AN ANALYTIC APPROACH TO STATISTICAL DATABASES

Ezio LEFONS
Alberto SILVESTRI[†]
Filippo TANGORRA

Istituto di Scienze dell'Informazione - via Amendola 173, Bari, Italy
[†]Istituto di Informatica - via Verdi 26, Trento, Italy

Abstract. In the commonly adopted data models (as in Chen's entity-relationship data model |1|, for example) an attribute is a mapping between an entity set or a relationship set and a value set. The intension of a mapping property is given implicitly or explicitly in the data models, but the extension can be generally represented by the set {<entity,value>}, as in the relational model. We propose an alternative data model for statistical databases, in which an attribute is represented by its analytic properties (the distribution function of the values of the attribute). These analytic properties are described by a set of parameters,which we call the *canonical coefficients* of the attribute. The canonical coefficients can be used to solve the usual statistical queries with no access to the data. In particular, we present: 1) the methods for computing and updating the canonical coefficients, 2) the use of the canonical coefficients for solving the main statistical queries, also in distributed statistical database environments. Besides, an application of such parameters to the query decomposition in distributed database environments is discussed.

INTRODUCTION

Statistical Data Bases (SDB) differ from those usually called Data Bases (DB) in:
- user query level: in SDB user queries are generally limited to statistical queries;
- system level: in SDB security methods and inference control mechanisms are very important, much more so than in a DB.

In our opinion another important level must be considered:
- data model level: until a data model for SDB, like a data model for DB, is adopted, security problems and execution-time (responsibility) problems cannot be satisfactorily overcome.

In order to clarify these concepts, we must recall some definitions.

This work was supported in part by the National Council of Research under Contracts 82.01556.97 and 82.00854.97

A Data Model (DM) is an intellectual tool used for understanding the logical organization of data and for modeling the real world of an enterprise (|2|).

A Data Base is a collection of stored operational data used by application systems of some particular enterprise (|3|).

On the contrary, a Statistical Data Base is a database that contains a large number of individually sensitive records but is intended to supply only statistical summary information to its users, not information referring to some specific individual (|4,5|).

These last two definitions implicitly outline the difference we have called user query level and system level. But the above definition of SDB also contains explicitly the assumption that an SDB *is* a DB: what changes is its use. Regarding this, we find a more explicit definition in |6|: "An SDB has been defined as one which returns statistical information, such as frequency counts of records satisfying some given criteria, as opposed to a database which returns details of an entity, for example, name and address of an employee".

On the other hand, statistical summary information, i.e. statistical operations such as SUM, PERCENT, COUNT, AVERAGE, MAXIMUM and MINIMUM, can be defined (|7|) as data base procedures and called aggregate functions, because they calculate a value which is not stored explicitly in the DB.

In some data models, aggregate functions (always referred to as statistical functions in the following) are an integral part of the data language, otherwise they are supported by statistical package interfaces (|8|) or use specialized hardware (database machines and their future plans |9-11|).

With regard to this, Kobayashi, Futagami and Ikeda (|12|) say: "The statistical packages enable non-programmers to analyze statistical data easily. It is not so easy to maintain the data in these packages. We have some statistical packages

which use a conventional database management system (DBMS). But it is still not sufficient to manage the whole environment around statistical data. A new system or package is required to support the statistical environment effectively. Such a system is often referred to as a statistical data base system".

This informal definition differs from the previous ones because it invokes a *new system* to support statistical environments.

Since it is inconceivable that a database sometimes (or for some users) has a database utilization and sometimes a statistical database utilization, in our opinion a wide distinction must characterize SDB from DB.

So we propose these extensive definitions:

- a statistical data model (SDM) is a description of the summary of the real world of an enterprise with, eventually, a description of the real world;
- a statistical data base is a collection of statistical entities finalized to help specialized people to make decisions on the development of an enterprise-world. (Another equivalent definition is given at the end of section 1.3).

So the main objective of an SDB is to develop decision-support facilities.

In what terms a so-defined statistical data model can be made effective is not *a priori* definable, depending on the user level types of an SDB.

In section 1 we introduce three types of user level. For two of these it is unnecessary and sometimes harmful (to the security and execution-time aspects, if statistical query solving is based on effectively stored data) to have also a description of the real world in the statistical data model.

Then we present our analytic approach proposal to statistical databases. The approach is based on the knowledge of the distribution function of data. Section 2 is devoted to illustrating the method for determining data distribution and its properties are discussed.

Applications of the method for solving statistical queries are presented in section 3, while section 4 illustrates an application to optimal query processing strategies in database and distributed database environments.

1. THE ANALYTIC APPROACH

1.1 The statistical database users

Statistical databases have a wide applicability to several user classes. For example they are widely utilized by scientists (in the analysis of the phenomena of Nature), by economists (in marketing analysis and planning), and by the politicians (in the analysis of social problems).

These possible SDB users are different from each other and they can exemplify the different levels of users. So, we distinguish three types of user levels for an SDB.

user level 1

At the first level (the lowest) we put the user who also creates and manages the data in an SDB. Such a user also designs the statistical application programs or the statistical package interfaces and finally analyses the results of its applications.

Users of this kind are very frequent in scientific environments and generally use statistical packages (for a good survey, see [8]). The difference between this kind of user and the more commonly known SDB users is that a user at level 1 can have access to individual information (in physics experiments, for example).

We refer to user level type 1 as the user who performs the statistical analysis of data in a database.

User level 1 usually works on static SDB's (i.e. those SDB's in which the data are not frequently updated). For this user level, an SDB does not come up against any security problems. Instead, and this has to be considered for all the users of an SDB, the SDB come up against execution-time problems.

user level 2

At the second level (the intermediate one) we put the users of dynamic SDB's.

These users can only obtain statistical results from their queries. (Here the dynamism of an SDB is intended to include the re-definition of the relevant data to be stored in the SDB).

It is at this level that the user utilizes a decision-support object.

Here, the Statistical Data Base Administrator (SDBA) can decide to use a data model or a statistical data model for the SDB. The choice depends on the following considerations:

a) if maximum information and accurate responses (precise knowledge of the properties of data) must be provided for the users, then a data model must be selected;
b) if a good approximate knowledge of the properties of data is sufficient (i.e. small relative errors can be tolerated in query responses[1])

[1] Note, however, that precise query responses are not always produced by inference control methods.

and the responsibility problem is an important one, the SDBA can use a statistical data model for SDB.

We call the user of level 2a a user who uses a DB in a statistical way, the user of level 2b one who uses a statistical database.

user level 3

At the third level (the highest) we put the user of a Distributed Statistical Data Base (DSDB) that is, the user who avails himself of all the decision-support facilities of the SDB's.

We say that this user uses statistical data bases in a statistical way.

At this level, the data model must be a (distributed) statistical data model.

We point out that each level needs its lower level.

1.2 Security and responsibility problems

Secure inference control mechanisms are required, thus making an SDB non-compromising. But, as is pointed out in |6|, "there is no such thing as absolute security because there are many unknowns in the system, e.g. users' knowledge. Absolute security is defined formally that no individual information can be inferred *solely* from the history of the answered queries".

