Web-Based Systems that Disseminate Information but Protect Confidential Data

Alan F. Karr and Ashish P. Sanil

National Institute of Statistical Sciences
PO Box 14006, Research Triangle Park, NC 27709-4006
{karr,ashish}@niss.org

Abstract

We describe two Web-based query systems being developed by the National Institute of Statistical Sciences (NISS) that disseminate, in the form of statistical analyses, information derived from confidential databases, but protect the confidentiality of the data.

Keywords: Data confidentiality, privacy, Web dissemination, digital government, aggregation, contingency tables.

Statistical agencies in the US and elsewhere have longstanding concern over confidentiality of their data. At the same time, agencies must also report information to the public. This tension between confidentiality and dissemination of statistical information is heightened by the emergence of the World Wide Web as a means of communication. Although confidentiality is threatened by advances in information technology (IT), other new technologies not only protect confidentiality, but also meet user needs in innovative ways.

In this article, we describe two prototype systems. The first was developed for the National Agricultural Statistics Service (NASS). It will be used to disseminate data on usage of agricultural chemicals (fertilizers, fungicides, herbicides and pesticides) on farms. The central mechanism for preserving confidentiality is geographical aggregation: data from adjacent counties are aggregated to the level of disclosable "supercounties."

The second system is being developed for agencies whose target database is a large cross-classified table of counts or sums (contingency table). In this case, data are disseminated in the form of marginal sub-tables in response to user queries. Distinguishing features (and major challenges encountered in the implementation) of these *table servers* include the incorporation of a release-history dependent dissemination policy, and the scaling of standard statistical techniques for risk evaluation to higher-dimensional contingency tables.

1 The NASS System

The data consist of on-farm use of various chemicals (referred to as active ingredients (AI)) on various crops in the years 1996–1998. For our purposes, each data record can be thought of as consisting of Farm ID, farm size in acres, crop, AI, pounds of AI applied, state, county and year.

The database contains 194,410 records collected from 30,500 farms, with information on the rates of usage of 322 AIs on 67 crops (field crops, fruits and vegetables).

The user community for the system is interested primarily in *application rates* (pounds applied per acre) of certain AIs on particular crops.

The confidentiality concern of NASS is to protect the identities of farms in the survey. Thus, information cannot be disclosed that would enable a user to estimate accurately the chemical usage on a particular farm. To protect confidentiality, NASS currently releases AI usage information only aggregated at the state level. The system we describe produces intermediate aggregations that are more informative than state-level data but still preserve confidentiality.

1.1 Aggregation for Disclosure-Risk Reduction

NASS employs two rules to determine if the data for a particular county pose a confidentiality risk. The first is the N-rule: if only one or two farms were sampled in the county (thus, $N = 3$), the possibility of re-identification is too high and the application rate for that county is undisclosable.

Second is the p-rule: a county-level application rate is undisclosable if the sample contains a dominant farm, whose size (in acres) is more than $p\%$ of the total size of all farms sampled in that county (we use $p = 60\%$). The rationale is that a farm which dominates in a sample is susceptible to both identity and attribute disclosure risks.

We refer to these risk criteria collectively as the *N-p rule* (a form of what is referred to in the statistical data disclosure literature as the (n, k) rule [2, 4]). There is also an additional level of protection arising from the user's not knowing which farms are included in the survey.

The trouble with implementing the N-p rule ($N = 3$, $p = .6$) at the county level is that, for most user queries, over 50% of the counties are undisclosable. Simply refusing to answer such queries would lead to unacceptable user frustration. Our remedy is to aggregate undisclosable counties with neighboring counties to form disclosable "supercounties." As a result, NASS can release data at the finest level of detail consistent with the risk criteria.

Aggregations must be computed automatically, since there too many (crop, AI, year, state) combinations to permit manual aggregation on a case-by-case basis. We employ heuristic, "greedy" algorithms (see [1] for details) based on the following procedure: Visit the undisclosable (super)counties in a random order and merge them with a neighboring (super)county according to some criterion for desirability of merging with the selected neighbor. Continue until only disclosable (super)counties remain.

The algorithms differ only in the rule that governs the merging process. We have developed two rules. The pure rule directs the merging of counties in a manner that favors leaving the disclosable counties alone (preserving the "purity" of their data), instead merging the undisclosable counties among themselves as far as possible. Although this procedure succeeds in preserving the "purity" of as many counties as possible, its strong tendency to force disclosable counties to remain unmerged leads to large supercounties comprised of many undisclosable counties.

