NISS

Table Servers: Protecting Confidentiality in Tabular Data Releases

Alan F. Karr, Adrian Dobra, and Ashish P. Sanil

Technical Report Number 120
November, 2001

National Institute of Statistical Sciences
19 T. W. Alexander Drive
PO Box 14006
Research Triangle Park, NC 27709-4006
www.niss.org

Table Servers: Protecting Confidentiality in Tabular Data Releases

Alan F. Karr, Adrian Dobra, Ashish P. Sanil
National Institute of Statistical Sciences, Research Triangle Park, NC 27709-4006, USA

Introduction. Federal statistical agencies must balance concern over confidentiality of data with their obligation to report information to the public [5]. Advances in information technology threaten confidentiality, but also new technologies can protect confidentiality while meeting user needs in innovative ways.

Here we describe *table servers* being developed by the National Institute of Statistical Sciences (NISS) that disseminate tabular summaries of statistical data in response to user queries for marginal sub-tables of a large (e.g., 40 dimensions with 4 categories each) contingency table containing counts or sums. Table servers *evaluate disclosure risk dynamically*, in light of previously answered queries.

Abstractions. The query space $\mathcal{Q}$, which contains all 2^K sub-tables of a K-way table, is partially ordered by set inclusion of variables in subtables. The set $\mathcal{R}(t)$ of all tables released through some time t contains direct releases in response to queries and indirect releases (previously unreleased children of direct releases); $\mathcal{R}(t)$ is specified by the *released frontier* $\mathcal{RF}(t)$ of its maximal elements (Figure 2).

Underlying dynamic release decisions is a *risk criterion* **RC** defined on subsets of $\mathcal{Q}$: at all times the system must satisfy $\mathbf{RC}(\mathcal{R}(t)) \leq \alpha$, where α is a risk threshold set by the operators. A typical risk criterion is accuracy of bounds based on $\mathcal{R}(t)$ for sensitive (small count) cells in the full table. Bounds can be computed using network methods [3, 9] and the "shuttle algorithm" [2]. There are also exact techniques for special cases. For example, if the released sub-tables constitute the minimal sufficient statistics of a decomposable graphical model [8] (Figure 1), then bounds can be expressed as *explicit* functions of these sub-tables [4].

Whenever an answered query releases previously unreleased information, other queries become unanswerable. Consequently (Figure 2), at t there is an *unreleasable set* $\mathcal{U}(t)$ of sub-tables whose release would be too risky, with an *unreleasable frontier* $\mathcal{UF}(t)$ of its minimal elements.

Release rules determine which requests for unreleased tables will be fulfilled. The simplest is the *myopic rule* of releasing T at t as long as $\mathbf{RC}(\mathcal{R}(t) \cup T) \leq \alpha$. To prevent the table server from taking excessively large steps, one can allow only tables adding but one variable to a previously released table to be eligible for release. To prevent a single user (or a set of colluding users) from driving the table server into a region of $\mathcal{Q}$ that suits their needs but not those of other users, release rules can biased against releases that add large numbers of tables to $\mathcal{U}(t)$. Rules can also incorporate the *value* of releasing T [6, 12].

System Design and Prototypes. A prototype table server, written as a Java application, is shown in Figure 2. Its principal strength is the engaging (but non-scalable) visualization of $\mathcal{Q}$.

Figure 5 shows the architecture of a more powerful table server written using the Java 2 Enterprise Edition [10] platform, with HTTP processing performed by Java Servlets [11]. This prototype uses a 14-dimensional, 300,000,000-cell, but extremely sparse, table derived from the Current Population Survey [1].

Figure 3 shows the user input screen. If the requested table lies on or below $\mathcal{RF}(t)$, it is provided immediately, ordinarily via downloaded XML. Releases are governed by the myopic and "at most one step away from $\mathcal{R}(t)$" rules, and disclosure risk is evaluated in real time. The query history database, with tables for users, queries and the time trajectories of $\mathcal{RF}(t)$ and $\mathcal{UF}(t)$, is maintained in a MySQL database server [13]. A frontier display facility (Figure 4) monitors evolution of $\mathcal{RF}(t)$.

The system employs data structures based on hash tables for storing tables and algorithms that exploit sparsity and the fact that $\mathcal{R}(t)$ and $\mathcal{U}(t)$ are characterized completely by $\mathcal{RF}(t)$ and $\mathcal{UF}(t)$. The risk criterion is narrowness of cell bounds computed via a generalized shuttle algorithm.

References

[1] C. L. Blake and C. J. Merz. UCI repository of machine learning databases, 1998.

[2] L. Buzzigoli and A. Giusti. An algorithm to calculate the lower and upper bounds of the elements of an array given its marginals. In *Proceedings of the Conference on Statistical Data Protection*, pages 131–147, Luxembourg, 1999. Eurostat.