Regarding this, however, it must be noted that for almost any SDB a general tracker predicate[2] can always be found (|13|) and that the problem of maximizing the amount of information to the users without compromising the SDB is *NP*-complete (|6|).

There are six approaches to the inference control problem: 1) partitioning the SDB |14|; 2) modifying the results by query modification |15|, output perturbation |16|, data distortion |17| or random sampling |18|; 3) controlling the query set (number of tuples satisfying a query) |19,20|; 4) controlling the overlap between query sets |21|; 5) allowing security constraints in the data model definition |7,22|; and auditing |6|. Methods 1 to 5 are suitable for large SDB's; method 6 works on small ones.

The responsibility property is the response-time independence from the quantity of data involved. Data compression |23|, random sampling |24|, and derived files (HSDB system) |12| approaches are examples of improvements in responsibility, but they have some disadvantages. For example, multistage and stratified sampling have some strategic parameters, so unreliable results can be produced if parameter specification is at fault.

Using the above-mentioned derived files of HSDB, responsibility improves, but security problems are partially solved. An HSDB user can obtain statistical information of an attribute in real time, due to the use of expanded data dictionaries[3], whereas interactive statistical analyses are performed on derived files, while more detailed analyses process the original files. No inference control is contained in this methodology: the 'degree' of security depends only on how the derived files are defined.

In general, these approaches do not have an easy maintenance in dynamic SDB's.

1.3 Advantages of an analytic approach

The analysis of the statistical information contained in an SDB is mainly that of the statistical properties of the stored data. These statistical properties of the data must be expressed in terms of general properties and can be represented by several forms, for example by synthesis data (i.e. normalized summary data) or by analytic data derived from the stored data (i.e. statistical quantities of data; see those mentioned in footnote 3). In the following, we shall refer to the statistical information contained in the data as the statistical data properties.

In some cases a precise knowledge of the statistical data properties is required. For example, in marketing analysis, the retail mean prices of some goods can be required on pre-established days. This information is obtainable only by an effective access to the stored individual prices.

In other cases it is sufficient to have a good approximate knowledge of the statistical data properties. These latter cases generally arise for large and very large SDB's, for which small relative errors can be tolerated. In the previous marketing analysis example, the curve of the state of the prices can be required twice a year. This information is obtainable only if synthesis data representation is effectively stored.

---

[2] A general tracker is a predicate that permits one to find the answer to any inadmissible query, as opposed to an individual tracker that is a specific inadmissible query.

[3] The expansion consists of adding the following attributes: established data, missing values, unit, precision, discrete or continuous domain, the number of actual records within a presented field, theoretical distribution, max, min, medium, mode, mean, variance, skewness, kurtosis and percent points.

So, three situations are possible: an SDB which contains only the individual data, one which also contains the statistical data properties, or one containing only the statistical data properties.

In an analytic approach we deem that the statistical data properties of data must be themselves be stored (see fig.1).

user level		data model	individual data storing	statistical data storing
1		DM	yes	yes/no
2	a	DM	yes	yes/no
	b	SDM	no/yes	yes
3		SDM	no	yes

Figure 1. Situations for an SDB

The major advantage in doing this is the maximum responsibility obtainable. Answers to user queries only require accesses to the stored representations of statistical data properties and not to the stored data.

A second, but no less important advantage, is that new inference control methods can be approached. They work directly on statistical data properties or on their representations.

The analytic approach we propose has some other specific advantages (in the following we identify statistical data properties with their representations):

1) statistical data property evaluation requires only one read-in *per* relation (at the storing-time, for example).
2) Statistical data property representations require a small amount of storage. For example, as we show in the following, for a single attribute, statistical data properties consist of min, max, and an n-tuple of real numbers in (-1,+1) (this n-tuple represents the data distribution).
3) Statistical data properties are not oriented to any particular class of applications: they can handle the usual statistical queries such as PERCENT, COUNT, AVERAGE, and so on, as well as compute other statistical quantities such as the moments of the data distribution (up to a degree 'n', see the previous point), or they can plot or tabulate the data distribution and make histograms of the data.
4) Statistical data properties are scale-independent (e.g. a histogram classifies the data regardless of any particular scale of the range of data).
5) Statistical data properties are independent from the units of measure of stored data and this supports the integration of the queries in distributed environments. (For another approach, based on a data definition language extension, see |25|).
6) Data updating induces a simple immediate statistical data property updating.
7) Finally, statistical data properties can be made known with a high level of accuracy. From this aspect, our experiments have always given satisfactory results. Up to now the tests have concerned predicates on one or two domains.

The analytic approach can be used at every user level, although at user level 1 the stored data must be accessible.

So, we can give an equivalent definition of an SDB as a collection of data consisting of statistical information derived from time-varying data not (necessarily) stored in the database.

2. THE DATA DISTRIBUTION FUNCTION

In this section,we describe the analytic method we utilize to determine the statistical information contained in an SDB. The statistical data properties consist in the knowledge of the distribution function of the data.

At first, in section 2.1, we consider the problem of how to represent the distribution of a random variable (representing the values of a single attribute) and in section 2.2 we discuss the properties of the method.

After that, in section 2.3 we extend the method in order to evaluate the distribution of two random variables (representing the values of two attributes) and discuss it in section 2.4. This extension can be generalized to three or more variables.

In the following we assume that the values of an attribute are numerical values in any range of variability (a,b). That is because any value of an alphanumerical attribute can be mapped into a numerical range by an opportune isomorphic mapping.

We shall call:

R and S generic relations of cardinalities N and M

D a generic domain of R defined on a real and bounded range (a,b)

$X_1, X_2, \ldots$ some homogeneous attributes on D with cardinalities $N_1, N_2, \ldots$

$X, Y, \ldots Z$ some generic attributes of R with ranges $(a_X,b_X),(a_Y,b_Y), \ldots (a_Z,b_Z)$ respectively and the same cardinality N

$x_1, x_2, \ldots x_N$ the value occurrences of X, at a certain time.

The problem of determining the analytic distribution function of an attribute X has several solutions.

The first solution consists in determining the frequency histogram of X. This solution has a relevant disadvantage: the distribution is determined in a static way, i.e. is based on an *a priori* classification of the range (a,b). So, this method clashes with the requirement in point 4) of the section 1.3.

All the other methods are based on the numerical approximation of the real distribution of X.

The interpolation and least squares methods are often utilized, but it is well known that they are not efficient regarding the data updating problem. If polynomials are used, these methods are conveniently used on digital computers only for small values of the approximation polynomial degree. In our opinion, however, the use of polynomial approximation is better than the other functional approximations, such as the trigonometric and exponential approximations, due to the fact that these latter functions are more time-consumingly computable than polynomials. Furthermore, some statistical quantities, such as the moments of data, are more easily derived from polynomial representation of the distribution function.

2.1 The distribution function of an attribute

The method we use approximates the data distribution by orthogonal polynomials and is an alternative method to determining a least squares polynomial approximation.

Here we present the formulae to compute the distribution function of the values of an attribute X on (a,b), by means of a polynomial approximation up to a degree 'n'. Details of their derivations are given in Appendix.

