ON THE ESTIMATION AND USE OF SELECTIVITIES IN DATABASE PERFORMANCE EVALUATION

Stavros Christodoulakis

Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1

Research Report CS-89-24

June, 1989

ON THE ESTIMATION AND USE OF SELECTIVITIES IN DATABASE PERFORMANCE EVALUATION

(Extended Version of a talk presented in the
Workshop on Query Optimization, Portland, May 30, 1989)

Stavros Christodoulakis
Department of Computer Science
University of Waterloo
Waterloo, Ontario
N2L 3G1

Abstract: This presentation is not a survey on the estimation of selectivities. It is rather a biased selection of topics on selectivity estimation reflecting our own opinions and preferences on what are some worthwhile research areas in selectivity estimation, what has been done in the past, and what should be investigated in the future. We apologize for not including all research areas or results in this short presentation. The interested reader can find an extensive survey on selectivity estimation in [Manino, Chu, and Sager 88]. We study the behaviour of several cost functions for varying probability distributions in the attributes of relations. Several new results related to selectivity estimation and data base performance evaluation are presented. Specific results include join costs, projection costs, buffer costs, sample sizes, and file organizations.

1. Introduction

We define record selectivity to be the number of records qualifying in a query, block selectivity to be the number of blocks containing the qualifying records, average record selectivity to be the average number of qualifying records in a set of queries, and average block selectivity to be the average number of blocks containing qualifying records for each of a set of queries.

Selectivities have been extensively used in performance analysis of file organizations, physical data base design (such as index selection and attribute partitioning or attribute clustering), and query optimization. The requirements for each problem may be different. Although in this presentation we emphasize the use of selectivities in query optimization, the research topics described are also applicable to other areas of performance evaluation.

2. Selectivities in Query Optimization

A statistical profile describes the statistics kept or estimated for a relation. Usual statistical profiles frequently assumed are variations of the original query optimization research in System R and they typically include:

1. The number of tuples in a relation.
2. The number of distinct values of each domain.
3. The number of bytes for a value for each domain.
4. The number of distinct values (or the range of values) currently in an attribute.
5. The average number of records per block.

Based on these statistics a typical optimizer should calculate:

1. The cost of individual operations (selections, projections, joins, semi-joins).
2. A new statistical profile for the relation derived from a single operation. This statistical profile is needed in order to calculate the cost of subsequent operations.
3. The cost of a sequence of operations. This is done by utilizing the cost of individual operations and the statistical profiles of the intermediate relations.

Note that the accurate calculation of the statistical profile resulting from a single relational operation is very important if a sequence of operations is taking place. This problem has been discussed only in passing in the current literature. We present some new results in this paper.

3. More Detailed Models

The statistical profiles described in the previous section assume that the tuples of a relation are uniformly distributed over the attribute values of an attribute, and that attribute values of different attributes are independent. This is not a realistic assumption in many environments. Attribute values are frequently skewed, and attributes are placed together in a relation precisely because there is some relationship (dependency) among them. It is therefore reasonable to investigate more accurate statistical profiles than those assuming uniformity and independence of attribute values.

3.1 Parametric Techniques

Parametric techniques model the distribution of data points in a multivariate space by using a member of a class of distributions. A particular class of distributions is characterized by a (small) set of parameters. A member of the class is selected by calculating the parameters of the class using the existing set of data points. The advantages of parametric techniques are:

1. They are simple and intuitive. Typically only a few moments are calculated. Since the parameters are typically easy to understand and even give an approximate estimate in case that all the data points are not known, or expensive to calculate, or update on-line, parametric techniques may be useful in environments that the user is responsible for providing information about the distributions followed. This may often be the case in data base design for example, or in systems where the user is responsible for inserting or updating statistics. Finally, they may be useful in performance studies where the system has to be tested under certain possible workloads. The results can be interpreted easily in such a case due to the intuitive nature of the parameters of the parametric models.

2. When a multivariate space is described with a parametric model, it is easy to find the distributions followed in each subspace. This is useful, since only a subset of the dimensions (attributes of a relation) may be of interest (specified in a user query).

3. Typically, it is easy (requires one file pass) to generate the parameter values for such models from an existing set of data points. It is also typically easy to update the parameter values by considering the new set of data points only. (The old points are not needed.)

A major limitation of the parametric techniques is that the shape of a particular class of distributions can not change arbitrarily by tuning the parameters of the model. For example all multivariate Pearson type 2 and type 7 distributions are unimodal.

This limitation may be overcome if clusters are detected in the multivariate space and each cluster is approximated by a parametric distribution. The probability distribution at a point in space will be described as the weighted sum of the probabilities of each cluster at this point. This technique can be used successfully to model a multivariate space with high concentrations of points in particular locations. Such is the case with many populations. However, good cluster detection and update are expensive. Cluster updates, if they are not too many, can be done easily on top of the existing clusters. However, periodic re-clustering is needed in this case.

3.2 Non-Parametric Techniques

Non-parametric techniques can be either **algebraic** or **histogram based.**

Algebraic techniques use a polynomial to approximate the density of the distribution of points in a multivariate space. The polynomial has variables that correspond to each attribute. The higher the degree of the polynomial the better (hopefully!) the approximation. For a given degree of the polynomial, the coefficients of terms are calculated from the set of data points so that certain error criteria are satisfied. The coefficients typically have no intuitive interpretation.

The calculation of the coefficients can be done in one file pass in some models, and updating the coefficients of the polynomial can be done by only considering the new set of data points. The approximation of the multivariate space using this technique can be very good. However, non-numeric attribute values have to be converted to numbers before parameter estimation or selectivity estimation. (This is also the case with parametric techniques). The conversion can be done by either using a table or a hashing function. The table may be expensive to maintain and search. The hashing function approach may destroy existing clusterings of data points in the multivariate space and therefore reduce the accuracy of approximation with a given number of terms. Finally, in order to find the distribution followed in a subset of the multidimensional space an integration is needed.

Histogram based techniques use a histogram in each dimension to approximate the distribution of points. The histogram may be constructed by utilizing equal width intervals or equal height intervals [Manino 88]. Multidimensional histograms can be constructed in an analogous manner.

A major advantage of the histogram based techniques is that they avoid the conversion table needed by parametric and polynomial techniques for the non-numeric attributes. This is important for large data bases with many (possibly non-numeric) attributes, as opposed to typical statistical experiments where this aspect is ignored.

One-dimensional histograms are easy to construct. The point density in higher dimensionalities will have to be approximated assuming independence of attributes. Higher dimensionality histograms are typically more expensive to construct especially for large relations. Updates may also be expensive because they may result in different subdivisions of the space.

3.3 The Principle of Maximum Entropy

Jaynes first ([Jaynes 57a], [Jaynes 57b]) proposed that the Shannon's measure of uncertainty (entropy [Shannon 48]), be used to define the values of probabilities when only limited information about the probability distribution is known. This proposal resulted in the foundations of "The Maximum Entropy Principle" ([Levine and Tribus 78]).

