THE CASE FOR THE RECTANGULAR ATTRIBUTE CARDINALITY MAP IN QUERY OPTIMIZATION: MODELING, PROTOTYPE VALIDATION AND TESTING

Murali Thiyagarajah and B. John Oommen

TR-99-05 February 1999

School of Computer Science, Carleton University
Ottawa, Canada, KIS 5B6

The Case for the Rectangular Attribute Cardinality Map in Query Optimization: Modeling, Prototype Validation and Testing*

Murali Thiyagarajah† and **B. John Oommen**‡
School of Computer Science
Carleton University
Ottawa, Canada K1S 5B6
{murali, oommen}@scs.carleton.ca

Abstract

Current business database systems utilize histograms to approximate frequency distributions of attribute values of relations. These are used to efficiently estimate query result sizes and access plan costs and thus minimize the query response time for business (and non-commercial) database systems. In a recent work [12] we proposed a new form of histogram-like technique called the Rectangular Attribute Cardinality Map (R-ACM) that gives much smaller estimation errors than the traditional equi-width and equi-depth histograms currently being used by many commercial database systems. We also provided a fairly extensive mathematical analysis for its average and worst case errors for its frequency estimates which was verified for synthetic data.

This paper demonstrates the earlier claim that the R-ACM is indeed a viable tool for query optimization. It, first of all, presents the R-ACM model and reports a prototype validation for the R-ACM for query optimization in real-world database systems. By investigating the performance of the scheme on an extensive set of experiments using real-life data [1, 2], we demonstrate that the R-ACM scheme is much more accurate than the traditional histograms for query result size estimation. We anticipate that, due to its high accuracy and low construction costs, it could become an invaluable tool for query optimization in the future database systems.

1 Introduction

Modern-day database management systems (DBMS) provide competitive advantage to businesses by allowing quick determination of answers to business questions. Intensifying com-

*A preliminary version of some of the results of this paper can be found in the Proceedings of the 1999 International Conference on Business Information System, Poznan, Poland, April, 1999 [16].

†Supported by the Natural Sciences and Engineering Research Council of Canada.

‡Senior Member, IEEE. Partially supported by the Natural Sciences and Engineering Research Council of Canada.

Vendor	Product	Histogram Type
IBM	DB2-6000 (Client-Server)	Compressed(V,F) Type
IBM	DB2-MVS	Equidepth, Subclass of End-Biased(F,F)
Oracle	Oracle7	Equidepth
Sybase	System 11	Equidepth
Tandem	NonStop SQL/MP	Equidepth
NCR	Teradata	Equidepth
Informix	Online Data Server	Equidepth

Table 1: Histograms used in commercial DBMSs.

petition in the business marketplace continues to increase the sizes of databases as well as the level of sophistication of queries against them. This has resulted in a greater focus (both academically and in the industrial world) to develop systems with superior DBMS functionalities that would, in turn, minimize the response times for business and other queries.

The problem of minimizing query response time is known as Query Optimization, which has been one of the most active research topics in the database and information system fields for the last two decades. Query optimization for relational database systems is a combinatorial optimization problem, which requires the estimation of query result sizes to select the most efficient access plan for a query based on the estimated costs of various query plans. As queries become more complex (such as those found in modern-day business systems), the number of alternative query evaluation plans (QEPs) which have to be considered explodes exponentially. For example, for a join of 10 relations, the number of different QEPs is greater than 176 billion!. A typical inquiry that a bank officer runs many times a day, say, to retrieve the current market value of a customer's mutual fund/stock portfolio, usually carries out a join of many relations behind the scene.

Query result sizes are usually estimated using a variety of statistics that are maintained in the database catalogue for relations in the database. Since these statistics approximate the distribution of data values in the attributes of the relations, they represent an inaccurate picture of the **actual** contents of the database. It has been shown in [6] that errors in query result size estimates may increase exponentially with the number of joins. This result, in light of the complexity of present-day queries, shows the critical importance of accurate result size estimation.

Several techniques have been proposed in the literature to estimate query result sizes, including histograms, sampling, and parametric techniques [5, 9, 11, 14] and it is impossible to survey them in this paper. Of these, histograms are the most commonly used form of statistics, which are also used in commercial database systems such as Microsoft SQL Server, Sybase, Ingres, Oracle and DB2.A more comprehensive list is shown in Table 1.

We recently proposed a new catalogue based non-parametric statistical model called the *Rectangular Attribute Cardinality Map* (R-ACM) that can be used to obtain more accurate estimation results than the currently known estimation techniques. A detailed description of this technique can be found in [12]. The goal of this paper is to demonstrate the power of the R-ACM for query result size estimation by extensive experiments on real-world data and to confirm our analytical results that the R-ACM is superior to the current state-of-the-art techniques used in the commercial database systems.

By using two popular real-world databases, namely the U.S. CENSUS and the NBA Player Statistics, as the test-bed for our experiments, we conduct an extensive prototype validation and testing of the R-ACM and compare its estimation accuracy with that of the traditional equi-width and equi-depth histograms. We also augment the data distributions from the above mentioned databases with frequency values generated from two well-known mathematical distributions, namely, the Zipf and the multifractal distributions to obtain a wide range of data skews for the experiments.

2 Previous Work

In the interest of brevity, it is impossible to give a good review of the field here. Such a review is found in [15]. However, to present our results in the right perspective a brief survey is given.

Equi-width histograms for single-attributes were considered by Christodoulakis [3] and Kooi [9]. Since these histograms traditionally have the same width, they produce highly erroneous estimates if the attribute values are not uniformly distributed. The problem of building equi-depth histograms on a single attribute was first proposed by Piatetsky-Shapiro and Connell [14]. This was later extended as multi-dimensional equi-depth histograms to represent multiple attribute values by Muralikrishna and Dewitt [11].

Ioannidis and Christodoulakis took a different approach by grouping attribute values based on their frequencies [6, 7]. In these serial histograms, the frequencies of attribute values associated with each bucket are either all greater or all less than the frequencies of the attribute values associated with any other bucket. They also considered optimal serial histograms that minimize worst case error propagation in the size of join results [6, 7]. The serial histograms provide optimal results for equality join estimations, but less than optimal results for range queries. Faloutsos *et al* [5] proposed using a multi-fractal assumption for real-data distribution as opposed to the uniformity assumptions made within current histograms.