The probability distribution function

Let g(x) be the distribution function (probability density function, pdf) approximation of X. Then we have [26,27]:

$$(1)\quad pdf(X) \cong g(x) = \sum_{i=0}^{n} (2i+1)\cdot c_i \cdot p_i(x)$$

where

$$(2)\quad c_i = \frac{1}{b-a} \cdot \frac{1}{N} \cdot \sum_{j=1}^{N} p_i(x_j) \qquad i=0,1,..n$$

and the p_i's are the Legendre polynomials, computable by the recursive relations:

$$(3)\quad p_0(x) \equiv 1\ , \quad p_1(x) \equiv x$$
$$(i+1)\cdot p_{i+1}(x) = (2i+1)\cdot x \cdot p_i(x) - i \cdot p_{i-1}(x)$$

(The Legendre polynomials are defined on the interval (-1,+1). So, for computing $p_i(x)$ ($i \geq 1$) by formulae (3), each $x \varepsilon X$ must previously be mapped from (a,b) into (-1,+1) by a trivial isomorphism).

The cumulative distribution function

The knowledge of the cumulative distribution function (cumulative density function, cdf) of attributes is very important in DB and SDB environments (see sections 3 - 4). The cdf of X is suitably derivable from the pdf of X (formula (1)) as

$$(4)\quad cdf(X) \cong G(x) \equiv \int_a^x g(y)\,dy$$
$$= \frac{x+1}{2} + \frac{b-a}{2} \cdot \sum_{i=1}^{n} c_i \cdot (p_{i+1}(x) - p_{i-1}(x))$$

2.2 The analytic properties of the method

The canonical coefficients

The calculation of the $c_0, c_1, .. c_n$ coefficients (formula (2)) occurs only once at the creation of the database or, if the data are already stored, it requires only one sequential read-in of each attribute X.

Furthermore, the algorithmic procedure that computes the c_i's consists of few instructions, because it is based on the recurrence relations (3).

Also the computation of the distribution function g (formula (1)) or of the cumulative function G (formula (4)) requires a simple procedure based on formulae (3).

Since the c_i's contain all the information on the distribution of an attribute X, we call them the canonical coefficients of the attribute X.

The additive property

If an attribute X is updated, its c_i' updating does not require a re-read-in of X, but it is immediately performed by the additive property:

$$c_i = \frac{1}{N \pm 1} \cdot (N \cdot c_i \pm p_i(x)/(b-a)) \qquad i=0,1,..n$$

('+' sign holds for insertion of a datum x,
'-' sign holds for deletion of x).

In general the additive property can also be applied to determine the global distribution (i.e. the canonical coefficient $\{c_i\}$) of two or more homogeneous attributes $X_1, X_2, \ldots X_r$, if the respective canonical coefficients $\{c_i^p \mid p=1,..r; i=0,..n\}$ are known:

$$c_i = \left(\sum_p N_p \cdot c_i^p \right) / \sum_p N_p \qquad i=0,1,..n$$

This latter application of the additive property will be discussed later: its main peculiarity shows up in distributed DB and SDB environments.

The performance of the approximation method

The method has been experimented on several attribute samples and in many real cases. It has given a very good performance both for large or very large attributes and for small attributes.

The accuracy of the approximation function depends [28] upon some factors such as:

- the approximation polynomial degree: our tests have indicated a small value (always less than 20, generally 9÷15). The use of higher values is not profitably practicable: it can cause a rounding error propagation increase.
- The rounding error propagation in formula (2): for large values of N, the finite machine precision affects the value of the sum. A relative error damping is obtainable if negative and positive values are separately added.

2.3 The joint distribution function of attributes

Let us call X,Y, ...Z a set of attributes.

If there are not any functional or multivalued dependencies among X,Y,... and Z (i.e. they are mutually independent), the distribution function $g_{X,Y,..Z}(x,y,...\ z)$ and the cumulative function $G_{X,Y,..Z}(x,y,...\ z)$ are directly derivable from $g_X(x)$, $g_Y(y)$, ... and $g_Z(z)$, as in theory of probability for independent variables:

$$g_{X,Y,..Z}(x,y,..z) = g_X(x)\cdot g_Y(y)\cdot\ldots\cdot g_Z(z)$$
$$G_{X,Y,..Z}(x,y,..z) = G_X(x)\cdot G_Y(y)\cdot\ldots\cdot G_Z(z)$$

So, the problem of the calculation of the multi-dimensional distribution of two or more attributes regards only those domains that are mutually dependent.

First we discuss the case of the 2-dimensional distribution of X and Y.

Let us suppose $X \to Y$. Then we can fix a base β of classification of X (or, equivalently, the range (a_X,b_X)) into β intervals $\beta_1,\beta_2, \ldots\beta_\beta$ (in the following denoted as β-intervals) not necessarily of the same length.

The classification of X, based on β, is chosen so that the following is reasonable:

Assumption. For each β_r-interval, if $x_i \varepsilon\ \beta_r$ and $x_j \varepsilon\ \beta_r$, then $g_{Y|x_i} \sim g_{Y|x_j}$.

So, it is possible to compute the β distributions of Y *vs* each β-interval. The distribution $g_{X,Y}(x,y)$ is obtained from the conditional distribution $g_{Y|X}(y|x)$ (represented by the $(n+1 \times \beta)$ matrix $C_{Y|X}$ of the canonical coefficients of the β distributions) and from the distribution function $g_X(x)$ of the independent domain.

The distribution $g_Y(y)$ can either be computed at the same time as $g_X(x)$ and $g_{Y|X}(y|x)$ computations, or directly derived from $g_{Y|X}$: that is its canonical coefficients are the sum of the values on the rows of $C_{Y|X}$.

We point out that g_X, g_Y and $g_{Y|X}$ computations require only one read-in of (X,Y).

In a similar way, the method can be generalized to more than two attributes. For example, for three attributes, X,Y and Z, the distribution function $g_{X,Y,Z}$ is obtainable from:

g_X, $g_{Y|X}$ and $g_{Z|X,Y}$ if $X \to Y \to Z$ and from

g_X, g_Y and $g_{Z|X,Y}$ if $(X,Y) \to Z$.

Example. Let EMPLOYEE be a relation defined on domains E# (employee number), NAME (employee name), AGE (employee age), SALARY (employee salary) and CT (contract type). If we assume a dependence of SALARY attribute on AGE and CT attributes, the distribution of SALARY values, conditioned by (AGE,CT) values, can be obtained by partitioning the ranges of the AGE and CT values into k×l subranges $\{(a_i \leq AGE < a_{i+1}\ ;\ CT_j) \mid i=1,..k;\ j=1,..l\}$ and by computing the β SALARY distributions, where $\beta = k\times l$. In this case, the conditional matrix $C_{SALARY|AGE,CT}$ is a three-dimensional $(k\times l\times n+1)$ matrix.

2.4 Remarks on the multi-dimensional distribution

The method for determining the distribution function of two or more non mutually independent domains needs some clarifications on:

1) the large amount of storage required for a conditional matrix of canonical coefficients;
2) the arbitrariness (or uncertainty) of the classification β.

These points are not real disadvantages, in fact:

1) Large SDB's require a very large amount of storage to memorize the data, in any case. Since the proposed method consists of an analytic approach for solving the user queries (satisfied only by means of canonical coefficients, see the next section), then no storage is required for data. So the storing of the canonical coefficients of the data is largely compensated by the unnecessariness of storing the data.