Let $H = -\int p(\mathbf{x})\ell n p(\mathbf{x}) d\mathbf{x}$ be the entropy function, where $\mathbf{x}$ is a point in the multivariate space. Assume that certain constraints are known about $P(\mathbf{x})$ (e.g. ranges, means, averages, etc.). The maximum entropy principle states that if the probability density function is not known, the probability density function which maximizes the entropy of the random variable subject to any known constraints is the logical choice ([Levine and Tribus 78], [Tou and Gonzales 74, pp.134-135]). Application of this principle leads to the minimum bias solution, since any other function would show a bias toward information available from known data.

The maximum entropy probability density function is particularly easy to determine when all known constraints are in the form of averages, such as means or variances for the probability density function. For example if the range is only known a uniform density would be chosen by the application of this principle, and if mean and variance are known, a normal probability density would be chosen. Parametric and non-parametric techniques developed in statistics frequently use the maximum entropy principle as a starting point for approximating probability densities.

3.4 Future Research

Future research in the area of providing more detailed multivariate statistical models for the approximation of the distribution of a data point set is difficult. This difficulty arises because such research has extensively been done in the area of statistics and pattern recognition. There is a serious danger of effort duplication. On the other hand, if such models are directly adopted from other disciplines, the estimation of the various selectivities and statistical profiles should be very straightforward. It is also intuitive that the more detailed the model the more accurate the estimation of the costs of operations. More research could possibly emphasize the computational aspects (e.g. efficiency of calculations and updates of the statistics for **large** relations).

The problem that we are after is how to encode maximum information about the distribution of data points in a minimum amount of space. There are alternative ways of viewing the problem at hand: For example, we may want to incorporate some information about the distribution of points only if this results in a significant acquisition of information. The fact that a certain attribute value involves a large number of tuples gives us a lot of **information** in the information theoretic sense. Not only we know that this value involves many tuples, but also that the remaining values involve a small number of tuples.

We suggest here an **information theoretic approach** to the problem of deciding what statistics to incorporate in the statistical profile.

Let $\mathbf{P} = (p_1, \ldots, p_r)$ be a probability vector. The information that is revealed from the knowledge of the value of a particular component of the vector is

$$I_q = -p_q \log p_q$$

It is obvious that extreme values of probabilities reveal more information about the contents of the data base. This easily extends to multivariate spaces.

Formal research is needed in the area to incorporate information theoretic approaches to the selection of statistical profiles.

4. Sensitivity Analysis and Error Propagation

A **Statistical Profile** is the instance of a statistical model (parametric or non-parametric) with a given choice and number of terms (parameters in the case of parametric models, coefficients in the case of algebraic models, etc.). A statistical profile always **approximates** the distribution of points in a multivariate space. Given that it is only an approximation, an important question to ask is what is the impact of this approximation in the quality of the performance estimates that we get using this particular statistical profile. This is loosely called **sensitivity analysis.**

There are some results that have been derived in the area of sensitivity analysis for the case of the "usual" simple model of selectivities that assumes uniformity and independence of attribute values, random placement of qualifying tuples among the blocks of a relation, and constant number of tuples per block.

It has been shown [Rosenthal 80] that the calculation of the size of the join assuming uniformity and independence can also produce accurate results for certain distributions that do not satisfy the uniformity and independence assumptions. This more general class of distributions however is not well characterized or easy to detect. We show a related general result on the average join size later on.

4.1 The Majorization Theory, Its Importance, and Probability Approximations for Database Performance Evaluation

Consider two decreasing probability density vectors $\mathbf{P} = (p_1, \ldots, p_n)$, and $\mathbf{q} = (q_1, \ldots, q_m)$. A branch of Mathematics (theory of majorization) can frequently be used to compare values of functions of vectors that are of a specific kind (Schur functions). In the theory of majorization it is said that **P** majorizes **q** if $\sum_{i=1}^{j} p_i \geq \sum_{i=1}^{j} q_i$ for all $j < n$ and $\sum_{i=1}^{n} p_i = \sum_{i=1}^{n} q_i$. Intuitively if **P** majorizes **q**, **P** is "more skewed" that **q**. If a function $\phi(\mathbf{x})$ is a **Schur convex** (or Schur **increasing**) and **P** majorizes **q**, then $\phi(\mathbf{P}) \geq \phi(\mathbf{q})$. If a function $\phi(\mathbf{x})$ is a **Schur concave** (or **Schur decreasing,**) and **P** majorizes **q**, then $\phi(\mathbf{P}) \leq \phi(\mathbf{q})$. Thus it is easy to describe how a Schur function will behave for more uniform or more skewed probability distributions.

Precise conditions for testing if a function is a Schur function are described in [Marshall and Olkin 79]. For continuously differentiable functions ϕ a necessary and sufficient condition for ϕ to be Schur convex (concave) is that ϕ is symmetric, and that for all $i \# j$, $(x_i - x_j)\left(\frac{\partial \phi}{\partial x_i} - \frac{\partial \phi}{\partial x_j}\right) \geq (\leq) 0$.

In [Christodoulakis 81] and [Christodoulakis 84] the theory of majorization and Schur functions has been associated with data base performance estimates for the first time. It was shown that several performance functions used in data base design as well as performance functions used in query optimization are Schur functions. As a result the performance estimated by these functions is **not the average performance, nor even has some random deviation** from the average performance. It is a **bound** on performance in the sense of the theory of majorization.

Consider a simple example to illustrate the concept. Let a file F of N records, and an attribute A of F. Let $\mathbf{P}(n_1, \ldots, n_M)$ be a vector describing the number of record occurrences n_i for the ith value of attribute A. Assume that the file is in main memory and that user queries are uniformly distributed over all the values of the file. The expected cost is $\frac{1}{M} \sum_{i=1}^{M} n_i = \frac{N}{M} = \frac{1}{M} \left(\frac{N}{M} + \frac{N}{M} + \ldots + \frac{N}{M} \right)$. The last cost is precisely the cost of a uniform probability distribution of attribute values. In this case, using a uniform approximation to the attribute value probability distribution introduces no errors on the average (is exact). However, under the same assumptions, if the file resides on secondary storage, record accesses have to be mapped into block accesses. In this case the expected cost is different for different vectors $\mathbf{P}$. It can be shown that the cost function in this case is a Schur concave function and that among all possible vectors $\mathbf{P}$, the uniform vector results in a maximum expected cost. In this respect then using a uniform approximation to the probability distribution results in a bound (upper bound in this case) of the expected cost.

In any scientific discipline it is very important to understand precisely what we are doing when we use approximations and heuristics. This is also true in performance evaluation. The theory of inequalities in general, and the majorization theory in particular, can help understand better the effect of the approximations involved in every step of database performance evaluation.

In the same paper ([Christodoulakis 84]) it was also pointed out that entropy is a Schur concave function of the probability distribution, thus associating the usual methods of probability density approximation which are based on the maximum entropy principle with the Schur cost functions.

Note that these results are not only applicable to finding an extreme (maximum or minimum in performance) which can be also found by finding the extreme of the cost function subject to the known constraints (e.g. using Lagrange multipliers to show that uniformity results in an extreme of performance [Christodoulakis 81]). This is a special case only. The application of majorization theory can also be used to compare the expected performance when any two non-uniform probability vectors describe the probabilities in two different attributes or files for example. In the previous example, if attribute A has a more skewed probability distribution than attribute B, under the same assumptions, the expected cost for A will be less than the expected cost for B. In the extreme case that a uniform or a single-valued vector are used, bounds of the performance for all possible distributions are obtained.