[3] L. H. Cox. Network models for complementary cell suppression. *J. Amer. Statist. Assoc.*, 90:1453–1462, 1995.

[4] A. Dobra and S. E. Fienberg. Bounds for cell entries in contingency tables given marginal totals and decomposable graphs. *Proc. Nat. Acad. Sci.*, 97(22):11885–11892, 2000.

[5] G. T. Duncan, V. A. de Wolf, T. B. Jabine, and M. L. Straf. Report of the Panel on Confidentiality and Data Access. *J. Official Statist.*, 9:271–274, 1993.

[6] G. T. Duncan and S. Keller–McNulty. Mask or impute? *Res. Official Statist.*, 2001. To appear.

[7] S. Keller–McNulty and G. T. Duncan. A progress report to the National Center for Education Statistics: Disclosure-limited statistical analysis of confidential data to support NSF-sponsored Digital Government grant. Technical Report LA-UR-01-1673, Los Alamos National Laboratory, 2001.

[8] S. L. Lauritzen. *Graphical Models*. Clarendon Press, Oxford, UK, 1996.

[9] S. Roehrig. Auditing disclosure in multi-way tables with cell suppression: Simplex and shuttle solutions. In *Proc. Joint Statistical Meetings*, Alexandria, VA, 1999. American Statistical Association.

[10] Sun Microsystems, Inc. Java 2 Platform, Enterprise Edition Specification. Information available online at java.sun.com/j2ee/.

[11] Sun Microsystems, Inc. Java Servlet Technology. Information available on-line at java.sun.com/products/servlet/.

[12] M. Trottini. A decision-theoretic approach to data disclosure problems. In *2nd Joint ECE/Eurostat Work Session on Statistical Data Confidentiality*, Luxembourg, 2001. Eurostat.

[13] R. J. Yarger, G. Reese, and T. King. *MySQL and mSQL*. O'Reilly, Sebastopol, CA, 1999.

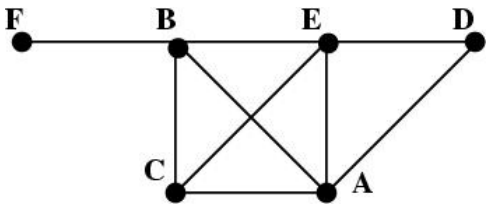


Figure 1: Independence graph associated with three marginals [BF], [ABCE] and [ADE] of a six-way table [ABCDEF], visualizing the dependency patterns induced by the released marginals. The graphical model with minimal sufficient statistics [BF], [ABCE] and [ADE] is decomposable.

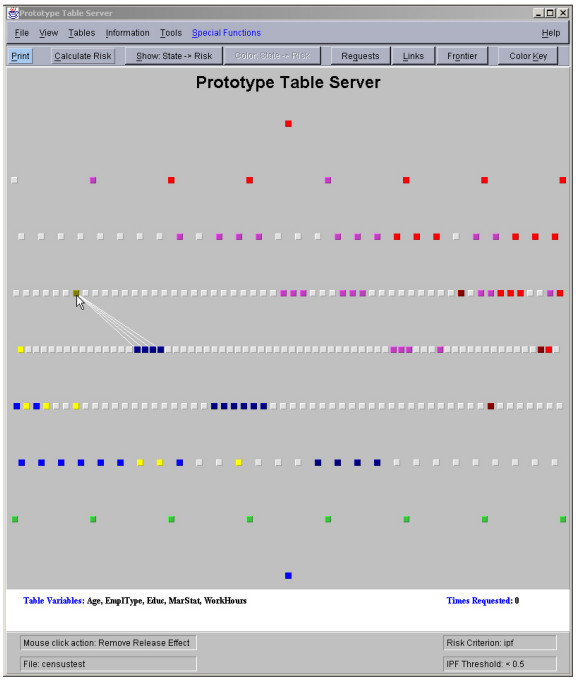


Figure 2: Java table server prototype. The visualization of the query space $\mathcal{Q}$ shows direct releases (yellow), indirect releases (blue), unacceptably risky releases (red) and the potential effect (dark blue, magenta and dark red) of releasing the 5–way table indicated by the cursor. The released (unreleasable) frontier lies at the top of the lower left (bottom of the upper right) portion of the visualization.

