ACCURATE ESTIMATION OF THE NUMBER OF TUPLES SATISFYING A CONDITION

Gregory Piatetsky-Shapiro
Department of Computer Science, New York University and
Advanced Database Systems Division, Strategic Information,
80 Blanchard Road, Burlington, Mass 01803.

Charles Connell
Department of Computer Science, Boston University and
Advanced Database Systems Division, Strategic Information.

Abstract

We present a new method for estimating the number of tuples satisfying a condition of the type *attribute rel constant*, where *rel* is one of "=", ">", "<", "$\geq$", "$\leq$". Our method gives highly accurate, yet easy to compute, estimates. We store information about attribute values as a list of *distribution steps* (histograms where buckets, instead of having equal width, have equal height). These distribution steps provide an upper bound on the error when estimating the number of tuples satisfying a condition. The estimation error can be arbitrarily reduced by increasing the number of steps. We analyze desirable conditions that such estimates should satisfy. Based on the distribution steps, we derive a set of estimation formulas which minimize the worst-case error. We also present another set of formulas which reduce the average-case error. Finally, we show how to use sampling to compute a close approximation of the distribution steps very quickly. The major applications of our method are in query optimization and in answering statistical queries.

1. INTRODUCTION

[1] The cornerstone of a relational DBMS is the ability of the system to select the cheapest evaluation method for any given query. The precise estimation of the cost of an evaluation method is complex [Blasgen 77, Selinger 79, Yao 79] but approximately the cost is proportional to the number of I/O-s, which is a monotonic function of the number of tuples retrieved [Yao 77, Whang 83, Luk 83]. An efficient query evaluation method therefore minimizes the number of tuples retrieved at each step. Thus estimation of the number of tuples satisfying a condition is a prerequisite for, and a basis of, query optimization.

[1]

© 1984 ACM 0-89791-128-8/84/006/0256 $00 75

File Industrial Compustat has about 50,000 tuples - 20 years of financial information for about 2,500 companies traded on New-York stock exchange. Consider, for example, a query on that file

```
Find the number of tuples where
SALES > 20 billion  AND YEAR = 77
```

Two of the possible evaluation methods are:

- using an index on SALES, get a list of pointers to all tuples with SALES > 20 billion, for each pointer get the tuple and check if YEAR = 77 .
- using an index on YEAR, get a list of pointers to all tuples with YEAR = 77, for each pointer get the tuple and check if SALES > 20 billion.

To decide whether to use an index on YEAR or on SALES, we need to estimate how many tuples satisfy SALES > 20 billion and how many satisfy YEAR = 77. We then choose the index which returns the smallest pointer list. This will be the faster evaluation method.

The minimum value of SALES[2] in the relation is 0, the maximum is 108 billion. The values of YEAR are spread almost equally among 20 values: 63, 64, ..., 82. The System R optimizer [Selinger 79] would estimate that only 1/20 (5%) of all tuples satisfy the condition on YEAR, while 77/108 (71%) of all tuples satisfy the condition on SALES. It chooses the index on YEAR. Query evaluation by this method took 137 seconds of elapsed time, 41 seconds of CPU time and 2414 disc I/O-s.

Using distribution steps and the estimation formulas presented later in this paper, the query optimizer for the FASTSCAN Query System [Piatetsky 84] guessed that only 2% of the tuples satisfy the condition on SALES (the correct number is 0.5%) and chose that index. Query evaluation by this method took 8 seconds of elapsed time (16 times faster), 2.5 seconds of CPU time and 108 disc I/O.

Definition: *Selectivity* of a condition *E*, denoted SEL(E), is the fraction of tuples satisfying this condition. For example, SEL(sex = 'female') is about 0.5 in the general population.

In this paper we develop a new and detailed theory for estimating the selectivity of comparisons:

[2] more information on distribution of SALES is given in sec 9

attribute = *const*
attribute > *const*   *attribute* < *const*
attribute ≥ *const*   *attribute* ≤ *const*

Using selectivity estimates for these simple conditions, we can accurately estimate the cost of different query evaluation schemes - even when the total query is more complex than a simple condition. Often we can make the *best* choice of a query evaluation method, with substantial savings of computational resources. Our estimation scheme itself uses very little storage and cpu time.

Other uses of the estimations are in those applications (e.g., statistical analysis) where users are frequently interested only in the approximate size of the query result. If the estimate is accurate enough, it can be given to the user before or instead of the potentially expensive query evaluation. Moreover, using selectivity we may determine that *no* tuples or *all* tuples satisfy the query - again skipping the query evaluation.

Most of the work described in this paper was carried out as a part of the Ph.D. research by Gregory Piatetsky-Shapiro on Self-Organizing Database Systems [Piatetsky 84].

Notation: Generally, we will use *X*, *Y* or *const* to denote constants. *T* will denote the number of tuples in a relation. *A*, *attr* will denote an attribute.

When the attribute under discussion is unambiguous, we we will use SEL(*rel X*) to denote the selectivity of comparison *attr rel X*. For example, SEL(>3500) is the selectivity of *attr* > 3500.

SEL~(rel X) will denote an estimate of SEL(*rel X*).