Ioannidis and Poosala [8] discussed the design issues of various classes of histograms and of strategies for balancing their practicality and optimality in query optimization. They investigated various classes of histograms using different constraints (V-Optimal, MaxDiff, Compressed, and Spline-based) and sort and source parameters (Frequency, Spread, and Area). They also provided various sampling techniques for constructing the above histograms and concluded that the V-optimal histogram is the most optimal one for estimating the result sizes of equality-joins and selection predicates.

3 Attribute Cardinality Maps

The Attribute Cardinality Maps (ACM) are histogram-like techniques for query optimization [12]. Since they are based on the philosophies of numerical integration, query result size estimations based on these models have been analytically shown to be much more accurate than the traditional equi-width and equi-depth histograms. There are two types of ACMs, namely, the Rectangular ACM (R-ACM) and the Trapezoidal ACM (T-ACM). Since this paper is about modeling, prototype validation and testing of the R-ACM, we briefly describe it below. Observe that since this paper is not intended to be of a theoretical flavor, we shall merely allude to the theoretical properties of the R-ACM. For a more detailed mathematical treatment of both the R-ACM and T-ACM structures, the reader is referred to [12, 13].

3.1 Rectangular Attribute Cardinality Map

The Rectangular Attribute Cardinality Map (R-ACM) of a given attribute, in its simplest form, is a one-dimensional integer array that stores the count of the tuples of a relation corresponding to that attribute, and for some subdivisions for the range of values assumed by that attribute. The R-ACM is, in fact, a modified form of the histogram. But unlike the two major forms of histograms, namely, the equi-width histogram, where all the sector widths are equal, and the equi-depth histogram, where the number of tuples in each histogram bucket is equal, the R-ACM has a variable sector width, and has varying number of tuples in each sector. The sector widths or subdivisions of the R-ACM are generated according to a rule that aims at minimizing the estimation error within each subdivision.

Definition 1 A One dimensional Rectangular ACM: *Let $\mathcal{V} = \{v_i : 1 \leq i \leq |\mathcal{V}|\}$, where $v_i < v_j$ when $i < j$, be the set of values of an attribute X in relation R. Let the value set $\mathcal{V}$ be subdivided into s number of sector widths according to the range partitioning rule described below. Then the Rectangular Attribute Cardinality Map of attribute X is an integer array in which the j^{th} index maps the number of tuples in the j^{th} value range of the set $\mathcal{V}$ for all j, $1 < j \leq s$.*

Rule 1 Range Partitioning Rule: *Given a desired tolerance value τ for the R-ACM, the sector widths, $l_j, 1 \leq j \leq s$, of the R-ACM should be chosen such that for any attribute value X_i, its frequency x_i does not differ from the* **running mean of the frequency** *of the sector by more than the tolerance value τ, where the running mean is the mean of the frequency values examined so far in the current sector.*

For example, consider the frequency set $\{8, 6, 9, 7, 19, 21, 40\}$ corresponding to the attribute values $\{X_0, X_1, X_2, X_3, X_4, X_5, X_6\}$ of an attribute X. Using a tolerance value $\tau = 2$, the attribute value range will be partitioned into the three sectors, $\{8, 6, 9, 7\}, \{19, 21\}, \{40\}$ with sector widths of 4, 2, and 1 respectively..

Since the ACM only stores the count of the tuples and not the actual data, it does not incur the usually high I/O cost of having to access the base relations from secondary storages. Secondly, unlike the histogram-based or other parametric and probabilistic counting estimation methods in use currently [10], ACM does not use sampling techniques to approximate the data distribution. Each cell of the ACM maintains the *actual* number of tuples that fall between the boundary values of that cell, and thus, although this leads to an approximation of the density function, there is no approximation of the number of tuples in the data distribution.

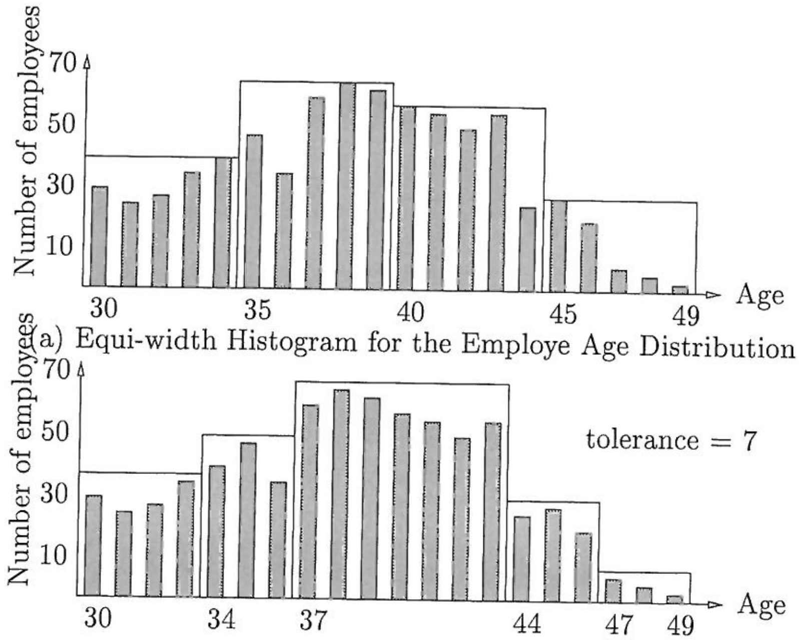
The one-dimensional R-ACM as defined above can be easily extended to a multi-dimensional one to map an entire multi-attribute relation. The multi-dimensional ACM, which can also be used to store the multi-dimensional attributes that commonly occur in geographical, image, and design databases, is currently being investigated.

The algorithm for generating the R-ACM is given below. Although this is found in [12], we have included it here to ensure that our results can be verified and duplicated by interested researchers.