The application of majorization theory can also be used to study the performance of algorithms for a class of distributions without resorting to exhaustive simulations. A small number of test probability vectors can be used (including the extreme allowed vectors such as the uniform). If the vectors majorize each other the performance observed will monotonically increase or decrease, and the performance for all vectors "in between" two of those vectors (in between according to majorization) will be bound from the performance observed for the two vectors. This observation may save exhaustive simulation studies and very large tables describing results.

Note also that the theory of majorization is not necessarily restricted to showing that uniform distribution approximations of the probability distributions result in performance bounds. Even if more accurate approximations were used, such a possibility exists. This is frequently a consequence of the method used for deriving approximations of the probability densities.

Intuitively, the reason that this may be the case is the fact that the entropy is Schur concave function [Christodoulakis 84]. As we have stated before the maximum entropy principle is frequently the starting point for approximating probability density functions. The approximations used maximize the entropy subject to known constraints (range, mean, variance, etc.). If there are any other constraints (unknown, or not taken into account due to space limitations imposed) the entropy of the actual probability distribution will be less than the entropy of the computed probability distribution. The only majorization relationship that can exist in this case between any actual probability vector $\mathbf{P}$ and the computed one $\mathbf{q}$ (which is an approximation subject to the known constraints) is

P majorizes **q**. Therefore for any performance cost function S that is Schur concave (convex) it will be $S(\mathbf{P}) \leq (\geq) S(\mathbf{q})$. In this sense, $S(\mathbf{P})$ will be a bound on performance.

This argument implies that the computed value may be a bound of the cost even if a more detailed (than uniform) probability density function is used. It is expected however that, the more constraints incorporated (the more information about the probability distribution is known) the less the difference of the computed entropy from the actual entropy (or the computed performance cost from the actual cost).

What makes the observations in [Christodoulakis 84] important is that majorization theory has many applications in data base performance evaluation.

4.2 Applications in Data Base Performance Evaluation

In [Christodoulakis 84] the following approximations are shown to result in upper bounds of cost estimates (block accesses) under certain conditions:

- uniformity and independence of attribute values
- random placement of qualifying records in the file
- constant number of records per block

It is also shown that uniformity and independence of attribute values may result in upper bound estimates of the selected number of distinct values in a joining domain (useful for semi-join cost estimates in distributed systems). In section 4.2.1, we study costs of joins and projections as well as profile estimation problems with applications in query optimization.

Since the first application of majorization to show that certain performance estimates were in fact performance bounds [Christodoulakis 81] a number of other estimates were also shown to be bounds of performance: Zahorian, Bell, and Sevcik observed that when the probabilities of record requests are assumed to be uniform over the records of a life, an (upper) bound of the cost will be calculated [Zahorian, Bell and Sevcik 83]. Piatiensky-Shapiro [Piatiensky-Shapiro 85] used majorization and Schur functions to show that when sampling is used for estimating the number of distinct values of an attribute a **lower bound** of the number of distinct values (cost) will be calculated if uniform distribution of attribute values is assumed. In section 4.2.4, we describe some other cost functions involved in sampling.

Two more performance problems that may result in bound estimates are described in [Christodoulakis 81, pp. 138]. The first is the problem of calculating the expected cost in the case that transactions update objects in a data base. It is intuitively understood that if transactions are assumed to uniformly lock objects in the data base (e.g. the probability that an object is updated is uniform) this will result in a **lower** bound of performance. The reason is that if the probability of an object is high there will be queueing delays for transactions that wait until an object is unlocked. A proof for this result has been difficult. However recently some efforts for such a proof have been made ([Singhal 86], [Singhal 88]).

The second problem proposed in [Christodoulakis 81] is the problem of calculating the expected cost when a large main memory buffer is used. It is expected that when block reference is uniform an upper bound in the performance will be reached. We give in section 4.2.2. a formal proof using majorization.

In [Christodoulakis 84], it is conjectured that uniformity is good (e.g. gives lower bounds of performance) in certain file organizations. In section 4.2.3. we study the impact of non-uniformity in file organizations. The approach is interesting in that it shows the generality of the applicability of Schur functions in performance, and it shows how complex Schur functions can be combined.

4.2.1. Applications to Query Optimization

In this section we describe applications of majorization theory to query optimization. We describe applications related to the cost of joins and projections. The case of selections, semi-joins and other operations has been described in [Christodoulakis 84].

Consider first the case of the join. We will investigate the cost of the join, the size of the result of the join, and the number of distinct values in the join result (new profile), as a function of the distribution of the attribute values of the joining attributes.

Consider first the cost of the join. We do not hope to examine the cost of all join algorithms here. We will examine two well known general cases. In the first case, both relations are sorted on the joining attributes, in the second case one or both relations are not sorted on the joining attributes.

When both relations are sorted on the values of the joining attribute the cost is proportional to the cost of the sequential retrieval of the blocks of each relation from the secondary storage. (We consider the CPU cost of merging to be zero.) In this case, the cost is $\frac{n_1}{b_1} + \frac{n_2}{b_2}$, where n_1 and n_2 are the relation sizes, and b_1 and b_2 are the block sizes for each relation. Therefore the cost is independent on the probability distribution on the joining domains.

When one or both relations are not sorted and an index is used for accessing the values of the joining attribute(s), the cost depends on the probability distributions in the two domains. Consider the case that one of the relations is not sorted on the joining attribute, and an index is used to access its tuples. The cost can be expressed as

$$\frac{n_2}{b_2}+\sum_{i=1}^{N}\left(1-\frac{C_{n_1p_i}^{n_1-b_1}}{C_{n_1p_i}^{n_1}}\right).$$

(We have not included the cost of accessing the index itself.) This cost function is Schur concave [Christodoulakis 84] and therefore the more uniform the probability distribution of the indexed attribute is, the higher the expected cost.

The same conclusion holds for the second attribute if the second attribute is indexed as well. In this case the cost is expressed as

$$\sum_{i=1}^{N}\left(1-\frac{C_{n_1 p_i}^{n_1-b_1}}{C_{n_1 p_i}^{n_1}}\right)+\sum_{i=1}^{N}\left(1-\frac{C_{n_2 q_i}^{n_2-b_2}}{C_{n_2 q_i}^{n_2}}\right)$$

where $\mathbf{P} = (p_1, \ldots, p_N)$ and $\mathbf{Q} = (q_1, \ldots, q_N)$ are the probability distributions of the two attributes. When a uniform probability distribution in each attribute is assumed (instead of the actual one) an upper bound of the expected cost is calculated.

In summary, we examined only the cost of two possibilities for the join. Merging sorted relations residing on secondary storage, and using indexes to access the tuples of one or two unsorted relations from secondary storage. In the first case the cost is independent on the probability distribution on the joining domains, in the second case the cost is a Schur concave function of the probability distribution of the joining domains, and therefore it increases for more uniform distributions (according to majorization). The uniform distribution results in an upper bound of the expected cost.