2. OVERVIEW OF THE PAPER

We begin by analyzing how to store information about attribute values distribution. One possible scheme is a histogram: divide the range of attribute values into *K* buckets and count the number of tuples falling into each bucket. We show that this scheme is not very good for estimating selectivity, since it frequently leads to estimates not much better than those obtained by random guessing. We argue that this happens because traditional histograms control the wrong parameter - they force buckets to have equal width. Instead, we should have buckets of equal height.

This leads us to the second scheme, called *distribution steps.* In this scheme we collect the values for a given attribute from all the tuples in a relation and sort them according to the intrinsic ordering of the domain. After choosing the number of steps S, we find the S values for that attribute such that there are an equal number of tuples between each step. The maximum selectivity estimation error from distribution steps is approximately $1/S$. Thus, it can be arbitrarily reduced by increasing S.

Next, we analyze conditions that are desirable (in general) on selectivity estimates and give the axioms of *consistency*, *correctness on ends* and *monotonicity*.

We then examine selectivity estimates based on our distribution steps and give two sets of formulas for getting these estimates from the distribution steps. The first minimizes the worst-case estimation error. The second, used in the FASTSCAN query system, has a much smaller average error in important cases, at the cost of a slightly increased worst-case error.

Lastly, we show how to construct a close approximation of true distribution steps using sampling, and examine the relationship between sample size and the selectivity estimation errors.

3 REVIEW OF THE PREVIOUS WORK

One of the first attempts at estimating comparison selectivity was done by the designers of System R [Selinger 79]. Their scheme relies on the minimum and maximum attribute values (denoted *minval* and *maxval*), and the number of different attribute values (denoted *ndiff*). Their estimates are:

$$\mathrm{SEL}{\sim}(=X) = 1 \,/\, \mathit{ndiff}$$

$$\mathrm{SEL}{\sim}(<X) = \frac{X - \mathit{minval}}{\mathit{maxval} - \mathit{minval}}$$

$$\mathrm{SEL}{\sim}(>X) = \frac{\mathit{maxval} - X}{\mathit{maxval} - \mathit{minval}}$$

$$\mathrm{SEL}{\sim}(\leq X) = \mathrm{SEL}{\sim}(<X)$$

$$\mathrm{SEL}{\sim}(\geq X) = \mathrm{SEL}{\sim}(>X)$$

These formulas are obtained by assuming that attribute values are linearly and uniformly distributed between *minval* and *maxval*. When, as is often the case, the distribution is neither uniform nor linear, the formulas can be grossly inaccurate with an upper bound on the estimation error being 100%; the estimate can be close to 0, while the actual selectivity is close to 1, and vice versa.

Yu and others [Yu 78] examined estimation of the number of tuples satisfying a query in the context of text retrieval. In their model all attributes have binary values (either 0 or 1), queries are conjunctions of attributes, and tuples are stored in clusters of similar records They have derived estimates for the number of tuples in a cluster having K or more attributes in common with a query.

Christodoulakis suggests histogramming for single-attribute conditions [Christodoulakis 81]. He also derives a very good estimation of sizes of joins and block transfers in [Christodoulakis 83]. In [Demolombe 80] only conjunctions of equalities are considered and the main thrust is toward quantitative estimation of inter- and intra-relation attribute dependency.

A novel approach is taken in [Rowe 83] - statistical functions are computed not bottom-up, but top-down. The functions (such as MIN, MAX, AVG) are precomputed on a collection of initial database subsets, called the *database abstract*. Then the function values on other subsets are estimated by inferring (using a production system) their upper and lower bounds from their values on initial subsets.

However, none of these authors describe accurate estimation of comparison selectivity with a guaranteed precision and none analyze the desirable conditions on the estimation scheme.

4. DESCRIBING ATTRIBUTE VALUE DISTRIBUTION

Attribute value distributions rarely have any closed functional description, such as Zipf distribution [Zipf 49] which was observed (very approximately) in last names in a telephone book [Samson 83]. In most cases the only description is the distribution itself - the list of values of that attribute across all tuples. Since that takes too much space, we should look for more compact schemes. We examine two alternatives in the following subsections. Our criterion for comparison is the the worst-case, (i.e., maximum possible) estimation error.

4.1. Scheme 1: a Histogram

One possible approach is to use a histogram. We divide the range of attribute values into *K* equal-width buckets and count the number of values falling into each bucket.

Let us consider a (hypothetical) distribution of attribute AGE in a relation with information on 100 employees of a small software company. AGE ranges from 10 (super-bright whiz kid) to 60 (company president). Let the histogram divide the range into 10 buckets (see next figure).

```
          |           40   40
          |          XXXX XXXX
    # of  |          XXXX XXXX
employees |       5  XXXX XXXX  5     5
          |  1  XXXX XXXX XXXX XXXX  XXXX  1    1    1    1
          +---------------------------------------------------
          10   15   20   25   30   35   40   45   50   55   60

                                  A G E
```

Figure 4-1: Scheme 1: Histogram

A histogram is cheap to compute since it requires only a single pass through the relation. Its precision, however, is not always good. For example, all it tells us about SEL(<29) (the percentage of the employees who are younger than 29) is