Additionally, this safeguards further the data base from snooper-inspections: solving queries by statistical methods while guaranteeing statistically accurate responses. On the other hand it impedes the deduction of confidential information by inference, because it is based on the analytic properties of data and not on the data themselves.

2) Regarding the arbitrariness of the choice of a classification β, our tests on large experimental samples of data have indicated that, even for small values of β (10÷20), a good performance is obtainable in a statistical sense. So the assumption made in the previous section is not a restrictive one. (We note that the analysis of a conditional matrix of canonical coefficients can furnish some *a posteriori* indications to the SDBA on the dependency existing among the attributes, by comparing the canonical coefficients of each β-interval by means of an opportune norm).

Moreover, for the user queries based on a classification β' different from β, it is easy to adapt β to β'. The adaption is performed by opportunely weighing the canonical coefficients of the β-conditional matrix *vs* β', by means of the distribution functions of the independent domains (that are exact functions).

Finally, the SDBA is not advised to use a finer classification as the basis of the conditional matrix: not only due to storage considerations, but mainly because it could allow a higher degree of accuracy, so that some individual or general tracker predicates could be profitably found and the deduction of information by inference could not be avoided.

3. APPLICATIONS

In this chapter we show how the statistical functions are computable in SDB environments using the canonical coefficient knowledge.

We deal with statistical queries on a single domain (section 3.1) and on two domains (section 3.2), whose generalization on more than two domains can be similarly obtainable. In section 3.3 other applications based on canonical coefficients such as the calculus of data distribution moments and data report (plotting, tabulating and making histograms of data) are described. Finally, we present some experimental tests of the analytic approach proposed in section 3.4 while the application to distributed statistical databases is discussed in section 3.5.

3.1 Statistical queries on the domain X

Let us indicate with $I \equiv (x_i, x_{i+1})$ a generic subinterval of values of X referred to by a query. (Computable formulae are given in Appendix).

$$PERCENT(x;I) \equiv p(x|x\epsilon I) = G(x;I) \equiv G(x_{i+1}) - G(x_i)$$
$$COUNT(x;I) = N \cdot PERCENT(x;I)$$
$$AVERAGE(x;I) = \left(\int_I x \cdot g(x)\, dx \right) / G(x;I)$$
$$SUM(x;I) = AVERAGE(x;I) \cdot COUNT(x;I)$$

Let us indicate with $J \equiv (y_j, y_{j+1})$ a generic subinterval of values of Y.

Case 1. X and Y are independent domains.

$$PERCENT(x,y;I,J) = PERCENT(x;I) \cdot PERCENT(y;J)$$
$$COUNT(x,y;I,J) = N \cdot PERCENT(x,y;I,J)$$

Case 2. X and Y are not independent domains (X↛Y).

In this case we utilize the conditional matrix C of the canonical coefficients of Y *vs* X in basis β:

$$\begin{vmatrix} C(0,1) & C(0,2) & \dots & C(0,\beta) \\ C(1,1) & C(1,2) & \dots & C(1,\beta) \\ \vdots & & & \\ C(n,1) & C(n,2) & \dots & C(n,\beta) \end{vmatrix}$$

In order to compute a PERCENT function on intervals I and J, we must evaluate the canonical coefficients $\{c_k(y|x;I) \mid x\epsilon I;\ k=0,1,..n\}$ of the distribution $g(y|x)$ on the interval I and then integrate it on the interval J.

By comparing I with the β-intervals it is possible to determine the inferior (inf) and superior (sup) limits of the β-intervals overlapping I.

There are two possible cases:

a) sup-inf=1; i.e. I is entirely contained in one β-interval;
b) sup-inf≧2; i.e. I crosses two or more β-intervals. In this case there are exactly two β-intervals partially overlapping I: we denote x_I and x_F the extremum of these two β-intervals such that: $inf \leq x_i \leq x_I \leq \dots x_F \leq x_{i+1} \leq sup$ (if sup-inf=2, then $x_I = x_F$) and call $\underline{I} \equiv (x_i, x_I)$ and $\bar{I} \equiv (x_F, x_{i+1})$.

In either case the canonical coefficients of $g(y|x)$ on I are defined as the sum of the canonical coefficients of eventual β-intervals entirely contained in I by adding the quota-part of the canonical coefficients (corresponding to the partial intervals: I for case a, $\underline{I}$ and $\bar{I}$ for case b) weighed with respect to the distribution of X on these sub-intervals.

Formally, we have, for k=0,1,...n,:

case a $\quad c_k(y|x;I) = C(k,inf) \cdot PERCENT(x;I)$

case b $\quad c_k(y|x;I) = C(k,inf) \cdot PERCENT(x;\underline{I}) + C(k,sup) \cdot PERCENT(x;\bar{I}) + \sum_{l=inf+1}^{sup-1} C(k,l)$

So, if we call $G(y|x;I)$ the cdf of $g(y|x;I)$, we have:

$$PERCENT(x,y;I,J) \equiv G(y|x;I,J) = \int_J g(y|x;I)\, dx$$
$$COUNT(x,y;I,J) = N \cdot PERCENT(x,y;I,J)$$

(Computable formulae are given in Appendix).

3.3 Other statistical applications

1) The moments of the data distribution are easily calculable as linear functions of the canonical coefficients. For example: for an attribute X on range (a,b), we have:

$$\begin{aligned} & M_0 = (b-a)\cdot c_0 \\ \mu(X) = & M_1 = (b-a)\cdot c_1/3 \\ & M_2 = (b-a)\cdot(2c_2+5c_0)/15 \\ \sigma^2 = & M_2 - M_1^2 \\ & M_3 = (b-a)\cdot(2c_3+7c_1)/35 \\ & M_4 = (b-a)\cdot(8c_4+36c_2+63c_0)/315 \\ & \text{and so on.} \end{aligned}$$

2) The histograms of attribute values can be obtained with a high degree of accuracy by their cumulative distribution functions, by applying the COUNT function recursively to all the intervals of a required classification of the ranges of the attributes.

3) The distribution function of an attribute can be plotted or tabulated by applying formulae (3) to:

$g(x)$ for monodimensional distribution
$g(x)\cdot g(y|x)$ if $X \rightarrow Y$
$g(x)\cdot g(y|x)\cdot g(z|x,y)$ if $X \rightarrow Y \rightarrow Z$
and so on.

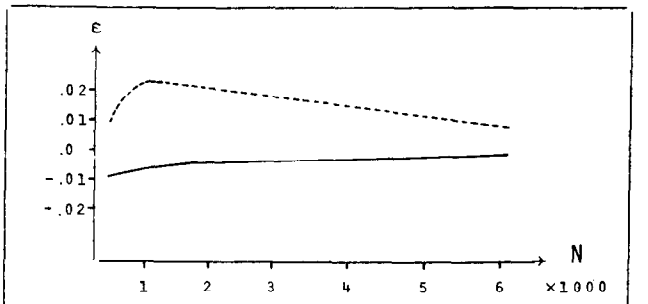
We point out these dynamic facilities in obtaining histograms and tabulations of data. In fact, parameters of a COUNT function are definable at run-time and scaling can use a variable step. Moreover, regarding the plotting or the tabulation of the distribution of data, it is possible to choose a sub-division of the data range by means of any scale function. For example, a logarithmic sequence s_i is definable, for m points in (a,b), as $s_i = k\cdot 10^i$ (i=1,2,..m), where k is such that $a \leq 10\cdot k$ and $b \geq 10^m\cdot k$. So, the tabulation is based on $g(s_1)$ ... $g(s_m)$.