Algorithm 1 Generate_R-ACM

```
  Input:  tolerance $\tau$, frequency distrib. of $X$ as $A[0 \dots L-1]$
  Output:  R-ACM
  begin
     Initialize_ACM;      /* set all entries in ACM to zero */
     current_mean := $A[1]$; $j := 0$;
     $ACM[j] := A[1]$;
     for i:=1 to $L-1$ do      /* for every attribute value */
        if abs($A[i]-$ current_mean) $< \tau$
           $ACM[j] := ACM[j] + A[i]$;
           current_mean := (current_mean*$i + A[i]$)/($i$+1); /* running mean */
        else begin
           $l_j := i-1$;      /* set the sector width */
           $j++$;      /* move to next sector */
           current_mean := $A[i]$;
           $ACM[j] := A[i]$;
        end;
  end;
end Algorithm.
```

The input to the algorithm are the tolerance value τ for the ACM and the actual frequency distribution of the attribute X. The frequency distribution is assumed to be available in an integer array A, which has a total of L entries for each of the L distinct values of X. For simplicity reasons, we assume that the attribute values are ordered integers from 0 to $L-1$. The output of the algorithm is the R-ACM for the given attribute value set.



(a) Equi-width Histogram for the Employe Age Distribution

(b) Corresponding Rectangular ACM

Figure 1: An Example Rectangular Attribute Cardinality Map

3.2 Properties of the R-ACM

In order to provide the right perspective for our experimental results, we give below a brief catalogue of the major properties of the R-ACM.

(1) The R-ACM is neither an equi-width nor an equi-depth histogram. This is due to the partitioning strategy based on the tolerance.

(2) The frequency distribution of any attribute value within an R-ACM sector obeys a Binomial distribution with mean $\mu = \frac{n}{l}$ and variance $V = \frac{n(l-1)}{l^2}$, where n is the number of tuples within the sector and l is the sector width.

(3) For a one-dimensional R-ACM, the maximum likelihood estimate of the number of tuples for a given value X_i of attribute X is given by,

$$\hat{x}_{ML} = \frac{n}{l}$$

where n is the number of tuples in the sector containing the value X_i and l is the width of that sector.

(4) For a one-dimensional R-ACM, the maximum likelihood estimate of the number of tuples for a given value X_i of attribute X falls within the range of,

$$\frac{(n+1)}{l} - 1 \leq \hat{x}_{ML} \leq \frac{(n+1)}{l},$$

where n is the number of tuples in the sector containing the value X_i and l is the width of that sector.

(5) The error in self-join estimation from the R-ACM is given by,

$$\epsilon = Var(ACM) + \sum_{j=1}^{s} \left\{ \sum_{k=1}^{l_j} x_k^2 - \frac{n_j^2 + n_j l_j - n_j}{l_j} \right\}$$

which is $O(Var(ACM))$, where self-join is the operation of joining the relation with itself and the $Var(ACM)$ is the variance of the entire R-ACM.

(6) The variance of the entire R-ACM is given by,

$$Var(ACM) = N - \sum_{j=1}^{s} \frac{n_j}{l_j},$$

where N is the total number of tuples mapped by the R-ACM, n_j is the number of tuples in the j^{th} sector, l_j is the j^{th} sector width, and s is the number of sectors in the R-ACM.

(7) If the attribute value X_i falls in the j^{th} sector of an R-ACM, then the number of occurrences of X_i is,

$$\frac{n_j}{l_j} - \left| \tau \left[\ln \left(\frac{l}{i-1} \right) - 1 \right] \right| \leq x_i \leq \frac{n_j}{l_j} + \left| \tau \left[\ln \left(\frac{l}{i-1} \right) - 1 \right] \right|$$

where n_j and l_j are the number of tuples and the sector width of the j^{th} sector and i is the location of the attribute value within the sector.

For example, consider an R-ACM sector of width 10 containing 124 tuples. Using a tolerance value $\tau = 3$, we see that the attribute value X_3 falls in the following range:

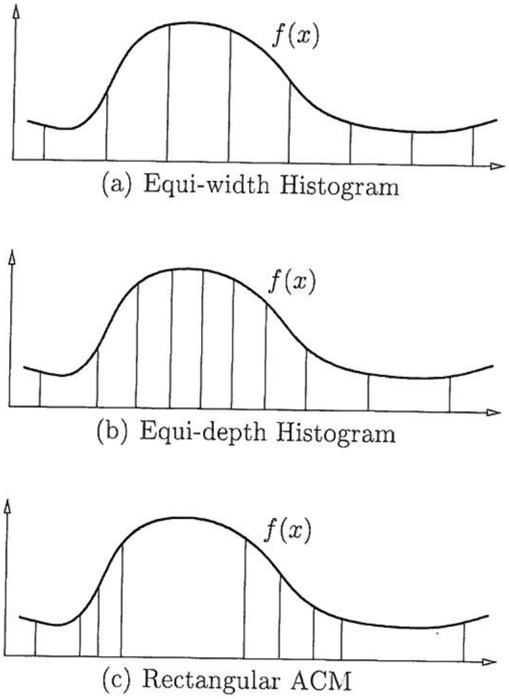
$$10.57 \leq x_3 \leq 14.23.$$

The power of the R-ACM is obvious when we observe that the corresponding range for the equi-width and the equi-depth histograms are,

$$0 \leq x_3 \leq 124.$$

(8) The average-case error in estimating the frequency of an attribute value is always bounded by 2τ.

The proofs of the above assertions are found in [12]. Also found in [12] are various expressions for the join estimation error, average-case and worst-case errors for both equality-select and range-select operations using the R-ACM. However, for the interest of the practitioner, we provide below a brief rationale for why the R-ACM is advantageous over the traditional histogram techniques.



(a) Equi-width Histogram

(b) Equi-depth Histogram

(c) Rectangular ACM

Figure 2: R-ACM and Traditional Histograms. Note in (b), the areas of the sectors are equal.

4 Rationale for the Rectangular ACM

Without loss of generality, let us consider an arbitrary continuous frequency function $f(x)$. Figure 2 shows the histogram partitioning of $f(x)$ under the traditional equi-width, equi-depth methods and the R-ACM method.