We will examine next the size of the result of the join. The size of the result enters the cost of a sequence of operations in two ways. First, if the result is written temporarily on secondary storage. Second, if a subsequent join is to be performed, the size of the result of the first join enters the cost formula as described above. The size of the result of course is independent on the join algorithm and therefore this cost is entering the cost formulae for all different join algorithms. The size of the join is given by

$$n_1 n_2 \sum_{i=1}^{N} p_i q_i$$

where n_1 and n_2 are the sizes of the two relations and $\mathbf{P} = \left(p_1, p_2, \ldots, p_N\right)$ and $\mathbf{Q} = \left(q_1, q_2, \ldots, q_N\right)$ are the probability distributions followed by the two joining attributes.

All costs that we considered so far are for secondary storage data bases. Note however that in the case of main memory data bases the cost of the join algorithm itself may frequently be proportional to the size of the join result

[Weddell 87]. When such is the case, the analysis of the size of the join that we provide below also applies to the cost of the join algorithm for main memory resident data bases.

From the above formula we can see that when the probability distribution in one of the two attributes is uniform, the result of the join does not depend on the probability distribution in the joining domains. (The expected cost in this case is $n_1 n_2 \frac{1}{N}$.) We will investigate what happens in more complex situations.

Consider next the case where both probability distributions are the same, both increasing or decreasing, and "not too different" e.g. $\mathbf{P} \approx \mathbf{Q}$. In this case the cost becomes

$$C = n_1 n_2 \sum_{i=1}^{N} \left(p_i\right)^2 .$$

This cost function is symmetric and satisfies $(p_i - p_j)\left(\frac{\partial C}{\partial p_i} - \frac{\partial C}{\partial p_j}\right) \geq 0$. Therefore it is Schur convex (or Schur increasing). As a result, the more skewed the probabilities $\mathbf{P}$, the higher the cost. The cost is minimized (among all distributions $\mathbf{P} \approx \mathbf{Q}$) when $\mathbf{P}$ and $\mathbf{Q}$ become uniform.

However, it is typically the case that the probability distributions of the two attributes are not the same. When the probabilities are not the same, the cost may not be minimized for uniform probabilities in each attribute.

In fact, it can be shown, that, given attributes A and B with probabilities $\mathbf{P}$ and $\mathbf{Q}$ respectively, the join size (and the cost of their main memory join) is less or equal to the cost of the join of two attributes that have as probabilities the average probabilities of A and B, e.g. $\sum_i p_i q_i \leq \sum_i \frac{p_i + q_i}{2} \frac{p_i + q_i}{2}$. (This is true because for a constant sum the product of two numbers maximizes if the numbers are equal). In the special case that $\frac{p_i + q_i}{2} = \frac{1}{N}$ for all i it holds that

$$\sum_i p_i q_i \leq \sum_i \frac{1}{N}\frac{1}{N}.$$

In other words, the cost is lower than the cost of uniform approximations for all those vectors **P** and **Q** that have a uniform average.

It is more difficult to describe what happens in the general case when **P** and **Q** are not the same, and they do not have a uniform average. Two theorems from the theory of inequalities are applicable in this case. The first states that for $p_1 \leq p_2 \cdots \leq p_N$ (e.g. if the values are rearranged so that probabilities are increasing for **P**) then

$$\sum_{i=1}^{N} q_i p_i \leq \sum_{i=1}^{N} q'_i p_i$$

for all **Q** and **Q'** so that

$$\sum_{i=1}^{k} q_i \geq \sum_{i=1}^{k} q'_i, \; k = 1, \ldots, N-1$$

and

$$\sum_{i=1}^{N} q_i = \sum_{i=1}^{N} q'_i.$$

In other words, if **P** is increasing, then the "less decreasing" the **Q** is, the higher the size of the join [Marshall and Olkin 79, pp. 445].

The second theorem is the well known theorem of Hardy, Littlewood and Polya [Hardy, Littlewood and Polya, 1952, pp. 261] that states that

$$\sum_{i=1}^{n} p_{[i]} q_{[n-i+1]} \leq \sum_{i=1}^{n} p_i q_i \leq \sum_{i=1}^{n} p_{[i]} q_{[i]}$$

where $p_{[i]}$,$q_{[i]}$ indicates the rearranged probabilities so that both $p_{[i]}$ and $q_{[i]}$ appear in decreasing order. This inequality states that if the two probability vectors are not both increasing or decreasing, the expected cost will be somewhere in between the cost of arranging both probability vectors in the same order (both increasing or both decreasing, in which case the cost is maximized) and the cost of arranging **P** in increasing order and **Q** in decreasing order (when the cost is minimized).

In summary, the size of the join is always between the size calculated by the product of an increasing rearrangement for **P** and **Q** (maximum size for all rearrangements) and an increasing rearrangement for **P** and a decreasing one for **Q** (minimum size for all rearrangements). If the values of **P** appear in an increasing value order then the "more increasing" according to majorization the **Q** is, the higher the expected size. If both **P** and **Q** follow approximately the same distribution then the expected cost is Schur convex function and therefore the more skewed **P** or **Q** is the more the expected cost. If one of **P** or **Q** is uniform the expected cost is independent on the distribution of the other attribute. Finally, among all **P** and **Q**, the expected cost is maximized when **P** and **Q** are the same permutation of the vector (1,0, . . . ,0). The expected cost is minimized when the non-zero values of **P** correspond to zero values of **Q**. For example $\mathbf{P} = \left(\frac{1}{4},0,\frac{1}{4},0,\frac{1}{2}\right)$ $\mathbf{Q} = \left(0,\frac{1}{2},0,\frac{1}{2},0\right)$.

The above results state that the expected size of the join does not only depend on how skewed the probabilities in each attribute are, but also on the permutation (order) of the probabilities within the probability vectors.

One interesting question to consider is what is the expected size of the join for all permutations of two given probability vectors e.g., given **P** and **Q**, what is the expected cost of the join for all permutations of the components of **P** and all permutations of the components of **Q**. The answer to this question will give us an intuition on how the skewness of the probability distributions affects the size of the join. Note also that it is much easier for the optimizer to keep information on the skewness of the sorted components of a probability vector, than to keep information on the probability height of individual values.

The expected size of the join for all permutations of **P** and **Q** is:

$$n_1 n_2 \sum_i p_i \frac{\sum_i q_i}{N} = \frac{n_1 n_2}{N}.$$

Therefore the expected size of the join for all permutations of two given vectors

P and **Q** is independent on the probability distributions **P** and **Q**, and it is equal to the expected size of the join of two uniform probability distributions. The skewness of the probability distributions on A and B has no effect on the average (for all permutations of values) on the size of the join (or the cost of joins in main memory).

Consider next the change in the statistical profile as a result of a join operation. In particular we want to examine the effect of distributions on the expected number of values in the joining domain. In the case that several joins are performed, the number of values selected by the join may be a very important criterion for query optimization. It may be desirable to select even an expensive join to be performed early in the query processing stage, if this join is to reduce significantly the remaining values in the joining domain. This may have a significant impact on the reduction of the overall cost of the sequence of joins. In that respect this cost may be even more important to study than the size of the join since it may result in more joins or more expensive joins later on in a sequence of operations.

