is equal to the sum of the demands in the destination partition.

Figure 4 Network Structure of the LP



Note. Example cell entries in parentheses.

Observation 1. Suppose the constraint equations arising from the projection T_{+jk} are multiplied by -1. Then the constraints of our problem can be represented by a collection of DC subnetworks. Specifically, (a) each constraint in the LP is a node in a subnetwork, (b) if the right-hand side of the constraint is positive then it is a source node and if it is negative then it is a destination node; (c) each variable is represented by an arc whose source is a constraint in which its coefficient in the LP matrix is $+1$ and whose destination is a constraint in which its coefficient is -1. Since each variable is present in only two constraints, each arc has a unique source and unique destination in the subnetwork.

Observation 2. The objective function of the linear programming formulation is a linear combination of as many variables as there are subnetworks. Further, since each variable in the objective function appears in only one subnetwork, the linear program can be decomposed into separate single variable optimization problems.

This is illustrated in Figure 4, where each doctor defines an optimization problem.

Observation 3. The optimal objective function value of the linear program is the sum of the optimal values of each subnetwork.

We next present two procedures, VAP-1 and VAP-2 (VAP stands for value assignment procedure), that compute the maximum and minimum flows in a densely connected network; then we prove the optimality of these procedures.

```
Procedure VAP-1(arc, network)\\compute max flow on arc in a
DC network
    S := supply(arc, network)
    D := demand(arc, network)
    MaxV := min(S, D)
    return MaxV
End VAP-1
```

To illustrate VAP-1, consider the subnetwork for Doctor 1 at the top of Figure 4. VAP-1 gives the maximum flow on the arc joining D_1T_1 and D_1P_1 as the minimum of D_1T_1 and D_1P_1.

Proposition 2. *VAP-1 computes the maximum flow on an arc in a densely connected network.*

Proof. Follows directly from the bipartite and fully connected nature of the network. If the supply S exceeds the demand D, the maximum possible flow is D, since no arc leaves the destination node. A flow of D is feasible since the sum of the network supplies equals the sum of the demands. If $D \geq S$, a flow of S is the maximum possible, and is feasible, once again because the sum of supplies equals the sum of demands. □

Corollary 2. *Let $S_1, \ldots, S_J$ be the J densely connected networks equivalent to the constraint matrix of the linear programming formulation. Applying* VAP-1 *to each of these J instances and summing the result gives the maximum value of the objective function of the LP.*

Proof. This follows directly from the decomposability of the overall network. □

Similarly, we define an algorithm that determines the minimum flow on an arc in a densely connected network.

```
Procedure VAP-2(arc, network) \\ compute min flow on an arc in
a DC network
    i := index of destination node of arc
    S := supply(arc, network)
    D[j] := demand of destination node j
    minV := S
    for j ≠ i
        minV := minV − D[j]
    if minV > 0
        return minV
    else
        return 0
End VAP-2
```

Referring again to Figure 4, the minimum flow along the arc joining D_1T_1 and D_1P_1 can be determined by VAP-2. Since the supply at D_1T_1 cannot be completely consumed by destination nodes D_1P_2 and D_1P_3, the minimum flow is $8 - (2 + 5) = 1$.

Proposition 3. *VAP-2 computes the minimum flow on an arc in a densely connected network.*

Proof. The flow given by the procedure is clearly feasible, since all the remaining source nodes are free to supply the sink node of the given arc. It is also the minimum flow, since if the demands of all other nodes have been met, the remainder must be directed to the arc's sink node. □

Corollary 3. *Let $S_1 \cdots S_J$ be the J densely connected networks equivalent to the constraint matrix of the linear programming formulation P. Applying* VAP-2 *to each of these J instances and summing gives the minimum value of the objective function of the LP.*

Proof. Again, this follows directly from the decomposability of the network. □

In this section, we have shown that the linear programming formulation can be interpreted as a collection of network flow problems, each having an especially simple solution. While disclosure detection via linear programming can, at least theoretically, be done in polynomial time, the new algorithms given here are attractive because they eliminate the need for translation of the detection problem to LP format, and can, as we will demonstrate further, considerably reduce the computational burden.

