A Model of Data Distribution Based on Texture Analysis

Nabil Kamel
Roger King

Department of Computer Science
University of Colorado
Boulder Colorado, 80309

Abstract

To estimate the number of tuples satisfying a certain query, a data distribution model is proposed The model is based on a discrete approximation of the data space and belongs to the class of nonparametric models Using texture analysis techniques applied to the multi dimensional data space, it is proposed that a segmentation of this space be obtained as a means of obtaining a discrete approximation Thus the space is divided into a number of homogeneous regions which can be later queried to obtain good estimates of the size of the response set To obtain this segmentation, a new function to assess the homogeneity of a bit pattern is proposed Test results performrd for this function are presented to show the inverse correlation between its value and the resulting estimation errors

1. Introduction

Database performance studies often require estimates of the number of tuples satisfying a certain query To derive such estimates, a model of data distribution in a database is needed One way of constructing such a model is to use a multidimensional bit map where each dimension corresponds to one attribute and each bit represents one possible combination of attribute values Exact modeling using a bit map approach however, requires astronomical amounts of storage In this paper we propose an approximate model for relations which makes use of texture analysis ideas from the field of pattern recognition to reduce the storage requirements while maintaining reasonable accuracy

The discussion which follows is based on the relational model of data but similar results can be obtained for other models (e g network and hierarchical) [MERR79] presents a distribution model for relations which approximates a multidimensional bit map in a way which allows some control over the tradeoff between the storage requirements and the accuracy of the model The method however, does not exploit any natural data distribution characteristics which might exist in the database in selecting the sectors

To improve the accuracy of the model [PIAT84] propose a different approach for selecting the sectors based on equal heights instead of equal width as in [MERR79] Their ideas however, are limited to modeling the distribution of a single attribute In this paper we propose a new model for multi-attributed relations which is intended to return selectivity estimates for partial range selection queries (see footnote # 1) In our model, the sectors are not selected independently, but in combinations based on homogeneity of their contents It is shown that this produces a distribution model which requires less storage while providing more accurate results

The main idea is to divide the data space into unequal cells which contain homogeneously distributed data Figure 1 1 illustrates the effect of applying the texture analysis model (a) as opposed to dividing the data space into equal size cells (b) To use the model to estimate the tuple selectivity of a query, one has to remember that a partial range selection query has a response set which can be represented as a rectangle like the dashed one in figure 1 1-a The problem is now reduced to answering the question of which cells intersect with our query

After this question is answered (see section 5), the proportion of each one of those cells which also lies within our query space can be easily computed by comparing the coordinates of the cell and those of the query The number of tuples contributed by each cell to the main query are then obtained and summed to become our estimate of the number

This work was supported by IBM under a faculty development award

© 1985 ACM 0-89791-160-1/85/005/0319 $00 75

of tuples qualifying in our query

2. Motivation

In order to estimate the number of tuples satisfying a certain query in a given page of secondary storage, we need a way to model record distributions within files In cases where the real world phenomenon underlying the creation of tuples follows some known probabilistic pattern, a multivariate probability distribution can be an adequate way to describe the distribution of tuples in the relations of the database See [CHRI83] and [CHRI84a] In such cases, adding a few parameters which define this distribution can greatly enhance the accuracy of performance estimates [CHRI84b] An example of such a situation exists in conjunction with word frequency distributions in natural languages Such distributions have been shown to follow the Zipf distribution [SILE76] Distribution models which are built using this approach are known as ***parametric models*** Another method for finding a suitable probability function when knowledge about the underlying mechanisms of the database is incomplete is described using the principle of maximum entropy in [CHRI84b] In general, however, it is very difficult to find suitable distributions for data which is not random in nature and is dynamically evolving In this paper, we adopt a nonparametric approach to modeling our data, based on a discrete approximation

