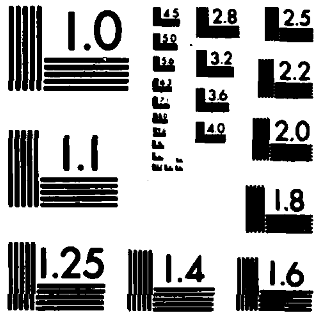
AD-A138 318   TOP-DOWN STATISTICAL ESTIMATION ON A DATABASE(U) NAVAL   1/1
POSTGRADUATE SCHOOL MONTEREY CA   N C ROWE JAN 84
NPS52-84-001

UNCLASSIFIED   F/G 12/1   NL

END
DATE
FILMED
3 84
DTIC


1.0
4.5
5.0
5.6
6.3
7.1
8.0
9.0
10.0
2.8
3.2
3.6
4.0
2.5
2.2
2.0
1.1
1.8
1.25
1.4
1.6

2

NPS52-84-001

NAVAL POSTGRADUATE SCHOOL

Monterey, California

TOP-DOWN STATISTICAL ESTIMATION ON A DATABASE

Neil C. Rowe

January 1984

Approved for public release; distribution unlimited

Prepared for:

Chief of Naval Research
Arlington, VA 22217

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Commodore R. H. Shumaker
Superintendent

D. A. Schrady
Provost

The work reported herein was supported in part by the Foundation Research Program of the Naval Postgraduate School with funds provided by the Chief of Naval Research.

Reproduction of all or part of this report is authorized.

This report was prepared by:

Neil C. Rowe
NEIL C. ROWE
Associate Professor of
Computer Science

Reviewed by:

David K Hsiao
DAVID K. HSIAO, Chairman
Department of Computer Science

Released by:

Kneale T. Marshall
KNEALE T. MARSHALL
Dean of Information and
Policy Sciences

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NPS52-84-001	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Top-down Statistical Estimation on a Database		5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Neil C. Rowe		8. CONTRACT OR GRANT NUMBER(s) N00039-82-G-0250
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61152N; RR000-01-10 N0001484WR41001
11. CONTROLLING OFFICE NAME AND ADDRESS Chief of Naval Research Arlington, VA 22217		12. REPORT DATE January 1984 13. NUMBER OF PAGES 20
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

databases, statistics, counts, means, exploratory data analysis, sampling, artificial intelligence, expert systems, production systems, interval mathematics, estimation, database query languages, data abstraction, rule-based systems

20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The size of data sets subjected to statistical analysis is increasing as computer technology develops. Quick estimates of statistics rather than exact values are becoming increasingly important to analysts. We propose a new technique for estimating statistics on a database, a "top-down" alternative to the "bottom-up" method of sampling. This approach precomputes a set of general-purpose statistics on the database, a "database abstract", and then uses a large set of inference rules to make bounded estimates of other, arbitrary statistics requested by users.

(continued on next page)

unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Continued from block 20

The inference rules form a new example of an artificial-intelligence "expert system". There are several important advantages of this approach over sampling methods.

Top-down statistical estimation on a database

Neil C. Rowe
Department of Computer Science
Stanford University
Stanford, CA 94305

This work is part of the Knowledge Base Management Systems Project, 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.

This paper is to appear in the Proceedings of the ACM-SIGMOD International Conference on Management of Data, May 1983.

Table of Contents

Abstract

The size of data sets subjected to statistical analysis is increasing as computer technology develops. Quick estimates of statistics rather than exact values are becoming increasingly important to analysts. ~~We~~ The author proposes a new technique for estimating statistics on a database, a "top-down" alternative to the "bottom-up" method of sampling. This approach precomputes a set of general-purpose statistics on the database, a "database abstract", and then uses a large set of inference rules to make bounded estimates of other, arbitrary statistics requested by users. The inference rules form a new example of an artificial-intelligence "expert system". There are several important advantages of this approach over sampling methods.

1. Introduction

In statistical analysis of data it is often the storage and access that are critical to performance, not the statistical analysis routines themselves [1, 19]. For very large data sets this suggests special attention to characteristics of secondary storage devices. Forthcoming devices such as videodisks, bubble memories, and special processors may improve time and space efficiency in the years ahead, but it will be difficult for such improvements to surpass those in processing power due to mere integration and scaling down of components in VLSI designs. It seems likely that secondary storage access will continue to be the bottleneck in statistical analysis of computer data.

An idea, however: perhaps we can trade off processing speed for storage. Statistical databases often have much redundancy in attribute values and statistics that can be predicted by other attribute values and statistics. If we can formulate this redundancy computationally, we may be able to put it into cheap processors and programs instead of expensive secondary storage. We may be able to create a much smaller database, a "database abstract" [13], that preserves most of the information content of the original data. We can then use a number of "reasonable guess" rules from statistics (together with some "special case" checks) to infer (impute) statistical characteristics of the original data from the abstract -- that is, by an artificial-intelligence "expert system".