The Matrix Comparative Assignment Approach. The logic embedded in procedures VAP-1 and VAP-2 can be recast as simple, but original, matrix operations. Consider the basic operation used in the VAP-1 procedure. In each subnetwork the minimum of the supply value and demand value associated respectively with the source and destination of the variable being maximized is computed. The value of the objective function for the LP (by Corollary 2) that determines the maximum value of a cell in the confidential table is the sum of the values returned by VAP-1 applied to each subnetwork structure. The maximum value that a cell in the confidential table can take is the sum of the minima of "supplies" and "demands" of subnetwork structures. These supplies and demands are simply cell values of the accessible tables. This insight (and a complementary one for VAP-2) suggests two new matrix operators, Cell-Maxima $\bar{\otimes}$ and Cell-Minima $\underline{\otimes}$. They encode the logic embedded in VAP-1 and VAP-2. These matrix operations are not only straightforward and intuitively appealing, but they are fast to compute and yield insights to disclosure limitation through cell suppression.

Definition 2. Let $A = [a_{ij}]$ be an $I \times J$ matrix and $B = [b_{jk}]$ be a $J \times K$ matrix. The Cell-Maxima operator $\bar{\otimes}$ is a binary operator on (A, B) that yields an $I \times K$ matrix C^U defined by ik entries

$$C^U_{ik} = \sum_j \min(a_{ij}, b_{jk}),$$

for $i = 1, \ldots, I$, $k = 1, \ldots, K$.

Note the analogy with ordinary matrix multiplication: For the cell-maxima operator we use the sum of the minima rather than the sum of the products. Note also that the cell values a_{ij}, b_{jk}, which are arguments to the minima operator, are just those supplies and demands referred to in the VAP-1 procedure.

Definition 3. With the same conditions as in Definition 2, the Cell-Minima operator $\underline{\otimes}$ *is a binary operator on* (A, B) that yields a matrix C^L of dimension $I \times K$ defined by ik entries

$$C^L_{ik} = \sum_j \left[a_{ij} - \sum_{p \neq k} b_{jp} \right]^+$$

for $i = 1, \ldots, I$, $k = 1, \ldots, K$, where $[\cdot]^+$ is the maximum of zero and the argument.

The cell-minima operation is the logic encoded in VAP-2 to compute the surplus supply, if any, at the source node of the variable being minimized. Since the cell-max operation is the same as VAP-1 and the cell-min operation is the same as VAP-2, they yield the same bounds as the LP approach.

Applying the two matrix operators defined above to the medical example, we find

$$\begin{aligned} PT^U &= PD \;\bar{\otimes}\; DT \\ &= \begin{bmatrix} 14 & 1 & 8 \\ 2 & 7 & 1 \\ 5 & 2 & 4 \end{bmatrix} \bar{\otimes} \begin{bmatrix} 8 & 12 & 1 \\ 0 & 9 & 1 \\ 4 & 7 & 2 \end{bmatrix} \\ &= \begin{bmatrix} 12 & 20 & 4 \\ 3 & 10 & 3 \\ 9 & 11 & 4 \end{bmatrix}, \end{aligned}$$

$$\begin{aligned} PT^L &= PD \;\underline{\otimes}\; DT \\ &= \begin{bmatrix} 14 & 1 & 8 \\ 2 & 7 & 1 \\ 5 & 2 & 4 \end{bmatrix} \underline{\otimes} \begin{bmatrix} 8 & 12 & 1 \\ 0 & 9 & 1 \\ 4 & 7 & 2 \end{bmatrix} \\ &= \begin{bmatrix} 1 & 7 & 0 \\ 0 & 6 & 0 \\ 0 & 1 & 0 \end{bmatrix}. \end{aligned}$$

We note that the cell-minima operator can be expressed in terms of the cell-maxima operator and standard matrix operators. Specifically,

$$A \underline{\otimes} B = AE - A \bar{\otimes} (B(E - I)), \qquad (1)$$

where E is a matrix of all ones and I is the identity matrix, each of appropriate dimensionality.

Using these operators we develop the *matrix comparative assignment* (MCA) algorithm to find the bounding matrices for a confidential table. This algorithm is given for two-dimensional accessible tables, but may be extended to higher dimensions under certain conditions.

Algorithm MCA.

1. Identify two jointly confidential attributes, i and k, for example (Patient and Treatment). Identify all accessible tables which have one or the other of these attributes (Patient-Doctor and Doctor-Treatment). From that set, choose a pair of tables which have a nonconfidential attribute in common, for example, $R(i, j)$ and $R(j, k)$.

