Binomial Multifractal Curve Fitting for View Size Estimation in OLAP

Thomas P. NADEAU
EECS, University of Michigan
Ann Arbor MI, USA
nadeau@engin.umich.edu

Kanda RUNAPONGSA
EECS, University of Michigan
Ann Arbor MI, USA
krunapon@eecs.umich.edu

Toby J. TEOREY
EECS, University of Michigan
Ann Arbor MI, USA
teorey@eecs.umich.edu

ABSTRACT

On Line Analytical Processing (OLAP) aims at gaining useful information quickly from large amounts of data residing in a data warehouse. To improve the quickness of response to queries, pre-aggregation is a useful strategy. However, it is usually impossible to pre-aggregate along all combinations of the dimensions. The multi-dimensional aspects of the data lead to combinatorial explosion in the number and potential storage size of the aggregates. We must selectively pre-aggregate. Cost/benefit analysis involves estimating the storage requirements of the aggregates in question. We introduce a useful diagram illustrating rows in the fact table versus rows in an aggregate. We demonstrate predictable trends in these diagrams. We present an original curve-fitting approach to the problem of estimating the number of rows in an aggregate. We test the curve-fitting algorithm empirically against three published algorithms, and conclude the curve-fitting approach is the most accurate at small sample sizes.

Keywords: OLAP, View Size Estimation, Materialized Views, Data Warehouse, Binomial Multifractal.

1. MOTIVATION

Accumulation of data in industry and organizations has led to large archives of data in recent years. Quick access to the information in these archives has become critical for decision making. The need to excel has given rise to new data models and decision support systems. Typically the queries posed involve operations of aggregation such as sum or count. The queries also typically include "group by" expressions. For example, the CEO of a book manufacturing company may want to examine trends in profitability of different types of books over time. The answer could be found by doing a sum of the cost and sell values of jobs, grouped by bind style and quarter. Data warehouses have been engineered to answer queries of aggregation with "group by" expressions efficiently.

Data warehouses are commonly organized with one large central fact table, and many smaller dimension tables. The fact table is keyed by the attributes to be used in "group by" expressions. The fact table also contains measure attributes, the values to be aggregated. Each attribute of the fact table key is typically a foreign key matching the primary key of a dimension table.

In an actual data warehouse, the fact table may contain many millions of rows, and processing a single aggregate can require significant resources. To improve the quickness of response to queries, pre-aggregation is a useful strategy. Pre-aggregation requires the result to be saved to disk. The number of possible aggregates is exponential in the number of dimensions. Faced with combinatorial explosion and limited disk space, we must decide which aggregates to calculate in anticipation to queries. The cost/benefit analysis involves estimating the storage requirements of the aggregates in question.

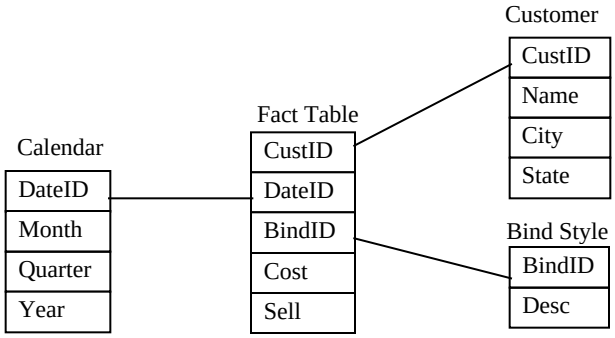

Fig. 1. A simple star schema for a data warehouse. The *CustID*, *DateID* and *BindID* together make up the primary key of the *Fact Table*. Thus there are three dimensions. Notice that a dimension can also have a hierarchy. For example, time can be grouped by *DateID*, *Month*, *Quarter* or *Year*

This paper focuses on estimating the space required for an aggregate. We introduce a useful paradigm for understanding the data trends: A curve diagram shows rows in the fact table versus rows in an aggregate. The trends are examined in both synthetic and real world data. Original curve-fitting approaches to the problem of estimating the number of rows in an individual aggregate are presented and tested against real and synthetic data sets.

2. RELATED WORK

This section first outlines the sources of ideas used in this paper for estimating the number of rows in an aggregate. Then some relevant papers are described which address the larger problem of materialized view selection.