3.4 Tests on the performance of the analytic approach

As examples of the performance of the presented formulae in the applications to experimental and real data, we show some results referring to the COUNT function, as all the other statistical functions are based on it.

Figure 2 illustrates, for a single attribute X, the mean relative error ε obtained in applying the COUNT function while varying the cardinality N of X.

Table 1 illustrates an experimental example of the application of the COUNT function used to obtain a bi-dimensional statistical histogram of



approximation degree n=5 (dotted line)
approximation degree n=15 (continuous line)

Fig. 2. COUNT function (selectivity of selection)

attributes X and Y (X→Y) that can be compared with the real histogram of (X,Y) shown in table 2 (the values refer to a ratio $\beta/\beta' = 0.286$). The first row and the last column for each table report the partial sums on the columns and the rows respectively. (Eventual discordancies of partial sum values with the respective row or column values in table 1, are due to the fact that the reported values have been rounded to integer values).

3.5 Distributed statistical databases

A peculiarity of distributed statistical databases with respect to distributed databases is that almost all the user statistical queries can be solved with no transmission of data in the network. In fact, almost all the statistical functions (except the linear correlation between two (distributed) domains, for example) are linear functions of their parameters.

So, in the horizontal fragmentations of data, the analytic approach proposed does not require the analysis (read-in) of the data. It is sufficient to apply the additive property to the canonical coefficients of the attributes involved in order to solve the COUNT, PERCENT, ..., SUM queries and to make histograms or to plot the distributions of distributed data.

Also in the cases of vertical fragmentations of data, it is sufficient to transmit the canonical coefficients of those attributes involved in a query, if the query regards only remote local data and does not require any relationship among data belonging to different nodes.

4. FURTHER APPLICATIONS

In this section we indicate another application based on the knowledge of the canonical coefficients of opportune domains of a database.

	0	0	0	0	1	1	4	18	45	95	168	253	333	397	420	402	339	255	166	88	40	17	8	6	2	0	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	4
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
270	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	5
	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	8
216	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	10
	0	0	0	0	0	0	0	0	0	1	1	1	1	2	2	2	1	1	1	0	0	0	0	0	0	0	0	0	13
	0	0	0	0	0	0	0	0	0	0	1	1	2	2	2	2	2	1	1	0	0	0	0	0	0	0	0	0	16
	0	0	0	0	0	0	0	0	0	0	1	2	3	3	3	3	2	2	1	0	0	0	0	0	0	0	0	0	22
162	0	0	0	0	0	0	0	0	0	0	1	3	4	5	5	4	3	2	1	1	0	0	0	0	0	0	0	0	29
	0	0	0	0	0	0	0	0	1	1	2	5	6	7	7	5	5	3	2	1	1	0	0	0	0	0	0	0	47
	0	0	0	0	0	0	0	0	1	2	4	6	8	10	9	8	7	5	3	2	1	0	0	0	0	0	0	0	66
	0	0	0	0	0	0	0	1	1	3	5	7	10	11	11	10	9	6	5	3	1	0	0	0	0	0	0	0	83
108	0	0	0	0	0	0	0	1	2	3	6	8	11	13	14	13	11	8	6	3	1	1	0	0	0	0	0	0	101
	0	0	0	0	0	0	0	1	2	4	7	10	13	15	16	16	13	10	7	4	2	1	0	0	0	0	0	0	121
	0	0	0	0	0	0	0	1	2	5	8	12	16	20	21	21	17	13	9	5	2	1	0	0	0	0	0	0	156
	0	0	0	0	0	0	0	1	3	7	11	17	22	27	29	28	24	18	13	7	3	1	0	0	0	0	0	0	211
54	0	0	0	0	0	0	0	2	4	9	16	23	30	36	40	39	33	25	16	9	4	2	0	0	0	0	0	0	288
	0	0	0	0	0	0	0	2	6	11	20	29	39	46	52	52	44	33	21	11	5	2	1	1	0	0	0	0	375
	0	0	0	0	0	0	1	3	7	14	25	38	50	59	66	65	55	41	26	14	6	3	2	2	1	0	0	0	478
	0	0	0	0	0	0	1	4	10	21	35	52	69	82	86	82	69	52	34	18	8	3	3	2	1	0	0	0	633
0	0	0	0	0	0	0	2	2	6	12	22	34	45	53	54	49	42	31	19	10	4	2	2	1	0	0	0	0	389
	25				32				39				46				53				60				67				

Table 1. Statistical histogram of a gaussian data distribution *vs* an exponential one

	0	0	0	0	2	3	4	14	45	96	189	252	303	419	404	437	312	257	166	88	43	22	5	6	0	0	0	0
	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	4
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
270	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	3
	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	5
	0	0	0	0	0	0	0	0	0	1	0	0	1	2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	5
	0	0	0	0	0	0	0	0	1	0	1	0	0	1	1	0	2	1	0	0	0	0	0	0	0	0	0	0	7
216	0	0	0	0	0	0	0	0	0	1	0	1	3	1	0	1	1	0	1	0	0	0	0	0	0	0	0	0	9
	0	0	0	0	0	0	0	0	0	1	1	1	2	4	0	0	1	0	0	0	1	0	0	0	0	0	0	0	11
	0	0	0	0	0	0	0	2	0	0	0	2	0	2	4	4	3	1	1	0	0	0	0	0	0	0	0	0	19
	0	0	0	0	0	1	0	0	0	0	0	3	4	4	3	5	2	2	0	0	1	1	1	0	0	0	0	0	27
162	0	0	0	0	0	0	0	0	1	2	2	6	3	2	2	4	0	5	2	0	0	0	0	0	0	0	0	0	29
	0	0	0	0	0	0	0	0	0	4	2	2	9	7	11	2	5	2	2	1	1	0	0	0	0	0	0	0	48
	0	0	0	0	0	0	0	0	0	1	4	6	8	11	8	8	6	3	2	2	0	1	0	0	0	0	0	0	60
	0	0	0	0	0	0	0	0	2	1	6	8	8	8	12	10	12	3	5	4	1	0	0	0	0	0	0	0	80
108	0	0	0	0	0	0	0	0	1	4	8	7	4	17	17	17	10	12	4	1	2	1	0	0	0	0	0	0	105
	0	0	0	0	0	0	1	0	4	4	9	10	9	16	21	14	9	14	9	6	1	0	0	0	0	0	0	0	127
	0	0	0	0	0	0	0	0	0	7	11	12	9	24	23	21	18	10	14	4	0	2	0	0	0	0	0	0	155
	0	0	0	0	0	0	0	1	3	4	13	15	27	26	17	25	33	20	7	5	4	3	0	1	0	0	0	0	204
54	0	0	0	0	0	0	0	1	3	8	18	17	30	36	40	46	26	37	18	9	6	1	0	0	0	0	0	0	296
	0	0	0	0	0	0	0	2	4	13	27	30	40	51	46	53	36	35	22	14	5	2	0	1	0	0	0	0	381
	0	0	0	0	0	0	1	2	8	14	30	47	42	53	66	72	46	38	23	16	5	3	0	2	0	0	0	0	468
	0	0	0	0	1	1	2	4	9	19	34	57	66	84	81	92	51	48	35	19	9	7	2	2	0	0	0	0	633
0	0	0	0	0	1	1	0	2	9	11	22	26	36	68	50	61	39	26	19	7	6	1	2	0	0	0	0	0	387
	25				32				39				46				53				60				67				

Table 2. Real histogram of data referred to by table 1

The suggestions which follow are referred to the database and distributed database management, regarding an application to query decomposition and optimization.

