Statistical Databases: Query Restriction

Nina Mishra

January 21, 2004

Introduction A statistical database typically contains information about n individuals where n is very large. A statistical database system gives users the ability to both obtain statistical information (like average, median, count) and preserve the privacy of any individual. Examples include census and medical databases.

There are many challenges to designing a statistical database system. The greatest challenge is to properly define privacy. Other challenges include: users may pose multiple queries, users may have prior knowledge, collusion between users is possible, static vs. dynamic databases, and centralized vs. decentralized databases.

There are many methods for designing a statistical database system. These methods are typically broken into three categories [AW89]. (1) Query Restriction - where queries are answered exactly, but not all queries are permitted (2) Input Perturbation - where queries are answered according to a perturbed database (3) Output Perturbation - where perturbed responses are given to queries.

We give an overview of the Query Restriction methods. One way to define the Query Restriction problem is as follows:

Problem 1 *Given a database $D = \{x_1, \ldots, x_n\}$ where each element $x_i \in R$, given $S = \{S_1, \ldots, S_m\}$ where each $S_i \subset D$, and given integers $b_1, \ldots, b_m$ where $\sum_{x_i \in S_j} x_i = b_j$, the* Query Restriction Problem *is to determine if there is a variable x_k that is uniquely determined. In such a case, the database is said to be compromised.*

In the Query Restriction category, there are many methods that have been considered. The following methods are discussed Query-Set-Size Control, Query-Set-Overlap Control, Auditing, Partitioning, and Cell Suppression.

Query-Set-Size Control If we allow queries that cover a small number of items, i.e., subsets S_i with just one element, then the database can be compromised since any element can be determined. One natural solution is to lower bound the number of elements in the query, i.e., $|S_i| \geq k$, also known as Query-Set-Size Control.

However, just limiting the size of the query is not sufficient to prevent compromise, since if we want to compromise the value x_1, we can just take the difference between the queries $\sum_{i=1,\ldots n} x_i$ and $\sum_{i=2,\ldots n} x_i$.

Query-Set-Overlap Control The source of the compromise in the previous example is the overlap in the queries. A possible solution then is to limit the amount of overlap allowed between queries.

However, compromise is still possible if both the size and the overlap of the queries is bounded. An example where each query contains at least k items and each pair of queries overlaps in at most one point demonstrates a compromise [DJL79].

Auditing Beyond ensuring a minimum query size and a maximum allowable overlap, the actual trail of queries posed by a user needs to be followed if one wants to solve the Query Restriction Problem.

For a database where each element x_i is real-valued it is possible to determine if a database has been compromised by applying linear programming [CO82] and/or a sequence of row/column manipulations on the matrix where the rows correspond to the subsets S_i and the columns correspond to x_j.

However, for a database where each x_i is discrete-valued, i.e., $x_i \in \{0, 1\}$, it is coNP-complete to determine if a database has been compromised [KPR00]. This problem is known as the Boolean Auditing Problem.

The above hardness result assumes that queries can be posed over arbitrary subsets of points. For the restricted class of queries of the form $\sum_{i=j,\ldots,k} x_i$, also known as 1-dimensional queries, there is a polynomial time algorithm for determining if a database has been compromised [KPR00]. However, for higher-dimensional queries, the problem is known to be coNP-complete.

Partitioning In the Partitioning problem, the goal is to cluster the database into similar groups. "Privacy" is achieved by replacing the s elements of a cluster with s copies of a cluster centroid. If a cluster is too tight then

privacy can be breached since, in the extreme case, all points in the cluster are identical, so replacing all points with a centroid reveals the original points. Thus, there is a tradeoff between the goodness of a clustering and the privacy guarantees [CO81].

Cell Suppression One way to define the cell suppression problem is as follows:

Problem 2 *Given a two-dimensional table D and given the sum of the cell values in row i and the sum of the cell values in column i (also known as marginals), and given a collection of table entries S to suppress, the* Cell Suppression Problem *is to identify a minimum-sized set of additional cells S' to suppress (also known as complementary cells) so that S cannot be deduced from the unsuppressed cells and the marginals.*

For a given S and S', one can determine in polynomial time if S can be deduced [Gus89]. However, for a given S, it is NP-hard to find a minimum sized set S' that should also be suppressed to protect S [Gus89]. For follow-up work, see [Kao96, HK97].

Relationship to Secure Multi-Party Computation The problem addressed by statistical databases is conceptually different from secure-multiparty computation. In statistical databases, we have specific data that we do not wish to reveal, yet we're obligated to give out statistical information. In secure multi-party computation, there is a function we must compute, and we allow the output of that function to dictate what is revealed. Thus, in secure multi-party computation, what is revealed changes depending on the function computed.

Discussion and Open Problems The current modeling of queries as subsets of items is too powerful since in general, users cannot determine the sum of an arbitrary subset of elements. So we need to restrict the subset of items that can be queried. One way to accomplish this is to postulate a second attribute-column, and allow selection on that column to generate a subset. The relation between this selection column and the summed-column is assumed to be hidden from the user. In addition, it might be good to think about queries other than Sum, like Median, Max, etc.

The Query Restriction/Auditing problem doesn't seem like the right problem to solve since remembering the sequence of queries posed by a

user is space-intensive and solving linear programs based on those queries is compute-intensive. Furthermore, collusion is a huge issue for auditing.

This big open question is: What is the right problem description and privacy definition? Perhaps the right model is to postulate query syntax restrictions and explore its impact on disclosure/compromise?

Other items:

- Explore relation to TRAPP work especially the work on medians
- Relation to k-anonymity?
- Hector's Taulbee-Salary-Survey problem?
- Constant-factor approximation algorithm for the cell suppression problem?

References

[AW89] N. Adam and J. Wortmann. Security-control methods for statistical databases: A comparative study. *ACM Computing Surveys*, 21(4):515–556, 1989.

[CO81] F.Y. Chin and G. Ozsoyoglu. Statistical database design. *ACM Trans. on Database Systems*, 6(1):113–139, 1981.

[CO82] F. Chin and G. Ozsoyoglu. Auditing and infrence control in statistical databases. *IEEE Transactions on Software Eng.*, 8(6):113–139, 1982.

[DJL79] D. Dobkin, A. Jones, and R. Lipton. Secure databases: Protection against user influence. *ACM Transactions on Database Systems*, 4(1):97–106, 1979.

[Gus89] D. Gusfield. A graph theoretic approach to satistical data security. *SIAM J. Comput.*, 17(3):552–571, 1989.

[HK97] T. Hsu and M. Kao. Security problems for statistical databases with general cell suppressions. In *Ninth International Conference on Scientific and Statistical Database Management, Proceedings, August 11-13, 1997, Olympia, Washington, USA*, pages 155–164, 1997.

[Kao96] M. Kao. Data security equals graph connectivity. *SIAM Journal on Discrete Mathematics archive*, 9(1):87–100, 1996.

[KPR00] J. Kleinberg, C. Papadimitriou, and P. Raghavan. Auditing boolean attributes. In *Symposium on Principles of Database Systems*, pages 86–91, 2000.