2. Find $x_j^U(i, k) = R(i, j) \bar{\otimes} R(j, k)$ and $x_j^L(i, j) = R(i, j) \underline{\otimes} R(j, k)$. These are the upper and lower bounds for $x(i, k)$ obtained through j.

3. Repeat Step 2 for each available j.

4. The tightest MCA bounds for $x(i, k)$, denoted $x^U(i, k)$ and $x^L(i, k)$, are given by

$$x^U(i, k) = \min_j x_j^U(i, k), \qquad x^L(i, k) = \max_j x_j^L(i, k).$$

Note that the MCA approach is not limited to inference over a single pair of tables. It is possible to use sequences of pairs of tables to discover bounds. For example, if the Patient-Doctor table is not available, but a Patient-Condition table and Condition-Doctor table are, a snooper can use the latter two to arrive at bounds for the Patient-Doctor table. This, coupled with a Doctor-Treatment table, give bounds for the Patient-Treatment table. However, the resultant bounds for this indirect approach may be considerably looser than can be achieved by more direct means. That there can be no improvement by the indirect method is substantiated by Proposition 4, whose proof is straightforward given that $(PD \bar{\otimes} DC) \bar{\otimes} CT \geq PC \bar{\otimes} CT$ for any CT of appropriate dimensionality.

Proposition 4. *MCA implemented indirectly through an intermediate table cannot produce tighter bounds than direct MCA.*

Computational Complexity and Execution Time. Let $A = [a_{ij}]$ be an $I \times J$ matrix and $B = [b_{jk}]$, a $J \times K$ matrix. Then, by the definition of the operator $\bar{\otimes}$, the calculation of each cell of $A \bar{\otimes} B$ requires J comparisons and $J - 1$ additions. Because $A \bar{\otimes} B$ is of dimension $I \times K$, a total of IJK comparisons and $I(J - 1)K$ additions are needed for computing the upper bounds on *all* cells of the sensitive table. A similar analysis shows that the operation $A \underline{\otimes} B$ requires $J(K - 1)$ additions/subtractions and J comparisons for each sensitive cell, giving a total of $IJ(K - 1)K$ additions/subtractions and IJK comparisons for the entire sensitive table.

As a concrete example, using a 300 MHz Pentium II and code written in C++, MCA requires less than a second to compute all 62,500 upper bounds for a 250 × 250 sensitive table. Less than a minute is required to compute the same number of lower bounds, using the algorithm in Definition 3. Depending on the speed of multiplication relative to addition/subtraction (which is hardware dependent), the formulation in Equation (1) may be considerably faster. This shows that the matrix operators are quite fast, suitable in fact for most dynamic database applications with very large tables.

Extensions to Higher Dimensions. The MCA approach is extensible to higher dimensions under certain conditions. For example, if the two tables, Patient-Doctor-Symptom and Doctor-Symptom-Treatment, were available, then a three-dimensional analog of MCA could compute bounds on the confidential Patient-Treatment table. Intuitively, the maximum number of treatments T_1 that patient P_1 could have had, for example, is the sum over all doctors and *symptoms* of the minimum of the number of times P_1 visited a doctor with that symptom and the number of times that doctor prescribed T_1 for that symptom. Since in this case we have that the number of patients seen by doctor j with symptom k must equal the number of treatments given by doctor j for symptom

k, this upper bound is feasible. Similar reasoning yields a higher-dimensional version of the lower-bounding operator.

More generally, we can define the following.

Definition 4. Let $A = [a_{ijk}]$ be an $I \times J \times K$ array and $B = [b_{jkl}]$ be a $J \times K \times L$ array. The Cell-Maxima operator $\bar{\otimes}$ is a binary operator on (A, B) that yields an $I \times L$ matrix C^U defined by il entries

$$C^U_{il} = \sum_{j,k} \min(a_{ijk}, b_{jkl}),$$

for $i = 1, \ldots, I$, $l = 1, \ldots, L$.

Definition 5. With the same conditions as in Definition 4, the Cell-Minima operator $\underline{\otimes}$ is a binary operator on (A, B) that yields a matrix C^L of dimension $I \times L$ defined by il entries

$$C^L_{il} = \sum_{j,k} \left[a_{ijk} - \sum_{p \neq l} b_{jkp} \right]^+,$$

for $i = 1, \ldots, I$, $l = 1, \ldots, L$.