The problem of query decomposition and optimal processing strategies in distributed database environments |29| is strictly related to a selected cost function. A cost function defines the distributed database system cost and is generally obtained by opportunely weighing:

a) time-based cost factors: total time, response time, network traffic, I/O operations and parallel computing;
b) dollar-based cost factors: I/O operations, CPU utilization and lease of communication lines.

Of course these factors are not independent, that is, any factor cannot be minimized separately from the others. So the definition of a query processing strategy requires, first of all, the choice of appropriate ratios for the weighing factors. Optimizing algorithms (cf. |30-36|) have been generally produced on the basis of the time-based cost factors. In |37| algorithms for minimizing response time and total time are presented.

But the network traffic is almost always considered be the bottle-neck of the systems and it weighs the most.

Therefore, almost all the algorithms take into account the network traffic cost expressed in terms of amount of transmitted messages (mainly, number of tuples to be sent among the computers of a DDB). So, the governing cost of a query processing depends on the cardinalities of the relations or sub-relations (to be transmitted) resulting from the projection, selection, and join operations. However, in |38|, a distinction is made for join operations in terms of their 'simplicity'. For 'simple joins' (one tuple to one tuple joins, for example) the message transmission factors must weigh at least 90% of the total cost function (the remaining percent is due to I/O operations). Instead, for 'complex joins' (i.e. those requiring a high number of pages fetched) the prevalent cost is due to CPU utilization and I/O operations (more than 80%). So, in |34| and |38| the cost function is based on local costs (I/O request + CPU request) and on communication costs (message transmission). However, it must be noted that the communication system assumptions, data rate and access delay, in |38| are quite different from those in |29|.

However, as in |35|, the general conclusions on the algorithms are:

- limited search algorithms (i.e. 'greedy' heuristic algorithms, as in |30,31,33 and 36|) do not perform very well as global search algorithms (i.e. exhaustive algorithms, as in |34| and |38|);
- accurate estimates of temporary result sizes are crucial;
- run-time methods (as in |32| and |33|) are no better than compile-time methods (like that of System R), if relation sizes are known accurately.

Finally, in our opinion, those methods based on replicated relations (as in |36|), while permiting a reduction of network traffic, do not consider the additional cost of the DDB system, due to replicated relation storage, their maintenance (updating propagation) and management (replication transparency). (For the updating propagation problem, see |39|).

However, the accurate knowledge of cardinalities of intermediate relations is always regarded as an important problem in every case.

A distributed data dictionary, |40|, that contains the canonical coefficients of the domains of the relations in a DDB, can permit an accurate forecast of the selectivity factor for the selection and join operations.

In fact:

- selectivity of a selection operation is generally provided by PERCENT function (as defined in sections 3.1 and 3.2, see fig. 2);
- selectivity of a join operation between two relations R (N tuples) and S (M tuples) on attributes X ε R and Y ε S, both defined on the range (a,b), is provided, with respect to N•M, by:

$$\sum_{i=1}^{k} \text{PERCENT}(x;I_i) \cdot \text{PERCENT}(Y;I_i)$$

where the set of intervals $I_1, I_2, \ldots I_k$ is a partition of (a,b).

Each interval I_i can have a different amplitude from the others and is defined in such a way that all the values in it can be properly assumed join-equivalent. This assumption (i.e. the definition of an equivalence relation on (a,b)) defines the semantics to be assigned to a join operation.

Figure 3 shows the accuracy of the estimate of the selectivity of natural join in an experimental case, by using canonical coefficients, while varying the polynomial approximation degree 'n'. (The two considered attributes had 2500 values each, and a gaussian and an exponential distribution respectively).

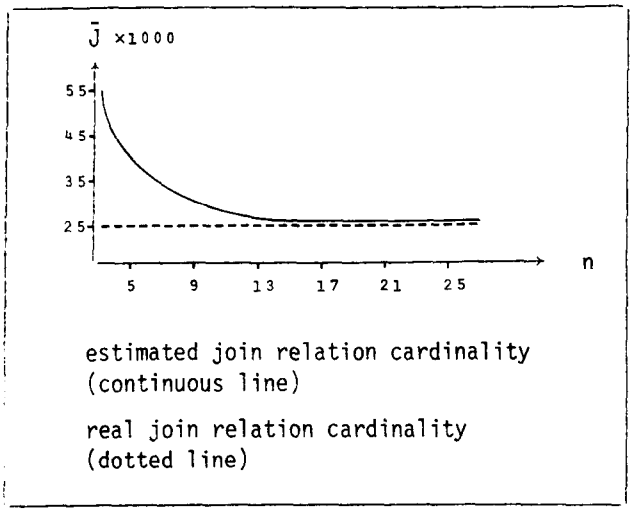


Fig. 3. Selectivity of join

CONCLUSIONS

In the author's opinion a wide distinction must be made between a database approach and a statistical database approach. This distinction does not only regard the differences existing at the level of admissible queries and of inference control problems, but also at the level of the statistical data model definition.

In fact, in this paper, a methodology has been presented, in order to obtain a suitable representation of the statistical information contained in the data of an SDB. Statistical queries can be satisfied only on the basis of (analytic) statistical information (which we represent with a tuple of values called the *canonical coefficients* of data), and no access to stored data is necessary.

So, particularly for large SDB's, it is suitable to furnish a statistical user with a statistical view of the data, instead of directly with a usual schema of the data (as in the ANSI/SPARC architecture). The analytic statistical data model, which permits the definition of each attribute in terms of the distribution function of its current values, is not alternative to the data model definition. Also the storing of the canonical coefficients of the attributes can be made in addition to the storing of the data. However, a detailed description of a statistical data model definition is referred to another paper.

An additive property holds for the canonical coefficients of an attribute, so, stored data updating induces a simple canonical coefficient update. The canonical coefficient updating can be performed at the data updating time or by using checkpoint methods. In this latter case, in fact, it is reasonable to assume that the statistical properties of data, for large data, do not vary quickly.

The performance of the method in solving statistical queries on an attribute is very satisfactory: the differences with respect to the results obtainable by analyzing the stored data are relatively small, and can be considered negligible in the statistical sense. Furthermore, the response time is extremely low: it is independent of the quantity of the data involved because no data is read. However, the method can be suitably extended to the analysis of two or more attributes.