This approach shares some ground with "exploratory data analysis" (EDA) [20]. We wish to capture the essence of a large set of data, and to summarize that essence; a number of EDA techniques will help. But also we can support EDA activities by creating small, partial database abstracts for data analysts to explore rather than full databases. The advantages are speed of query answering and storage savings (perhaps allowing employment of a small personal computer); the disadvantage is that answers to statistical questions may be approximate rather that exact. But in EDA, estimates are often perfectly satisfactory.

2. Overview of the approach

Figure 2-1 shows our approach. We start with a user and a database. The database is preprocessed to create a "database abstract", a collection of simple statistics (mean, maximum, mode frequency, etc.) on important and frequently asked-about sets in the database. The user talks through an interface to the database abstract, asking it the same statistical questions he would ask the full database if he had more time (or space). If an answer is not in the database abstract, an estimate and bounds on that estimate are inferred from rules. These rules, several hundred in number, form an artificial-intelligence production system [3], and represent distinct pieces of domain-independent knowledge about statistical estimation from a variety of of very different sources. Some (e.g. EDA rules) are justified on intuitive criteria, but most can be derived mathematically, by theorem-proving methods, non-linear optimization, and maximum entropy theory. Rules can also be suggested by analogies to other rules.

We have an implementation; see the Appendix for a demonstration. Walker [21] has also studied database abstracts, but his work differs in two fundamental ways: (a) it ignores statistical aggregates, and (b) it addresses the conditions for exact answers, not what you *can* say about an inexact answer.

3. What's wrong with sampling

Our approach provides a new alternative to random sampling for exploring a large data population at low cost. There are several serious disadvantages to sampling:

1. The data may already be partially aggregated in means, counts, etc., as when there are large amounts of data from instrument readings in laboratory experiments. Sampling then is tricky, and may not be possible without detailed information about the preaggregation.

2. Sampling is inefficient in paging. Suppose we randomly sample m items from a database of p pages with k items per page on the average. We can model the apportionment of sample items to pages as a Poisson process where the expected number of pages examined in sampling is $p(1-e^{-m/p})$, approximately $p(1-(1-m/p)) = m$ for small m. This will generally be considerably larger than m/k, the sampling ratio. So a sample of one thousandth of a database with a hundred items to a page will access one tenth the pages, not one thousandth.

   Because of this, Morgenstein [10] has proposed randomized page assignment for databases. But this faces many problems. Randomization is very complex when joins are involved. Randomization is difficult to maintain on updates without reorganizing the whole database. Most important of all, randomization of database pages is antithetical to optimization, and will markedly degrade performance for nonstatistical queries on the same database. Thus it is inappropriate for most databases, which are multi-purpose.

3. Random sampling is also inefficient with indexes when they are used. Since one has no assurance that indexes list items in a statistically random order, usually one must assemble pointers to all the items in the set to choose randomly among them. This may require much temporary storage space for the pointers, and many index page accesses, depending on how the index is stored.