This observation led us to formulate the small rule, which favors forming small supercounties by merging an undisclosable region with a neighboring region most likely to form a disclosable region. The resulting aggregation is minimal in the sense that it produces many small supercounties.

In practice, judging by visual inspection, the small rule produces satisfactory aggregations.

Both algorithms, because of the random order in which candidate mergers are considered, can produce supercounties that may be decomposed into smaller disclosable supercounties (see the example in. We guard against such suboptimality by using a two-step process. The small algorithm is run first to produce an initial aggregation, and then the pure algorithm is run within each supercounty produced by small, in order to determine if these can be decomposed.

This composite procedure works remarkably well. It is also very fast – test runs and simulations for aggregation of states containing 100 counties take less than a millisecond per run, even on a 233MHz Pentium PC running Linux.

The aggregation problem can be formulated as a (NP-hard) combinatorial optimization problem over the edge-set of the adjacency graph of the counties in a state. One advantage of doing so is that the optimization framework allows explicit incorporation of "goodness" of aggregations (for instance, allowing preference for aggregating counties that lie within the same watershed). The combinatorial optimization problems can be solved using computationally intensive simulated annealing methods, but long running times make it infeasible to use simulated annealing in practice. In test cases where we have used both methods, there is no significant difference between the characteristics of the aggregations they produce, and we conclude that the heuristic methods are adequate for the application we describe here.

1.2 NASS System Operation

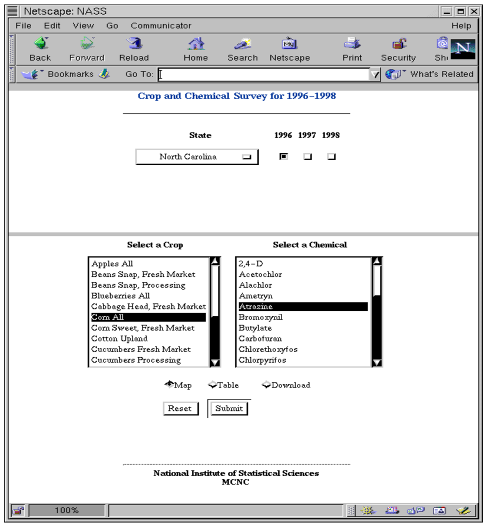
As shown in Figure 1, the user first selects from an HTML form a State containing the counties for which information is desired. Once a State is selected, the user selects among only the crops grown and chemicals used in the selected State.

The result of a query consists of a list of supercounties that arise from the running of the aggregation algorithm. Each supercounty has associated with it a set of component counties and the application rate of the selected AI on the selected crop in the selected year(s).

The default mode for reporting query results to the user is a map. One map of the State is displayed for every year requested. Supercounties are colored according to the application rate of the chemical. Both supercounty boundaries and the individual county boundaries are discernible. In order to give users a sense of the numerical values involved, a color key is displayed with the average pounds per acre for the State marked on it (see Figure 1). The user may also view in tabular form the numerical values of the application rates underlying the map or download the values in XML format. The DTD for the XML download is well-suited to the hierarchical nature of data aggregated at the supercounty level.

2 Table Servers

The target database of a table server is a large (e.g., 40 dimensions with 4 categories each) contingency table, containing counts or sums. The essential characteristic of the table servers being developed by NISS is that they are dynamic: user queries for marginal sub-tables of the full table


Netscape: NASS
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security
Bookmarks Go To: What's Related
Map Table Download
Reset Submit
100%

Crop and Chemical Survey for 1996–1998

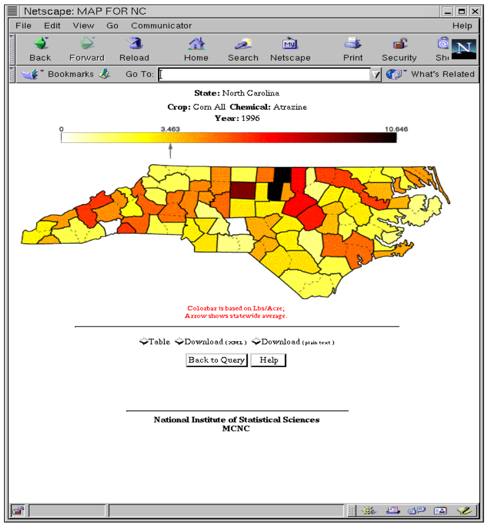
State: North Carolina

1996 [✓] 1997 1998

Select a Crop: Apples All, Beans Snap, Fresh Market, Beans Snap, Processing, Blueberries All, Cabbage Head, Fresh Market, [✓ Corn All], Corn Sweet, Fresh Market, Cotton Upland, Cucumbers Fresh Market, Cucumbers Processing