These definitions clearly extend to higher dimensions, and allow us to characterize those situations in which MCA is applicable. Let T be an N-dimensional table with two attributes i and l which, if appearing together in any projection, constitute a disclosure. Then we may apply the MCA algorithm to any two projections A and B (of dimension $N - 1$ or lower) of T provided i is in A but not B, l is in B but not A, and all other attributes appearing in A also appear in B.

Likelihood of Disclosure. The Matrix Comparative Assignment (MCA) approach also gives insight into how often a database $[T_{ijk}]$ is vulnerable to inferential disclosure from related, but accessible, tables. In principle, this question is unanswerable since millions of databases exist, each with different characteristics. Access to a meaningful number of these databases is impractical, for confidentiality reasons among others. However, we can use plausible probability models for database entries T_{ijk} and derive suggestive results. Broadly, the models we consider are *sparse-table models* and *mixture models*. We focus our attention on one type of disclosure: lower bounds greater than zero for cell entries in the sensitive table $[T_{i+k}]$, based on the released tables $[T_{ij+}]$ and $[T_{+jk}]$.

The MCA approach provides direct evidence of when a cell lower bound is greater than zero. From the structure of the cell-minima operator $\underline{\otimes}$ as given in Definition 3, we can deduce that a nonzero disclosure will occur for cell ik when

$$X = T_{ijk} > Y = \sum_{r \neq i} \sum_{p \neq k} T_{rjp}, \quad \text{for any } j = 1, \ldots, m.$$

For simplicity, take $\{T_{ijk}\}$ to be exchangeable, and let p be the probability that $X > Y$. Further assume that the T_{ijk} are mutually independent random variables. Now j is different for each of the events defined by the above inequality so the left- and right-hand side variables are independent. Therefore the m events over j are independent. Disclosure occurs if any one of the events occur. Hence the probability of a disclosure in the ik cell is 1 minus the probability of no disclosure for any of the events, or $q = 1 - (1 - p)^m$.

Sparse-Table Models. The original database $[T_{ijk}]$ is sparse if many of the cell values are 0 and values greater than 1 never or very rarely occur. It might seem that with generally small entries, a disclosure because of a lower bound on a cell entry above zero would rarely occur. This intuition overstates. To establish this, we model T_{ijk} as independent and identically distributed Bernoulli random variables with parameter π. A cell is then 0 with probability $1 - \pi$. For simplicity, we take $I = J = m$, say. In this case, $X \sim$ Bernoulli(π) and $Y \sim$ Binomial($(m - 1)^2$, π). Then

$$P(X > Y) = P(X = 1, Y = 0) = \pi(1 - \pi)^{(m-1)^2}.$$

In general, the probability of disclosure is maximized for $\pi^* = 1/((m - 1)^2 + 1)$. For this π, the probability of nonzero disclosure for a cell and the expected number of nonzero disclosures in the table can be relatively high. As the size of the table m increases, the expected number of nonzero disclosures decreases, approaching a limiting value of $\frac{1}{e}$ (where $e = 2.718 \cdots$). Figure 5 presents results for different values of m.

Mixture Models

For tables that are not sparse, a reasonable and common probability model for the cell entries is that of independent Poisson distributions where the mean λ

Figure 5 **Probability of Disclosure in Tables of Size $m \times m \times m$ Under Sparse-Table Model**

m	2	3	5	10
π maximizing P(disclosure)	0.50	0.20	0.059	0.012
Maximum P(disclosure)	0.25	0.082	0.022	0.005
Expected number of disclosures	1.00	0.74	0.56	0.45

varies from cell to cell. We take the variation in cell means to have a probability distribution $F(\lambda)$. A standard and useful approach is to take F to be a gamma distribution with parameters α and β, for example. With λ then having mean $\alpha\beta$ and variance $\alpha\beta^2$, the mean and variance of X are $\alpha\beta$ and $\alpha\beta(\beta + 1)$, while the mean and variance of Y are $(m - 1)^2\alpha\beta$ and $(m - 1)^2\alpha\beta(\beta + 1)$. The difference $Y - X$ has mean $m(m - 2)\alpha\beta$ and variance $(m^2 - 2m + 2)\alpha\beta(\beta + 1)$. A disclosure occurs when $Y - X < 0$. Based on this model, Figure 6 gives the expected number of disclosures, based on a normal approximation for $Y - X$, in the sensitive table, for some combinations of α and β, and m between 2 and 10. In general, the expected number of disclosures is high with small values of m and with a small value of α, corresponding to a large coefficient of variation $(1/\sqrt{\alpha})$ of the gamma distribution, and is relatively insensitive to β.