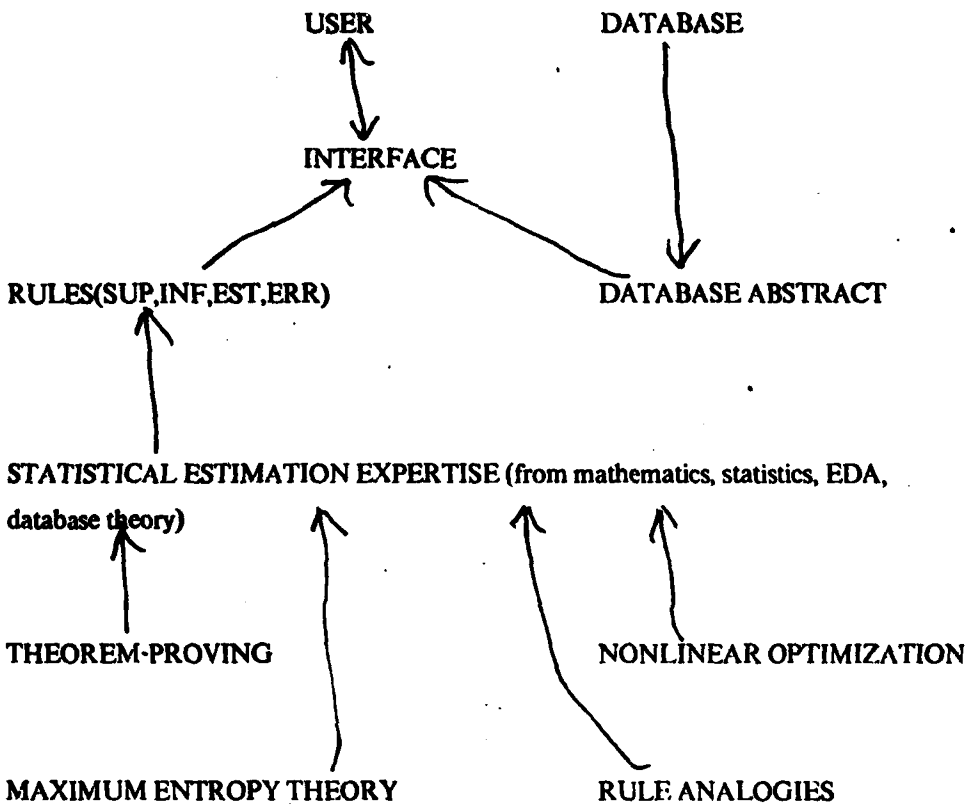


Figure 2-1: Block diagram of the system

4. Sampling is a poor way to estimate extremum statistics like maximum, mode frequency, and bounds on distributional fits. Many applications can exploit such statistics.

5. Similarly, sampling is poor for obtaining absolute bounds on statistics.

6. Sampling is "brittle": given a sample, it is hard to speculate about properties of a subset, superset, or sibling of that set. This is serious because an EDA user may not be sure what he wants to look at, or want to explore clusters of related sets in the course of data analysis, or may simply make mistakes.

   A random sample of a set is unlikely to be a random sample of a superset or sibling. (We may be able to *use* it in a stratified random sample design for the superset or sibling, but such designs are highly data-specific.) So if we choose a set too restrictive in our initial query to a database, we must sample all over again for a superset, with all the paging inefficiences doubled. On the other hand, we must also resample if we choose too large a set to start with, for otherwise we're sampling a sample, a poor statistical design. (Erroneous generalizations are suggested from the accidental correlations of a sample of a sample.)

7. A random sample doesn't have semantics. That is, it is of interest only as a random sample and not as an entity in its own right as a set created by set intersections might be.

8. Sampling in the artificial world of a database makes less sense than in natural, real world populations. Databases are "closed worlds" containing finite amounts of well-characterized data, *where each datum is (in principle at least) equally accessible.* Real world populations, as for example populations of people, have fluid boundaries and members of differing accessibilities. The idea of knowing the mean exactly for some set only makes sense in a (nonsampling) database -- we can only be more or less sure of the mean in the real world.

The architecture of a database abstract of precomputed statistics plus some inference rules compares favorably:

1. The database abstract *is* aggregated data.

2. Once set up, the database need not be paged at all. Paging of the abstract will be low (see 7.2). And setup can be quite efficient -- each page can be fetched in turn, and every item on a page examined.

3. Database index pages are used efficiently for the same reasons.

4. Inference rules can handle extremum statistics (e.g. maximum) nearly as well as normative statistics (e.g. mean).

5. Inference rules explicitly infer bounds.

6. Inference rules gracefully handle extensions of a set to supersets and restrictions of a set to subsets; many rules explicitly address these cases.

7. Sets in the database abstract have an explicit semantic

8. Inference explicitly takes into account the closed world nature of databases.

4. About rules

We now discuss the inference rules used with the database abstract.

4.1. The querying language

We use a set-descriptive language for queries, in the style of relational algebra (as opposed to predicate calculus). Figure 4-1 gives a formal specification. Queries S(C,F,R) consist of a statistical aggregate operator S applied to three arguments: a database relation R, a class C (or set) of items within that relation, and an attribute F (or field) of those items. Rules are substitutions for a query of a particular form by a mathematical function of the results of other queries.

4.2. Describing answers

Rules may give exact or, usually, inexact answers for queries. When inexact for a statistic S, we describe the probability distribution with four items of information (and always four items):

- An upper bound on S (alias SUP-S)
- A lower bound on S (alias INF-S)
- An estimate of S (alias EST-S)
- The expected standard error of that estimate (alias ERR-S)

The bounds are absolutely guaranteed. The estimate is guaranteed (see 8.2) within some criterion (say 10%) on some finite number of queries chosen by the system designer.

4.3. The rule taxonomy

There are on the order of five hundred rules for this domain. Rules can be categorized along four different dimensions:

- the statistic dimension (whether it is mean, maximum, standard deviation, mode frequency, etc.)
- the characteristic dimension (whether it is an exact answer, a bound, an estimate, or a standard error of an estimate)
- the computational dimension (what form of queries it applies to) -- see Figure 4-2
- the derivation dimension (where we got it) -- see Figure 4-3

An arbitrary single-statistic query is in the form S(C,F,R) where

- S is a single-field aggregate statistical operator
- C is a set of tuples (a "class")
- F is field (which may be virtual)
- R is a relation

S, C, R, and F are specified as follows:

- S :: SIZE | MAX | MIN | MEAN | SIGMA | MEDIAN | MODE | SIZEUNIQUE | MODEFREQ | MODEFREQ2 | LEASTFREQ | MAXGAP | MINGAP | MAXEVENDEV | MINEVENDEV | KEYS
- C :: first-order-class | NOT(C) | AND(C,C) | OR(C,C)
- F :: schema-fieldname | ARITHOP(F) | ARITHOP2(F,F) | ABSTRACTION(F) | VECTORIZE(F,F)
- R :: relation-name | JOIN(R,R,F)
- ARITHOP :: SQUARE | SQRT | LOG | ANTILOG | RECIPROCAL | ABS
- ARITHOP2 :: PLUS | DIFFERENCE | TIMES | QUOTIENT | MAX | MIN

Figure 4-1: The query language

1. answer-syntax rules

2. other-statistic rules

3. tuple class decomposition

    a. subset inheritance

    b. set intersection

    c. set union

    d. set complement

4. field expression decomposition

    a. unary 1-to-1 operators

    b. other unary operators

    c. binary operators

    d. vectorization of corresponding values

    e. operations with constants

5. join decomposition

6. unusual inheritances

    a. upwards inheritance

    b. lateral inheritance

    c. diagonal inheritance

    d. value-level inheritance

    e. rule inheritance

7. canonical query rearrangement

8. handling nulls

Figure 4-2: The computational dimension of rules

1. basic mathematics

2. laws of probability and statistics

    a. definitions

    b. theorems

    c. extrema of definitions and theorems

        i. bounds on values

        ii. independence and linearity assumptions

        iii. exploratory data analysis

        iv. nonlinear optimization

        v. maximum-entropy assumptions

3. database theory

    a. functional dependencies

    b. theory of inference compromise

        i. small samples and trackers

        ii. exploiting uniqueness

        iii. Diophantine (integer) equations

4. sampling theory on database

    a. load time

    b. run time

5. reasoning from prototypical examples

Figure 4-3: The derivation dimension of rules

As an example, consider the rule that the largest item in the intersection of two sets cannot be any larger that the minima of the maxima of the two sets for some numeric attribute. On the statistic dimension, this is a rule for a maximum statistic; on the characteristic dimension, this is a SUP; on the computational dimension, this is a set intersection rule of the tuple-class-decomposition type; and on the derivation dimension, this is a theorem derivable from basic mathematics. Or consider the rule that the mode of a set can be estimated as the mode of its largest subset. This is a mode rule on the statistic dimension; an EST on the characteristic dimension; an upwards inheritance rule [14] on the computational dimension; and a maximum entropy rule on the derivation dimension.

4.4. A production system architecture

The production-system architecture frequently used in artificial intelligence expert systems [3] is strongly suggested here by classical signs:

- The derivation and computational dimensions of the taxonomy show much variety.
- Rules represent highly modular pieces of knowledge.
- So much heterogeny enables synergistic effects where several very different rules together lead to surprising results that could not be foreseen by examining any of the rules independently.
- There is no "complete" set of rules. There are always special cases formulatable in a more powerful additional rule, perhaps automatically (see 6.4); additional rules can improve performance for particularly common or important queries. And in moving to a smaller computer we may want to remove rules that aren't sufficiently cost-effective.
- Production rules let us make a conceptually clean break between database-independent knowledge (the rules) and database-dependent (the database abstract).

5. A processing example

To make things clearer, we show an example of a number of quite different rules contributing to answering a query. Suppose a query asks the number of American tankers in a database of ship information. Assume that basic statistics on American ships and tankers separately are available, but none for American tankers. With this information, we cannot uniquely determine the size of the set intersection. But for a user who is satisfied with an estimate, we can try the following lines of reasoning:

1. An intersection of two sets can't be any larger than the smaller of the two. For 1000 American ships and 5000 tankers, there cannot be more than 1000 American tankers (a SUP).
2. Set sizes are nonnegative, so there are no fewer than zero (an INF).
3. For an estimate, use a simple log-linear model. If there are 20,000 ships in the database, expect 5000 * 1000 / 20000 = 250 American tankers (an EST).

4. For the standard error (ERR) of this estimate, find the standard deviation of a truncated exponential distribution (the maximum-entropy distribution) consistent with SUP, INF, and EST. The math is complicated and we won't go into it here.

5. If the mode tanker nationality occurs 140 times, there cannot be more than 140 American tankers. Analogously, find the mode frequency of the ship type field for the set of American ships.

6. Similarly, if the least frequent ("antimode") tanker nationality occurs 5 times, there cannot be fewer than 5 American tankers.

7. Occasionally we may know a superset containing a set intersection. For instance, all American tankers may have identification codes of a certain type, hence the number of ships with those type codes is an upper bound on the number of American tankers.

8. We can infer bounds and estimates from "range analysis" of arbitrary numeric attributes. Suppose the length range of tankers is from 300 to 1000 feet and that of American ships 50 to 400 feet. Then American tankers must have lengths 300 to 400 feet.

