2008

Diophantine Inference on a Statistical Database

Rowe, Neil C.

Monterey, California. Naval Postgraduate School

Information Processing Letters, 18 (1984), 25-31.
http://hdl.handle.net/10945/36437

Diophantine Inference on a Statistical Database

Neil C. Rowe

Department of Computer Science, Code CS/Rp
Naval Postgraduate School, Monterey, CA 93940
ncrowe at nps.navy.mil

This work is part of the Knowledge Base Management Systems Project at Stanford University, under contract #N00039-82-G-0250 from the Defense Advanced Research Projects Agency of the United States Department of Defense. The views and conclusions contained in this document are those of the author and should not be interpreted as representative of the official policies of DARPA or the US Government. Thanks to Kyu-Young Whang.

This paper appeared in *Information Processing Letters, 18* (1984), 25-31. The equations were redrawn in 2008 and some corrections made.

1. Introduction

A statistical database is said to be compromisable if individual data items can be inferred from queryable values of statistical aggregates (mean, maximum, count, etc.) ((Denning82), ch. 6). We discuss here some methods, which while only leading to compromise of individual records on occasion, do lead to powerful inferences of other statistical characteristics of a database which may also be sensitive information. These methods use a new technique that has not apparently heretofore been explored, solution of simultaneous Diophantine (integer-solution) equations.

We review some previous research in part 1. Then in part 2 we formulate the problem in Diophantine terms, and in part 3 review the standard methods for solving such equations. In parts 4 and 5 we discuss how to improve equation-solving performance by getting additional equations and additional constraints. In part 6 we discuss countermeasures, in part 7 another application of these ideas, and in part 8 some conclusions.

2. Previous work

A good deal of previous work has addressed the issue of protecting the confidentiality of individual data values in a database, given that certain statistics on that database may be obtained (Denning79, KamUllman77, Cox80, ChinOzsoyoglu81, Denning83). This work demonstrates that a variety of techniques must be guarded against. However, the emphasis of this work is on protection of individual data values, and does not address protection of other statistics on a database which may also be sensitive information (e.g., the number of alcoholics employed by a company). More importantly, none of this previous work addresses situations where there are only a few possible values for certain attributes (e.g. employee sex, number of children) and hence implicitly many fewer variables to contend with in solving equations for inferring data values (though (KamUllman77) does address a related question of inferences on data values known to be drawn from a certain limited integer range, but their database model is highly restrictive).

3. Problem definition

There are two important situations for Diophantine inference which we address. The "unknown-counts" one is when exact statistics queryable on some class (set) of items (objects) include:

the class size n
the mean with respect to the values of some attribute μ

the set of all possible values v_i for that attribute for the set, and
(practically speaking) the number of such values is small compared to
the size of the set

The "unknown-values" problem arises when what are known about a class (set) are:

the mean with respect to the values of some attribute μ

the number of occurrences n_i of each possible value for that attribute for the set, and (practically speaking) the number of such values is small

In either case we can write the linear Diophantine (as it is classically referred to) equation

$$(n_1 v_1 + n_2 v_2 + n_3 v_3 + \ldots) / D = \mu n / D$$

(where n is the set size, μ the mean, the v_is the possible values, and the n_is their occurrence counts. D is the greatest common divisor of the coefficients (the v_is in the unknown-counts case, the n_is in the unknown-values case); it is introduced to ensure make this is a Diophantine (integer-solution) problem for the unknowns. So for instance if the v_is are 3.14, 6.5, and 1.52, and μ is 531, then D is 0.02 and the equation can be written $107n_1 + 325n_2 + 76n_3 = 26550$.

In the unknown-counts case we can also write an additional equation to be solved simultaneously with the above:

$$n_1 + n_2 + n_3 + \ldots = n$$

In the unknown-counts case the n_is are unknown; in the unknown-values case, the v_i's. In this paper we shall be more concerned with unknown-counts problems than unknown-values. Usually values are more fixed than counts of sets that have those values, and often they can be looked up in books and found in other ways than querying the database

The unknown-counts and unknown-values problems can be cascaded together, allowing inference of the *mean* of an attribute. Suppose you know the values of one attribute, the number of items having specific values for a second attribute, and you know that the corresponding attribute values are in one-to-one correspondence, as will be discussed in 3.2. You can infer the mean (as well as median, mode, standard deviation, etc.) of the second attribute. Note also that both the unknown-counts problem and the unknown-values problem are always *guaranteed* to have at least one solution, the one representing the actual state of the data values. This means these problems are a bit nicer to solve than many important combinatorial problems, which may have no solution satisfying constraints.

