Analytic-based Estimation of Query Result Sizes

Carlo DELL'AQUILA, Ezio LEFONS, Filippo TANGORRA
Dipartimento di Informatica
Università di Bari
Via E. Orabona 4, I-70125 Bari
ITALY

Abstract: - In this paper, a method for estimating the size of relational query results is proposed. The approach is based on the estimates of the attribute distinct values. On the basis of our method, a set of parameters, the so-called *Canonical Coefficients*, can be derived from actual data; they allow us to approximate both the multivariate data distribution and distinct values of attributes. In particular, the capability of analytic method to estimate selectivity factors of relational operations is considered. Some experimental results on real databases are also presented which show the promising performance of our analytic approach.

Key-Words: - Data Bases, estimation, query optimization, join selectivity, projection selectivity.

1 Introduction

Solution of the well-known problem of query optimization in database systems has been widely applied also to many others non-traditional environments, including data warehouses and decision support systems. Whereas in past the hardware limits (small buffering memory-storage sizes and low processor speed) required efforts in establishing efficient strategies in order to minimize access to large data sets, at present, even if hardware limitation have been overcome, the same needs arise because modern applications run on huge amount of data in network context. A knowledge of data profile can yield advantages in activities such as data in warehouses are continually collected from different large databases, as data mining applications in decision support systems require to access large data sets for complex exploratory activity, and as decisions on how to move data in network for minimizing the data traffic. In fact, opportune metadata in data profiles permit the system service software (the query optimizer in database systems) to select the optimal query execution among different equivalent query execution strategies. In centralized and distributed environments, query optimizer parses and analyses the query, constructs possible access paths and estimates the cost of each path in order to determine the least expensive; in fact, in such systems, query plans diverge much in cost due to the database's volume. This path is then selected as query execution plan. Because cost function always considers selectivity factor for relational operation, the performance of query execution module depends on accuracy of selectivity factor estimation [1], which proceeds in turn on estimation of number of tuples produced from a relational operator.

Metadata describing data characteristics are collected in data profiles maintaining statistics about data as the form of data distribution, the data cardinality, the number of distinct values, and so on, in order to determine accurate estimates of the query counts and of the selectivity factor.

In addition, to estimate the query selectivity factor, data profiles are also important in data mining applications and data analysis activities, where the user explores various hypotheses in data and would prefer an approximated count to a query in real time, rather than waiting for an exact count. This research field has conduced to the definition of AQUA systems which involve the estimate of summary data and the execution of aggregate queries with approximate answers derived by the data profile. To obtain reasonable cost estimates of query execution, the query optimizers needs estimates of sizes of the final and intermediate relations involved, depending from selectivity factor. The selectivity factor corresponds to the fraction of tuples which satisfy the query. The selectivity factor of restriction operation is the ratio of the cardinality of the result to that of the base relation. The selectivity of projection is the ratio of tuple size reduction due at the fact that some attributes are being removed. However often the duplicates are also removed. The join selectivity of one relation with another defines the ratio of the attribute values that are selected in one of the relations. Joins on more than two relations are usually considered as sequences of joins on two relations at a time.

Traditionally, the estimate methods of the actual query selectivity factor are classified into *parametric* and *nonparametric* methods. The distinction is based on the method for determining data distributions. Here, the knowledge of attribute distributions plays a crucial role in estimating both selectivity factors and aggregate functions.

Parametric methods assume that the distribution has a known form except for a few parameters (*e.g.* mean and standard deviation). By making this assumption, one only needs to estimate the parameters which complete the description of the shape of the distribution. Parametric methods are of interest because they summarize the distribution with a few parameters [2-4]. On the contrary, they do not provide accurate estimates when the actual data do not fit the assumed theoretical distribution.

Nonparametric methods do not make *a priori* assumptions about the form of the distribution. Therefore, the distribution can be more difficult to estimate, and more storage is required than with parametric methods but these methods provide more accurate estimates.

Several nonparametric methods have been proposed. We classify them into sampling-based [5-8] and histograms (equal-width, equal-height, variable-width and wavelets-based histograms are well known examples [9-12]).

Sampling based estimators describe faithfully the actual data distribution and increasing samples provide more accurate estimates. However, more time will be required to obtain run-time sampled information in addition to current query processing time which generally can not be reused for subsequent queries. Minimizing the sample size while maintaining estimate accuracy is the main objective of research in this area [4].