We note that in the equi-width case, regardless of how steep the frequency changes are in a given sector, the sector widths remain the same across the attribute value range. This means even widely different frequency values of all the different attribute values are assumed to be equal to that of the average sector frequency. Thus there is an obvious loss of accuracy with this method. On the other hand, in the equi-depth case, the area of each histogram sector is the same. This method still results in sectors with widely different frequency values and thus suffers from the same problem as the equi-width case. In the R-ACM method, we note that whenever there is a steep frequency changes, the corresponding sector widths proportionally decrease (or in other words, the number of sectors proportionally increases). Hence the actual frequencies of all the attribute values within a sector are assured to be closer to the average frequency of that sector. This partitioning strategy obviously increases the estimation accuracy. Figure 3 shows a comparison of probability estimation errors obtained on all three estimation methods on synthetic data.

The rationale for partitioning the attribute value range using a tolerance value is to

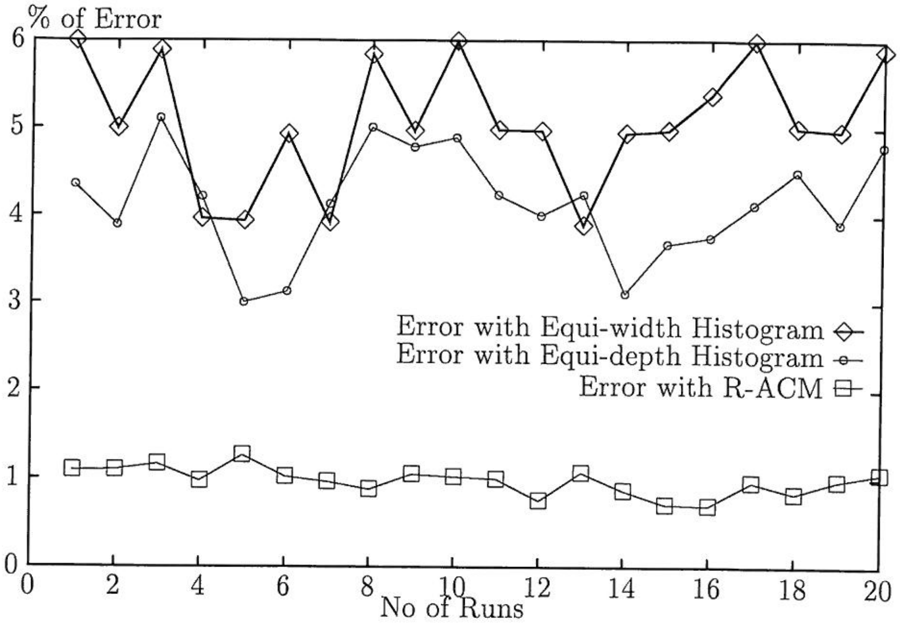


Figure 3: Comparison of Equi-width, Equi-depth Histograms and the R-ACM for Probability Estimation: Each experiment was run 500,000 times to get the average percentage of errors in the estimated occurrence of the attribute values. Estimation errors are given for the exact match on a random distribution with 100,000 tuples and 1000 distinct values. For the R-ACM, the tolerance was $\tau = 3$.

minimize the variance of values in each ACM sector, and this, as we shall see, has the effect of minimizing the estimation errors. Since the variance of an arbitrary attribute value X_k is given as $Var(X_k) = E[(x_k - \mu_k)^2]$, forcing the difference between the frequency of a given value and the running mean of the frequencies to be less than the tolerance τ, (i.e: $|x_k - \mu_k| \leq \tau$, will ensure that the variance of the values falls within the acceptable range.

Having described the properties of the R-ACM in the previous section, we shall now provide a comparison of the worst-case and average-case errors of this new technique to that of the traditional histograms. This is summarized in Table 2. It can be seen from this table that the R-ACM is much more accurate for query result size estimation than the traditional equi-width and equi-depth histograms. The demonstration of this fact for real-world databases follows.

5 Experiments on Real-World Databases

Because synthetic data is usually generated as random numbers or on the basis of some mathematical distributions, it is impossible to simulate real-world data distributions using synthetic data. Consequently, we have resorted to conduct our experiments on two real-world databases, namely, the United States CENSUS database and the database on the NBA

Histogram	Worst-case Error	Average-case Error
Equi-width	$\max\left(n_j - \frac{n_j}{l}, \frac{n_j}{l}\right)$	$\max\left(n_j - \frac{n_j}{l}, \frac{n_j}{l}\right)$
Equi-depth	$\frac{2}{3l_j}$	$\frac{1}{2l_j}$
R-ACM	$\tau \left\| \ln\left(\frac{l_j}{i-1}\right) - 1 \right\|$	2τ

Table 2: Comparison of Histogram and R-ACM Errors.

players. In addition to conducting our experiments using the data directly from the above two databases, we also resort to the use of two powerful mathematical distributions, namely the *Zipf distribution* and the *multi-fractal distribution*. We apply the frequencies generated from these two distributions randomly to the original value domains of the relations from the U.S. CENSUS and NBA databases, thus obtaining data distributions with wide range of skews.

5.1 Overview of the Zipf Distribution

G.K. Zipf first proposed a law, called the Zipf's law, which he observed to be approximately obeyed in many of the real-world domains, such as physics, biology, income distributions [17]. Zipf's law is essentially an algebraically decaying function describing the probability distribution of the empirical regularity. Zipf's law can be mathematically described in the context of our problem as follows.

For an attribute value X of size N with L distinct values, the frequencies generated by the Zipf distribution are given by,

$$f_i = N.\frac{1/i^z}{\sum_{i=1}^{L} 1/i^z}, \quad \text{for } 1 \leq i \leq L.$$

The skew of the Zipf distribution is a monotonically increasing function of the z parameter, starting from $z = 0$, which is the uniform distribution. We have plotted the frequency sets of several Zipf distributions with different z values in Figure 4. These frequency distributions were all generated for $N = 2000$ and $L = 10$.

One of the common claims in database literature is that many attributes in real-world databases contain a few attribute values with high frequencies and the rest with low frequencies [4], and hence can be modeled satisfactorily by Zipf distributions. Statistical literature abounds with information on modeling real-life data by Zipf distributions. This is why we have also resorted to using Zipf distribution to generate frequencies for the value domains in the relevant real-world databases.