As we said, these are methods for inferences on statistical databases, and only occasionally do they lead to compromise of individual data values. This occurs in the unknown-counts problem when a data value is found to occur only once in a set. But it is often hard for an individual attempting compromise to find a set that has one and no more than one occurrence of the data value. Other small counts can lead to small-set inferences (Denning et al 79). Similarly in the unknown-values problem, if we know the size of a set is 1, then if we know what item is represented, we know its value when we solve the Diophantine equations.

4. Solving the equations

Linear Diophantine equations (unlike most other Diophantine equations (Mordell69)) can be solved algorithmically. It can be shown that solutions in the unconstrained case (no bounds on any variables) form a set of evenly spaced lattice points in hyperspace. (Observe that if x_1 and x_2 are two solutions to a set of linear Diophantine equations, then $x_1 + k(x_2 - x_1)$, where k is an integer, is also a solution.) Thus they can be completely characterized by a single point x_0 and a set of basis vectors, multiples of which represent offsets from x_0 in each dimension. (ChouCollins82) provides an good overview of the state of art in solving linear Diophantine equations and sets of them, presenting two very different algorithms which are both refinements of a history of mathematical development. Generally speaking, the difficulty of solving the equations and number of solutions are very hard to predict, and vary markedly even among situations with the same number of equations on the same number of variables. At worst, the problem is exponential in the number of operations performed as a function of the number of terms in the equations, but algorithms can significantly improve this. Actually, the critical issue in such algorithms is usually space, in the form of "intermediate expression swell" of the coefficients, not time in the form of operations; see (ChouCollins82) for more details. But we do not really need the full generality of Diophantine solution vectors found by these algorithms; all we really want is the set of values that each component takes on in at least one solution, and some shortcuts may be possible in these methods.

However, the precise algorithms used to solve linear Diophantine equations, and their execution times, are not important issues. As with other compromise research, one can often assume that an individual attempting compromise has access to a good deal of computational power, and the important issue is the absolute prevention of vulnerability.

5. Additional evidence: multiple equations

There are a number of ways that additional equations can be generated for the same variables, thus providing even smaller solution sets and faster solution times.

5.1. Equations from moments

If for some set we know the standard deviation in addition, i.e. we know:

```
the set size n
the set mean with respect to some attribute μ
the set standard deviation with respect to the same attribute
all possible values for that attribute σ
```

we can write an additional linear Diophantine equation on the same variables:

$$n_1(u_1 - \mu)^2 + n_2(u_2 - \mu)^2 + n_3(u_3 - \mu)^2 + \ldots = n\sigma^2$$

to solve simultaneously with our original two. And if we can also get higher-order moments on that attribute, they generate additional linear equations as well. (Note these are all linear equations for the unknown-counts problem, but polynomials for the unknown-values problem.)

5.2. Equations from linked attributes

Another way to get an additional equation is when we know that the values for two numeric attributes u and v of some object are in one-to-one correspondence -- that is, when a particular value for u logically determines the value for v, and vice versa. (This does not necessarily mean a functional dependency in both directions, as the term is used in database research (Ullman80); the relationship may be an "extensional functional dependency", a one-to-one mapping in only a particular database state.) Then if we know for some class:

```
the set size n
the mean μu with respect to attribute u
the mean μv with respect to attribute v
all possible values ui for attribute u
all possible values vi for attribute v
the single fixed pairings of u values with v values
```

We can write three linear Diophantine equations to solve simultaneously:

$$n_1 + n_2 + n_3 + \ldots = n$$

$$(n_1u_1 + n_2u_2 + n_3u_3 + \ldots) / D_u = \mu_u n / D_u$$

$$(n_1v_1 + n_2v_2 + n_3v_3 + \ldots) / D_v = \mu_v n / D_v$$

If the u values are statistically independent of their correlated v values, the selectivities will tend to multiply in solving the combined equation. That is, the number of points satisfying the constraints and both of the two last equations will be the product of the number satisfying the constraints and each of the last two equations separately, divided by the number of points satisfying only the constraints.