The expected number of values selected by the join on attribute A with probability vector **P** and on vector attribute B with probability **Q** is

$$C = \sum_{i=1}^{N} \left(1-(1-p_i)^{n_1}\right)\left(1-(1-q_i)^{n_2}\right)$$

where n_1 and n_2 are the number of tuples in the two relations, and N is the number of distinct values in the joining domain.

It is easy to verify that when one of the probability distributions is uniform the cost C becomes a symmetric function on the other. Let **P** be uniform, e.g. $p_i = \frac{1}{N}$. Then the cost is

$$C = \left(1-(1-\frac{1}{N})^{n_1}\right)\sum_{i=1}^{N}\left(1-(1-q_i)^{n_2}\right)$$

It can be verified that $\left(\frac{\partial C}{\partial q_i}-\frac{\partial C}{\partial q_j}\right)(q_i-q_j) \leq 0$. Therefore C is Schur concave and it maximizes for a uniform probability distribution **Q**.

In many cases the cost function C can be simplified

$$C = \sum_{i=1}^{N} \left(1-(1-p_i)^{n_1}\right) \left(1-(1-q_i)^{n_2}\right)$$

$$= \sum_{i=1}^{N} \left(1-(1-p_i)^{n_1}-(1-q_i)^{n_2} + (1-p_i)^{n_1}(1-q_i)^{n_2}\right)$$

Keeping the highest terms we can have

$$C_1 \approx \sum_{i=1}^{N} \left(1-(1-p_i)^{n_1}-(1-q_i)^{n_2}\right)$$

or

$$C_2 = \sum_{i=1}^{N} 1-(1-p_i)^{n_1}-(1-q_i)^{n_2}+1-n_1 p_i-n_2 q_i .$$

It can be easily verified that the cost function in this case is symmetric, and that it is Schur concave on **P** and **Q**. The cost is increasing for more uniform probabilities.

In summary, we showed that frequently the result of the join has a number of distinct values which increases for more uniform probability distributions and it maximizes if a uniform probability distribution in the joining domain is assumed. (The conditions under which this statement is true where described above.) In comparison to the join size, the number of distinct values selected in the join is more dependent on the shape of the probability distribution in each domain than on the particular permutation of the probability values of a vector. As a result, it may be easier to predict accurately the number of remaining values than the result size. More intuition on this is given in the section of sampling (4.2.4).

We consider next the expected number of distinct values selected in the join result for all permutations of two given probability vectors. E.g. given **P** and **Q** we consider what is the expected number of distinct values selected in the join result for all permutations of the components of **P** and **Q**. This will give us an intuition on the effect of skew of the probability distribution, independent on the particular permutation of the probability vectors.

Given a permutation of **P**, the probability that a particular value A_i of A exists is

$$1-\left(1-p_i\right)^{n_1}.$$

For this permutation of **P**, and this value A_i, the probability that the value appears in the result when all permutations of **Q** are considered is

$$\left(1-(1-p_i)^{n_1}\right)\frac{\sum_j \left(1-(1-q_j)^{n_j}\right)}{N}.$$

The expected number of distinct values in the resulting relation for all permutations of **P** and **Q** therefore is

$$\sum_i \left(1-(1-p_i)^{n_1}\right)\frac{\sum_j \left(1-(1-q_j)^{n_j}\right)}{N}$$

$$= \frac{1}{N}\left(\sum_i \left(1-(1-p_i)^{n_1}\right)\right)\left(\sum_j \left(1-(1-q_j)^{n_2}\right)\right)$$

This cost function is Schur concave in **P** or **Q**, and therefore it increases for more uniform distributions for **P** or **Q**. The number of distinct values selected maximizes for uniform distributions in both attributes. The skewness of the probability distributions therefore has an impact on the expected number of distinct values selected by the join, and this fact can be easily exploited by query optimizers.

In summary, we have shown that the join size and the number of distinct values selected by the join depend not only on the skewness of the sorted probability distributions of the two attributes but also on the particular permutation of the probability vector components (to a lesser degree for the number of the selected values in the join result). On the average, for all permutations of the probability vectors **P** and **Q**, the expected size of the join is independent on the skewness of the probabilities **P** and **Q** (and equal to the expected size of the join of two uniform probability distributions). However, on the average for all permutations of the probability vectors **P** and **Q**, the expected number of distinct values in the join result depends on the skewness of **P** and **Q**, and it increases for more uniform **P** or **Q**.

The size of the result relation has an immediate impact on the cost of the next operation. The number of distinct values in the joining domain, however, may have a more lasting impact. E.g. it may be desirable to perform a join first even though it may have a large output size when the join is going to be very selective (e.g. reduce greatly the number of distinct values in the joining domain). This is because the next join to be performed may result into an empty relation. To study such optimizations it may be desirable that more detailed statistics is kept on the joining domains. If such information is available, it may be worth in some cases for the optimizer to investigate if performing joins on very skewed domains first would reduce the number of joins required and/or the overall cost of a sequence of joins. When statistics on individual values are not kept, but only information on the skewness of the probability distributions is available, the above results suggest that a heuristic that could be used is to perform early joins on skewed attributes (the size of the relations is also an important factor).

Next we analyze the case of projections. The size of projections has been investigated by several authors. There are two problems of concern. First, given a projection on an attribute A, what is the expected number of tuples remaining in the relation (after duplicate elimination)? Second, given a projection on an attribute A, what is the number of distinct values selected on an attribute B? The latter is needed if B is going to be used as a joining attribute for subsequent joins.

Both problems can be modelled as selection problems in a multidimensional space. Consider the two-dimensional case for simplicity (without a loss of generality). Let A and B be two attributes of a binary relation of n tuples and let p_{ij} be the probability distribution in the two-dimensional space which is defined by the values of A and B. Let i follow the dimension that corresponds to A, and j follow the dimension that corresponds to B. Consider a projection on A. The size of the resulting relation is equal to the number of distinct values of B

selected by the projection.

The probability that a given value j of B is selected by any of the n tuples is

$$1-\left(1-\sum_i p_{ij}\right)^n$$

The expected number of values of B selected by the projection is

$$\sum_j \left(1-(1-\sum_i p_{ij})^n\right)$$

It is clear that this cost function is Schur concave on $p_j = \sum_i p_{ij}$. As a result, the more uniform the probability distribution of p_j is the higher the expected cost.

The expected size of the projection is maximized when a uniform probability distribution is assumed in each attribute. Since the estimation of the number of values selected in an attribute as result of a projection can be done using a similar formula, this cost function is also maximized for uniform distributions.

In summary, both performance estimates in the case of projection (the size of the resulting relation, as well as the number of distinct values of an attribute selected as a result of the projection) are Schur concave functions, and they are maximized when uniform approximations of the probability distributions in each attribute are assumed. In that respect projection is similar to selection. It was shown in [Christodoulakis 84] that the average selection cost (for secondary storage relations) maximizes for uniform distributions, and that the number of distinct values of another attribute that exist in the relation resulting from a selection is maximized when uniformity and independence of attribute values is assumed.