Also, in the distributed environments, the analytic method is advantageously applicable, because the data transmission consists only of transmitting the canonical coefficients of the data involved. This happens both in the horizontal and vertical fragmentations of the attributes. The only restrictions for the vertical fragmentations is that no associations can be required among attributes stored in different sites.

Other applications are possible in the usual database and distributed database management:

1) the prevision of the storage amount required for the results of relational operations, such as selection and join operations, can be made.
2) The parameters resulting from the previous point 1) can be suitably utilized in order to determine an optimal decomposition strategy of a query in a distributed database environment.
3) A very efficient use of the knowledge of the distribution function of an attribute is allowed, in order to apply a unique distributive method for hashing and sorting.

   In fact, the distributive mapping $x \rightarrow \mathrm{COUNT}(x; a,x)$, (where $x \in (a,b)$), produces a data structure, which is both an ordered data structure and a direct access data structure.

   This application is widely discussed in the referenced author's bibliography.

APPENDIX

The distribution function approximation method

Let $g^*(x)$ be the distribution function of the values of an attribute X and $g(x)$ its *orthonormal* polynomial approximation up to a degree 'n':

$$g^*(x) \cong g(x) \equiv \sum_{i=0}^{n} c_i \cdot P_i(x)$$

From the first mean-value theorem, |41|, we have:

$$\langle f,X\rangle = (f,g^*) \cong (f,g) \equiv \int_a^b f(x)\cdot g(x)\,dx$$

where f(x) is any continuous bounded function on the range (a,b) of X, and <f,X> represents the mean value of f on (a,b) (or, equivalently, on X).

In particular, if we choose |26|:

$$f(x) = \sum_{i=0}^{n} P_i(x)$$

from the linearity property of the inner product and the orthonormality of the P_i's, we have:

$$\langle f,X\rangle = \langle \textstyle\sum_j P_j,\ X\rangle \cong (\sum_j P_j,\ \sum_i c_i \cdot P_i)$$

$$= \textstyle\sum_{i,j} c_i\cdot(P_j,\ P_i) = \sum_i c_i$$

In the expansion in orthonormal polynomials the coefficients c_i do not change at the varying of the approximation degree 'n'. So, by induction on 'n', for any coefficient c_i, (i=0,1, ...), we have:

$$c_i = \langle P_i,\ X\rangle$$

that is computable as

$$c_i = \frac{1}{N} \cdot \sum_{j=1}^{N} P_i(x_j)\ .$$

But the Gram-Schmidt orthonormalization method |42| is not efficient to be implemented on computer. So, we substitute the orthonormal P_i's by a set of *orthogonal* polynomials $\{p_i\}$ (we use the Legendre polynomials) that are easily computable by the recursive relations:

$$p_0(x) \equiv 1$$

$$p_1(x) \equiv x$$

$$(i+1)\cdot p_{i+1}(x) = (2i+1)\cdot x\cdot p_i(x) - i\cdot p_{i-1}(x)$$

The Legendre polynomials are defined as

$$p_i(x) = (i+\tfrac{1}{2})^{-\frac{1}{2}}\cdot P_i(x)$$

on the interval (-1,+1), so we have:

$$(p_i,\ p_i) = (i+\tfrac{1}{2})^{-1}$$

and, consequently,:

$$\frac{2}{2i+1} \cdot c_i = \langle p_i;\ -1,+1\rangle$$

Because the isomorphism t:(a,b) → (-1,+1), defined by t(x) = (2x-a-b)/(b-a), allows the representation of orthogonal polynomials over (a,b) in terms of orthogonal polynomials on (-1,+1):

$$p_i(x;a,b) = \sqrt{2/(b-a)} \cdot p_i(t(x);-1,+1)$$

then, we have:

$$\frac{b-a}{2i+1} \cdot c_i = \langle p_i,\ X\rangle\ .$$

Note that in this paper we use x for t(x).

So, we can finally obtain a suitable formula to determine the distribution function of X on (a,b) |27|:

$$g^*(x) \cong g(x) = \sum_{i=0}^{n} (2i+1)\cdot c_i \cdot p_i(x)$$

where

$$c_i = \frac{1}{b-a} \cdot \langle p_i,X\rangle = \frac{1}{b-a} \cdot \frac{1}{N} \cdot \sum_{j=1}^{N} p_i(x_j)$$

The cumulative distribution function

The cumulative distribution function G(x) of X is defined as

$$G(x) \equiv G(x;\ a,x) \equiv p(y|a\le y\le x)$$

$$\cong \int_a^x g(y)\,dy = \frac{b-a}{2} \cdot \int_{-1}^{t(x)} g(y)\,dy$$

By using, for brevity, x for t(x) and because

$$p'_{i+1}(x) = (2i+1)\cdot p_i(x) + p'_{i-1}(x) \qquad i\ge 1$$

holds (cf. |42|), we have:

$$G(x) \cong \frac{b-a}{2} \cdot \sum_{i=0}^{n} (2i+1)\cdot c_i \cdot \int_{-1}^{x} p_i(y)\,dy$$

$$= \frac{b-a}{2} \cdot \sum_{i=0}^{n} c_i \cdot \left[(p_{i+1}(y) - p_{i-1}(y))\right]_{-1}^{x}$$

Since

$$c_0 = \frac{1}{b-a}\ ,\quad p_{-1}(y) \equiv 0,\ \text{and}\ p_i(\pm 1) = (\pm 1)^i \quad \forall i$$

we finally have:

$$G(x) = \frac{x+1}{2} + \frac{b-a}{2} \cdot \sum_{i=1}^{n} c_i \cdot (p_{i+1}(x) - p_{i-1}(x))$$

In particular, for a generic sub-interval $I = (x_k, x_j) \subseteq (a,b)$, it results[4]:

$$G(x;I) = \frac{x_j - x_k}{2} + \frac{b-a}{2} \cdot \sum_{i=1}^{n} c_i \cdot \left[(p_{i+1}(x) - p_{i-1}(x)) \right]_{x_k}^{x_j}$$

Computable formulae for statistical queries

AVERAGE query

Let $I = (x_k, x_j)$ be a sub-interval of (a,b). It results:

$$AVERAGE(x;I) = \frac{\int_I x \cdot g(x)\, dx}{G(x;I)} = \frac{\sum_{i=0}^{n} c_i \cdot \int_I (2i+1) \cdot x \cdot p_i(x)\, dx}{G(x;I)}$$

If we call $\Psi_i(x)$ the indefinite integral $\int (2i+1) \cdot x \cdot p_i(x)\, dx$, then we have[4]:

$$AVERAGE(x;I) = \frac{\frac{b-a}{2} \cdot \sum_{i=0}^{n} c_i \cdot \left[\Psi_i(x) \right]_{x_k}^{x_j}}{G(x;I)}$$

By using the recurrence relations of Legendre polynomials and those of their integrals, we finally have, for i=0,1, ...n,:

$$\Psi_i(x) = \alpha_{i+1} \cdot x \cdot p_{i+1}(x) - \gamma_i \cdot p_i(x) - \beta_{i-1} \cdot x \cdot p_{i-1}(x)$$

where $\alpha_i = \frac{i}{i+1}$, $\beta_i = \frac{1}{\alpha_i}$, $\gamma_i = \alpha_i - \beta_i$.