   If we have statistics on length subdivisions of ships, we can bound the size of this set. Statistics for any range that includes 300-to-400, like 260-to-420, will do too (we may partition on deciles, etc.). We can lower this upper bound if we know the maximum distance between successive American tankers in this range, or the maximum deviation of values across the range from some standard distribution. For instance, we may know that values are never off more than 10% of range from where they would be in a perfectly even distribution, in which case for our 260-to-420 example range, the 300-to-400 range can contain no more than 100/160 + .1 + .1 = 82.5% of the points in the 260-to-420 range.

9. We can set up Diophantine (integer-solution) equations for mean computations and find sets of distinct points (as opposed to ranges) consistent with those equations. Suppose there is a numeric code for each of four basic ship types (freighters, tankers, bulk carriers, and miscellaneous) -- say 2, 3, 8, and 11 respectively. Suppose the mean of this code field for American ships is 2.3. Then we can solve simultaneously the two equations:

   $$2n_1 + 3n_2 + 8n_3 + 11n_4 = 2.3 * 1000$$
   $$n_1 + n_2 + n_3 + n_4 = 1000$$

   where variable n_2 represents the number of American tankers, what we are looking for. For these equations there can only be a number of American tankers which is a multiple of 3 up to 300, excluding 297.

   In general, there are many ways to include extra information leading to fewer solutions. We discuss this inference method in detail in [15].

Once obtained, this query answer is useful for many other related queries. The size of the union of two sets is the size of the first set plus the size of the second set minus the size of their intersection. The size of the intersection of three sets can be obtained from statistics on the intersection of two of those sets first, together with statistics on the third.

6. More about rules

6.1. Conflict resolution

As with other production systems, we must specify action when more than one rule applies to the same query. Bounds rules are easy -- we just apply all of them separately, and intersect the answer ranges. With estimates and standard errors we have two options: (1) define priorities among rules (perhaps always give the most specific priority), or (2) execute the rules separately, model the results as truncated normal distributions, assume statistical independence, and use the standard formulas, which in the two-rule case are:

$$\sigma^2 = 1 /[(1/\sigma^2{}_1) + (1/\sigma^2{}_2)]$$
$$x = \sigma^2 (x_1/\sigma_1^2 + x_2/\sigma_2^2)$$

6.2. Subqueries and caching

Queries may invoke many subqueries. To avoid infinite loops, we check subqueries against a stack of active queries, terminating analysis if a match is found. For efficiency in addition, we check new queries and subqueries against cache of previous queries and their answers. Cached answers may be saved over a session if users tend to concentrate on particular sets, kept at the end of a session for the next one, or pooled among a user group with similar interests.

6.3. Rule compilation, lower level

"Levels of knowledge" occur frequently in artificial-intelligence expert systems [5]. Upper levels represent general but hard-to-use knowledge that can be compiled in a computationally expensive operation into a more efficient form.

As suggested by part 5, rules can work on different subqueries in parallel. Each can be assigned a processor, and subquery answers pooled in a common cache. In addition, rule condition testing can be made more efficient by a decision tree [9]. Usually large classes of rules can be eliminated by inspection; for instance, a query involving only intersection of sets and a database attribute doesn't need rules on set unions or complements, or virtual attributes.

6.4. Rule compilation: upper level

Another kind of rule compilation is the creating of rules from underlying theories. There are four basic approaches: rearrangement of an existing rule or functional composition of existing rules, symbolic optimization via theorem-proving to get bounds rules, entropy maximization to get estimates and standard-error rules, and analogies to previous rules to get both. All can be automated in a symbolic algebra system to varying degrees.

Given the maximum, minimum, mean, and median of a set, what is the largest possible value of the standard deviation? That is a SUP question for our system, but it is also a quadratic optimization problem. There are standard solution techniques [4] given exact values for certain statistics. But with only approximate values denoted by ranges we require a kind of symbolic optimization which is much trickier, more like theorem-proving, and whose difficulty varies markedly from case to case.

Rules for estimates and standard errors are another thing altogether. In information systems in general, the best guess for a parameter is that with least information content [18]. Formalizing this leads to the calculus of variations and a general solution involving Lagrange multipliers (see appendix to [18]). For instance, if we know the maximum, minimum, mean, and standard deviation of a set, the maximum entropy distribution is of the form $ae^{b(x-c)^2}$, where the constants a, b, and c can be determined uniquely. From this concrete distribution we then calculate any statistic we want to estimate.

Rules can also be found by analogy. The most obvious examples are maximum and minimum rules, where additions are replaced with subtractions, maxima with minima, and minima with maxima in the text of the rule. [8] gives many ideas along these lines. But analogies can be misleading, and rules postulated must always be rigorously checked.