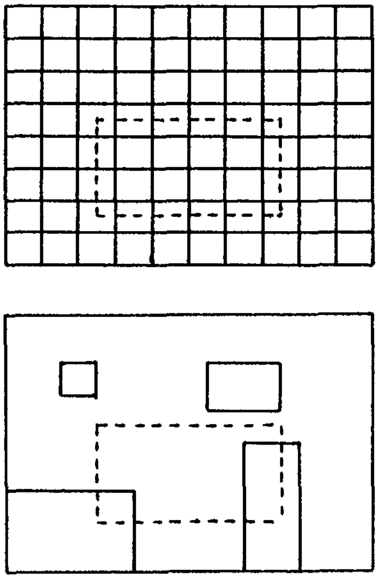


Figure 1 1
Equal and unequal sized cells in a distribution model

The ideal way to model the distribution of existing attribute values among all possible values is to use a bit map The bit map consists of a multidimensional boolean array with each element representing one possible combination of attribute values of one tuple Each element takes the value 1 if a tuple with that particular combination of values exists in the relation being modeled, and takes the value 0 otherwise

In fact, such a bit map obviates the need for any extra structures for the relation itself [NIEV84], insertions and deletions would be a matter of flipping one directly addressable bit Unfortunately such a representation is plagued by the curse of dimensionality The amount of storage needed for even a moderately large relation is beyond current technology (a relation of 12 attributes, each assuming any one of a 100 possible values will require 100^{12} bits to store it)

Merrett and Otoo [MERR79] propose a distribution model for relations which sacrifices the exactness of the bit map for the sake of storage reduction The degree of the tradeoff is controllable by the user Their approach is to divide each dimension into a suitable number of equal length sectors This will result in dividing the entire bit map space into a large number of adjacent hyper rectangles The information in each hyper rectangle is then replaced by a summary of it, namely a count of all the one valued bits which fall within that hyper rectangle

If the number of different possible values in each dimension were divided into a number c of equal width sectors of width d, then the bit map will be reduced in size by a factor of

$$\frac{d^D}{\lceil D \log_2 d \rceil}$$

where D is the number of dimensions in the distribution and equals the number of attributes in the relation Let us consider a relation with 12 attributes where each may assume any of 100 different possible values If we divide each dimension into 25 sectors, each having 4 different values, the above equation evaluates to a reduction factor of about 700,000 This will leave us with an inexact distribution model which still requires an astronomical amount (about 1 8 × 10^{17} bytes) of storage Working our way backwards, it can be shown that a sector size of at least 50 is needed to bring the storage requirements to under 1 MB

On the other hand, the larger the sectors get, the more crude the model becomes Since the purpose of the model is to estimate the size of the response set of any given partial range query[1], we will measure the crudeness of the model in terms of the maximum possible error in making such estimates Note that in reality, one does not expect a distribution model like the one proposed in [MERR79] to store all the cells, only the nonempty ones should be stored This will not impact the error resulting from such model, but will lower the storage requirements The model proposed here exploits the data distribution patterns which are likely to exist in many databases to further reduce the storage requirements while at the same time lowering the expected error

In the next section, we derive lower and an upper bounds on the estimation error resulting from using a model based on equal size sectors as in [MERR79] By applying this to our example 12-attributed relation, we show that in the worst case, the upper bound on the error is unacceptably high In the section following that, we present our approach to modeling the distribution of tuples in a relation and show its advantages

3 Error Analysis in a Model Based On Equal Width Sectors

To estimate the size of the response set of a certain query using the model described in [MERR79], we first note that the model divides the space of all possible tuples into equal size D-dimensional cells Each cell consists of d^D binary bits, where d, and D are as shown in figure 3 1 One basic assumption made about the distribution of actual tuple values within one cell is that of uniformity Thus, if a certain cell is labeled as having n tuples, it follows that the expected number of tuples associated with any one tuple value is

$$\frac{n}{d^D}$$

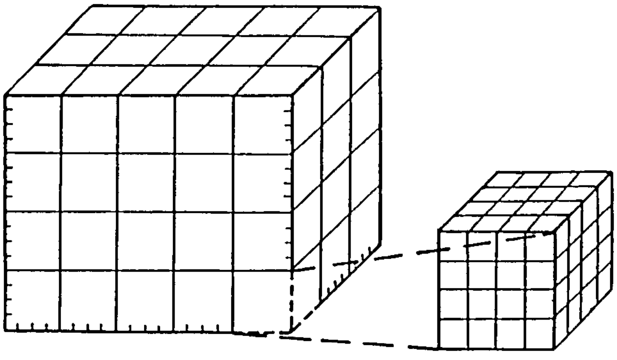


Figure 3 1
A three dimensional distribution having $d = 4$

In general, a query will define some random hyper rectangle in the D-dimensional space which properly includes a number h of whole cells and intersects with another number p of cells While no error arises from the h whole cells, all the estimation error is attributable to the fractional cells contained within our query space