0 46 ≤ SEL(<29) ≤ 0.86

The true fraction of tuples with AGE < 29 is anywhere from 0.46 to 0.86. If we take our estimate of SEL(<29) as the mid-point in this range (0.66), then the estimate can be wrong by 0 20

The maximum error in estimating SEL(<*X*) is half the height of the bucket in which *X* falls. For an unlucky distribution of attribute values (where the tallest bucket contains almost 100% of the tuples) a selectivity estimate from a histogram can be wrong by almost 0.5.[3]

Such a situation is quite common. For example, for SALES in the Industrial Compustat relation, the minimum value is 0 and the maximum value is 108,108 million. Yet 95% of the companies have SALES < 2,613 million, which is 2.4% of the maximum value. So if we divide the range of SALES into 10 or 20 or even 40 equal steps, almost all tuples will fall into the first bucket. Then for any *X* < 2,613 million, the maximum estimation error on SEL~(<*X*) is close to 0.5 .

From this discussion we see that the way to control the maximum estimation error is to control the *height* of each bucket. Thus, we should try to create a "histogram" where all buckets have equal height (instead of equal width). Such scheme is described in the next section.

[3] We note that we can get an estimate with a maximum error of 0 5 with no work at all by always setting SEL~(<X) = 0 5

4.2. Scheme 2: Distribution Steps

Consider an attribute A of some relation R. Distribution steps STEP(0), STEP(1), ..., STEP(S) are values such that the fraction of tuples with A < STEP(I) is less than or equal to I/S. For example, suppose we choose S=10 STEP(0) is the minimum value of A, and STEP(10) is the maximum value of A. STEP(3) is a value such that about 30% of all tuples have A < STEP(3).

Practically, we compute distribution steps by the following procedure:

1 Collect the values of A from all the tuples in a relation and sort them in ascending order according to the *intrinsic* ordering of the domain. We note that this ordering must exist and be unique, for otherwise comparison A < X is not meaningful.

2. Select, depending on the desired accuracy and available storage, the number of distribution steps S. Select S+1 positions (including the first and the last) in the sorted list of attribute values, such that there is the same number of attribute values between any[4] two succesive positions. These positions are 1, 1+N, 1+2N, ..., 1+(S-1)*N, 1+S*N=T, where N = (T-1)/S.

3. Take values found in these positions in the sorted list of all values and let them be the distribution steps STEP(0), STEP(1), ..., STEP(S).

From the construction process we see that

> for every step I,
> the number of tuples with A < STEP(I)
> is $\leq$ I*N.

The data from the previous example, as it would be represented using 10 distribution steps, is shown in the following figure.

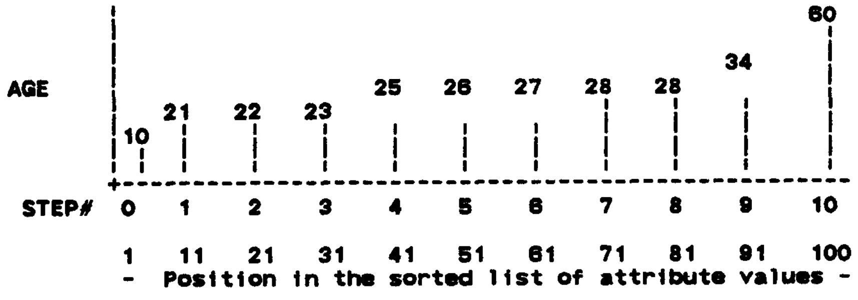

Figure 4-2: Scheme 2: Distribution Steps

To estimate the same selectivity - SEL(<29) - we first find where "29" falls relative to distribution steps. Since

[4] If that is not possible, we can have fewer values between just the last two positions All the following formulas would still hold

$$\mathrm{STEP}(8) = 28$$

we know that more than 80 employees are 28 or younger, so $\mathrm{SEL}(<29) > 0.80$. Since

$$\mathrm{STEP}(9) = 34$$

we know that 90 or fewer employees are younger than 34, so $\mathrm{SEL}(<29) \leq 0.90$. Therefore

$$0.80 < \mathrm{SEL}(<29) \leq 0\ 90$$

Again choosing the midpoint of the range (0.85) as our estimate of SEL(<29), the maximum possible error is 0 05, 4 times less than in scheme 1.

The error in estimating $\mathrm{SEL}(<X)$ depends on where X falls relative to the distribution steps. If X falls between the steps, (as X=29 above), then the maximum error is half of the fraction of values falling between the steps, i.e., $\leq 1/2S$.

If X is equal to one or more step values, then (as we will show later in this paper) the maximum error is about $1/S$. In both cases, the maximum error can be reduced to an arbitrarily small number by choosing a sufficiently large S.

The exact computation of distribution steps is more expensive than that of a histogram because it requires sorting of all attribute values. A cheaper way to compute distribution steps is to use an index on the field, if it exists. Finally, a very cheap method can be used when we do not need the completely guaranteed accuracy of this scheme. Then we can closely approximate the distribution steps, retaining high average accuracy, by using only a small sample of the tuples (see section 9).