As an example, suppose we know there are 58 ships in the British fleet in the South Atlantic. Suppose further that we know there are only three kinds of ships in that fleet, with lengths 440, 210, and 80 feet, and beams 52, 27, and 20 feet respectively. Suppose the mean length in the fleet is exactly 190, and the mean beam exactly 26. Then the equations are:

$$n_1 + n_2 + n_3 = 58$$
$$(440n_1 + 210n_2 + 80n_3)/10 = 190 * 58/10$$
$$(52n_1 + 27n_2 + 20n_3)/1 = 26 * 58/1$$

5.3. Equations from one-way maps

The relationship between attributes u and v need not be one-to-one to use this approach. A functional mapping in either direction is sufficient, provided as before that we know which values map onto which other values. We need twice as many variables as previously, however: those for u and those for v. Assuming the mapping is from u onto v, we then write an additional set of equations equating the count for each variable in v with the sum of the counts variables of u that correspond. So if variables 1 and 2 of u are the only ones that map onto variable 6 of v, we write $n_{v6} = n_{u1} + n_{u2}$. We then solve simultaneously the Diophantine set of the u equation, the v equation, the sum-of-counts equation, and these u-v interrelationship equations. (The exact mapping of values is often "domain knowledge" that can be looked up in books or inferred by common sense. If not, something still can be done. Solve for the u and v count variables separately in phase 1. Then in phase 2, the generation of solutions, throw out any u solution which can't be summed in some way to get at least one v solution. But this is slow.)

5.4. Equations from nonnumeric attributes

Even nonnumeric attributes can be used to calculate means if type checking is not enforced on a database. For instance, computing the mean of a character-string field may involve converting the first two characters into a 16-bit integer. This gives a one-way functional mapping which can be exploited with the above ideas if the nonnumeric attribute itself is the object of a functional mapping from the target attribute.

5.5. Equations from virtual attributes

As if the above methods weren't enough, there are even more powerful ways of generating equations. Attributes need not be stored directly, but can be derived by numerical operations on data like logarithms, exponentials, square roots, squares, reciprocals, absolute values; and for pairs of attributes, operations like sums, differences, products, quotients, etc. So a database system that permits such operations before calculating means provides many equations. And even more if operations can be cascaded: you could take the mean of the sum of the logs, the mean of the log of the sums, as well as the mean of the sum, the mean of the logs, etc.

5.6. Equations from database updates

Inserts, deletes, and updates involving single items of classes followed by recalculation of the mean do not provide any new equations. For instance, inserting an item with value b for some attribute into a class of size n with mean μ for that attribute always gives a new mean of $(n\mu + b)/(n+1)$. But if subclasses of indefinite size are inserted, deleted, or updated, powerful compromises are possible, extending (Ozsoyoglu and Ozsoyoglu 81) to our Diophantine domain. For example, the number of items with a certain value in a class is the number of items in the class minus the number of items in the set formed by deleting all items with that value from the class. Relational joins ((Ullman80), ch. 4) can disguise such conditions when they are permitted the user. For instance, to find how many items in a class have a given attribute, join its relation to another relation where that attribute is a key. Let the second relation have another field f which is numeric. Then by computing the mean μ_0 of the field f in the join, changing the entry in f for the target from v_0 to v_1, and finding the new mean μ_1, we can determine the target value as $n(\mu_1 - \mu_0)/(v_1 - v_0)$ where n is the size of the joined relation (and also the size of the original first relation).

Note this is a different sort of compromise-by-update than previously discussed (ChinOzsoyoglu79); here known updates compromise unknown a priori circumstances, not the other way around. Note also that mere inserts to a secondary relation can also have much the same effect as the abovementioned updates. Hence this compromise is a realistic possibility in a practical application, since most databases are designed to permit routine clerical transactions to add records easily. As an example, suppose we want to infer the salary distribution of a set of 10,000 employees. Suppose we know a set of values certain to include all salaries. For instance, it might be all multiples of |1000 from |5000 to |100,000, if we know it is company policy to only pay in such multiples. We then create a new relation whose key is this salary value and whose dependent part is a single attribute whose value is always 0 except for the first entry, for which it is 1. We then compute the mean of this attribute for the join of the main and new relation -- suppose it is .0074. Then there must be 10000 * (.0074-0) / (1-0) = 74 employees with that first salary value.

5.7. Equations from multiple factorizations

Even when sophisticated kinds of processing such as virtual attributes and updates are not permitted on a statistical database, and no functional mappings appear between attributes, additional evidence may be obtained another way. Contingency tables (as the term is used in statistics) may be set up, and solved for the values in cells from the marginals that represent sums of rows and columns (and levels, etc. if the tables are more than 2-dimensional). This requires a database in which an attribute whose means can be queried has itself a class partitioning whose means for other numeric attributes can be queried. Number theoretic investigations of magic squares are relevant here. As an example, suppose we have a military ships database with attributes ship nationality, ship class, and length. Assume a ship's nationality and class uniquely determine its length (i.e., there is a functional dependency from them to length). Our unknowns are the number of ships of each nationality-class pair. For each ship nationality we can write a linear Diophantine equation using the number of ships of that nationality and their mean length. For each ship class we can write a linear Diophantine equation using the number of ships of each class and their mean length. Thus we have two sets of equations involving the same unknowns in different ways.