The worst case error will occur as a result of the unfortunate coincidence of asking the worst possible query in a relation which has the worst possible distribution of tuples While the above worst case can be discarded as being very improbable, it is useful for comparison In addition, no estimate of the expected error can be obtained without knowing the probabilities of different queries, and the computation of the expected value of the error then becomes too complicated

It can be shown that for one cell of D dimensions, the worst possible combination of query and tuple distribution is obtained when exactly half the cell is filled with one valued bits, while the query overlaps the other half

Suppose that the query overlaps a bits out of a total of A bits in the cell Obviously, the maximum possible error is obtained when all the a bits of our query differ in value from all the remaining $A-a$ bits Assuming all the a bits are zeroes and all the remaining $A-a$ bits are ones, the estimation error p is given by

$$\rho = | \textit{estimated number of tuples} - \textit{actual number of tuples} |$$

$$\rho = | \frac{A-a}{1} \times a - 0 |$$

to obtain a maximum, we differentiate and equate to 0

$$\frac{\partial \rho}{\partial a} = A - 2a = 0$$

Differentiating a second time, we obtain

$$\frac{\partial^2 \rho}{\partial a^2} = -2 \qquad \textit{a negative quantity}$$

Thus, a maximum error of $\frac{A}{4}$ occurs at $a = \frac{A}{2}$ This means that the maximum error occurs when exactly one half of the cell is filled with ones while the query contains the other half

[1] A partial range query is one that specifies ranges for a subset of the set of the attributes in the relation

The above maximum error applies in the case of one D-dimensional cell We now give a worst case error estimate for a general query which can include and overlap any number of cells since the estimation error arises from the fractional cells which the query might intersect Intuitively, the worst case will occur when the query intersects as many fractional cells as possible with as many of these intersections following the worst case of cell intersection as possible

Figure 3 2 shows a two dimensional distribution with each dimension divided into 4 sectors The worst case for the estimation error is realized by the query whose response set is enclosed by the dashed square This occurs when the space between the dashed and the solid squares is filled with ones, while the rest of the distribution is filled with zeroes Generalizing from figure 3 2 to the case of D dimensions, we obtain an upper bound on the total estimation error P

$$P = 2^D \times abs(a - \frac{a^2}{A}) + 2D(c - 2)\frac{A}{4}$$

where

c is the number of sectors per dimension

d is the sector width

D is the number of dimensions

$$a = \left(\frac{d}{2}\right)^D$$

$$A = d^D$$

The first of the two terms of P represents the error resulting from the corner cells, while the second term represents the error caused by all the other half cells If the relation does not contain enough tuples to fill all the space outside the worst case query boundary, then the error will not be quite as large In fact it will be bounded by $\frac{N}{2}$ where N is the number of tuples in the relation Thus the total maximum error is bounded by

$$E_{\max} = \max\left(\frac{N}{2}, P\right)$$

Substituting for the case of our 12-attributed relation we find

$$E_{\max} = \max\left(\frac{N}{2}, 2\ 4 \times 10^{20}\right)$$

which means that it is possible to have an absolute error in any one query equal to half the relation size regardless of the size of the response set of the query Note that a distribution model based on equal sized cells does not take advantage of any special distribution patterns, like the existence of homogeneous or clustered regions Under such scheme, it is possible for the worst case of query/cell pattern discussed above to occur and thus contributing to the overall error in the selectivity estimates Since this error is unacceptable, we look for other ways to compress the data in the distribution model without sacrificing the accuracy too much