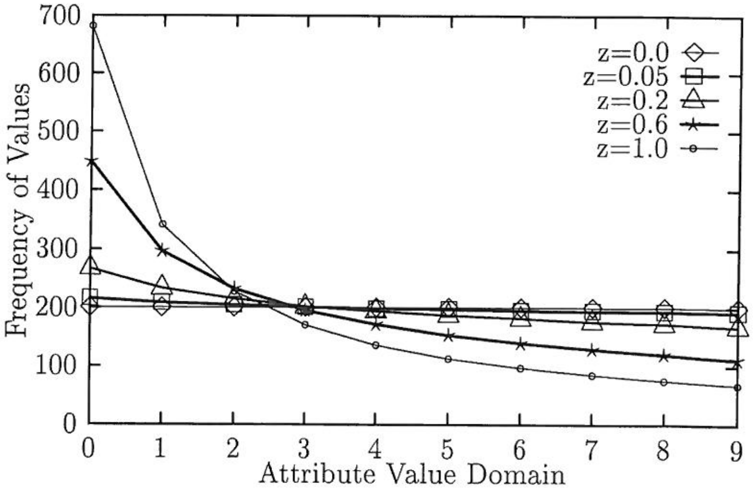


Figure 4: Zipf Distributions for Various z Parameters

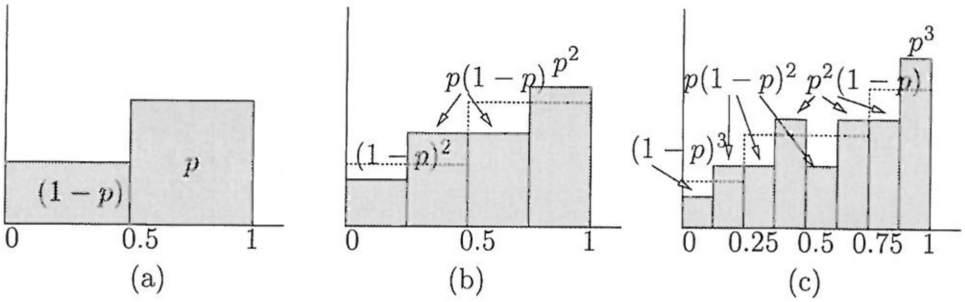


Figure 5: Generation of a Multi-fractal Distribution - First three steps

5.2 Overview of the Multi-fractal Distribution

The relationship of multi-fractals with the "80-20 law" is very close, and seems to appear often. Indeed, several real-world distributions follow a rule reminiscent of the 80-20 rule in databases. For example, photon distributions in physics, or commodities (such as gold, water, etc) distributions on earth etc., follow a rule like "*the first half of the region contains a fraction p of the gold, and so on, recursively, for each sub-region.*" Similarly, arguably, financial data and people's income distributions follow an analogous pattern.

With the above rule, we assume that the attribute value domain is recursively decomposed at k levels; each decomposition halves the input interval into two. Thus, eventually we have 2^k sub-intervals of length 2^{-k}.

We consider the following distribution of probabilities, as illustrated in Figure 5. At the first level, the left half is chosen with probability $(1-p)$, while the right half is with p; the process continues recursively for k levels. Thus, the left half of the sectors will host $(1-p)$ of the probability mass, the left-most quarter will host $(1-p)^2$ etc.

For our experiments we use a *binomial multi-fractal* distribution with N tuples and parameters p and k, with 2^k possible attribute values. Note that when $p = 0.5$, we have the uniform distribution. For a binomial multi-fractal, we have

Count	Relative Frequency
C_0^k	p^k
C_1^k	$p^{(k-1)}(1-p)^1$
...	...
C_a^k	$p^{(k-a)}(1-p)^a$
...	...

In other words, out of the 2^k distinct attribute values, there are C_a^k distinct attribute values, each of which will occur $N * p^{(k-a)}(1-p)^a$ times. Thus for example, out of the 2^k distinct attribute values, there is $C_0^k = 1$ attribute value that occurs p^k times.

5.3 Queries Used in the Experiments

The *select* and *join* operations are the two most frequently used relational operations in database systems. Thus for our experiments we will use queries that use these two operations.

For estimating the result sizes of select operations, we will actually use two types of select operations, namely the *exact-match select* and the *range select*. The exact-match select operation, denoted $\sigma_{X=X_i}(R)$, retrieves all the tuples from the relation R, for which the attribute X has the value X_i. The range select operation retrieves all the tuples falling within an attribute value range. For example, the query $\sigma_{X \leq X_i}(R)$, retrieves all the tuples from the relation R, for which the attribute value X has values less than X_i. For the join operation, we will use the most frequently encountered equi-join operation. The equi-join operation, denoted $R \bowtie_{X=Y} S$, combines all the tuples in the relations R and S whenever the value of attribute X from relation R is equal to the value of attribute Y from relation S.

5.4 Experiments on U.S. CENSUS Database

The CENSUS database contains information about households and persons in the United States for the years 1993 to 1995 [2]. Most of the relations in this database contains hundreds of attributes, both scalar-typed (such as a person's sex and type of job) and numerical (such as salary and age). The Data Extraction System (DES) at the U.S. Census Bureau allows extracting records and fields from very large public information such as governmental surveys and census records. The DES produces custom extracts in selectable data file formats that can be later analyzed by any statistical software package. Tables 3 and 4 describe the set of relations and attributes chosen from this database for our experiments. The data distributions of some of the selected attributes from CENSUS are plotted in Figure 6. The queries for our experiments consisted of either (a) equality join (b) equality selection or (c) range selection operators.

Relation Name	No of Tuples	Description
cpsm93p	155197	Population survey for 1993 - Person
cpsm94p_1	83455	Population survey for 1994 (Set 1) - Person
cpsm94p_2	150943	Population survey for 1994 (Set 2) - Person
cpsm95f	63756	Population survey for 1995 - Family
cpsm95h	72152	Population survey for 1995 - Household
pums905h	828932	Decennial Census Microdata Samples

Table 3: Some of the Relations in the CENSUS Database

Relation	Attribute	No of Distinct Values	Description
cpsm93p	hours	95	No of hours worked/week
	industry	48	Industry code
	wages	31321	Person's wages
cpsm94p	income	8143	Total income
cpsm94p_2	age	91	age
	hours	100	Hours worked/week
cpsm95f	income	32026	Annual income
	persons	15	No of persons/family
cpsm95h	state	51	State code
	wages	10496	Total wages
pums905h	wages	34668	Total wages

Table 4: Attributes in the CENSUS Database

The first group of experiments were conducted on equi-width, equi-depth histograms and the R-ACM. In each of our experimental runs, we chose different build-parameters for the histograms and the R-ACM. The build-parameters for the equi-width and equi-depth histograms are the sector width and the number of tuples within a sector respectively. The build-parameter for the R-ACM is the tolerance value, τ.