Figure 6 **Expected Number of Disclosures in Tables of Size $m \times m \times m$ Under Gamma Mixture of Poisson Model**

		m			
α	β	2	3	5	10
0.1	1	2	3.4	5.2	2.4
0.2	1	2	3.0	3.1	0.3
0.5	1	2	2.3	0.9	0.0
1.0	1	2	1.5	0.1	0.0
0.1	5	2	3.0	3.2	0.3
0.2	5	2	2.5	1.4	0.0
0.5	5	2	1.6	0.2	0.0
1.0	5	2	0.9	0.0	0.0
0.1	100	2	3.0	3.1	0.3
0.2	100	2	2.5	1.3	0.0
0.5	100	2	1.5	0.1	0.0
1.0	100	2	0.8	0.0	0.0

5. Disclosure Detection for Linear Combinations

In this section, we show that bounds on linear combinations of sensitive cells can be obtained that are tighter than those obtained using aggregations of single-cell optimizations. In our previous development, determinations of upper and lower bounds were done univariately, i.e., by considering only one confidential cell at a time. In this section we extend this development to bounds on functions of more than one cell value. We show by way of an example that this more efficient estimation is indeed possible. We find bounds on linear combinations of sensitive data that are tighter than those obtainable from a univariate analysis.

The data for this example were obtained from a Carnegie Mellon University student database. Using aggregation operations which would typically be considered nonrevealing of sensitive data, this relatively large database was condensed to the tables presented here in Figure 7. Thus it is a practical example of what a serious data snooper might be able to achieve.

Two tables were generated—a Professor-Student (PS) table showing the number of times a student had taken courses with a professor and a Professor-Grade (PG) table showing the grading patterns of the professors. These tables were public information.

We calculated univariate upper and lower bounds on the grades each student could have received. From these bounds, we used a counting scheme to obtain

Figure 7 **Professor-Student and Professor-Grade Tables**

PS	S1	S2	S3	S4	S5	S6
P1	1	2	1	2	1	1
P2	2	0	2	0	0	0
P3	0	1	0	1	1	1
P4	0	1	0	1	2	2

PG	B	B+	A−	A	A+
P1	0	0	5	3	0
P2	0	0	0	2	2
P3	1	3	0	0	0
P4	0	4	0	2	0

upper and lower bounds on students' grade point averages (GPA). Specifically, a student's univariate maximum GPA was calculated by assuming that she actually received the number of A+ grades equal to the univariate maximum for A+, the number of A grades equal to that maximum, and so forth until her grade total equaled her course total. The univariate minimum was computed analogously, starting instead from B.

A multivariate estimation of GPA was calculated using the LP approach with all accessible cell values as constraints. An expression representing student GPA was used as the objective function. A comparison of the bounds in Figure 8 shows that the multivariate approach can give tighter bounds than the univariate approach.

6. Disclosure Limitation

When disclosure detection methods flag a confidentiality risk problem, data can only be released after the application of appropriate disclosure limitation methods. For any proposed limitation method, a disclosure audit should be performed. This section shows the advantages of the LP method and the MCA method in a disclosure audit of the tables protected through disclosure limitation.

Disclosure limitation methods for tabular data include rounding, random rounding (Nargundkar and Saveland 1972), controlled rounding (Fellegi 1972, Kelly et al. 1990), cell suppression (Carvalho et al. 1994), interval protection (Gopal et al. 1998), and perturbation (Duncan and Fienberg 1998). In most of these procedures, marginal totals are maintained or nearly maintained. Consider rounding to base *b* (every cell entry is rounded to the nearest integer multiple of *b*). In the LP formulation, instead of equality constraints, inequalities would be introduced. These inequalities reflect the imprecision in the snooper's knowledge of the actual values. The goal in disclosure limitation is to increase the size of the base *b* until all bounds on confidential cell entries are adequately wide. The LP approach permits a disclosure audit for each base *b*.