The histogram based methods store tables in the data profile for computing accurate estimates depending on the number of update operations, since frequent updates in database will change the data distribution and render expensive updating histograms which must be computed periodically to fit actual data distribution.

We have presented an analytic approach based on the approximation of the actual multivariate data distribution of attributes by a series of orthogonal polynomials [13,14]. The method is a special case of least squares approximation by orthonormal functions and summarizes all the information on the data distribution by few data—the computed coefficients of the polynomial series. We have called them the *Canonical Coefficients* of the data, for they allow the selectivity factors and the main data statistics to be easily derived and efficiently computed with no access to the data warehouse. Moreover, data updates can immediately propagate to the canonical coefficients based on their so-called *additive property*.

According to the encouraging results of the performance for the selectivity factor of the restriction operation in multidimesional environment [14], we apply the analytical approach to estimate the selectivity factor of join and projection operations. In these relational operations, the knowledge of distincts of database attributes plays a crucial role in improving the performance of the method in estimating the selectivity factor. The analytic method, opportunely adapted, permits to enrich the descriptive contents of data profile with few metadata relative to the distribution of distances of distincts in the attribute value ranges.

We describe the application of the analytic method to determine the selectivity factor of join and projection operations using a new approach based on the estimate of attribute distincts, which we have reported in our earlier work on this topic [15].

2 Related Works

2.1 Join selectivity factor

Join is used in relational algebra to match two relation on compatible attributes and it is defined, in terms of primitive operators, as a Cartesian product followed by a selection. The selection condition specifies a Boolean expression on attributes of relations and is called the join condition. The most common use of join involves join conditions with equality comparisons only and is called *equijoin* operation and we will refer to this operation using only the term "join".

When we consider the join T between relations R and S, the canonical formula giving the cardinality of join is *card*(T)= *j*ρ × *card*(R) × *card*(S) where *j*ρ is the join selectivity factor and represents the fraction of tuples of Cartesian product which satisfy the join condition.

There are several methods for join cardinality estimating [16-21]. Some methods are based on integrity constraints of the relational schema and provide limits for join cardinality estimation (eg., Card(T)≤Card(R×S)); these methods don't provide accurate estimations. Other methods assume arbitrarily that join attributes are uniformly distributed and that the number of distinct values of attribute T.X is min(*dist*(R.X), *dist*(S.Y)) [22]. However, it is very rare that attributes are uniformly distributed in real cases and the uniformity assumption leads to pessimistic results when data attributes follow to others distribution types as Normal, Gamma, Zipf, etc.

Other nonparametric methods have been proposed. In worst case assumption method, join cardinality is estimated as *card*(R) × *card*(S). Worst case divided 2 method minimizes errors produced by the worst case method. Method of perfect knowledge doesn't produce errors but all occurrences of attributes values must be stored, so it is very expensive in memory and time consuming for updating [16]. Piece-wise uniform method has been proposed and used in [19] to estimate frequency distribution of join attributes by equal-width histograms. In this approach, attribute domain is divided into intervals and number of tuples holding values which fall into each is stored. Attribute distribution is approximate by a piece-wise linear curve which interpolates some point by a relative frequency histogram, which is then used to estimate size of relational query and parameters of intermediate results. Join range is divided in a number of intervals whose scale is function of number of distinct values of join range. To determine number of distinct values of join range, number of distinct values of joining attributes are to be known, but no method has been proposed to estimate them. So a way to obtain them, is to sort join attributes' values and to eliminate duplicates, but this method is impractical due to the high cost of processing and maintenance on updating. Our approach estimates the resulting size of the relational join by evaluating distincts and using the estimation of actual distribution of join attributes.

2.2 Projection selectivity factor

The project operation reduces the number of attributes in a relation; it can be thought of as cutting it down vertically. The result of the project operation is a relation with a reduced number of attributes. The selectivity of projection is the ratio of tuple size reduction. Indeed, not only some attributes are removed in a projection, the duplicate tuples are removed also.

In general, the approach for estimating the size of a projection consists into the ability of data statistical profile to fit actual multidimensional data distribution in order to derive estimates of the marginal distribution of the projected attributes.