4.2.2 Buffered IO Processing

Consider a main memory buffer of size B blocks. The probability that a block i is not in the buffer at any point in time is $(1-p_i)^B$, where p_i is the probability of block i. The probability therefore that a request for block i is given next, and the block i does not exist in the buffer is

$$(1-p_i)^B p_i$$

Therefore the expected cost in a sequence of N block retrievals is

$$C = \sum_{i=1}^{N} (1-p_i)^B p_i$$

secondary storage block accesses. Differentiating we obtain

$$\begin{aligned}\frac{\partial C}{\partial p_i} &= \left(1-p_i\right)^B - Bp_i(1-p_i)^{B-1} \\ &= (1-p_i)^B \left(1-B\frac{p_i}{1-p_i}\right)\end{aligned}$$

This is a decreasing function of p_i. Therefore

$$\begin{aligned}(p_i-p_j)\left(\frac{\partial c}{\partial p_i}-\frac{\partial c}{\nu p_j}\right) &= (p_i-p_j)\left[\left(1-p_j\right)^B\left(1-B\frac{p_i}{1-p_i}\right)-\left(1-p_j\right)^B\left(1-B\frac{p_j}{1-p_j}\right)\right] \\ &\leq 0\end{aligned}$$

and thus C is Schur concave. Therefore the more "uniform" the block probability vector according to majorization, the higher the expected cost. In the special case where the block probabilities are uniform the cost is maximized.

4.2.3. File Organizations

Majorization theory may have many applications in file organization performance analysis. In [Christodoulakis 84] it is conjectured that uniformity is desirable (e.g. should lead to **lower bound** estimates in several file organizations. As examples superimposed coding and hashing are stated.)

It is easy to see that more uniform distributions of k bits across the M blocks of a superimposed coding file organization will result in a lower cost. Let n_i be the number of bits in block i, $\sum_{i=1}^{n} n_i = k$. The false drop probability for block i is $\left(\frac{n_i}{b}\right)^r$ where b is the number of bits per signature block [Christodoulakis and Faloutsos 84] and r is the number of bits set "on" by a single word. The total cost then can be expressed as

$$\sum_{i=1}^{M} \left\lceil \frac{n_i}{b} \right\rceil^r .$$

It can be easily verified that this cost function is Schur convex and therefore more uniform distributions of bits per signature block will result in better performance.

The case of hashing is much more interesting because it illustrates an application of compositions of Schur functions with many other potential applications.

The analysis of the expected cost of file organizations frequently uses a multivariate probability distribution to calculate the probability that a given bucket will receive x records. The cost is typically calculated by multiplying the probability of having x records in the bucket times the cost for x records, and summing over all values of x, for all buckets of the file. For example analysis of hashing organizations frequently employs a Multinominal distribution or a Poisson distribution to describe the probability of an allocation of records in a file described by a vector $\mathbf{x} = (x_1, \ldots, x_M)$ where M is the number of buckets in the file. The cost for a given vector $\mathbf{x}$ is typically a sum of costs of individual buckets, e.g. it has the form $\phi(x_1, \ldots, x_M) = \sum_{i=1}^{M} g(x_i)$ where $g(x_i)$ is may be a convex function of $g(x_i)$.

For example consider the case of hashing where the primary bucket can hold up to b records, and each overflow record is stored in separate buckets. Using the Multinomial probability distribution the overflow cost can be written as

$$C = \sum_{i}^{M} \phi(c_1, \ldots, c_M) \begin{pmatrix} N \\ x_1, \ldots, x_M \end{pmatrix} \prod_{i=1}^{M} \lambda_i^{x_i}$$

with $\phi = \sum_{i=1}^{M} c_i$, $c_i = x_i - b$ for $x_i > b$, zero otherwise. In this cost function λ_i is the probability of bucket i, x_i the number of records that hash in bucket i, and N the total number of records.

Using the Poisson approximation the cost can alternatively be expressed as

$$C = \sum_{i=1}^{M} \phi(c_1, \ldots, c_M) \prod_{i=1}^{M} \frac{e^{-\lambda_i} \lambda_i^{x_i}}{x_i!}$$

In both cases the cost function for a given vector **x** is $\phi(c_1, \ldots, c_M) = \sum_{i=1}^{M} c_i$, with $c_i = x_i - b$ for $x_i > b$, zero otherwise. This function is a special case of function of the form $\phi(c_1, \ldots, c_M) = \sum_{i=1}^{M} f(x_i)$ with $f(x)$ being a convex function. It has been shown by Hardy Littlewood and Polya [Hardy, Littlewood and Polya 1929] that any function $\phi(c_1, \ldots, c_M) = \sum_{i=1}^{M} f(x_i)$ with $f(x_i)$ being convex is a Schur convex function. It has also been proven [Rinnott 73] that the expectation for a Multinomial or a Poisson probability distribution of a Schur convex function is also a Schur convex function. Therefore the functions

$$C = \sum_{i=1}^{M} \phi(c_1, \ldots, c_M) \prod_{i=1}^{M} \frac{e^{-\lambda_i} \lambda_i^{x_i}}{x_i!}$$

and

$$C(\lambda_1, \ldots, \lambda_M) = \sum_{i=1}^{M} \phi(c_1, \ldots, c_M) \begin{pmatrix} N \\ x_1, \ldots, x_M \end{pmatrix} \prod_{i=1}^{M} \lambda_i^{x_i}$$

are Schur convex functions with respect to $\lambda = (\lambda_1, \ldots, \lambda_M)$. Both cost functions for hashing are therefore Schur convex with respect to bucket probabilities. As a special case this shows that the uniform fix probability $\lambda_1 = \lambda_2 \cdots = \lambda_M$ will minimize the expected cost for hashing.

Several more probability density functions preserve the Schur property when they are used to find the expected value of a Schur cost function. Such probability densities for example are the binomial, hypergeometric, negative multinomial, negative hypergeometric, etc. [Marchal and Olkin 79]. These probability functions frequently appear in many performance analysis problems. For example, it has recently been shown that the probability distribution followed in ISAM organizations is negative binomial [Christodoulakis, Manolopoulos, and Larson 89]. It is this generality of applicability of the majorization theory to stochastic majorization, and to probabilistic and statistical applications that makes the theory important to performance studies.

4.2.4. Sampling

Sampling has been used as a way to estimate the parameters of a distribution. Since sampling requires only part of the data base it may result in more inexpensive calculation of the parameters.

When sampling there are three important considerations. First, how the results of sampling with or without replacement compare. Second, how the expected number of distinct values selected by a sample relates to the total expected number of distinct values in the population. Third, how the expected number of distinct values selected by two samples of the same size from different populations relate to each other.