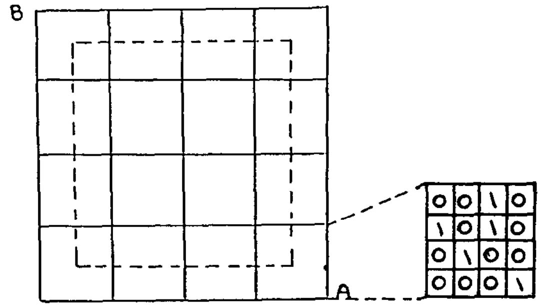


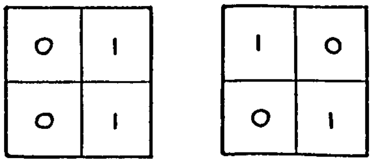
Figure 3 2
A two dimensional distribution for a relation of two attributes A and B

4. The Texture Analysis Model

Fortunately, a bit map which describes a real world relation is very likely to be extremely sparse This fact can be used to compress the data and reduce the map to a manageable size The problem of compressing large sparse multidimensional data arrays has been dealt with in at least two other fields, sparse matrix computations, and pattern recognition Those however, typically deal with no more than 3 dimensions and with considerably smaller data sizes In [NIEV84], the authors point out that matrix techniques used in numerical applications are inapplicable, since they are not compatible with the general file access operations, FIND, INSERT, and DELETE

We envision a new distribution model for relations based on ideas related to texture analysis in the area of pattern recognition Texture analysis is used to identify regions of interest in an image In our case, the bit map discussed above is analogous to an image The only difference is that an image is typically limited to 3 dimensions [HARA73] describes some easily computable textural features One of these textural features is of special interest to us, the homogeneity feature

We are interested in identifying the homogeneous areas of the distribution because these areas can be compressed easily without producing large estimation errors To illustrate this idea, let us consider the very simple example of the two 2-dimensional distributions shown in figures 4 1(a,b)



0	1
0	1

1	0
0	1

Figure 4 1
Two possible patterns in a 2×2 distribution

Suppose that we reduced these distributions in the same manner of [MERR79] This means that we will no longer view them as having distinct patterns, instead, we will think of them as being perfectly homogeneous with all cells having the same number of tuples This model for distributions is a form of data approximation which results necessarily in some loss of information Basing the selection of the cells on the homogeneity of their contents is a way to minimize this loss

In the last section we analyzed the quantization error resulting from replacing distribution cells by the number of ones in them We have always assumed the worst case In reality however, the worst case has a very low probability of occurring Instead, we typically have a large number of "bad" cases but not necessarily the worst

The model we propose divides the distribution space into cells, as described in [MERR79], but picks the cells in such a way as to avoid as many of these bad cases as possible A logical starting point is to classify all possible cells into good ones and bad ones We do so, but in a slightly different manner We will try to find some function which can be computed quickly and which places an ordering on all possible distribution patterns according to their goodness inasfar as the loss of information due to quantization, is concerned

Intuitively, the more homogeneous the original cell is, the more closely is it represented by the perfectly homogeneous model For example, the distribution of figure 4 1-a is more likely to produce larger errors than the one in figure 4 1-b because it is less homogeneous

To divide the data space into (nonuniform sized) cells which minimize the expected error in our selectivity estimates, we first divide the data space into a mesh of equal sized cells Then, the homogeneity of each cell is calculated The function which is used is

$$\sum_{i=1}^{n} (row(i) - u_r)^2 + \sum_{j=1}^{m} (col(j) - u_c)^2$$

where n is the number of columns in the cell,
m is the number of rows,
$row(i)$ is the number of ones in row i,
$col(j)$ is the number of ones in column j,
u_r is the average of $row(i)$ for all i, and
u_c is the average of $col(j)$ for all j

(Below, we present test evidence which indicates that this function performs well)

Once this information is obtained for all the equal sized cells, adjacent cells which show similar homogeneity are lumped together Typically, the result is the isolation of some number of isolated cells which have a homogeneity which is distinctly different from its surroundings The remainder of the data space is broken into a number of large cells This leads to a distribution model which is more accurate and which requires less space to store the information about the cells