In [22] is described a non parametric approach based on the multidimensional histogram divided in equal-sized buckets, assuming uniformity within the buckets. Parametric methods described in [23] assume that attributes are uniformly distributed and independent. Under these assumptions authors studied the probability distributions of the sizes of the projections testing various hypothesis. The assumptions of the independence and uniformity of attributes result inadequate to correctly represent many actual database instances. However, they simplify and speed the computations of parameters for run-time query optimizer in evaluating access plans. The model presented in [24] takes into account that values of attributes determine a time dependent active domain (the distinct values that are actually assumed, at a certain time). Others approach are based on fast ways to estimate distinct values [25-27].

Our approach estimates the resulting size of the relational project by evaluating distincts of active domains of attributes involved in the operation.

3 The Analytic Method to Approximate Data Distribution

Let R be a relation of cardinality N and let X be an attribute of R. Suppose *dom*(X)=[a,b] and let $x_1, x_2, ..., x_N$ be the occurrences of X in R.

We approximate the probability density function g(x) of the attribute X with

$$g(\mathrm{x}) = \frac{1}{\mathrm{b-a}} \sum_{i=0}^{n} (2i+1)\, c_i\, P_i(\mathrm{x}) \qquad (1)$$

for all x∈X and for the opportune n. For each i = 0,1, .., n, P_i(x) is the Legendre orthogonal polynomial of degree i, and coefficient c_i is the mean value of P_i(x) on the instances of X. That is,

$$c_i = \frac{1}{N} \sum_{\mathrm{x} \in \mathrm{X}} P_i(\mathrm{x})\,. \qquad (2)$$

c_0, c_1, ..., c_n's are computed with simple recursive formulae [14] and they are called the *Canonical Coefficients* of X.

The approximation of the cumulative distribution function G(x) of g(x) is

$$G(\mathrm{x}) = \int_{\mathrm{a}}^{\mathrm{x}} g(y)\,\mathrm{d}y = \frac{1}{\mathrm{b-a}} \sum_{i=0}^{n} c_i \left(P_{i+1}(\mathrm{x}) - P_{i-1}(\mathrm{x})\right). \qquad (3)$$

Let I = $[x_1,x_2] \subseteq$ [a,b] be a generic query-range of X. We denote with *count*(x;I) or N × *percent*(x;I) the number of tuples of R whose x value belongs to interval I. *count*(x;I) can be approximated by $N \times (G(x_2)\text{-}G(x_1))$.

4 Join selectivity estimation

In this Section, we present our approach to estimate the cardinality of join operation between two relations R and S on respective attributes X and Y using the analytic model described in the previous Section.

Let T=R⋈$_{X=Y}$S, to estimate $card(T) = j\rho \times card(R) \times card(S)$, or equivalently for $j\rho$, let $x_1, x_2, ..., x_{dist(X)}$ and $y_1, y_2, ..., y_{dist(Y)}$ be the ordered distinct values for attributes X and Y respectively. Then,

$$card(\mathrm{T}) = \sum_{i=1}^{dist(X)} \sum_{j=1}^{dist(Y)} \left[C_{\mathrm{X}}(x_i) \times C_{\mathrm{Y}}(y_j) \right]_{x_i = y_j} \qquad (4)$$

where $C_{\mathrm{X}}(x_i)$ and $C_{\mathrm{Y}}(y_j)$ denote respectively the number of the tuples of R and S satisfying the condition $x_i = y_j$. In general, an estimation of $\mathrm{C}_{\mathrm{X}}(x_i)$ and $\mathrm{C}_{\mathrm{Y}}(y_j)$ is obtained, without scanning R and S, using aggregate function *percent* as described in Section 3, as

$$C_{\mathrm{X}}(x_i) \cong count(x;\mathrm{I}x_i)=card(\mathrm{R})\times percent(x;\mathrm{I}x_i)$$

$$\mathrm{C}_{\mathrm{Y}}(y_j) \cong count(y;\mathrm{I}y_j)=card(\mathrm{S})\times percent(y;\mathrm{I}y_j)$$

were $\mathrm{I}x_i = [(x_i+x_{i-1})/2, (x_i+x_{i+1})/2]$ and $\mathrm{I}y_i = [(y_i+y_{i-1})/2, (y_i+y_{i+1})/2]$. So, the selectivity factor can be approximated as follows

$$j\rho \cong \sum_{i=1}^{dist(X)} \sum_{j=1}^{dist(Y)} \left[percent(x;\mathrm{I}_{x_i}) \times percent(y;\mathrm{I}_{y_j}) \right]_{x_i = y_j} \qquad (5)$$

The *percent* values depend on distinct values of attributes X and Y, which are often unknown. Many researchers suppose that differences between two adjacent domain values are approximately equal [20].