In the remainder of this paper we will assume that information about attribute values is stored in the form of distribution steps.

5. CONDITIONS ON THE SELECTIVITY ESTIMATES

Distribution steps provide upper and lower bounds on selectivity estimates $\mathrm{SEL}{\sim}(=X)$, $\mathrm{SEL}{\sim}(<X)$, $\mathrm{SEL}{\sim}(>X)$, $\mathrm{SEL}{\sim}(\leq X)$, $\mathrm{SEL}{\sim}(\geq X)$. There are many ways to compute the estimates within these bounds. One approach is to compute the estimates for each of the five types of conditions independently. We choose not to do that. Instead we postulate the following desired relationships between the selectivity estimates (these conditions are satisfied by the actual selectivity):

Consistency. For every X,

(C1) $\mathrm{SEL}{\sim}(<X) + \mathrm{SEL}{\sim}(=X) + \mathrm{SEL}{\sim}(>X) = 1$

(C2) SEL~($\le X$) = SEL~($<X$) + SEL~($=X$)
(C3) SEL~($\ge X$) = SEL~($>X$) + SEL~($=X$)

Additional desirable conditions, also satisfied by the actual selectivity, are:

Correctness on Ends. Let *minval, maxval* be the minimum and maximum values for the attribute, respectively We require

(E1) SEL~(>*maxval*) = 0
(E2) SEL~(<*minval*) = 0

Monotonicity.
For every X, Y, such that $X < Y$
(M1) SEL~($<X$) $\le$ SEL~($<Y$)

For every X
(M2) SEL~($=X$) ≥ 0

We will consider *consistency*, *correctness on ends* and *monotonicity* as the axioms of selectivity estimates. We believe that any estimation scheme should satisfy them. When an estimation scheme satisfies these conditions we can infer many other properties about the estimates.

Some of the implications are:

- From (C2), (C1) and (E1) we can infer SEL~($\le$*maxval*) = 1
- From (C3), (C1) and (E2) we infer SEL~($\ge$*minval*) = 1
- From (M2) and (C2) follows SEL~($<X$) $\le$ SEL~($\le X$) .

There are many ways to compute selectivity estimates which meet these conditions. In the next section we will present one way which uses the distributions steps and minimizes the worst-case error.

6. SELECTIVITY ESTIMATES WITH THE SMALLEST WORST-CASE ERROR

For any X, we want to estimate

SEL($=X$) ,
SEL($<X$) , SEL($\le X$)
SEL($>X$) , SEL($\ge X$)

The distribution steps give us upper and lower bounds on selectivities. For example, if

STEP(I) $< X <$ STEP(I+1) then we know that in the sorted list of all the attribute values, the value in position $1+I*N$ is STEP(I) (where $N = (T-1)/S$). Hence there are at least $1+I*N$ values less than X, and

$$\mathrm{SEL}(<X) \geq (1+I*T/S)/T > I/S.$$

Likewise, the value in position $1+(I+1)*N$ is STEP(I+1) – hence there are at most $(I+1)*N$ attribute values less than X, and SEL($<X$) $\leq (I+1)/S$. So

$$I/S < \mathrm{SEL}(<X) \leq (I+1)/S$$

Also, there are $T/S-1$ "intermediate" values between these steps They can all be equal to X (then SEL($=X$) $= 1/S - 1/T$), or none could be equal to X (then SEL($=X$) $= 0$) So

$$0 \leq \mathrm{SEL}(=X) < 1/S$$

Within these bounds we want to find estimates SEL~ that are *consistent*, *correct on ends*, and *monotonic*.

Let MaxErr(SEL~($rel\ X$)) be the maximum possible estimation error for SEL~($rel\ X$).

Then, for any X, we want to minimize

$$\begin{aligned}\max(\ & \mathrm{MaxErr}(\mathrm{SEL}\sim(=X)),\\ & \mathrm{MaxErr}(\mathrm{SEL}\sim(<X)),\\ & \mathrm{MaxErr}(\mathrm{SEL}\sim(\leq X)),\\ & \mathrm{MaxErr}(\mathrm{SEL}\sim(>X)),\\ & \mathrm{MaxErr}(\mathrm{SEL}\sim(\geq X))\ \ )\end{aligned}$$

Since

$$\mathrm{SEL}\sim(\geq X) = 1 - \mathrm{SEL}\sim(<X)$$

$$\mathrm{SEL}\sim(>X) = 1 - \mathrm{SEL}\sim(\leq X)$$

we have MaxErr(SEL~($\geq X$)) = MaxErr(SEL~($<X$)), and MaxErr(SEL~($>X$)) = MaxErr(SEL~($\leq X$)) Therefore it is enough to minimize MaxErr for SEL~($=X$), SEL~($\leq X$), SEL~($<X$). Furthermore, since

$$\mathrm{SEL}\sim(\leq X) = \mathrm{SEL}\sim(<X) + \mathrm{SEL}\sim(=X)$$

we only need to find formulas for SEL~($<X$) and SEL~($=X$) . These formulas depend on where X falls relative to the distribution steps. In the following subsections we analyze the different cases.