Consider a selection of n records with replacement. The expected number of values selected is given by

$$\begin{aligned} C_1 &= \sum_{i=1}^{M} \left[1 - \left(1 - \frac{n_i}{N} \right)^n \right] \\ &= M - \sum_{i=1}^{M} \left(1 - \frac{n_i}{N} \right)^n \end{aligned}$$

where $N = n_1 + n_2 + ... + n_M$ is the number of records in the file. The expected number of values selected by sampling without replacement is:

$$C_2 = M - \sum_{i=1}^{M} \frac{C_n^{N-n_i}}{C_n^N}.$$

It can be shown by expansion that $C_2 > C_1$. Therefore the non-replacement sampling results in higher expected distinct values selected than the sampling with replacement. It is also clear that in both models the number of distinct values selected is an increasing function of the sample size, and that for $n = N$ the non-replacement sampling results in selecting all the distinct values, while the replacement sampling results in less that M distinct values.

Consider now two different populations $\mathbf{P} = (p_1, \ldots, p_N)$ and $\mathbf{Q} = (q_1, \ldots, q_N)$. The expected number of values selected using sampling with replacement is

$$C(\mathbf{x}) = \sum_{i=1}^{N}(1-(1-x_i)^n)$$

Since this function is Schur concave with respect to $\mathbf{x}$, if $\mathbf{P}$ majorizes $\mathbf{Q}$ then the sample from $\mathbf{P}$ will select less distinct values than the sample from $\mathbf{Q}$ (in the expected case).

Sampling from a uniform distribution will produce the highest number of distinct values in the expected case.

Sampling relates to the problem of estimating the number of distinct values remaining in a domain after a relational operator is applied to a relation. According to the above, the more uniform the distribution is, the higher the expected number of distinct values selected. In the case of join, sampling happens in two attributes A and B. The result of sampling is two vectors of zeros and ones. The multiplication of the corresponding vector components gives the vector with the selected values. The more uniform distributions in A or B will result in a higher number of ones in the vectors representing the results of sampling. On the average over all permutations the product of the corresponding components will increase for more uniform distributions.

4.3 Limits of Applicability and Extensions

In [Christodoulakis 81, pp.138] it is also **explicitedly stated** that the implications of the assumptions (used in the probability approximations) have been studied **in isolation.** If a single cost function (one of those described above) completely characterizes the problem then the results described above can directly be applied. It is not however understood how performance estimates interact in a more complicated environment where several operations take place (as a sequence or in parallel). It is dangerous to make such generalizations without a formal theory to back them up. One reason is that the overall performance ratio (calculated versus actual) may depend heavily on the frequency of operations in the particular environment. Nevertheless experimental results for particular environments have been reported by various researchers. Results on the composition of cost functions similar to those described above for the file organizations may be applicable to more formal studies of the cost behavior in cases where a sequence of operations takes place.

It is clear that the results reported in [Christodoulakis 81] and [Christodoulakis 84] are applicable to data bases residing on magnetic disks since all cost functions map record accesses to block accesses. (The only exception is the semi-join size estimates). The same results **may not be applicable when other storage devices are used** for storing the data base. A particular example is main memory resident data bases. An extensive set of performance estimates for main memory resident data bases was reported by Weddell in his Ph.D. thesis [Weddell 87]. It was early observed that the cost of selections for main memory data bases is exact (e.g. not a bound as it was for secondary storage selections) even when attributes have non-uniform distributions and dependencies exist (provided that the queries are uniformly distributed) [Christodoulakis and Weddell 86]. An example for selections in main memory was given early in this paper.

The results of the size of the join presented earlier, and the number of distinct values selected in the join result are also applicable to main memory resident data bases. Moreover, the cost of performing a join is often proportional to the size of the join for main memory databases. In this case, the previous results also apply to the join cost. The results of projection are also applicable to main memory resident databases.

The effect of skew of the probability distributions in joining domains on various join algorithms in main memory databases with multiple processors has been extensively studied at IBM Watson Center and reported in [Lakshmi and Yu 88]. It has been found that skew has a major impact on performance and it should be used as one of the major parameters modeling performance. It is shown that when no skew exists performance is best, and it is suggested that straightforward generalizations of conventional join algorithms may not be adequate to handle the skew problem in environments with large number of processors. In this environment assuming uniform distributions in the joining domains may **lead in lower performance bounds.**

4.4 Future Research

More research is badly needed in the area. It is important to understand the import of the assumptions made on performance. It is also crucial to understand when to use or reject a particular profile. If a profile is unreliable it makes no sense to use it in query optimization. Most models evaluate a profile based on an average error minimization criterion using the data points of the distribution. The average error does not allow an error bound on an individual query. However, it is well known that in a given environment queries are highly skewed (80/20 rule). If it is the case that queries in an environment are directed towards values with large deviations from the profile estimate then the overall error may be high. Approaches that bound the error of individual value estimates when they calculate a particular profile may be more reliable.

In the case that multidimensional histograms are used, this suggests the construction of histograms based on a maximum error criterion [Christodoulakis 81]. In the case that a maximum error criterion is used the histogram based approach may result in certain high values of distributions stored separately (for skewed distributions).

Little is understood about the effect of the profile chosen on the cost of a sequence of operations. Not all cost estimates for the usual model are pessimistic (for example join estimates are not necessarily pessimistic). Combining joins with other operations may result in unpredictable (upper or lower) estimates.

If we consider a sequence of operations such as a sequence of joins it is worth knowing the **error propagation** in our estimates. The error propagation will tell us for how many operations we may use our selectivity estimates reliably. If the error in the estimation is large in some sense, alternative query optimization techniques that do not rely on selectivity estimates may be used. Note that the acceptable error may depend on the alternative query optimization strategies available.

We suggest here that formal research is needed in the area. Such research is well accepted and understood in other disciplines (for example numerical analysis). It is surprising to us that it has not yet been applied to selectivity estimation. We note also that the study error propagation in a sequence of operations may imply some restrictions in the choice of the original statistical model. We suggest here again that a more appropriate criterion for the design of multivariate approximations is a maximum error criterion as opposed to an average error criterion.

5. Capturing Attribute Dependencies and Estimating Block Selectivities

Attributes depend on each other. Attribute dependencies may result in serious errors in selectivity estimation if they are not captured by the model. Functional dependencies, multivalued dependencies, and correlations are well understood, frequently encountered, and it is relatively easy to capture them in a model (requires little space). There are many other forms of dependencies of attribute values which may be more difficult to detect and more space consuming to capture automatically. However, some of these dependencies may be captured as part of the data base design phase (during interviews). This is an advantage over traditional statistical approaches that derive all their information from existing data points.

Dependencies have serious impact in the quality of block selectivity estimates. Consider a relation clustered on the values of an attribute A, and selections on an attribute B. If there are dependencies between the values of A and B the cost estimates of selections may be highly pessimistic. For example, in a relation of employees where the employees are clustered on the values of the attribute Department, any employees with the skill of an engineer will be located in a small number of blocks since only one department or two may have engineers. To capture reliably the block selectivity in such a query, the dependencies between attributes A and B must be modeled.

Another case where the block selectivities may be overly pessimistic is when new records are always inserted at the end of a file. Such insertions create high correlations between time and block number. Attributes that correlate highly with time will also correlate highly with the block number. For example, if new employees are always inserted at the end of the file, then salary, age, number of years in the organization, rank and other attributes will highly correlate with the block number.