Select a Chemical: 2,4–D, Acetochlor, Alachlor, Ametryn, [✓ Atrazine], Bromoxynil, Butylate, Carbofuran, Chlorethoxyfos, Chlorpyrifos

National Institute of Statistical Sciences
MCNC


Netscape: MAP FOR NC
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security
Bookmarks Go To: What's Related
0
3.463
10.646
Table Download (XML) Download (plain text)
Back to Query Help

State: North Carolina
Crop: Corn All Chemical: Atrazine
Year: 1996

Colorbar is based on Lbs/Acre;
Arrow shows statewide average.

National Institute of Statistical Sciences
MCNC

Figure 1: Input-screen (left): Users first select the State from which the data are required. Next, they select a crop (or chemical) from the menu. Then, the adjacent menu is regenerated to contain only the chemicals (or crops) associated with the selected crop or chemical. Year(s) and output formats may be specified at any time before submitting the query. Output Screen (right): The map displays the pounds per acre of *Atrazine* used on *Corn* for all (super)counties in *North Carolina* in *1996*, using artificial data. The color key shows the numerical values, with a marker indicating the statewide average usage. The display allows users to identify both individual counties and parent supercounties.

are assessed for disclosure risk in light of previously answered queries. Potential responses include the requested table (as text, visualized or in XML), its "projection" onto the released frontier, risk-reduced modifications of the requested table and refusal of the query. The central challenges to building such a system are development of the abstractions that underlie it and of scalable algorithms that implement the abstractions.

2.1 The Query System

2.1.1 User's Perspective

The *target database* is a large (typically 40 dimensions $\times$ 4 categories each) contingency table, containing counts or sums. A request for a (low-dimensional, typically) marginal subtable of the full table is made by the user. The query is answered if the requested sub-table was released in the past, or if the sub-table can be derived from a previous release (i.e., if it is a marginal table of a previously released table).

If the sub-table is not immediately releasable, the user is offered the highest-dimensional releasable sub-table(s) that are nearest to the requested query. (This corresponds to a projection onto the releasable frontier.) The user may now choose one of the sub-tables offered or choose to submit the original query for risk evaluation. The risk evaluation module then evaluates the releasability of the requested sub-table according to the risk criteria (described below), and either answers the query or denies the request. (Later versions of the table server might include the option of releasing a risk-reduced modification of requested table.)

The results of a query are obtainable as tabular displays (for two- to four-dimensional tables) or as text/XML downloads.

2.1.2 Query History Database

A primary feature of the table servers is that they are *release-history dependent*: decisions about release is dependent on the history of what has been released in the past. This entails a careful and detailed recording of the query and response history of the system. This is done in the Query History Database (QHDB) which consists of tables for information on users, requests, sub-tables released, release frontier and unreleased frontier. (Frontiers are described in 2.1.3.)

2.1.3 Fundamental Abstractions

The abstractions underlying a table server include:

Query space $\mathcal{Q}$: In the case of a table server where the full table is K-dimensional, there are 2^K possible queries for sub-tables. Moreover, there is a natural partial order on the set of queries: $Q_i \prec Q_j$ if and only if the set of dimensions (variables) in Q_i is contained in the set of dimensions in Q_j. This partially ordered lattice structure of $\mathcal{Q}$ allows us to create a visual display of the space as shown in Figure 2 (where the top of the lattice represents the full table and the bottom is the "null" table or the grand total of the cells in the full table). This display provides an invaluable tool for the system operators, since this allows us to readily visualize the current state of the system as well as risk consequences of various releases.

Query Sets: There are a few distinguished query sets that capture a complete description of the current state of the system (and are therefore tracked by the QHDB):

- *Released Set* $\mathcal{R}(t)$ of all tables released (directly or indirectly) through time t.
- *Unreleasable Set* $\mathcal{U}(t)$ of all tables considered unreleasable (too risky to release) at time t.
- *Frontiers:* minimally sufficient representations of $\mathcal{R}$ and $\mathcal{U}$, which are necessary since both $\mathcal{R}$ and $\mathcal{U}$ could grow large and unwieldy very quickly. The Released Frontier, $\mathcal{RF}$(t) contains maximal elements of $\mathcal{R}$(t), and the Unreleasable Frontier, $\mathcal{UF}$(t) consists of minimal elements of $\mathcal{U}$(t).