6.1. Case A: X is between the steps

```
      !              STEP(I+1)
attr  !            X   #
value !  STEP(I)   !   #
      !     #      !   #
      !     #      !   #
      +-------------------------
      step#  I       I+1
```

Here

STEP(I) < X < STEP(I+1)

We know that $I/S < \mathrm{SEL}\sim(<X) \le (I+1)/S$

The estimate with the lowest maximum error is $(I+0.5)/S$ However, consider $\mathrm{SEL}\sim(\le X)$. For it the best estimate is also $(I+0.5)/S$ Yet, from consistency requirement,

$$\mathrm{SEL}\sim(\le X) = \mathrm{SEL}\sim(<X) + \mathrm{SEL}\sim(=X)$$

Thus we can write that

$$\mathrm{SEL}\sim(<X) = (I+0.5)/S - a*\mathrm{SEL}\sim(=X)$$
$$\mathrm{SEL}\sim(\le X) = (I+0.5)/S + (1-a)*\mathrm{SEL}\sim(=X)$$

where $0 \le a \le 1$. The maximum error for $\mathrm{SEL}\sim(<X)$ is then

$$0.5/S + a*\mathrm{SEL}\sim(=X)$$

and for $\mathrm{SEL}\sim(\le X)$ is

$$0.5/S + (1-a)*\mathrm{SEL}\sim(=X)$$

To minimize the maximum of those errors, a should minimize $\max(a, 1-a)$. Hence $a = 0.5$ and

$$\mathrm{SEL}\sim(<X) = (I+0.5)/S - 0.5*\mathrm{SEL}\sim(=X)$$
$$\mathrm{SEL}\sim(\le X) = (I+0.5)/S + 0.5*\mathrm{SEL}\sim(=X)$$

with the maximum error $0.5/S + 0.5*\mathrm{SEL}\sim(=X)$ in each case. Let $\beta = \mathrm{SEL}\sim(=X)$. Since $0 \le \beta < 1/S$, the error in estimating β is $\max(\beta, 1/S-\beta)$. The maximum error overall is

$$\max(0.5/S + \beta/2, \beta, 1/S - \beta)$$

This is minimized when $0.5/S + \beta/2 = 1/S - \beta$, or when $\beta = 1/3S$. Thus

$$\mathrm{SEL}\sim(<X) = \frac{I + 1/3}{S} \qquad \mathrm{SEL}\sim(=X) = \frac{1/3}{S}$$

with the maximum estimation error = $2/3S$.

6.2. Case B1: X is equal to one of the steps, but not to the first or last step

```
      !                    STEP(I+1)
attr  !          X=STEP(I)   #
value !  STEP(I-1)    #      #
      !     #         #      #
      !     #         #      #
      +-------------------------
      step#  I-1      I      I+1
```

Here $\mathrm{STEP}(I-1) < X = \mathrm{STEP}(I) < \mathrm{STEP}(I+1)$. Then,

$$(I-1)/S < \mathrm{SEL}{\sim}(<X) \le I/S$$

$$I/S < \mathrm{SEL}{\sim}(\le X) \le (I+1)/S$$

As in case A, we can say

$$\mathrm{SEL}{\sim}(<X) = I/S - a*\mathrm{SEL}{\sim}(=X)$$
$$\mathrm{SEL}{\sim}(\le X) = I/S + (1-a)*\mathrm{SEL}{\sim}(=X)$$

with the maximum estimation error for each formula being

$$1/S - \min(a,\ 1-a) * \mathrm{SEL}{\sim}(=X)$$

which is less than $1/S$. To minimize the worst-case error, we should set $a = 0.5$. Since $0 < \mathrm{SEL}{\sim}(=X) < 2/S$, the minimum-error estimate for $\mathrm{SEL}{\sim}(=X) = 1/S$ The final equations are:

$$\mathrm{SEL}{\sim}(<X) = \frac{I - 0.5}{S} \qquad \mathrm{SEL}{\sim}(=X) = \frac{1}{S}$$

with the maximum estimation error = $1/S$.

6.3. Case B2: X is equal to several steps, but not to the first or last step

```
        |                                       STEP(I+1)
attr    |          X=STEP(I)            X       #
value   | STEP(I-1)     #               #       #
        |      #        #               #       #
        |      #        #               #       #
        +--------------------------------------------
     step#     I-1      I               I+K-1   I+K
```

Here X is equal to values of K steps $(K>1)$:

$$\mathrm{STEP}(I-1) < X = \mathrm{STEP}(I) = \ldots = \mathrm{STEP}(I+K-1) < \mathrm{STEP}(I+K)$$

By reasoning analogous to Case B1 we get

$$\mathrm{SEL}{\sim}(<X) = \frac{I - 0.5}{S} \qquad \mathrm{SEL}{\sim}(=X) = \frac{K}{S}$$

with the maximum estimation error = $1/S$.

6.4. Case C: X is equal to one or more steps, including either the first or last step

We note that if X is equal to both first and last step, i.e., X is equal to minimum and maximum attribute values, then all values are equal to X, in which case estimation is trivial.