Future research in this area is promising. Frequent types of attribute dependencies should be identified. For example one type of dependency may be the maximum number of non-zero pairs of values of two attributes. Dependency types that are identified should be used to derive cost estimates and new profiles in sequences of operations.

6. Other Topics

Several other topics in selectivity estimation are worth investigating. Such topics include:

- Proposing systematic ways to deal with model adaptability and periodic changes. Such questions have already been discussed in the context of index selection.
- Investigate sampling as a means of verifying the reliability of selectivity estimates during the process of query evaluation. The system could dynamically modify the query evaluation strategy when significant discrepancies are observed.
- Estimate selectivities in large text dbms's. Selectivities in such an environment are important not only for query optimization but also for returning an estimate about the size of the response to the users.
- Provide selectivity estimates for accessing information from new storage devices (such as optical disks).

7. Conclusions

We have presented a highly biased view of topics in selectivity estimation for use in query optimization.

We have suggested what in our view are the most promising directions for research in this area and what topics should be avoided.

In summary information theoretic approaches, maximum error bound approximations, attribute dependency modeling and its use in block selectivities and other cost estimates, and error propagation studies in a sequence of operations are in our opinion the most important research directions in the area.

We have emphasized the applications of the theory of functional inequalities and in particular of majorization and Schur functions in data base performance problems. We have produced new results significantly extending the results in [Christodoulakis 84] showing the large range of applicability of this theory.

We studied the impact of the probability distributions on the cost functions used for joins, projections, and new profile calculation. The cost of join in general depends on the join algorithm. Merge joins for sorted relations on secondary storage are independent on the probability distribution in the joining domains while when an index is used for one or both attributes, more uniform distributions result in higher costs. The size of the join result is very sensitive to the probability distributions in the joining domains. The size increases when the peaks of the two distributions are rearranged to correspond to the same values and it decreases when the peaks of the first distribution are rearranged to correspond to the valeys of the second distribution. On the average however, overall permutations of the two probability vectors the size of the join is independent on the skewness of the probability distributions involved. The number of remaining values in the joining domain (new profile) is important when a sequence of joins is performed. This number increases in general for more uniform probability distributions in the two attributes. When all permutations of two probability vectors are considered, the number of distinct values selected in the join is a Schur concave function on the probability distributions involved, and therefore it increases for more uniform distributions in each attribute.

We showed that the size of the projection as well as the number of distinct values of an attribute remaining after the projection depend on the probability distribution of the attributes of the projection. The more uniform distributions result in higher costs in both cases.

We studied the effect of the assumption of uniform access probabilities to the blocks of a buffer and we showed that uniformity leads in an upper bound of the cost.

These results complement the results presented in [Christodoulakis 84], where the impact of the skewness of probability distributions was studied for the case of selections, semi-joins, and block accesses for non-randomly placed records was studied.

We proved that uniformity results frequently in a lower bound of the cost in file organizations such as hashing and superimposed coding, and we showed some applications of Schur functions in sampling.

Finally we suggested several important research directions for the estimation of selectivities.

References

[Christodoulakis, Manolopoulos and Larson 89] Christodoulakis, S., Manolopoulos, Y., and Larson, P.: "Analysis of Overflow Handling for Variable Length Records," Information Systems, 14,2, 1989.

[Christodoulakis and Faloutsos 84] Christodoulakis, S., and Faloutsos, C.: "Design Considerations for a Message File Server," IEEE Transactions on Software Engineering, SE-10, 2, March 84, 201-210.

[Christodoulakis 81] Christodoulakis, S.: "Estimating Selectivities in Data Bases", Report CSRG 136, Department of Computer Science, University of Toronto, 1981.

[Christodoulakis 83] Christodoulakis, S.: "Estimating Record Selectivities," Information Systems 8,2, 1983, pp.105-115.

[Christodoulakis 84] Christodoulakis, S.: "Implications of Certain Assumptions in Database Performance Evaluation," ACM TODS 9,2, June 84, pp.163-186.

[Christodoulakis 84a] Christodoulakis, S.: "Estimating Block Selectivities," Information Systems, 9,1, 1984.

[Christodoulakis and Weddell 86] Private Communication.

[Hardy, Littlewood and Polya 1952] Hardy, G., Littlewood, J., and Polya, G.: "Inequalities," 2nd edition, Cambridge University Press, London, 1952.

[Hardy, Littlewood and Polya 1929] Hardy, G., Littlewood, J. and Polya, G.: "Some simple inequalities satisfied by convex functions," Messenger Math., 58, pp.145-152.

[Jaynes 57a] Jaynes, E.T.: "Information Theory and Statistical Mechanics", Phys. Rev. 106, pp.620, 1957.

[Jaynes 57b] Jaynes, E.T.: "Information Theory and Statistical Mechanics", Phys. Rev. 108, pp.171, 1957.

[Lakshmi and Yu 88] M.S. Lakshmi and P.S. Yu: "Effect of Skew on Join Performance in Parallel Architectures", Proceedings ACM-IEEE International Symposium on Databases in Parallel and Distributed Systems, 1988.

[Levine and Tribus 78] Levine, R.D., and Tribus, M.: "The Maximum Entropy Principle", M.I.T. Press, 1978.

[Manino, Chu, and Sager 88] Manino, M., Chu, P., and Sager, S.: "Statistical Profile Estimation in Database Systems," ACM Computing Surveys: 20,3, Sept. 1988, 191.

[Marshall and Olkin 79] "Inequalities: Theory of Majorization and its Applications", Academic Press, 1979.

[Piatiensky-Shapiro 85] Piatiensky-Shapiro: "Estimating the number of distinct attribute values by using sampling," submitted.

[Rinnott 73] Rinnott, Y.: "Multivariate Majorization, and Rearrangement Inequalities with Some Applications in Probability and Statistics, Israel Journal of Math., 15, pp.60-77.

[Shannon 48] Shannon, C.E.: "A Mathematical Theory of Communication, The Bell System Technical Journal, 27, pp.379-623, 1948.

[Singhal 88a] Singhal, M.: "Proof of Lower Bound for the Probability of Conflicts Under Uniform Data Access Distribution for Databases", Technical Report, Department of Computer and Information Science, The Ohio State University, 2036 Neil Avenue Mall, Columbus, OH 43210.

[Singhal 88b] M. Singhal and Y. Yesha: "A Polynomial Algorithm for Computation of the Probability of Conflicts in Database Under Arbitrary Data Access Distribution", Information Processing Letters, 27, 2, 1988, pp.69-74.

[Teorey and Fry 82] Teorey, T. and Fry, J.: "Database Design", Prentice-Hall, 1982.

[Tou and Gonzalez 74] Tou, J.T. and Gonzalez, R.C.: "Pattern Recognition Principles", Addison Wesley, 1974.

[Weddell 87] G. Weddell, Ph.D. Thesis, University of Waterloo Department of Computer Science, "Physical Design and Query Optimization for a Semantic Data Model", 1987.

[Zahorian, Bell and Sevcik 83] Zahorian, J., Bell, B., and Sevcik, K.: “Estimating block transfers when record access probabilities are non-uniform”, Information Processing Letters 16,5, June 83, 249-252.