6.5. Algebra on quadruples

With inexact answers are expressed as quadruples, we need an algebra for arithmetic operations on them. For bounds, the first half of the quadruples, we can use ideas from interval analysis [11, 12], a branch of numerical analysis. For estimates and standard errors, we can interpret the quadruples as a truncated normal distribution (the maximum-entropy assumption), and use the formulas of [6]. (Some adjustments must be made to his formulas to model closed world effects, like the mean of the sum of corresponding values for two numeric fields being *exactly* the sum of the means of the two fields instead of just an approximation, because one is drawing without replacement from a population.)

6.6. The database as a last resort

The accuracy of estimates by these methods may vary considerably. When an estimate is unsatisfactory, the user should be able to go to the full database and get the exact answer. (And then cache it in the database abstract, possibly reducing future database querying.) The database abstract and rules could be at local nodes in a highly distributed system, say within "smart" terminals, with the database at a remote site.

6.7. Easy extensions

Nulls that represent unknown values can be treated by taking statistics on the nonnull portions of the set, and inferring upwards to characteristics of the full set including the nulls. Inexact data can be formulated as quadruples, and treated by our special algebra directly.

7. The database abstract as a database

A database abstract plus rules can provide significant savings in both space and time over use of a full database for statistical computation.

7.1. Storage space

Database abstract entries are best grouped by sets. A bit string header for each set can indicate which statistics are kept and to how many bits of accuracy, and the values can follow in a compact form. Sets can be accessed by an index on their names; any set with at least one stored statistic will be put in this index.

Rule storage should be comparatively negligible. Rules are short and simple, and use very few symbols which can be encoded in very few bits, using the taxonomy. Triggering conditions can be compacted in decision trees, with potential parallelisms indicated by a few additional pointers; since we anticipate 1000 or so rules in even sophisticated systems, pointers need not be large. The rule interpreter itself need not be large.

Though the database abstract is a compression of a database, other compression methods may apply too. We can store low order bits for a statistic value, and infer high order bits by inference rules. That is, use a statistic on American ships as a "base register" value, and keep only the "offset" to the American tankers value with the statistics for American tankers.

The number of database abstract entries necessary to achieve a certain level of answer accuracy is difficult to say, though we are developing a theory. But note information theory cannot be cheated: a database abstract can only contain a limited number of bits, for answering with limited accuracy a limited number of nonredundant queries.

7.2. Time considerations

Our system can be quite fast if desired. Cached information from previous queries makes a big difference. We expect page faults to be low for the database abstract because:

- it will be smaller (generally, much smaller) than the full database, and all or part of it might reside permanently in main memory

- usually there are not many sets relevant to a query (just the ones explicitly mentioned in it), hence many fewer retrievals than for the same query on the full database

- putting all the statistics for the same set on the same page greatly increases locality of references

8. Loading the database abstract

Choosing which statistics on what sets to store in the database abstract involves art as well as science, but some guidelines are possible.

8.1. Chosing the first-order classes

Sets representing simple concepts, e.g. "tankers", "American ships", "ships in the Mediterranean", are what we call first-order sets. While their statistics can sometimes be close to statistics on entire database relations (important things to have a lot of information for), it is statistically unlikely this will be a good estimate in general. So we must keep many statistics about first-order sets explicitly in the database abstract. Then when there is sufficient extra room we can include second-order sets (the intersections and unions of two first-order sets) and higher-order sets. Our approach works best for databases where attributes are correlated only is simple ways and few such higher-order sets are needed to capture subtleties. Note if there are n first-order classes there are $O(n^2)$ intersections of any two of those, $O(n^3)$ intersections of any three of those, and so on -- numbers increase rapidly.

First-order sets may be dictated by the needs of a user community. When choice is possible, they should be large enough to matter (say 10 items or more), and should represent reasonably even subdivisions of a database, consistent as possible with the way human beings cluster concepts.

8.2. Closed world reasoning

A useful trick for setup of the database abstract is to only enter "unusual" statistics, defined to mean that the rule-inferred value is within, say, 10% of its actual value. Since this involves effort in advance, we only check this for a limited "guarantee" set of queries. Generally this means only queries on sets larger than a certain minimum and relatively simple in description (like all queries involving three or fewer sets). So we have a new and powerful inference rule for answering queries, the "Closed World Rule": if a query is in the guaranteed query set and the answer is not in the database abstract, then the answer found by inference rules is within 10% (for else it would have been loaded).

8.3. Monotonicity

A complication of the preceding rule is that when new information is placed in the database abstract it may lessen the accuracy of answers to previous queries, causing what we call nonmonotonicity. For example, suppose we estimate the mean of the intersection of three sets by the average of the means of the three sets. Suppose we add to the database abstract the mean of the intersection of two of those sets, where this mean is far off from the mean of three sets taken together. We will get a poorer answer for the mean of the three sets after adding this new information. An answer that was within 10% before and did not need to be represented explicitly in the database abstract may now need to be, and the closed-world rule cannot hold.