If X is equal to K steps $(K \ge 1)$, including the *first* one:

```
      |                                  STEP(K)
attr  |    X=STEP(0)       X=STEP(K-1)   #
value |        #                #        #
      |        #                #        #
      |        #                #        #
      +---------------------------------------
    step#      0               K-1       K

      X = STEP(0) =       = STEP(K-1) < STEP(K)
```

then

$$\mathrm{SEL}\sim(<X) = 0 \qquad\qquad \mathrm{SEL}\sim(=X) = \frac{K - 0.5}{S}$$

If X is equal to K steps ($K \geq 1$), including the *last* one.

```
      |
attr  |           X=STEP(S-K+1)      X=STEP(S)
value |  STEP(S-K)     #                 #
      |        #       #                 #
      |        #       #                 #
      +-----------------------------------
    step#     S-K     S-K+1              S

      STEP(S-K) < X = STEP(S-K+1) =      = STEP(S)
```

then

$$\mathrm{SEL}\sim(<X) = 1 - \frac{K - 0.5}{S} \qquad\qquad \mathrm{SEL}\sim(=X) = \frac{K - 0.5}{S}$$

6.5. Case D: X is outside the range of attribute values

If X < STEP(0), then $\mathrm{SEL}\sim(<X) = 0$
$\mathrm{SEL}\sim(=X) = 0$

If X > STEP(S), then $\mathrm{SEL}\sim(<X) = 1$
$\mathrm{SEL}\sim(=X) = 0$

7. SELECTIVITY ESTIMATES WITH LOWER AVERAGE CASE ERROR

In the previous section we presented a set of formulas which satisfied all the axioms and for each X minimized the *worst-case* estimation error for SEL~(*rel* X). The major disadvantage of those formulas is that for X between the steps, SEL~(=X) = 1/3S is usually much higher than the actual SEL(=X).

In this section we present another set of formulas (also satisfying all the axioms) with a significantly smaller *average* estimation error for SEL~(=X) for X between the steps. The penalty incurred is the somewhat greater worst-case estimation error. In many applications, however, minimizing the worst-case error is less important than minimizing the average-case error. We find this to be especially true when using these selectivity estimates to choose an evaluation method for a query.

The formulas in this section are based on distribution steps and on an additional parameter called the attribute *density* These formulas, with some additions, are used in the query optimizer of FASTSCAN query system. We do not make any formal claim about the average estimation error of these formulas. In practice, however, they have proved to be very accurate - even more accurate than the formulas from the previous section.

Let N_i be the number of occurrences of value $value_i$ for a particular attribute throughout a relation.

Definition 1: The attribute *density* is defined as

$\Sigma\ (N_i^2)\ /\ T^2$
for all $value_i$ except those which
are equal to at least two step values.

Less formally, the attribute density is the average fraction of values that are the same. If there is only one value for the attribute in all the tuples, the attribute density is 1. If each tuple has a different value in the attribute, the density is $1/T$. If one value occurs in half the tuples, and two other values in one quarter each, the density will be between 1/4 and 1/2.

If all values occur equally often, then the density is equal to SEL(=X) for any X. We can also interpret the density as the average value of SEL(=X) over a set of queries $A = X$, where the query $A = value_i$ occurs N_i times. From these observations we see that density can be used as an approximation to SEL(=X). We conjecture that density is a better approximation than 1/*number_of_different_values* used by System R, because computation of density takes into account unequal distribution of attribute values.

We have excluded from computation of density all $value_i$ which stretched across more than one step, i.e., for which $N_i > T/S$. Therefore $density < 1/S$.

Let

$$\delta = \min(0.5/S,\ density)$$

With some exceptions, δ will be the difference between SEL~($\leq X$) and SEL~($< X$). δ is selected so that the following condition is partially satisfied:

if X < STEP(I) $\leq$ Y , then

SEL~($\leq X$) < SEL~($<Y$)

Below we present formulas for SEL~($=X$), SEL~($<X$) depending on where X falls relative to distribution steps. We do not show the cases where X is equal to more than one step, since for those cases we use the same formulas as in the previous section.

We have derived these formulas by fixing the value of SEL~($=X$) to be δ and then trying to minimize the errors for SEL~($<X$), SEL~($\leq X$) while obeying the axioms from section 5.

Case A: X is between the steps

Here

$$\text{STEP}(I) < X < \text{STEP}(I+1)$$

The estimates are:

$$\text{SEL\~}(<X) = \frac{I + 0.5}{S} - \delta/2 \qquad \text{SEL\~}(=X) = \delta$$

Case B1: X is equal to one of the steps, but not to the first or last step

Here

$$\text{STEP}(I-1) < X = \text{STEP}(I) < \text{STEP}(I+1)$$

The estimates are:

$$\text{SEL\~}(<X) = \frac{I}{S} - \delta/2 \qquad \text{SEL\~}(=X) = \delta$$

Case C1: X is equal to either first or last step

If

$$X = \text{STEP}(0) < \text{STEP}(1)$$

then

$$\text{SEL\~}(<X) = 0 \qquad \text{SEL\~}(=X) = \delta/2$$

If

$$X = \text{STEP}(S) > \text{STEP}(S-1)$$

then

$$\text{SEL\~}(<X) = 1 - \delta/2 \qquad \text{SEL\~}(=X) = \delta/2$$

8. A COMPARISON OF SELECTIVITY ESTIMATES ON A REAL WORLD EXAMPLE

In this section we compare the selectivity estimates using the distribution steps of attribute VOL - the volume of stock trading on the New York Stock Exchange. The data comes from a single day of trading. There are 15049 tuples in the relation, each tuple describing a single stock.