We obtained the relative estimation error as a ratio by subtracting the estimated size from the actual result size and dividing it by the actual result size. Obviously, the cases where the actual result sizes were zero were not considered for error estimation. We implemented a simple query processor to compute the actual result size. The results were obtained by averaging the estimation errors over a number of experiments and are shown in Table 5.

In the second group of experiments, we generated frequencies using the Zipf and multi-fractal distributions and applied them randomly to the value domains of the relations from the CENSUS database. Again the results were obtained by averaging the estimation errors over a number of experiments and are shown in Table 6.

In the third group of experiments, we compared the result estimates from the R-ACM

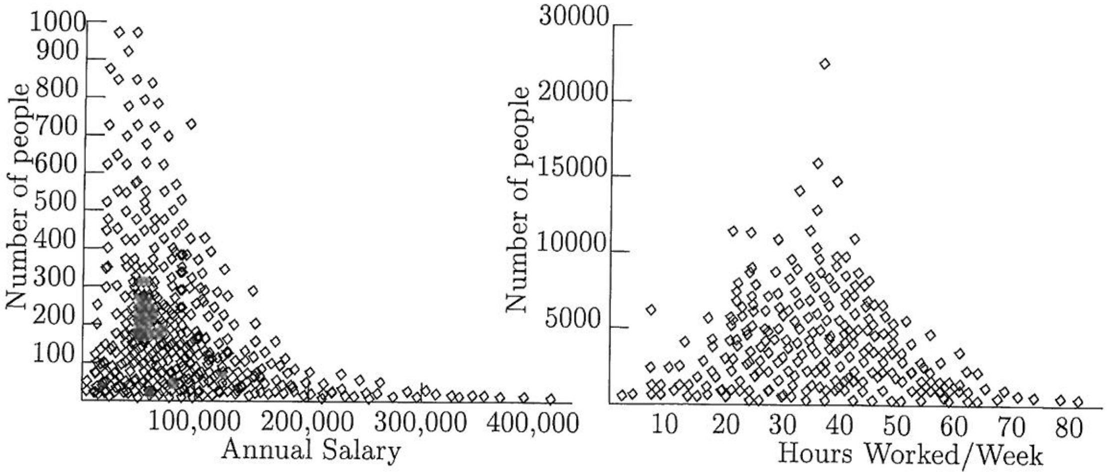


Figure 6: Frequency Distributions of Selected Attributes from the U.S. CENSUS

Operation	Actual Size	Equi-width		Equi-depth		R-ACM	
		Size	Error	Size	Error	Size	Error
Equi-select	1796	1279.1	28.8%	1365.6	23.4%	1702.3	5.23%
Range-select	32109	30008.3	6.5%	31214.9	2.8%	32319.2	-0.65%
Equi-join	720988	543908	24.6%	610482	15.3%	660183	8.43%

Table 5: Comparison of Equi-width, Equi-depth and R-ACM: U.S. CENSUS Database

for three different tolerance values. Again we considered (a) exact match select queries (b) range select queries and (c) equi-join queries and obtained the average estimation errors by comparing the estimates to the actual result sizes. The results are given in Table 7. We also repeated these experiments by applying the frequency values generated by the Zipf and multifractal distributions to the attribute value domains of the CENSUS database. The results of these experiments are given in Table 8.

5.4.1 Analysis of the Results

The results from the first two sets of experiments show that the estimation error resulting from the R-ACM is *consistently* much lower than the estimation error from the equi-width and equi-depth histograms. This is consequent to the fact the frequency distribution of an attribute value within an R-ACM is guaranteed to be close to the sector mean since the partitioning of the sectors is based on a user specified tolerance value and the deviation from the running sector mean. Thus we see that in the equi-select operation in Table 5, the percentage estimation error from the R-ACM is only 5.23%, but that of the equi-width and equi-depth histograms are 28.8% and 23.4% respectively, demonstrating an order of magnitude of superior performance. Such results are typical with both synthetic data and

Operation	Actual Size	Equi-width		Equi-depth		R-ACM	
		Size	Error	Size	Error	Size	Error
Equi-select	932	1182.5	26.9%	1127.1	20.9%	989.4	6.12%
Range-select	27180	29028.2	6.8%	28049.8	3.2%	27533.3	1.30%
Equi-join	589066	743990	26.3%	688029	16.8%	640904	8.80%

Table 6: Comparison of Equi-width, Equi-depth and R-ACM: U.S. CENSUS Database, using frequencies from the Zipf and Multifractal distributions.

Operation	Actual Size	Estimated Result			Percentage Error		
		$\tau = 4$	$\tau = 6$	$\tau = 8$	$\tau = 4$	$\tau = 6$	$\tau = 8$
Equi-select	1435	1338.5	1292.1	1143.6	6.75%	9.97%	20.34%
Range-select	26780	26219.1	24918.3	22098.9	2.09%	6.95%	17.48%
Equi-join	563912	610180	680953	719320	-8.2%	-20.8%	27.56%

Table 7: Result Estimation Using R-ACM: Data - U.S. CENSUS Database

real-world data. The power of the R-ACM is obvious!

The Third set of experiments illustrate that the accuracy of the R-ACM is inversely proportional to the tolerance value. Since smaller tolerance value results in a proportionally larger number of sectors in the R-ACM, there is obviously a trade-off of estimation accuracy and the storage requirements of the R-ACM. From Table 7, we see that for the tolerance value $\tau = 4$, the percentage error for the range-select query is only 2.09% whereas when the the tolerance value is increased to $\tau = 8$, the percentage error for the same operation becomes 17.48%. Such results are again typical.