Lumping together the adjacent cells which show similar homogeneity, suggests the use of some clustering method which takes into account two factors in determining the clusters, 1) physical proximity and 2) within-cluster homogeneity A similar problem arises when a segmentation of a digitized image is sought as an aid to pattern recognition This problem has been addressed by Rosenfeld and others in [BURT81] where a hierarchical computation scheme is proposed Experiments performed on several images (in two dimensions) indicate that the method produces good results and that it is reasonably fast Appendix A describes a multidimensional version of this scheme adapted for use in modeling database relations The method utilizes the homogeneity function proposed above to insure that the resultant segmentation will minimize the estimation error

An experiment programmed in FORTRAN under Unix has been used to examine the correlation between the homogeneity function above and the expected error The expected error for a certain size of cells (the experiment tests all 3 X 3 patterns) is computed by generating all the subrectangles of the cell, and summing the errors resulting from each subrectangle This is done by first calculating the expected number of ones (a one represents a tuple which actually exists in the database) for each subrectangle, as predicted by assuming a perfectly even distribution of ones The error of a subrectangle is then determined as the difference between that number and the actual number of ones falling within that subrectangle

Figure 4 2 is a scatter diagram generated by this experiment where the correlation between the function above and the expected error for all 3 X 3 patterns is evident Note, however, the existence of a number of exceptions To deal with this, a sampling method can be employed whereby the sample size is determined according to the desired accuracy Thus, actual error values for some sample of all the cells are calculated and taken to be representative of the error computed by taking all subrectangles into consideration Multidimensional distributions however, can have a very large number of bits and a sampling method may require a sample size which is too large for any practical computation So in cases where the number of dimensions is too high(i e for relations with many attributes), using a function like the one given in this paper appears to be the best approach

5. Accessing The Model

Another major question still lingers once the distribution model is known, how can it be used efficiently? Finding the cells which intersect with a query's response set is analogous to a common problem in geographical databases ([GUTT84]) Selecting the correct cells is similar to finding all the counties on a map which intersect with a particular rectangle The only difference is that a distribution model has more than two dimensions [GUTT84] addresses this multidimensional question and proposes a clever method for accessing multidimensional data This technique is called an R-tree An implementation of R-trees, written in C under Unix, has been obtained from the author Plans call for integrating our model and Guttman's technique

6. Conclusions

Multidimensional distribution models for relations which are based on a bit map approach are prohibitively large Dividing each dimension into a number of equal width sectors and summarizing the distribution within each one of its cells reduces storage considerably but does not exploit the natural clustering of data inherent in many databases It is shown that texture analysis can be used to select the cells of the distribution in a way which reduces the size of the distribution while maintaining reasonable accuracy Further research is needed to devise more efficient methods to query a distribution model built around these ideas Also in some special cases, the distribution of data can prove to be not easily amenable to homogeneity analysis Further study of these special distributions is needed

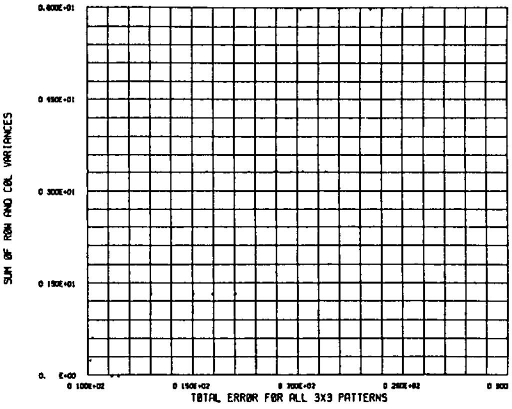


Figure 4 2
A scatter diagram showing the correlation between the function proposed and the quantization error

APPENDIX A

In this appendix we describe our adaptation of the pyramid linking techniques for picture segmentation to the problem of modeling the distribution of database relations We begin by dividing the data space into a large number of mutually exclusive and collectively exhaustive hyper rectangles This division is based on equal size cells and the cell sizes can be chosen to be a small number (*e g* 4^D *or* 8^D) where *D* is the number of dimensions Then the homogeneity of each one of those cells is estimated using either the function given in section 4, or by sampling Let us assume that the result can be arranged in a *D* dimensional array of size 2^{n^D} where 2^n is the number of points in each dimension