The approach followed in this paper doesn't make hypothesis about distinct values spacing distribution or their joining equivalence. Distinct values are approximated using canonical coefficient which can be easily computed and updated whenever a new distinct value is inserted in the database.

Let $\{x_1, x_2, ..., x_d\}$ be the distinct values of attribute X. The canonical coefficients $(d_i)_{1\leq i\leq n}$ up to degree *n,* that contain information about how distinct values are spaced, are computed as follows

$$d_i = \frac{1}{(d \times dns)} \sum_{j=1}^{d} \sum_{k=1}^{dns} P_i(x_j - \frac{\delta}{2} + \delta * rand(0,1)) \quad i = 0,1,...,n \qquad (6)$$

were we suppose that, for each x_j, in the interval $]x_j$-$\delta/2$, $x_j+\delta/2[$ there are distributed a sufficiently high number *dns* of random values, with $\delta= (b-a)/card(X)$. We assume that x is an approximation of a distinct value x_i, if it verifies the condition $count(x;\mathrm{I}) \approx dns$ in the interval I or, equivalently, $|\ count(x;\mathrm{I}) - dns\ | \leq \varepsilon$ for $\varepsilon = k \times dns$ $(0<k<1)$. The detailed derivation of the method is reported in [15]. Here we give the algorithm that provides approximations of distinct values and their number starting from the knowledge of canonical coefficient d_i:

Algorithm 1: determination of distincts.

```
$d$:=1; $x(d)$:=a; $x_1$=a+$\delta$/2; $x_2$= $x_1$+$\delta$; $eps$ = 0.05* $dns$
    while ($x_2$<b) do begin
        I:=[$x_1$,$x_2$]; $occ$:= $count(x$, I);
        if   $eps \geq | dns - occ |$   then begin
                                    $d := d+1$;
                                    $x(d) := (x_1+x_2)/2$ end
        $x_1$:=$x_2$; $x_2 := x_1+\delta$
        end
    $d := d+1$; $x(d)$ := b
```

When the above procedure terminates, value d approximates the number of distinct values for X in [a,b] and $x(1..d)$ contains approximations of distinct values.

Because of in computing canonical coefficient up to degree n in I_j=]x_j-δ/2, x_j+δ/2[we have distributed *dns* occurrences, then it is expected that $count(x, \mathrm{I_j}) \cong dns$. So, the Algorithm 1 analyses interval [a, b] searching sub-interval I which satisfies the condition of the required estimation accuracy at the given confidence level, defined empirically as the fraction 0,05 of the *dns* quantity in the interval I of amplitude δ.

Series of canonical coefficients $(d_i)_{0\leq i\leq \mathrm{n}}$ are different from series which approximates distribution of attribute X, and aggregate function *count* in Algorithm 1 is calculated using $(d_i)_{0\leq i\leq \mathrm{n}}$.

Figure 1 shows the number of distinct values of an attribute correctly estimated by the *Algorithm 1* for approximation degree n = 5, 6, …33 and *dns* = 50 in a real case relation with 80 distinct values

The performance of the shown example is to be considered as typical of the average behaviour of the *Algorithm* 1.

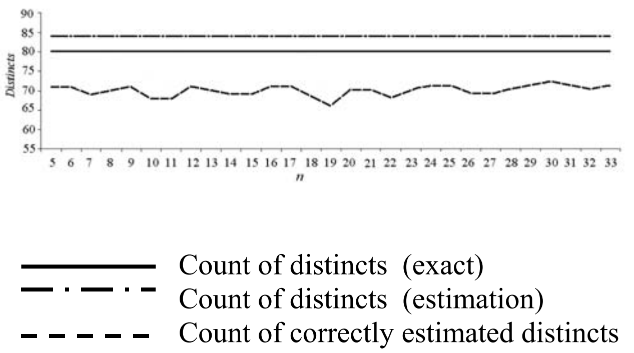


Fig.1- Performance of database distinct estimation.