Operation	Actual Size	Estimated Result			Percentage Error		
		$\tau = 4$	$\tau = 6$	$\tau = 8$	$\tau = 4$	$\tau = 6$	$\tau = 8$
Equi-select	2018	2121.3	2229.1	2443.4	5.12%	10.46%	21.08%
Range-select	39076	39849.7	41569.0	45054.6	1.98%	6.38%	15.30%
Equi-join	790528	866418	943495	997804	9.6%	19.35%	26.22%

Table 8: Result Estimation Using R-ACM: Data - U.S. CENSUS Database, using frequencies from the Zipf and Multifractal distributions for different tolerance values.

Our results from these three sets of experiments confirm our theoretical results, summarized in Table 2, and clearly demonstrate that the estimation accuracy of the R-ACM is superior to that of the traditional equi-width and equi-depth histograms.

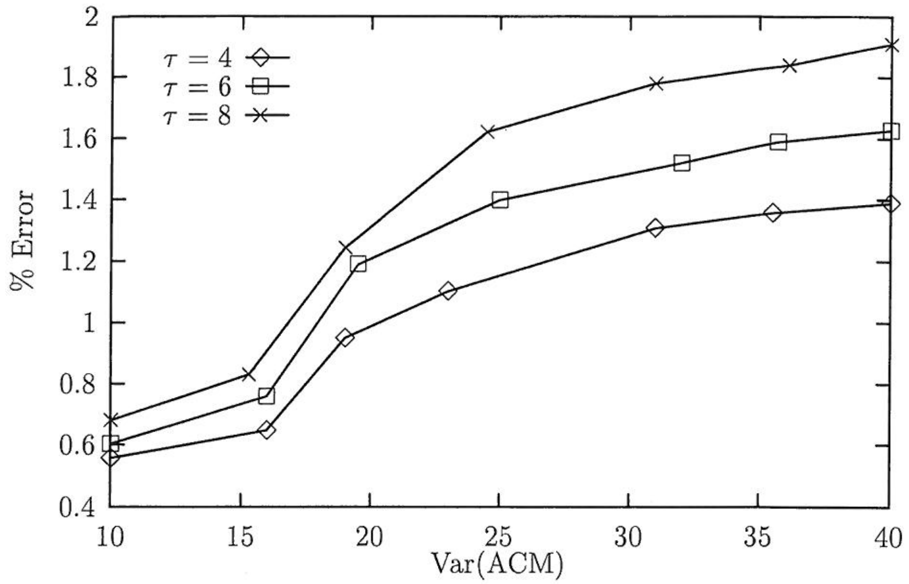


Figure 7: Estimation Error Vs Variance of the R-ACM: U.S. CENSUS Database

5.4.2 Estimation Errors and Variance of the R-ACM

The variance of the R-ACM was shown in our original paper [12] as $Var(R\text{-}ACM) = N - \sum_{j=1}^{s} \frac{n_j}{l_j}$, where N is the total number of tuples mapped by the R-ACM, n_j is the number of tuples within the j^{th} R-ACM sector and l_j is the sector width of the j^{th} R-ACM sector. One of our major results in [12] is that the R-ACM with smaller variance produces better query result size estimations. To demonstrate this results, we conducted a number of experiments on the U.S. CENSUS database for various attribute values and computed the variances of the R-ACM and the estimation errors. The errors between the estimated and actual size of random equality select queries are plotted against the computed variance of the ACM, and shown in Figure 7. These results, in addition to confirming the relationship between the variance of an R-ACM and the estimation errors, also confirm that the R-ACMs with lower tolerance values produce lower estimation errors.

5.5 Experiments on NBA Performance Statistics Database

The database on the NBA players is a performance statistics of NBA players for the year 1991-92 [1]. This database contains a relation with 50 attributes and 458 tuples. We have given the distributions of a few attributes from this relation in Figure 8.

In the first group of experiments, we compared the result estimations from equi-width, equi-depth, and the R-ACM. We considered (a) exact match select queries (b) range select queries and (c) equi-join queries. For the join queries, since the NBA database consists of a single relation, we generated various frequency values with the Zipf and multifractal distributions using the same value domain of the joining attribute from the NBA relation as

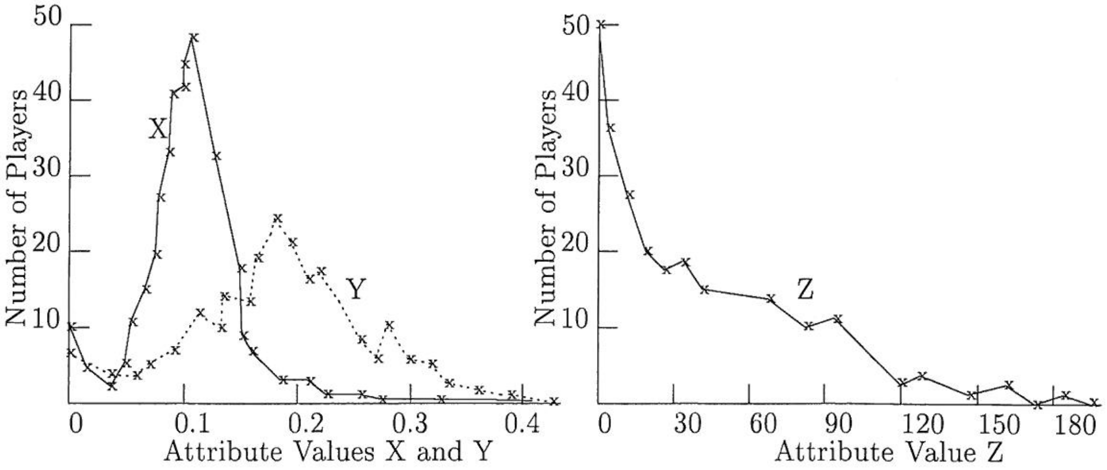


Figure 8: Frequency Distribution for Attribute Value No 9 from the NBA Statistics

the second relation and joined it with the corresponding attribute from the NBA relation. As in the case with the experiment on the CENSUS database, for each of our experimental runs, we chose different build-parameters for the histograms and the R-ACM.

We obtained the relative estimation error as a ratio by subtracting the estimated size from the actual result size and dividing it by the actual result size. Obviously the cases where the actual result sizes were zero were not considered for error estimation. Actual query result sizes were computed using a simple query processor. The results were obtained by averaging the estimation errors over a number of experiments and are shown in Table 9.