6. Additional evidence: multiple constraints

Additional equations simplify phase 1 of the Diophantine solution algorithm. Additional evidence may also help in phase 2 in the form of additional constraints to rule out solutions as they are generated. As an example, consider information about the frequency distribution of values for an unknown-counts problem. If a statistical database allows you to compute the mode of a class, it should also give the frequency of that mode in order to quantify its significance, analogously to a standard deviation quantifying the significance of a mean. A mode frequency not only lets you eliminate a variable from your Diophantine equations (the variable for the mode value's count), but puts an upper bound on all other count variables. If the system won't tell you the mode frequency, bounds on it and other frequencies may be obtainable from other counts queryable on the database -- so for instance the number of destroyers with "Russia" as the value of their nationality attribute can't be any more than the total number of Russian ships. Or you may be able to solve more easily a Diophantine problem on a superset of the target set (remember solvability is sensitive to low-order bits of coefficients, which are mostly independent of the size of the set considered) and use these counts as upper bounds on the corresponding counts for the target set. Even when these methods won't work, frequencies can be estimated with "reasonable guesses" by experts familiar with the domain of the database, and also by general rules-of-thumb like Zipf's "Law" that the frequency of the kth most common item is proportional to 1/k. There are also absolute, mathematically provable bounds that can apply. For instance, the mode frequency of a

class can't be any less than the number of items in the class divided by the number of distinct values. Order statistics like median and quartiles are another source of constraints. For instance, knowing the size and median of a class tells you the exact number of items above and below that median value. Maxima and minima can also give constraints. If you know that the distribution of values for some attribute is even to some tolerance, and you know the density of values, then you can put bounds on the size of a set given its maximum and its minimum. If you combine this with the fact that the maximum of the intersection of a bunch of sets is the minimum of the maxima, and the minimum of the intersection is the maximum of the minima, you can sometimes infer narrow ranges of values for the items of an intersection, and hence tight upper bounds on the sizes of intersection sets.

Counts (and bounds on them) can be inferred by a large variety of indirect methods besides these. Ongoing work (RoweSIGMOD) of which this research is a small part is directed towards constructing a rule-based system to make such estimates, and estimates of other statistics, by a large collection of methods. Integrated inference systems of this sort can demonstrate synergistic effects from rule combinations. Note that such reasoning involves almost exclusively the high-order bits of numbers, whereas Diophantine analysis emphasizes number-theoretic considerations sensitive to low-order bits, and hence selectivities of non-Diophantine and Diophantine restrictions will tend to multiply. Thus both together can be powerful.

7. Countermeasures

As we have seen, Diophantine inferences can exploit many different sources of information. Thus restrictions on the kinds of single queries a user of a database can ask may not be much of an impediment to inference or compromise. Furthermore, protection by restricting queries with extreme (very large or very small) counts and query pairs with excessive overlap between target classes (cf.(ChinOzsoyoglu81)) are nearly useless. Diophantine compromise can work on classes of any size (though it tends to work better for smaller classes), and need only involve queries on one class (set). A third possible countermeasure, database partitioning, doesn't prohibit Diophantine analysis, just restrict the number of sets to which it can be applied.

So there is only one countermeasure left in the list of (ChinOzsoyoglu81): random perturbations of data and/or query answers (Traubetal81). (For our purposes, data perturbations amount to the same thing as answer perturbations.) Note the perturbations must be pseudo-random, as opposed to truly random, or the user could ask the same question many times and average the results to get an answer with much less variance. Note also the perturbations must be large enough so the user cannot enumerate a small set of possibilities for set sizes and means, and then solve each possibility separately, throwing away possibilities that yield no solutions consistent with the constraints. With some equations and/or constraints, the average yield, based on selectivities, of the a priori information may be significantly less than 1 record, hence many possibilities for means and sizes will be inconsistent. With large classes, rounding and/or truncation will occur in finding means (and even class sizes). Thus a kind of random perturbation may occur "for free" in many statistical databases when large sets are queried, and specifically adding perturbations may not be necessary for protection. But first, the calculation error may not be sufficient for protection. Second, truncation and rounding are not random but deterministic operations in nearly all systems (they are usually handled in hardware, which is rarely designed with any concern for security issues), and may be analyzable. For instance, the values may be known to be sorted before they are added, or there may be only one particular value with low-order bits that will be significantly rounded; in both these cases large chunks of possibilities can be eliminated, meaning much less protection from compromise. Clever users can create complex compromise methods based on detailed scenarios for rounding and/or truncation.