Assuming that sets $X = \{x_i\}_{0\leq i\leq dist(X)}$ and $Y = (y_j)_{0\leq j\leq dist(Y)}$ estimated with Algorithm 1 are denoted with $\bar{X} = \{\bar{x}_i\}_{0\leq i\leq dist(X)}$ and $\bar{Y} = \{\bar{y}_j\}_{0\leq j\leq dist(Y)}$, in computing join selectivity instead of applying (5) we use

$$jp \cong \left[\sum_{i=1}^{dist(X)}\sum_{j=1}^{dist(Y)} percent(\bar{x},\bar{I}_{\bar{x}_i})\times percent(\bar{y},\bar{I}_{\bar{y}_j})\right]_{\bar{x}_i\cong\bar{y}_j} \qquad (7)$$

We consider $\bar{x}_i \cong \bar{y}_j$ if $|\ \bar{x}_i - \bar{y}_j\ |$ is lower than the mean distance among approximate distinct values respectively of X and Y [15].

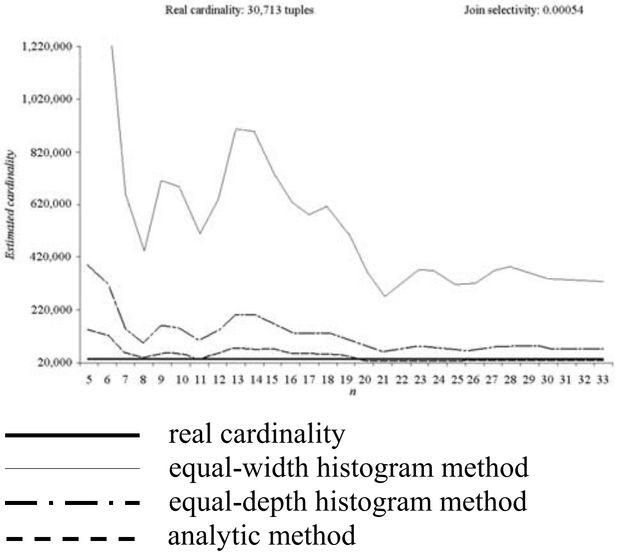


Fig.2- Comparison of the performance of methods for estimating join selectivity.

Figure 2 shows an example of the performance of join approximation in real database case. The comparison of analytic method with equal-width histogram method and equal-depth histogram is also reported.

5 Projection Selectivity Estimation

The model of Gardy Peuch [23], based on attribute independence and uniform distribution assumptions, is frequently used to estimate the size of projections. Using probabilistic argument, they derived formulas to compute the *l* tuples of projection $\pi_Y(R)$ from the *m* tuples of a relation R(X) with Y ⊆ X as expected value of all possible randomly generated projections. This value is computed as:

$$l = \frac{dx}{dy}\left[1 - \left(1 - \frac{dy}{dx}\right)^m\right] \qquad (8)$$

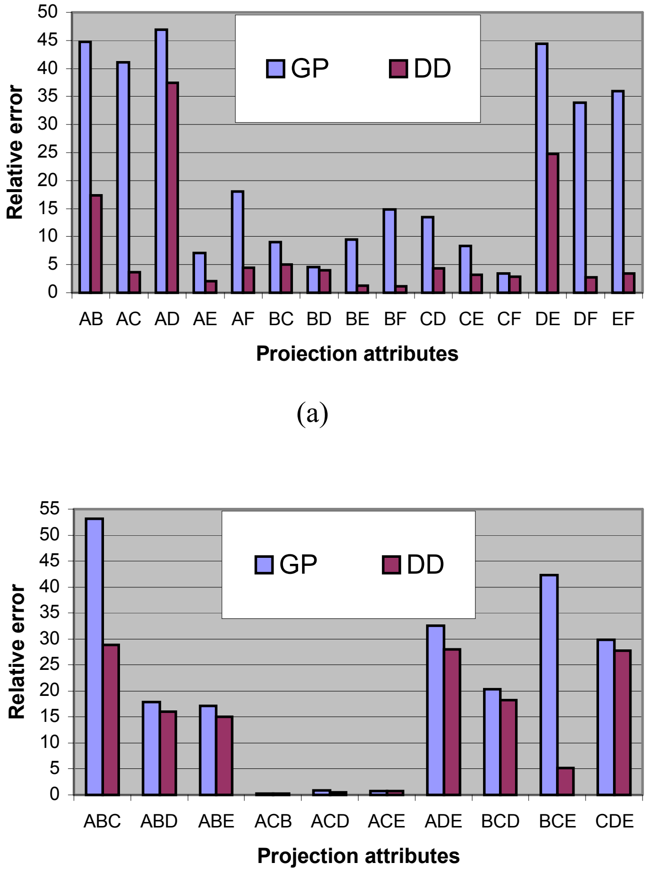
where d_X and d_Y are the product of distinct value numbers respectively of domains of attributes *X* and of attributes *Y*.