In some cases the AVERAGE formula can give unreliable results, due to approximation statistic errors. These cases can arise when elements in the sub-interval I do not exist or are clustered and very few with respect to the mean density of the elements in the entire range.

(However, we under-line that, in these cases, unreliable results are deliberately obtained by using many inference control methods).

PERCENT query

Let $J = (y_k, y_j)$ be a sub-interval of (a_y, b_y). It results[4]:

$$PERCENT(x,y;I,J) = \int_J g(y|x;I)\, dy = \frac{b-a}{2} \cdot \sum_{i=0}^{n} c_i(y|x;I) \cdot \left[p_{i+1}(y) - p_{i-1}(y) \right]_{y_k}^{y_j}$$

[4] We recall that x_r (resp. y_r) stands for $t(x_r)$ (resp. $t(y_r)$).

REFERENCES

|1| Chen P.P.
The entity relationship model: toward a unified view of data.
ACM TODS, 1, 1, 1976.

|2| Tsichritzsis D.C. and Lochovsky F.H.
Data base systems.
Academic Press 1977.

|3| Engles R.W.
A tutorial on data base organization.
Annual Rev. Autom. Programming, 7, 1, 1972.

|4| Denning D.E.
A review of research on statistical data base security.
In De Millo R.A., Dobkin D.P., Jones A.K. and Lipton R.J. (eds.) Fundations of secure computations.
Academic Press 1978.

|5| Date C.J.
An introduction to database systems. Vol.II
Academic Press 1983.

|6| Chin F.Y. and Ozsoyoglu G.
Auditing and inference control in statistical databases.
IEEE Trans. on S.E., 8, 6, 1982.

|7| Tsichritzsis D.C. and Lochovsky F.H.
Data models.
Prentice-Hall 1983.

|8| Hext G.R.
A comparison of types of database system used in statistical work.
Proc. in Comput. Statistics, vol.I, Toulouse
Physica-Verlag 1982.

|9| Babb E.
Implementing a relational database by means of specialized hardware.
ACM TODS, 4, 1, 1979.

|10| Maller V.A.J.
The content addressable file store - CAFS.
Proc. of IFIP WG 5.2 Work. Conference, Seehim 1981, North-Holland 1982.

|11| Epstein R. and Hawthorn P.
Design decisions for intelligent database machine.
Proc. NCC, vol.49, AFIPS 1980.

|12| Kobayashi Y., Futagami K. and Ikeda K.
Implementation of a statistical database system: HSDB.
Proc. in Comput. Statistics, vol.I, Toulouse Physica-Verlag 1982.

|13| Denning D.E., Denning P.J. and Schwartz M.D.
The tracker: a threat to statistical data base security.
ACM TODS 4, 1, 1979.

|14| Chin F.Y. and Ozsoyoglu G.
Security in partitioned dynamic statistical databases.
Proc. IEEE 3rd Intern. Conf. Comput. Soft. & Applications, 1979.

|15| Stonebraker M. and Wong E.
Access control in a relational data base management systems by query modification.
Proc. ACM Nat. Conference, San Diego 1974.

|16| Achungbue J.D. and Chin F.Y.
The effectiveness of output modification by rounding for protection of statistical data bases.
IFOR, 17, 3, 1979.

|17| Beck L.L.
A security mechanism for statistical databases.
ACM TODS, 5, 3, 1980.

|18| Denning D.E.
Secure statistical databases with random sample queries.
ACM TODS, 5, 3, 1980.

|19| Chin F.Y.
Security in statistical databases for queries with small counts.
ACM TODS, 3, 1, 1978.

|20| Denning D.E. and Schlorer J.
A fast procedure for finding a tracker in a statistical database.
ACM TODS, 5, 1, 1980.

|21| Reiss S.P.
Security in databases: a combinatorial study.
Journal ACM, 26, 1, 1979.

|22| Chin F.Y. and Ozsoyoglu G.
Statistical database design.
ACM TODS, 6, 1, 1981.

|23| Shoshani A. and Eggers S.J.
Efficient access of compressed data.
Proc. of the 6th Intern. Conf. on VLDB,1980.

|24| Burnett R.A. and Thomas J.J.
Data management support for statistical data editing and subset selection.
Proc. of the 1st LBL Workshop on Statistical Data Management, 1981

|25| Gehani N.H.
Databases and unit of measures.
IEEE Trans. on S.E., 8, 6, 1982.

|26| Lefons E., Silvestri A. and Tangorra F.
Evaluation of the distribution function of large datasets. An application to sorting.
Proc. AICA77, vol.II, Pisa 1977 (in Italian).

|27| Lefons E., Tangorra F. and Silvestri A.
A method to improve performance in statistical database management using data distribution approximation.
Proc. in Comput. Statistics, vol.II,Toulouse Physica-Verlag 1982.

|28| Lefons E., Silvestri A. and Tangorra F.
Performance of a hash technique for large databases.
Proc. AICA79, vol.II, Bari 1979 (in Italian).

|29| Rothnie J.B. and Goodman N.
A survey of research and development in distributed database management.
Proc. of the 3rd Intern. Conf. on VLDB,1977.

|30| Stonebraker M. and Neuhold E.
A distributed database version of Ingres.
Proc. of the 2nd Berkeley Work. on Distr. Data Manag. and Comput. Networks, 1977.

|31| Wong E.
Retrieving dispersed data from SDD-1.
Proc. of the 2nd Berkeley Work. on Distr. Data Manag. and Comput. Networks, 1977.

|32| Hevner A.R. and Yao S.B.
Query processing in a distributed data base systems.
IEEE Trans. on S.E., 5, 3, 1979.

|33| Mahmoud S.A., Riordon J.S. and Toth K.C.
Distributed data base partitioning and query processing strategies. In Bracchi G. and

Nijssen G.M. (eds.) Data Base Architecture. North-Holland 1979.

|34| Williams R. et al.
R*: An overview of the architecture.
IBM Research Report RJ3325, 1981.

|35| Epstein R. and Stonebraker M.R.
Analysis of distributed data base processing strategies.
Proc. of the 6th Intern. Conf. on VLDB,1980.

|36| Bracchi G., Baldissera C. and Ceri S.
Distributed query processing.
In Draffan I.W. and Poole F. (eds.)
Distributed Data Bases
Cambridge Univ. Press 1980.

|37| Apers P.M.G., Hevner A.R. and Yao S.B.
Optimization algorithms for distributed queries.
IEEE Trans. on S.E., 9, 1, 1983.

|38| Selinger P.G. and Adiba M.E.
Access path selection in distributed database management systems. In Deen S.M. and Hammersmey P.(eds.) International Conference on Data Bases. University of Aberdeen 1980.

|39| Coffman E.G., Gelenbe E. and Plateau B.
Optimization of number of copies in a distributed data base.
IEEE Trans. on S.E., 7, 1, 1981.

|40| Lefons E., Silvestri A. and Tangorra F.
A method for access strategy and data analysis in a distributed data base.
Proc.AICA80,vol.II,Bologna1980 (in Italian).

|41| Gradshteyn I.S. and Ryzhik I.M.
Tables of integrals, series and products.
Academic Press 1980.

|42| Isaacson E. and Keller H.B.
Analysis of numerical methods.
John Wiley 1966.