Figure 8 **Grade Point Average Range**

	Multivariate GPA	Univariate GPA
S1	3.40–3.85	3.40–3.85
S2	2.77–3.33	2.66–3.44
S3	3.40–3.85	3.40–3.85
S4	2.77–3.33	2.66–3.44
S5	2.66–3.33	2.55–3.44
S6	2.66–3.33	2.55–3.44

If the projection tables have been modified by cell suppression, the MCA method, with its inherent computational advantages, can be used. To see this, recall that in obtaining bounds for a given cell in the confidential table, pairs of values in the accessible projections are compared by either the cell-min or cell-max operators. Suppose that one of such a pair is suppressed. Then for the upper bound, which uses the cell-min operator, simply take the value in the suppressed cell to be indefinitely large. The minimum of this pair is then the value of the unsuppressed cell. Clearly the upper bound may only be increased as a result. If both cells in a min comparison are suppressed, the minimum is taken as infinite. In this case the overall upper bound will also be infinite. Similar remarks apply for the lower bound, substituting a value of zero for the suppressed cell or cells if it is in the *A* matrix (supply node) and infinity if it is in the *B* matrix (demand node). Note the lower bounds may only decrease when suppressions occur. These extensions are easily incorporated into the overall algorithm.

As further illustration, consider suppression to ensure a lower bound of zero for a sensitive cell. The cell-min operator yields a lower bound of zero for entry c_{ik} if and only if

$$a_{ij} \le \sum_{p \neq k} b_{jp}, \qquad \forall j.$$

For illustration, in the medical example in §4 the current lower bound is 7 for entry c_{12}. This value could be reduced to zero if a cell suppression pattern would allow that

$$a_{1j} \le b_{j1} + b_{j3}, \qquad j = 1, 2, 3.$$

For $j = 1$, this suggests suppressing cell a_{11} or (cell b_{11} or cell b_{13}); for $j = 2$, no cell need be suppressed; and for $j = 3$, this suggests suppressing cell a_{13} or (cell b_{31} or b_{33}). Since all three of the above inequalities must hold, there are nine possible cell suppression patterns suggested.

Any one of these cell suppression patterns are candidates as cell suppressions. Any of them would be adequate to drive the lower bound to zero for the sensitive cell entry if the suppressed cell entries in the A matrix could be taken to be small enough, zero, for example, or the cell entries in the B matrix large enough. However, they cannot be so taken because of certain other constraints on the tables. First, even though we assume that neither the row and column marginals, nor the grand total, are explicitly released, the grand total of the cell counts is known (by simple summation) if either original released table is left totally unsuppressed. Thus the cell suppression pattern (a_{11}, a_{13}) may not be adequate because both entries cannot be taken to be zero. Indeed their sum must be 22. But in order to have a lower bound on c_{12} of zero,

$$a_{11} \leq 9 \quad \text{and} \quad a_{13} \leq 6.$$

Since this is impossible, the cell suppression pattern is inadequate. Second, the column totals of table (matrix) A must equal the row totals of table (matrix) B. A similar analysis shows that the other identified cell suppression patterns are also inadequate. To deal with this problem, we suggest considering the identified cell suppression patterns as *primary* cell suppressions. *Complementary* cells need to be suppressed to mitigate the information gain to the data snooper of implicit knowledge of the marginal totals.

One approach to finding the complementary cells to suppress is to treat tables A and B separately, and use mixed integer programming methods (see, e.g., Kelly et al. 1992). Such methods typically use binary variables to flag whether a cell is a suppression or not. Then the sum of the binary variables is minimized under the constraints imposed by the unsuppressed cells. Although this would work, the procedure will identify more cells to suppress than necessary. To illustrate this, note that with (a_{11}, b_{31}) as primary cells to suppress, we can use as complementary suppression cells a_{13}, a_{31}, b_{11}, and b_{13}. Thus the task can be accomplished by suppressing a total of six cells. We do not need to suppress either cell a_{33} or cell b_{33}, cells which would in the two-way table framework have to be suppressed to complete a cycle of suppressed cells in each table (Willenborg and de Waal 1996, p. 92). Developing computationally efficient disclosure limitation methods in this context is a challenging problem. Some initial efforts in this direction are in Duncan et al. (1997).

7. The Relative Merits of MCA and LP

The MCA approach has two major advantages over the LP approach. The first advantage is computational; the second, conceptual. MCA assures optimal results with very fast computation when applied to tightly-linked projections of the original N-way table. Working through a maximum of $N - 2$ such procedures, the MCA algorithm converges on the tightest bounds. The MCA algorithm is a remarkably simple means for detecting disclosure potential in tables. Therefore it can be easily implemented for standard disclosure audits. As we have seen in §§5 and 6, MCA has the conceptual advantage that it identifies precisely how cell entries in the released tables influence the upper and lower bounds of the confidential table. This is useful in seeing how disclosures arise and in developing disclosure limitation procedures.