The attribute density is 0.008 . There are 144 distinct attribute values. δ = 0.008 . The next figure shows a 20-step attribute distribution, where *nn* % = *VVVV* means that step # *nn/5* has value *VVVV*, i.e., that nn% of the tuples have value ≤ *VVVV* in the attribute.

STEP	value	STEP	value	STEP	value
0 %=	0	35 %=	0	70 %=	800
5 %=	0	40 %=	0	75 %=	1500
10 %=	0	45 %=	0	80 %=	2800
15 %=	0	50 %=	0	85 %=	5200
20 %=	0	55 %=	0	90 %=	10900
25 %=	0	60 %=	100	95 %=	28400
30 %=	0	65 %=	400	100 %=	975800

Figure 8-1: 20-step distribution of attribute VOL

In the following figure **FASTSCAN estimate** is derived using formulas from section 7, and **Sm. w.-c. error estimate** is the smallest worst-case error estimate, as derived in section 6.

			Sm. w.-c.	
	Actual	FASTSCAN	error	System R
condition	selectivity	estimate	estimate	estimate
VOL < 1500	0.745	0.746	0.725	0.001
VOL = 1500	0.006	0.008	0.050	0.007
VOL > 1500	0.249	0.246	0.225	0.999
VOL < 5000	0.844	0.821	0.8125	0.005
VOL = 5000	0.003	0.008	0.025	0.007
VOL > 5000	0.153	0.171	0.1625	0.995
VOL < 0	0	0	0	0
VOL = 0	0.576	0.575	0.575	0.007
VOL > 0	0.424	0.425	0.425	1.0

Figure 8-2: Comparison of selectivity estimates

9. USING SAMPLING TO APPROXIMATE DISTRIBUTION OF ATTRIBUTE VALUES

Computing the distribution steps even for *one* attribute of a large relation is expensive, since it requires reading and sorting all the values for that attribute. Computing the distribution steps for all attributes in all relations (a typical example may be 50 relations with 100,000 tuples each with 100 attributes each) would take many days. This might be acceptable, if the database is read-only or if it changes very slowly over time However there are cases when the database changes rapidly, and there is no time to recompute all statistics in full.

A solution to this dilemma is to compute distribution steps using a *sample* of the tuples, rather than all of them We lose guaranteed precision but we are still able to maintain very good accuracy. We rely on the theory of nonparametric statistics (see, e.g., [Dixon 79], chapter 17) for analysis of estimation errors

Kolmogorov's statistic tells us that if we take a sample of size 1064 tuples and β is the fraction of the tuples in the sample with *attr* $< V$, then with confidence 99% the fraction of the tuples in the *entire* relation with *attr* $< V$ is in the interval $[\beta-0.05, \beta+0.05]$ If we take a sample of size 740 then the confidence would decrease to 95% with the same interval. The full relationship between the sample size, the confidence level and the interval is given by the table of Kolmogorov's statistic ([Dixon 79], p. 550) Interestingly, the sample size does *not* depend on the number of tuples in the entire relation.

We use a sample of 1064 tuples.[5] We compute distribution steps using just the tuples in the sample and use the same estimation formulas as we derived before.

Let $\mathrm{STEP}_s(I)$ denote the sample distribution step #*I*. Consider X such that $\mathrm{STEP}_s(I) < X < \mathrm{STEP}_s(I+1)$. Our formulas estimate SEL($<X$) assuming

$$I/S < \mathrm{SEL}(<X) < (I+1)/S$$

We are no longer sure that SEL will be within these bounds. We can, however, say that with confidence 99%

$$I/S - 0.05 < \mathrm{SEL}(<X) < (I+1)/S + 0.05$$

Similar analysis applies to other cases Thus, with 99% confidence, we can say that the error in selectivity estimates computed from sample distribution steps will be no more than 0.05 over the error in selectivity estimates computed from the distribution steps on the entire relation.

[5] If there are fewer tuples in the entire file, we use the entire file to compute distribution steps and avoid the loss of precision associated with sampling

In a forthcoming paper we plan to give a more detailed analysis of sampling and especially of how sampling affects estimation of attribute density.

In the following figure we show 20 distribution steps for attribute SALES from the Industrial Compustat relation. The column labelled **all** refers to steps obtained from reading all 48320 tuples in the relation, of which 38576 were used in computing distribution steps (the rest had an undefined value of SALES). The column labelled **sample** refers to steps obtained from a sample of 1064 tuples.