This array is again divided into overlapping hypercubes of sizes 4^D The amount of the overlap is 50 % in all directions as shown in figure A 1 for two dimensions The Array is viewed as being cylindrically closed in all directions thus, an end surface is considered adjacent to its opposing surface The number of one valued bits in each each one of those cubes is computed and divided by its volume (4^D) to obtain a new array of averages that has a number of elements which is less than those of the previous level by a factor of 2^D The process is continued until an exponentially tapering pyramid of values is obtained

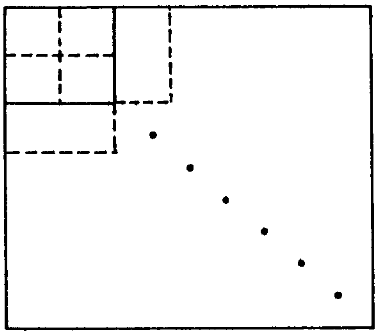


Figure A 1
Initial Division of the First Pyramid Level

The next phase in this process is a linking phase where each value on a given level is linked to one of the nodes on the higher level Each node on level $l-1$ participates in the average computation of 2^D nodes on level l Those 2^D nodes are taken to be potential fathers for the node on level l which participated in the computation of their values The father closest in value to that node is linked to it The process is repeated until the entire pyramid becomes linked

The third phase consists of recomputing the averages in the entire pyramid but this time, the average of a node is computed from the values of all its linked sons only This might introduce some parenthood changes which require relinking as was done in phase II

This process is repeated iteratively until the pyramid reaches a steady state (i e until we get two identical configurations in a row) Convergence of this method is guaranteed and test results done on two dimensional images indicate that only a few iterations are ever needed to reach steady state [BURT81]

A segmentation of the distribution is finally obtained with any one of several degrees of detail by projecting the values of any level to the base of the pyramid To project the values of certain level, the value of each node on this level is assigned to all its descendants on level 0 A higher level in the pyramid will produce a less detailed segmentation when projected on the base than if the projection were done from a lower level

BIBLIOGRAPHY

[BURT81] BURT, P , HONG, T , H and Rosenfeld, A , "Segmentation and Estimation of Image Region Properties Through Cooperative Hierarchical Computation", in *"IEEE Trans Syst , Man, Cybern ", Vol SMC-11, no 12, pp 802-809, 1981*

[CHRI83] Christodoulakis, S , "Estimating Block Transfers and Join Sizes", in *Proceedings of the SIGMOD International Conference,* 1983

[CHRI84a] Christodoulakis, S , "Estimating Block Selectivities", in *Information Systems,* Vol 9, 1984

[CHRI84b] Christodoulakis, S , "Implications of Certain Assumptions in Database Performance Evaluation", in *ACM Transactions on Databases,* June, 1984

[GUTT84] Guttman, A , "R-Trees A Dynamic Index Structure for Spatial Searching", in *Proceedings of Annual Meeting Of SIGMOD,* Boston, Ma , (June 1984), pp 47-57

[MERR79] Merrett, T , H , and Ekow Otoo, "Distribution Models of Relations", in *Fifth Int Conf on Very Large Databases,* Rio De Janero, Brazil,(October 1979), 418-425

[NIEV84] Nivergelt, J , Hinterberger, H , and Sevcik, K , C , "The Grid File An Adaptable, Symmetric, Multikey File Structure" in *ACM Transactions On Database Systems* vol 9, No 1, (March 1984), pp 38-71

[PIAT84] Piatetsky-Shapiro, G , and Connell, c , "Accurate Estimation Of The Number Of Tuples Satisfying A Condition" in *Proceedings Of The Annual Meeting Of The SIGMOD* Vol 14, No 2, Boston, Ma , (June 1984), pp 256-276

[SILE76] Siler, K , F , "A Stochastic Evaluation Model for Database Organization in Data Retrieval Systems", in *Communications of the ACM,* (February 1976), pp 84-95