2.1.4 Risk Criterion (RC)

The table server administrators use a risk criterion $\mathbf{RC}$ to evaluate the risk of releasing sub-tables. $\mathbf{RC}$ is a real-valued function defined on subsets S of $\mathcal{Q}$, indicating how risky it would be to release all of the sub-tables in S.

Such risk for a subset S of $\mathcal{Q}$ could be based on the following considerations:

- How accurately can one find upper and lower bounds for entries in the cells of the full table given that the marginal sums given in tables in S are known. It is possible to find such bounds using linear programming, heuristic methods or Fréchet bounds [3].
- How accurately can one reconstruct entries in the cells of full table entries in the cells of the full table given S. Techniques such as iterative proportional fitting (IPF) may be used for table reconstruction.
- The number of tables whose marginals coincide with all elements of $\mathcal{RF}(t)$ (or some other measure of the size of the set of all such tables)

2.1.5 Release Rules

After selecting a suitable $\mathbf{RC}$ we select an maximum acceptable level of risk, α. Then, release decisions can be made as follows. $T \in \mathcal{Q} \setminus \mathcal{R}$ *cannot* be released at t if

$$\mathbf{RC}(\mathcal{R}(t) \cup T) > \alpha.$$

Major issues related to the selection of release rules include:

- Choice of a suitable risk criterion, $\mathbf{RC}$ and cutoff level α to accurately reflect the release risk.
- Consideration of what regions of $\mathcal{Q}$ become too risky as a consequence of releasing T (this can be monitored using the query space visualization in Figure 2).
- The *value* of releasing T using, say, number of requests for T as an indicator.
- Whether system operators make periodic preemptive releases of certain subtables in order to prevent certain regions of $\mathcal{Q}$ from falling into $\mathcal{U}$.

2.2 Design Challenges

While many of the concepts underlying table servers are understood as theories, few have been implemented in scalable, functioning systems. Among the challenges that must be confronted are:

Scalability. Almost all the existing computational techniques for the analysis and risk computations for contingency tables have been developed for low-dimensional tables (usually for 2–3 dimensional tables). Since the table sizes and computational complexity grows exponentially with table dimension, näive extensions of existing techniques become infeasible for the size of tables we encounter in our case. Data structures that account for table sparsity and special structure are being used, and corresponding computational techniques that scale are under being employed and are the subject of active development.

Equity. It is important to prevent a small subset of users from steering the system to meet their needs, and thereby rendering large regions of $\mathcal{Q}$unreleasable for other legitimate user requests. Batch processing of requests based on a voting mechanism is one proposal being researched. Another possible solution is to have different user classes with only some users being allowed to request unreleased tables.

2.3 The Prototype Table Server

We are currently studying the characteristic behavior of table servers using a 14-dimensional table of Census data as a testbed.

We have developed versions of IPF-based table reconstruction, and a heuristic cell bounding algorithm that exploit the sparse nature of the table, and hence seem to be scalable to high-dimensional tables.

A Java application, shown in Figure 2, has been developed for visualizing the query space, and risk consequences of sequential table releases.

The QHDB schema and updating procedures have been developed within a standard RDBMS setting.

We are examining various risk criteria based on table reconstruction and cell-bounds. In particular, we are investigating if these criteria can correctly model disclosure risk, and how they relate to the evolution of the system (*vis-a-vis* movement of the two frontiers in $\mathcal{Q}$).

Acknowledgements

The research reported here is supported by the Digital Government program at NSF, under grant EIA–9876619 to the National Institute of Statistical Sciences. Carol House of the National Agricultural Statistics Service (NASS) stimulated development of this particular system, and Joseph Prusacki of NASS provided both data and knowledge about them. Joel Hernandez, Sousan Karimi, Hassan Karimi and Karen Litwin from MCNC implemented the NASS system.

References

[1] A. Karr, J. Lee, A. Sanil, J. Hernandez, S. Karimi, and K. Litwin. Web-Based Systems that Disseminate Information from Data but Protect Confidentiality. http://www.niss.org/dg/technicalreports.html – Expanded version of article in IEEE Computer, February 2001 issue., 2000.

[2] Federal Committee on Statistical Methodology. *Report on Statistical Disclosure Limitation Methodology*, May 1994.