The LP approach works directly on the N-dimensional data and generates the tightest bounds in one pass, but with substantially more computation. On the other hand, the LP approach is more flexible in modeling new situations. If the data snooper knows about one of the entries of the confidential table, then this knowledge can be captured in the LP approach by simply adding a new constraint to the problem. Also, as we saw in the previous section, the LP approach readily deals with disclosure audits of tables protected through rounding methods. Further, multivariate optimization is easily implemented using the LP approach, since it requires only a simple change in the objective function.

8. Conclusions

It is broadly understood that there can be no quick and easy solution to confidentiality and data access problems. Although design-time disclosure limitation methods do apply to direct disclosure, we have shown that there is no design-time or structural approach which comprehensively addresses the problem of inferential disclosure.

We have considered disclosure risk when multiple projections or views of an underlying database are published. Using linear programming to arrive at bounds on sensitive information is not new; however, the LP bounding procedure has been applied only to a single table, rather than multiple views. Our results establish that an additional level of disclosure checking is warranted. An organization collecting sensitive information must not only check each individual table it publishes; it must also look at the *cumulative* information contained in multiple published tables. This article provides an analysis of the probability of disclosure for tables.

A second contribution of this article is to the special and important case of published two-dimensional views of the higher-dimensional table. We have developed a fast detection algorithm (MCA) based on new matrix operations. We have also explained how this method relates to the general area of statistical database protection.

Future work in this area could focus on disclosure limitation for suites of published tables. Conventional single-table measures (cell suppression, rounding, and noise masking) can be expanded to cover the entire suite. For large datasets, the computational burden may be large; heuristics of the type now used in cell suppression, for instance, might be adapted to the multiple-table case. In any event, with increasing availability of data in numerous forms, defensive measures considered prudent yesterday need rethinking today to provide confidentiality tomorrow.[1]

[1] This research was supported in part by the National Science Foundation grant NSF IRI-9312143 and by the U.S. Army Research Office under grant DAAH04-94-6-0239. The authors thank Mark Kamlet for his suggestions leading to §5. They also thank the referees and the associate editor for their insightful comments.

References

Adam, N. R., J. C. Wortman. 1989. Security-control methods for statistical databases: A comparative study. *ACM Comput. Surveys* **21** 515–556.

Barquin, R., H. Edelstein, eds. 1997. *Planning and Designing the Data Warehouse.* Prentice Hall, New York.

Carvalho, F. D., N. Dellaert, M. S. Osorio. 1994. Statistical disclosure in two-dimensional tables: Positive tables. *JASA Theory and Methods* **89** 1547–1557.

Chvátal, V. 1980. *Linear Programming.* W. H. Freeman & Co., New York.

Cox, L. H. 1980. Suppression methodology and statistical disclosure control. *JASA* **75** 377–385.

——. 1987. New results in disclosure avoidance for tabulations. Internat. Statist. Institute, Proc. 46th Session. Voorburg, The Netherlands. 83–84.

——. 1992. Solving confidentiality protection problems in tabulations using network optimization: A network model for cell suppression in U.S. economic censuses. Internat. Seminar on Statist. Confidentiality. Eurostat, Dublin, Ireland 229–45.

——. 1995. Network models for complementary cell suppression. *JASA* **90** 1453–1562.

——, J. T. Fagan, B. V. Greenberg, R. Hammig. 1986. Research at the Census Bureau into disclosure avoidance techniques for tabular data. *Proc. Section on Survey Res. Methods.* American Statistical Association, Washington, D.C. 388–393.

Denning, D. E. 1980. Secure statistical databases with random sample queries. *ACM Trans. Database Systems* **5** 291–315.

De Vries, R. E. 1993. Disclosure control of tabular data using subtables. Report, Department of Statistical Methods, Statistics Netherlands, Voorburg, The Netherlands.

Duncan, G. T. 1997. Data for health: Privacy and access standards for a health care information ethics. Audrey R. Chapman, ed. *Health Care and Information Ethics: Protecting Fundamental Human Rights.* Sheed and Ward, Kansas City, MO.

——, S. E. Fienberg. 1998. Obtaining information while preserving privacy: A Markov perturbation method for tabular data. *Proc. Statist. Data Protection* '98, Eurostat, Lisbon, Portugal March.