There is a way out if we can impose a consistent partial ordering on all queries, where query A is partially ordered with respect to to query B if B is a subquery generated in processing A. If we then load the database abstract in some linearization of the partial ordering we cannot interfere with the closed-world rule. This means loading first-order set statistics before second-order, second-order before third, and so on. It also means ordering different statistics on the same sets, as mean before maximum, and standard deviation before mean.

A consistent partial ordering does require restriction of the class of possible inference rules. We are trading off the power of a number of miscellaneous "backwards" inference rules for one really powerful rule, the Closed World rule. (We can still use the backwards rules in answering queries, just not during loading.)

9. Handling updates to the database

When the full database is updated, the database abstract must remain consistent. This is not a problem for much statistical data since much of it is never updated [19] But we can handle it by a set of update rules; [7] provides a formal framework for developing them. Some updates are much easier to handle than others (e.g. to mean, standard deviation, set size), only requiring knowledge of the value updated, whereas others (e.g. median and mode) usually require expensive recomputation on the full database and hence may not be practical.

10. Evaluation

We do not have the room to discuss the complex issue of evaluation of our system here. In [16] we show for a particular database that our methods compare favorably to several simpler alternatives in regard to both space and time, without sacrificing large amounts of accuracy.

11. Extensions and applications

- We can add various kinds of correlation statistics and rules to estimate them.
- We can include reasoning about possible distinct values.
- We can use graphics or linguistic hedges to capture fuzziness of query answers.
- When sufficiently standardized, the database abstract and rules can be put into hardware.
- Our set-size rules can estimate selectivities for general query optimization (cf. [2]).
- Rules provide ideas for compromise methods for work in inference security of statistical databases [15]. But more importantly, our system provides a testbed for this research, much of which has focused on small numbers of inference rules in compromise, ignoring synergism between quite different sorts of rules.
- Rules raise important issues for artificial intelligence. They allow extension of the notion of inheritance to statistical properties, filling a key gap in representation-of-knowledge research, and demonstrating the adequacy of sets as the building block of a knowledge representation system [14].

12. Conclusions

This work explores a new area for rule-based expert systems, the domain of statistical computation on databases. We have made a start, but clearly further study and experimentation is needed to choose the most effective rules and most useful abstraction methods for an application. We are optimistic that this work will provide a new and powerful resource for statistical studies of data.

Appendix: The working program

The database abstract includes simple statistics on first-order (single-word-name) sets. No closed-world guarantees are offered. No correlations between attributes are exploited. The system does not actually understand English; the formal query has been paraphrased for understandability. For further details of the implementation, consult [17].

```
How many French ships of type ALI are there?
(GUESS: 6.2 GUESS-ERROR: 2.3 UPPER-LIMIT: 10 LOWER-LIMIT: 3)
(ACTUAL ANSWER IS 7)
[Which means: guess is 6.2 such ships, with associated error of 2.3; and
there are no more than 10 and no fewer than 3 such ships.]
```

```
What's the mean longitude of a Liberian tanker of type END?
(GUESS: 45.4 GUESS-ERROR: 34.5 UPPER-LIMIT: 168 LOWER-LIMIT: 3)
(ACTUAL ANSWER IS 47.4)
```

How many type ALI tankers are either French or Italian?
(GUESS: 12.6 GUESS-ERROR: 3.3 UPPER-LIMIT: 63 LOWER-LIMIT: 3)
(ACTUAL ANSWER IS 14)

What's the frequency of the most common tanker class
among the French, Italian, American, and British?
(GUESS: 18.5 GUESS-ERROR: 2.2 UPPER-LIMIT: 25 LOWER-LIMIT: 15)
(ACTUAL ANSWER IS 18)

What's the mean longitude for Liberian ships of type ALI
not in the Mediterranean?
(GUESS: 49.6 GUESS-ERROR: 42.4 UPPER-LIMIT: 176 LOWER-LIMIT: 6)
(ACTUAL ANSWER IS 44.75)

What's the mean distance of ALI-type ships from 30N5W?
(GUESS: 51.0 GUESS-ERROR: 12.3 UPPER-LIMIT: 57.1 LOWER-LIMIT: 6.0)
(ACTUAL ANSWER IS 42.34673)
[Here only statistics on latitudes and longitudes are known, and the
answer must be computed from them.]

What's the most common registration-nationality region
for type ALI ships currently in the Mediterranean?
(GUESS: 46.6 GUESS-ERROR: 9.3 UPPER-LIMIT: 78 LOWER-LIMIT: 26)
(ACTUAL ANSWER IS 37)
[We assume statistics on nationality-region are not kept, but the
function from nationality to nationality-region is fully defined.]