In the second group of experiments, we compared the result estimates from the R-ACM for three different tolerance values. As in the case with the U.S. CENSUS database, we considered (a) exact match select queries (b) range select queries and (c) equi-join queries and obtained the average estimation errors by comparing the estimates to the actual result sizes. The results are given in Table 10.

Operation	Actual Result Size	Equi-width		Equi-depth		R-ACM	
		Size	Error	Size	Error	Size	Error
Equi-select	38	23.3	38.68%	26.5	30.26%	36.2	4.7%
Range-select	119	111.0	6.7%	113.7	4.4%	116.4	2.18%
Equi-join	242	192.8	20.33%	204.1	15.7%	249.5	-3.09%

Table 9: Comparison of Equi-width, Equi-depth and R-ACM: NBA Statistics 1991/92

Operation	Actual Result Size	Estimated Result			Percentage Error		
		$\tau = 4$	$\tau = 6$	$\tau = 8$	$\tau = 4$	$\tau = 6$	$\tau = 8$
Equi-select	42.3	39.8	37.6	35.0	5.9%	11.11%	17.26%
Range-select	132.0	129.8	126.6	124.7	1.67%	4.09%	5.53%
Equi-join	318.2	301.8	285.6	264.1	5.15%	10.24%	17%

Table 10: Result Estimation Using R-ACM: Data - NBA Statistics 1991-92

5.5.1 Analysis of the Results

As in the case of the U.S. CENSUS database, the results from the first set of experiments show that the estimation error resulting from the R-ACM is a fraction of the estimation error from the equi-width and equi-depth histograms, again demonstrating the superiority of the R-ACM over the traditional histograms for query result size estimation. As we can see the percentage estimation error with the R-ACM for equi-select operation is only 4.7%, whereas for the same operation, the equi-width and equi-depth histograms result in 38.68% and 30.26% estimation errors - which is an order of magnitude larger!

As before, we see that our second set of experiments show that the accuracy of the R-ACM is inversely proportional to the tolerance value. The choice of the tolerance value is usually left to the database administrator as he/she would have the knowledge of how much trade-off the database system can afford in terms of the disk storage to achieve the desired response time for queries. In our experiment with the NBA database, we see that the R-ACM with the lowest tolerance value ($\tau = 4$) gives the smallest estimation error, 5.9% for the equi-select operation. When the tolerance value is increased to $\tau = 6$, we see that the estimation error for the equi-select operation is much higher at 11.11% which increases to 17% when $\tau = 8$. Such results are typical.

6 Conclusion

We recently introduced a new histogram like approximation strategy known as the Rectangular Attribute Cardinality Map and claimed that since this strategy is based on a user-defined tolerance for partitioning the sectors its worst-case and average-case errors are smaller than that of the traditional histograms. In this paper, we have presented a number of experimental results on the Rectangular Attribute Cardinality Map, including an extensive array of experiments using the U.S. CENSUS database and the NBA Statistics database. These modeling, prototype validation and testing results demonstrate and validate our theoretical analysis of the R-ACM and its superiority over the current state-of-the-art estimation techniques based on the traditional equi-width and equi-depth histograms. It is our hope that due to its high accuracy and low construction costs, it could prove to be a standard tool for query result size estimation in future database systems. Incorporation of these concepts in commercial databases is currently being considered.

References

[1] National Basket Ball Association. NBA Players Performance Statistics. ftp:olympos.cs.umd.edu, 1992.

[2] U.S. Census Bureau. U.S. CENSUS Database. 1997.

[3] S. Christodoulakis. Estimating selectivities in data bases. In *Technical Report CSRG-136*, Computer Science Dept, University of Toronto, 1981.

[4] S Christodoulakis. Implications of certain assumptions in database performance evaluation. In *ACM Transactions on Database Systems*, volume 9, pages 163–186, 1984.

[5] Christos Faloutsos, Yossi Matias, and Avi Silberschatz. Modeling skewed distributions using multifractals and the 80-20 law. In *Technical Report*, Dept. of Computer Science, University of Maryland, 1996.

[6] Yannis Ioannidis and S. Christodoulakis. On the propagation of errors in the size of join results. In *Proceedings of the ACM SIGMOD Conference*, pages 268–277, 1991.

[7] Yannis Ioannidis and Stavros Christodoulakis. Optimal histograms for limiting worst-case error propagation in the size of join results. In *ACM TODS*, 1992.

[8] Yannis Ioannidis and Viswanath Poosala. Balancing histogram optimality and practicality for query result size estimation. In *ACM SIGMOD Conference*, pages 233–244, 1995.

[9] R. P. Kooi. *The optimization of queries in relational databases.* PhD thesis, Case Western Reserve University, 1980.

[10] M.V. Mannino, P. Chu, and T. Sager. Statistical profile estimation in database systems. In *ACM Computing Surveys*, volume 20, pages 192–221, 1988.

[11] M. Muralikrishna and David J Dewitt. Equi-depth histograms for estimating selectivity factors for multi-dimensional queries. In *Proceedings of ACM SIGMOD Conference*, pages 28–36, 1988.

[12] B. John Oommen and Murali Thiyagarajah. The rectangular attribute cardinality map: A new histogram-like techniques for query optimization. Technical report, School of Computer Science, Carleton University, Ottawa, Canada, Jan 1999.

[13] B. John Oommen and Murali Thiyagarajah. The trapezoidal attribute cardinality map: A new histogram-like technique for query optimization. Technical report, School of Computer Science, Carleton University, Ottawa, Canada, Feb 1999.

[14] Gregory Piatetsky-Shapiro and Charles Connell. Accurate estimation of the number of tuples satisfying a condition. In *Proceedings of ACM SIGMOD Conference*, pages 256–276, 1984.

[15] Murali Thiyagarajah. PhD Thesis - In preparation, School of Computer Science, Carleton University, Ottawa, Canada.

[16] Murali Thiyagarajah and B. John Oommen. Prototype validation of the rectangular attribute cardinality map for query optimization in database systems. In *Third International Conference on Business Information Systems - BIS'99*, Poznan, Poland, April 1999.

[17] G. K. Zipf. *Human Behavior and the Principle of Least Effort.* Addision-Wesley, Reading, MA, 1949.