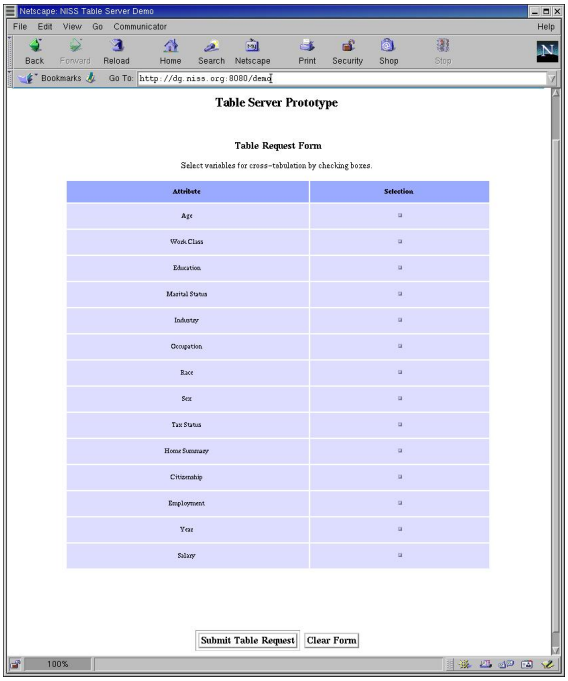
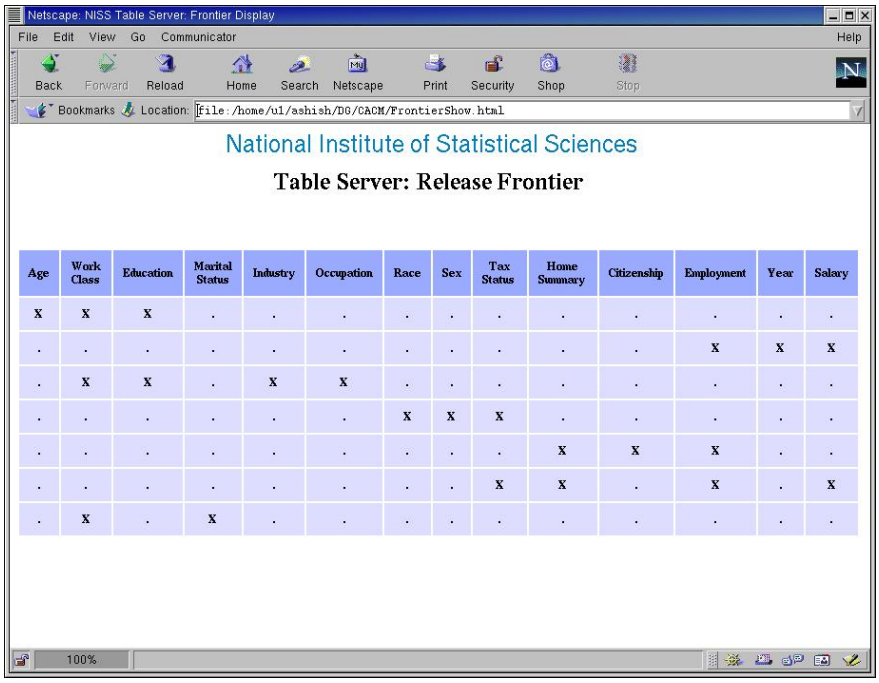


Figure 3: Table server prototype: User input screen. Queries are posed by selecting the variables in the desired sub-table.



Age	Work Class	Education	Marital Status	Industry	Occupation	Race	Sex	Tax Status	Home Summary	Citizenship	Employment	Year	Salary
X	X	X	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	X	X	X
.	X	X	.	X	X	.	.	.	.	.	.	.	.
.	.	.	.	.	.	X	X	X	.	.	.	.	.
.	.	.	.	.	.	.	.	.	X	X	X	.	.
.	.	.	.	.	.	.	.	X	X	.	X	.	X
.	X	.	X	.	.	.	.	.	.	.	.	.	.

Figure 4: Table server prototype: Released frontier display meant for system operators. The display lists the sub-tables comprising $\mathcal{R}(t)$.

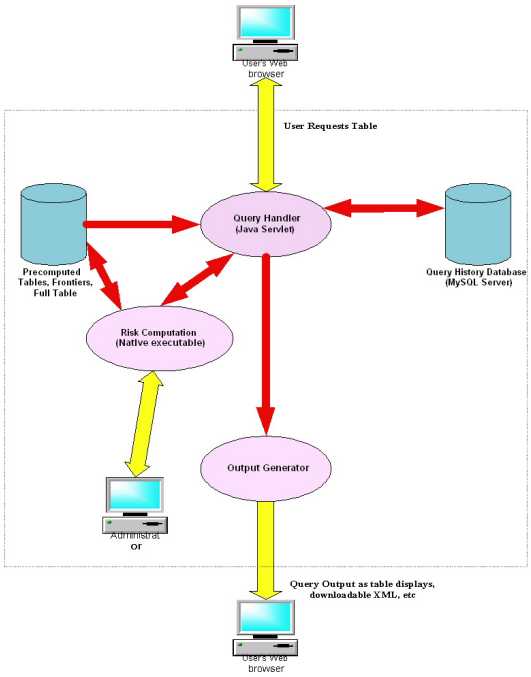


Figure 5: Table server prototype: System architecture. Output formats include screen display and XML.