What's the mean longitude for type ALI ships in the Adriatic?
(GUESS: 38.8 GUESS-ERROR: 27.8 UPPER-LIMIT: 179 LOWER-LIMIT: 3)
(ACTUAL ANSWER IS 21.93939)
[We assume nothing whatsoever is known about the Adriatic except that it
is a part of the Mediterranean.]

References

[1] Doug Bates, Haran Boral, and David J. DeWitt, .
A Framework for Research in Database Management for Statistical Analysis.
In *Proceedings*, pages 69-78. ACM SIGMOD International Conference on Management of Data, June, 1982.

[2] Stavros Christodoulakis.
Estimating Selectivities in Data Bases.
Technical Report CSRG-136, University of Toronto, December, 1981.

[3] Randall Davis and Jonathan King.
An Overview of Production Systems.
In E. W. Elcock and D. Michie (editors), *Machine Intelligence 8*, pages 300-334. Wiley, New York, 1976.

[4] Philip E. Gill, Walter Murray, and Margaret H. Wright.
Practical Optimization.
Academic Press, New York, 1981.

[5] Peter E. Hart.
Directions for AI in the Eighties.
ACM SIGART Newsletter (79):11-16, January, 1982.

[6] Edward B. Haugen.
Probabilistic Approaches to Design.
Wiley, New York, 1968.

[7] S. Koenig and R. Paige.
A Transformational Framework for the Automatic Control of Derived Data.
In *Proceedings of the 7th Meeting*, pages 306-318. International Conference on Very Large Data Bases, Cannes, France, 1981.

[8] Douglas B. Lenat.
The Nature of Heuristics.
Artificial Intelligence 19:189-249, 1982.

[9] J. McDermott, A. Newell, and J. Moore.
The Efficiency of Certain Production System Implementations.
In D. A. Waterman and F. Hayes-Roth (editors), *Pattern-Directed Inference Systems*, pages 155-176. Academic Press, New York, 1978.

[10] Jacob P. Morgenstein.
Computer Based Management Information Systems Embodying Answer Accuracy as a User Parameter.
PhD thesis, University of California at Berkeley, December, 1980.

[11] Karl Nickel.
Triplex-Algol and Its Applications.
In E. Hansen (editor), *Topics in Interval Analysis*, pages 10-24. Clarendon Press, Oxford, 1969.

[12] L. B. Rall.
Interval Analysis: A Tool For Applied Mathematics.
Technical Report 2268, University of Wisconsin Mathematics Research Center, 1981.

[13] Neil C. Rowe.
Rule-Based Statistical Calculations on a Database Abstract.
In *Proceedings*, pages 163-176. First LBL Workshop on Statistical Database Management, Menlo Park CA, December, 1981.

[14] Neil C. Rowe.
Inheritance of Statistical Properties.
In *Proceedings of the National Conference*, pages 221-224. American Association for Artificial Intelligence, Pittsburgh PA, August, 1982.

[15] Neil C. Rowe.
Diophantine Compromise of a Statistical Database.
In *Three Papers on Rule-Based Estimation of Statistics on a Database*, chapter 3. Stanford University Computer Science Department report 948, 1982.

[16] Neil C. Rowe.
Some Experiments in Evaluation of an Expert System for Statistical Estimation on Databases.
submitted to the Second Workshop on Statistical Database Management, February 1983.

[17] Neil C. Rowe.
Rule-based Statistical Calculations on a Database Abstract.
PhD thesis, Stanford University, June, 1983.

[18] John E. Shore and Rodney W. Johnson.
Properties of Cross-Entropy Minimization.
IEEE Transactions on Information Theory IT-27(4):472-482, July, 1981.

[19] Arie Shoshani.
Statistical Databases: Characteristics, Problems, and Some Solutions.
In *Proceedings of the 8th Meeting*, pages 208-222. International Conference on Very Large Data Bases, 1982.

[20] John W. Tukey.
Exploratory Data Analysis.
Addison-Wesley, Reading, Mass., 1977.

[21] Adrian Walker.
On Retrieval From a Small Version of a Large Data Base.
In *Proceedings of the 6th Meeting*, pages 47-54. International Conference on Very Large Data Bases, 1980.

INITIAL DISTRIBUTION LIST

Defense Technical Information Center 2
Cameron Station
Alexandria, VA 22314

Dudley Knox Library 3
Code 0142
Naval Postgraduate School
Monterey, CA 93943

Office of Research Administration 1
Code 012A
Naval Postgraduate School
Monterey, CA 93943

Chairman, Code 52Hq 40
Department of Computer Science
Naval Postgraduate School
Monterey, CA 93943

Associate Professor Neil C. Rowe, Code 52Rp 25
Department of Computer Science
Naval Postgraduate School
Morterey, CA 93943

Dr. Robert Grafton 1
Code 433
Office of Naval Research
800 N. Quincy
Arlington, VA 22217

Dr. David W. Mizell 1
Office of Naval Research
1030 East Green Street
Pasadena, CA 91106

ED

8