STEP	sample	all	STEP	sample	all
0%=	0	0	55%=	149.8	149.5
5%=	6.3	6.9	60%=	184.1	188.0
10%=	11.9	13.0	65%=	248.4	242.8
15%=	19.1	19.7	70%=	329.0	314.9
20%=	29.4	27.6	75%=	457.6	418.8
25%=	36.4	36.6	80%=	603.3	591.1
30%=	48.5	47.7	85%=	842.6	873.7
35%=	60.3	60.5	90%=	1561.8	1404.2
40%=	77.2	75.8	95%=	2613.3	2717.4
45%=	97.4	94.8	100%=	97172.9	108108.0
50%=	125.0	118.9			

Figure 9-1: Distribution steps - from full relation and from a sample

The following figure shows a comparison of selectivity estimates. Column **"Sm. w.-c. est. on all"** contains smallest worst-case error estimates from sec 6, computed using distribution steps found by reading all tuples. **"Sm. w.-c. est. on sample"** contains smallest worst-case error estimates, computed using distribution steps on a sample **"System R estimate"** is computed using the minimum and maximum attribute values.

condition	Actual sel.	Sm. w-c. est on all	Sm. w-c. est on sample	System R estimate
SALES<20	0.151	0.167	0.167	0.00018
SALES<200	0.612	0.617	0.617	0.0018
SALES<1500	0.906	0.867	0.917	0.014
SALES<2000	0.929	0.917	0.917	0.018
SALES<20000	0.997	0.967	0.967	0.18

Comparison of selectivity estimates

10. CONCLUSION

We have presented a new method for estimating the selectivity of conditions of the type *attribute rel constant*, where *rel* is one of "=", "<", ">", "≤", "≥". We argued that information about attribute values distribution should be stored as a list of *distribution steps*.

Then we presented axioms of *consistency*, *correctness on ends*, and *monotonicity* on selectivity estimates and analyzed their implications.

We showed two sets of selectivity estimation formulas which use distribution steps and satisfy the axioms. The first set is optimal in a sense that it has the smallest possible worst-case error. The second set has a slightly larger worst-case error, but a much smaller average case error for the selectivity of *attribute = const.*

We examined approximation of distribution steps using sampling. We analyzed the relationship between the size of the sample and the expected estimation error and showed, both theoretically and on several examples, that high accuracy is achieved using small samples.

The excellent precision and low overhead of our method of estimating selectivity makes it very useful for query optimization and other applications.

ACKNOWLEDGEMENTS

We are grateful to Nathan Goodman for his valuable suggestions which improved the clarity of presentation. We thank Boris Khazanov for suggesting the idea of sampling and Leonid Levin for helpful discussion of Kolmogorov's statistic. Finally, we thank Nancy Buchan for patiently handling X9700 tapes for innumerable versions of this paper.

REFERENCES

[Blasgen 77] Blasgen, M W., and Eswaran, K. P.
Storage and access in relational databases
IBM Syst. J 16(4), 1977.

[Christodoulakis 81]
Christodoulakis, S
Estimating selectivities in data bases.
Technical Report CSRG-136, Ph.D. thesis, University of Toronto, 1981.

[Christodoulakis 83]
Christodoulakis, S
Estimating block transfers and join sizes.
In *Proc of annual meeting*, pages 40-54. ACM-SIGMOD, May, 1983.

[Demolombe 80]
Demolombe, R.
Estimation of the number of tuples satisying a query expressed in predicate calculus language
In *Proc of 6th Int. Conf. on Very Large Data Bases*, pages 55-63. Montreal, Canada, 1980.

[Dixon 79] Dixon, W. J., and Massey, F. J.
Introduction to statistical analysis
McGraw-Hill Book Company, 1979.

[Luk 83] Luk, W. S.
On estimating block accesses in database organizations.
Commun. ACM 26(11):945-947, November, 1983.

[Piatetsky 84] Piatetsky-Shapiro, G. I.
A Self-Organizing Database System - a Different Approach to Query Optimization.
PhD thesis, Courant Institute, New York University, May, 1984.

[Rowe 83] Rowe, N C.
Top-down statistical estimation on a database.
In *Proc. of annual meeting*, pages 135-145. ACM-SIGMOD, May, 1983.

[Samson 83] Samson, W. B. and Bendell, A.
Rank order distributions and secondary key indexing, extended abstract.
In *Proc of 2nd Int Conf. on Databases.* Cambridge, England, September, 1983.

[Selinger 79] Selinger, P. G., et al.
Access path selection in a relational database management system.
In *Proc of ACM-SIGMOD Int Conf. on Manage. of Data.* 1979.

[Whang 83] Whang, K.-Y , Wiederhold, G. and Sagalowicz, D.
Estimating block accesses in database organizations: a closed noniterative formula.
Commun. ACM 26(11):940-944, November, 1983.

[Yao 77] Yao, S. B.
Approximating block accesses in database organizations
Commun ACM 20(4):260-261, April, 1977

[Yao 79] Yao, S. B.
Optimization of query evaluation algorithms.
ACM Trans. Database Syst. 4(2):133-155, September, 1979.

[Yu 78] Yu, C. T., Luk, W. S., and Siu, M. K.
On the estimation of the number of desired records with respect to a given query.
ACM Trans Database Syst 3(1) 41-56, March, 1978

[Zipf 49] Zipf, G. K.
Human Behaviour and the Principle of Least Effort.
Addison-Wesley, Cambridge, Mass., 1949.