In should be noted that census agencies (e.g. (Cox80)), as far as we are aware, do not check for Diophantine compromises in published tables of aggregate data, even though the are subject to this weakness in principle. They are protected for the most part by small-range-of-values practicality restriction given in section 3, which only applies rarely to census data, primarily in regard to artificial numeric codes.

8. Other uses of Diophantine inference

These compromise methods can be used for good as well as evil. As we discuss in (RoweSIGMOD), even simple statistical calculations like counts and means can take enormous amounts of time in a very large database. But as computer technology progresses, more and more data is being stored. Many users of such data would be gladly willing to accept approximate answers to some statistical queries on these huge databases, answers based on inferences including our Diophantine methods, in return for greatly increased access speed and/or greatly decreased storage requirements -- particularly in the early stages of exploratory data analysis (Tukey77).

9. Conclusions

Diophantine compromise represents a serious new threat to the security of statistical databases. It can be applied to find exact sizes of subclasses of a given class of items in a database, given the size, mean, and set of values for that class -- things which are not intrinsically suspicious. While solving a single such Diophantine equation may generate many solutions, too many to be very compromising, there are a wide variety of ways to cut down the number of possibilities by getting both additional simultaneous equations and constraints. One major countermeasure is possible, random perturbations of answers supplied to the user, but it has disadvantages, in particular its degradation of answer quality and being subject to clever exploitation if not applied carefully. Clearly this topic deserves detailed further investigation.

10. References

Denning82, author "D. E. Denning", title "Cryptography and Data Security", publisher "Addison-Wesley", address "Reading, MA", year "1982"

Denningetal79, author "D. E. Denning, P. J. Denning, and M. D. Schwartz", title "The Tracker: a Threat to Statistical Database Security", journal "ACM Transactions on Database Systems", year "1979", month "March", number "1", volume "4", pages "76-96"

KamUllman77, author "John B. Kam and Jeffrety D. Ullman", title "A Model of Statistical Databases and Their Security", journal "ACM Transactions on Database Systems", volume "2", number "1", pages "1-10", year "1977"

Mordell69, author "L. J. Mordell", title "Diophantine Equations", publisher "Academic Press", address "New York", year "1969"

ChouCollins82, author "Tsu-Wu Chou and George E. Collins", title "Algorithms for the Solution of Systems of Linear Diophantine Equations", journal "SIAM Journal of Computing", volume "11", number "4", year "1982", month "November", pages "687-708"

Ullman80, author "J. D. Ullman", title "Principles of Database Systems", publisher "Computer Science Press", address "Potomac MD", year "1980"

OzsoyogluOzsoyoglu81, author "Gultekin Ozsoyoglu and Meral Ozsoyoglu", title "Update Handling Techniques in Statistical Databases", organization "First LBL Workshop on Statistical Database Management", booktitle "Proceedings", pages "249-284", year "1981", month "December"

ChinOzsoyoglu79, author "F. Y. Chin and G. Ozsoyoglu", title "Security in Partioned Dynamic Statistical Databases", organization "IEEE COMPSAC", booktitle "Conference Proceedings", year "1979", pages "594-601", month "June"

ChinOzsoyoglu81, author "Francis Y. Chin and Gultekin Ozsoyoglu", title "Statistical Database Design", journal "ACM Transactions on Database Systems", year "1981", month "March", number "1", volume "6", pages "113-139"

Traubetal81, author "J. F. Traub, H. Wozniakowski, and Y. Yemini", title "Statistical Security of a Statistical Data Base", institution "Columbia University Computer Science Department", month "September", year "1981"

RoweSIGMOD, author "N. C. Rowe", title "Top-down statistical estimation on a database", organization "ACM-SIGMOD", booktitle "Proceedings of the Annual Meeting, San Jose, California", year "1983", month "May", pages "135-145"

Cox80, author "Lawrence H. Cox", title "Suppression Methodology and Statistical Disclosure Control", journal "Journal of the American Statistical Association", year "1980", month "June", volume "75", number "370", pages "377-385"

Tukey77, title "Exploratory Data Analysis", author "John W. Tukey", publisher "Addison-Wesley", address "Reading, Mass.", year "1977"

Go to paper index