We have used the same formula, but we estimate the distinct values with canonical coefficients considering the effective distinct values in the domain (*i.e.* the *active* domain). In fact, canonical coefficients allow to estimate current distincts for they are updated when a new/old occurrence is added/removed in/from database. For example, if values $(d_i)_{0\leq i\leq n}$, *dns*, and δ are stored in statistical profile at a certain time and an occurrence *x'* arrives for attribute X in database, the Algorithm 2 updates canonical coefficients d_i when it establishes that *x'* is a distinct value.

```
Algorithm 2: updating of canonical coefficients
        $(d_i)_{0\leq i\leq n}$ for distincts
    I = [$x' - \delta/2$, $x' + \delta/2$],
    if $|d*dns*percent(x,I) - dns| \leq eps$
        then $x'$ already exists in database
        else $x'$ is a new distinct value hence
    $d_i := \frac{d_i\times(d\times dns)+\sum_{k=1}^{dns}P_i(x'-\frac{\delta}{2}+\delta\times rand(0,1))}{(d+1)\times dns}$;
    i=0,...,$n$; $d := d + 1$
```

We have performed experiments on a real database considering the relation R(A,B,C,D,E,F) with *m* = 104828 tuples. According to empirical tests, the distinct values of the attributes are approximated using: *dns* = 50, ε = 0,005 × *dns* and degree *n* = 15.

Figures 3a and 3b report the performance of the projection selectivity respectively in the bidimensional and in tridimensional cases comparing the method of Gardy Peuch (GP) to the analytic method (DD).



(a)

(b)

Fig. 3 Comparison of the performance of GP and DD methods for estimating projection selectivity respectively in the (a) bidimensional and (b) tridimensional cases.

The errors obtained in applying the analytic method estimates to projection selectivity are in many cases significantly reduced with respect to GP method. We have observed that analytic method improves sensibly performances if the projection is performed on two attributes respect to the case of three attributes. This result offers a greater advantages if we consider that plans of optimizers normally privilege selection and projection operations before the more expensive operations. Therefore, the tentative of drastic reduction of query sizes using projection on few attributes can be supported by better accurate estimates.

6 Conclusion

Determining the selectivity factor estimates of relational operations is a useful task for optimizers to choice optimal path of execution of query processing. Traditional methods use parametric or nonparametric approaches for representing the data profiles which contains metadata describing the data distribution. The presented method follows an analytic approach and stores all information of data profile in canonical coefficients. It use the same method for representing both the multivariate data distribution and distinct values. The analytic method, already successfully tested in the estimate of selectivity factor for restriction operation, has been adapted and tested here for estimating join and projection operations. First experimental results show good performance and improvements with respect to other conventional methods. Moreover, its application is not limited to estimation of selectivity factor of relational operations. Several multidimensional aggregate functions and statistical quantities can be easily and accurately estimated using canonical coefficients. This application has been receiving attention in nontraditional emerging areas of database technology such as the approximate query processing field. It provide approximate answers to the queries very quickly and is particularly attractive for large-scale and exploratory activities in OLAP applications.

References

[1] S. Caudhuri and V.R. Narasayya, Automating Statistics Management for Query Optimizers, *IEEE TKDE*, Vol. 13, No. 1, 2001, pp. 7-20.

[2] S. Christodoulakis, Estimating Record Selectivities, *Information Systems*, Vol. 8, No. 2, 1983, pp 105-115.

[3] C.M. Chen and N. Roussopoulos, Adaptive Selectivity Estimation Using Query Feedback, *Proc. ACM SIGMOD Conf.*, 1994, pp. 161-172.

[4] Y. Ling and W. Sun, A Hybrid Estimator for Selectivity Estimation, *IEEE TKDE*, Vol. 11, No. 2, 1999, pp. 338-354.

[5] R. Lipton, J. Naughton and D. Schneider, Practical Selectivity Estimation Through Adaptive Sampling, *Proc. ACM SIGMOD Conf.*, 1990, pp. 1-11.

[6] P.J. Haas and A.N. Swami, Sequential Sampling Procedures for Query Size Estimation, *Proc. ACM SIGMOD Conf.*, 1992, pp. 341-350.