[3] Statistical Office of the European Communities. *Proceedings of the Conference on Statistical Data Protection*. Eurostat, 1999.

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

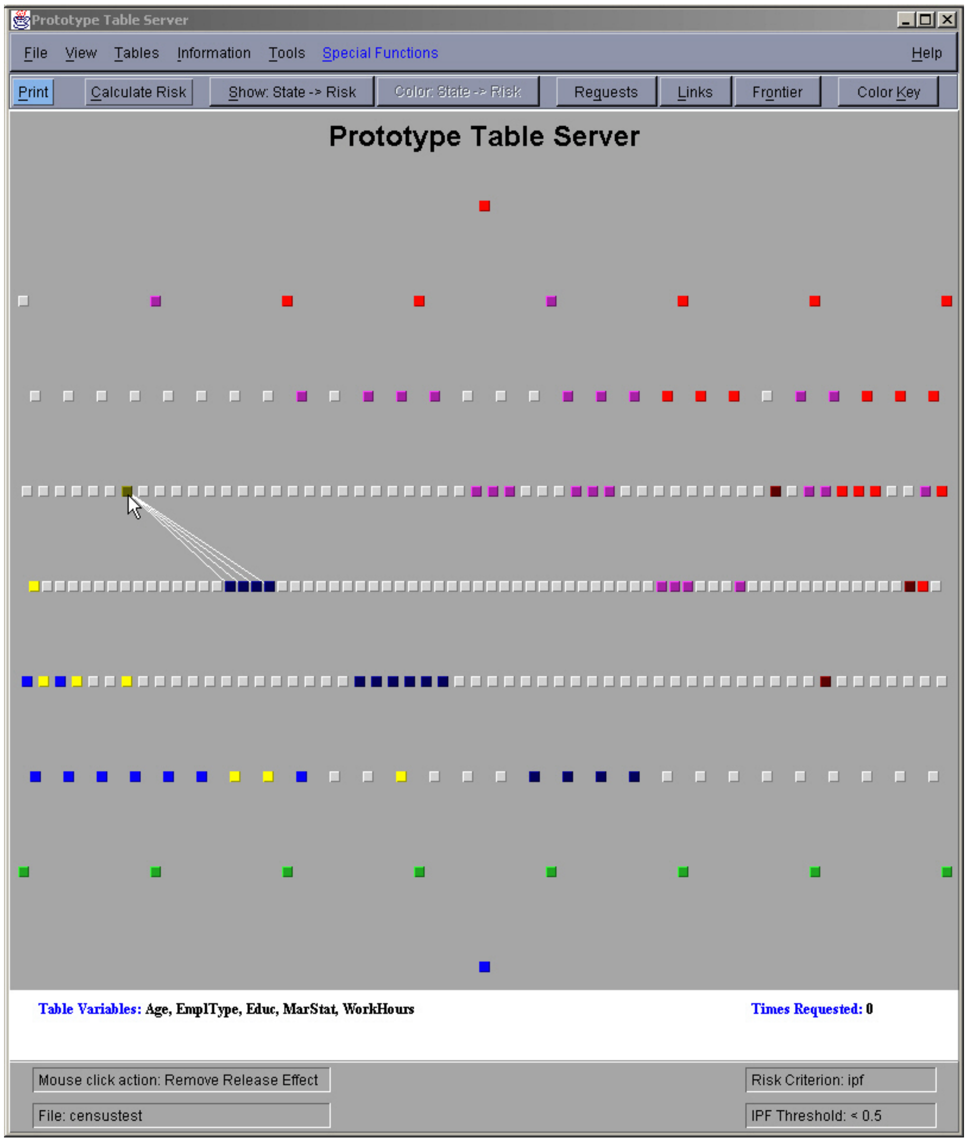


Table Variables: Age, EmplType, Educ, MarStat, WorkHours

Times Requested: 0

Mouse click action: Remove Release Effect

File: censustest

Risk Criterion: ipf

IPF Threshold: < 0.5

Figure 2: Visualization of the query space for a table server operating on an 8-dimensional table. The risk criterion is the accuracy with which the full table can be reconstructed using iterative proportional fitting (see §2.1.4). Shown are the previous releases (yellow and blue), *core releases* made when the system begins operation (green), and tables that are already too risky (red). The cursor highlights a particular proposed release. Releasing this table would, however, render all magenta and brown tables (one of which is even three-dimensional) unreleasable in the future.