——, D. Lambert. 1986. Disclosure-limited data dissemination. (With discussion by L. Cox, O. Frank, J. Gastwirth and H. Roberts). *JASA* **81** 10–28.

——, ——. 1989. The risk of disclosure of microdata. *J. Bus. Econom. Statist.* **7** 207–217.

——, R. W. Pearson. 1991. Enhancing access to data while protecting confidentiality: Prospects for the future. *Statist. Sci.* **6** 219–239.

——, R. Krishnan, R. Padman, P. Reuther, S. F. Roehrig. 1997. Cell suppression to limit content-based disclosure. *Proc. Thirtieth Ann. Hawaii Internat. Conf. System Sci.* Maui, Hawaii.

Ernst, L. 1989. Further applications of linear programming to sampling problems. SRD Report: Census/SRD/RR-89-05 (1989), Bureau of the Census, Washington, D.C.

Fellegi, I. P. 1972. On the question of statistical confidentiality. *JASA* **67** 7–18.

Fienberg, S. 1998. Fréchet and Bonferroni bounds for multiway tables of counts with applications to disclosure limitation. *Proc. Statist. Data Protection* '98. Eurostat, Lisbon, Portugal March.

Fréchet, M. 1940. *Les Probabilités Associées a un Système d'Événments Compatibles et Dépendants*. Priemiere Partie. Hermann & Cie, Paris, France.

Gopal, R., P. Goes, R. Garfinkel. 1998. Interval protection of confidential information in a database. *INFORMS J. Comput.* **10**(3) 309–322.

Gusfield, D. 1988. A graph theoretic approach to statistical data security. *SIAM J. of Comput.* **17** 552–571.

Kelly, T., S. Golden, A. Assad. 1990. Controlled rounding of tabular data. *Oper. Res.* **38** 760–772.

Kelly, J., ——, ——. 1992. Cell suppression: Disclosure protection for sensitive tabular data. *NETWORKS* **22** 397–417.

Kwerel, S. M. 1988. Fréchet bounds. S. Kotz, N. L. Johnson, eds. *Encyclopedia of Statistical Sciences*, Wiley & Sons, New York 202–209.

Lougee-Heimer, R. 1989. Guaranteeing confidentiality: The protection of tabular data. Masters Thesis, Department of Mathematical Sciences, Clemson University, South Carolina.

Muralidhar, K., D. Batra, P. Kirs. 1995. Accessibility, security, and accuracy in statistical databases: The case for the multiplicative fixed data perturbation approach. *Management Sci.* **40** 1549–1563.

Nargundkar, M. S., W. Saveland. 1972. Random rounding of tables to prevent statistical disclosure. *Proc. of the Social Statist. Section, Amer. Statist. Assoc.* Washington, D.C. 382–387.

Repsilber, D. 1991. Safeguarding secrecy in aggregative data. Proc. 1991 Internat. Seminar Statist. Confidentiality. Eurostat, Dublin, Ireland 353–368.

——. 1993. Preservation of confidentiality in aggregated data. *Second Internat. Seminar Statist. Confidentiality*. Luxembourg.

Robertson, D. A. 1994. Automated disclosure control at Statistics Canada. Working Paper, Statistics Canada, Ottawa, Ont.

Rowe, E. 1991. Some considerations in the use of linear networks to suppress tabular data. *Proc. Section on Survey Res. Methods*. American Statistical Association, Washington D.C. 357–362.

Sande, G. 1984. Automated cell suppression to preserve confidentiality of business statistics. *Stat. J. United Nations*. ECE 2 Geneva, Switzerland 33–41.

Schlörer, J. 1975. Identification and retrieval of personal records from a statistical data bank. *Methods Info. Med.* **15** 7–13.

Sullivan, C., L. Zayatz. 1991. A network flow disclosure avoidance system applied to the census of agriculture. *Proc. Section on Survey Res. Methods*. American Statistical Association, Washington D.C. 363–368.

Willenborg, L., T. de Waal. 1996. *Statistical Disclosure Control in Practice*. Springer-Verlag, New York.

Zayatz, L. 1992. Linear programming methodology used for disclosure avoidance purposes at the census bureau. *Proc. Section on Survey Res. Methods*. American Statistical Association, Washington, D.C.

Accepted by John C. Henderson; received June 11, 1996. This paper has been with the authors 28 months for 4 revisions.