[7] Y. Ling and W. Sun, A Comprehensive Evaluation of Sampling-Based Size Estimation, *Proc IEEE 11th ICDE Conf.*, 1995, pp. 532-539.

[8] Chaudhuri S., Das G., Datar M., Motwani R., and Narasayya V., Overcoming Limitations of Sampling for Aggregation Queries, *Proc. IEEE ICDE Conf.,* 2001, pp. 534-542.

[9] M. Muralikrishna and D.J. DeWitt, Equi-Depth Histogram for Estimating Selectivity Factors for Multi-dimensional Queries*, Proc. ACM SIGMOD Conf.*, 1988, pp. 28-36.

[10] M.V. Mannino, P. Chu, and T. Sager, Statistical Profile Estimation in Database Systems, *ACM Computing Surveys*, Vol. 20, No. 3, 1988, pp. 191-221.

[11] Y E. Ioannidis, V. Poosala, Balancing Histogram Optimality and Practicality for Query Result Size Estimation, *Proc. ACM SIGMOD. Conf.*, 1995, pp. 233-244.

[12] J.S. Vitter and M. Wang, Approximate Computation of Multidimensional Aggregates of Sparse Data Using Wavelets, *Proc. ACM SIGMOD Conf.,* 1999, pp. 193-204.

[13] E. Lefons, Silvestri A., and F. Tangorra, An Analytic Approach to Statistical Databases, *Proc. 9th VLDB Conf.,* 1983, pp. 260-274.

[14] E. Lefons, A. Merico, and F. Tangorra, Analytical Profile Estimation in Database Systems, *Information Systems,* Vol. 20, No. 1, 1995, pp. 1-20.

[15] C. dell'Aquila, E. Lefons, and F. Tangorra, Estimation of Database Unique Values, *WSEAS Trans. on Information Science and Applications,* Vol. 1, No. 1, 2004, pp. 280-285.

[16] D.A. Bell, D.H.O. Ling, and S. McClean, Pragmatic Estimation of Join Sizes and Attribute Correlations, *Proc. IEEE 5th ICDE Conf.*, 1989, pp. 76-84.

[17] Y. E. Ioannidis and S. Christodoulakis, Optimal Histograms for Limiting Worst-Case Error Propagation in the Size of the Join Results, *ACM TODS*, Vol. 18, No. 4, 1993, pp. 709-748.

[18] T. Mostardi, Estimating the size of relational SPθJ operation: an analytical approach, *Information Systems,* Vol. 15, No. 5, 1990, pp. 591-601.

[19] J. K. Mullin, Estimating the Size of a Relational Join, *Information Systems*, Vol. 18, No. 3, 1993, pp. 189-196.

[20] W. Sun, Y. Ling, N. Rishe, and Y. Deng, An Instant and Accurate Size Estimation Method for Joins and Selection in Retrieval-Intensive Environment, *Proc. ACM SIGMOD. Conf.*, 1993, pp. 79-88.

[21] P. J. Haas, J.F.Naughton, S. Seshadri, A.N. Swami, Selectivity and Cost Estimation for Joins Based on Random Sampling, *J. Computer Sys. Sci.* Vol. 52, No. 3, 1996, pp. 550-569.

[22] T.H. Merrett and E. Otoo, Distribution Models of Relations,. *Proc. 5th VLDB Conf.*, 1979, pp. 418-425.

[23] D. Gardy and C. Puech. On the Sizes of Projections a Generating Functions Approach. *Information Systems,* Vol. 9, No.s 3/4, 1984, pp. 231-235.

[24] P. Ciaccia and D. Maio. Domains and Active Domains: What This Distinction Implies for the Estimation of Projection Sizes in Relational Databases. *IEEE TKDE*, Vol. 7, No. 4, 1995, pp. 641-654.

[25] P.B. Gibbons, Distinct Sampling for Higly-Accurate Answers to Distinct Values Queries and Event Reports, *Proc. 27th VLDB. Conf.*, 2001, pp. 541-550.

[26] M. Charikar, S. Chaudhuri, R. Motwanni and V. Narasayya, Towards Estimation Error Guarantees for Distinct Values, *Proc. of ACM PODS*, 2000, pp. 268-279.

[27] P.J. Haas, J.F. Naughton, S. Seshadri, and L. Stokes, Sampling-Based estimation of the Number of Distinct Values of an Attribute, *Proc. 21th VLDB. Conf.*, 1995, pp. 311-322