View Size Estimation
There is a simple formula for estimating the number of rows in an aggregate. The approach is known as Cardenas' formula [1].

Let n be the number of rows.
Let v be the number of possible values.

$$\text{Expected distinct values} = v - v(1 - 1/v)^n . \quad (1)$$

Cardenas' formula assumes uniform distribution. However, the data distribution affects the number of rows in an aggregate. In order to capture the effect of data distribution, other methods have been developed.

Probabilistic counting was introduced as a new approach in [2]. A hashing function is applied to the values, and meta-data is gathered on the output. Probabilistic analysis is applied to the meta-data, determining an estimate of the number of distinct

values. The approach uses very little memory, but requires a full scan of the data.

A sampling approach based on the binomial multifractal distribution model is presented in [3]. Properties of the distribution are estimated from a sample. The number of rows in the aggregate for the full data set can then be estimated using the parameter values determined from the sample. Further details of this approach can be found in Section 4.1.

Three approaches are tested and compared in [6]. They examine Cardenas' formula, a sampling approach we call linear projection, and the probabilistic counting method. The probabilistic counting method is the most accurate of the three algorithms tested, for the given data sets.

Two algorithms, which are hybrids of Cardenas' formula and sampling approaches, are presented and tested in [5]. The proportional skew effect algorithm, and the sample frequency algorithm test favorably compared to Cardenas' formula and linear projection.

Materialized View Selection

The question of which views to materialize is addressed in [4]. A lattice structure captures the hierarchy of which queries can be answered from other views. The cost of materialization is based on the number of rows to be examined. The paper demonstrates that strategic selection of materialized views can yield dramatic benefits.

Strategic selection of materialized views using a data cube lattice is pursued further in [7]. The cost calculations used account for query frequency and the cost of update operations on materialized views.

3. EFFECT OF DATA SKEW

We gain insight if we diagram the number of rows in the fact table versus rows in an aggregate. Figure 2 shows the relationship for a data set with uniform distribution. The aggregate size was calculated with Cardenas' formula. There are 1000 possible rows in this aggregate.

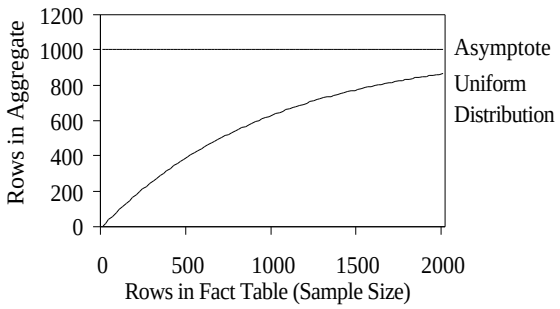

Fig. 2. This uniform distribution curve was calculated using Cardenas' formula. There are 1000 possible rows in the aggregate for this example

There are several properties of the curve to note here. The gradient is steeper when there are fewer rows in the fact table. When there are more rows in the fact table, the gradient levels out. Imagine you begin with an empty fact table. The first row added to the fact table will produce exactly one row in the aggregate. When the fact table is sparse, there is little overlap in the key values of the aggregate. The gradient will tend to be close to one near the origin. As you add more rows to the fact table, the probability increases that a corresponding row already exists in the aggregate. If a row already exists with matching key values, no row will be added to the aggregate. The increasing overlap of the key values leads to the curve leveling out. Note the number of possible values in the aggregate acts as an asymptote.

Data skew has a marked effect on the number of rows in an aggregate. Figure 3 illustrates the aggregate size for three binomial multi-fractal distributions. The binomial-multifractal formula for computing the expected number of distinct rows has three parameters. The first is the sample size, which we are showing as the horizontal axis. The second is a bias parameter P, which can range from 0.5 to 1.0. The P value for a uniform distribution is 0.5. The binomial multi-fractal formula will give the same answer as Cardenas' formula when P = 0.5. The value of P represents a measurement of the skewness (degree of clustering) of the data. The higher the value of P, the more skewed the data is. Figure 3 shows three levels of skewness: P = 0.5, 0.7 and 0.9. The third parameter is k, which is related to the number of possible rows in the aggregate. The number of possible rows in the aggregate for this diagram is 1024. The k value used in the calculations was 10. For this example, k = 10 because $\log_2 (1024) = 10$.

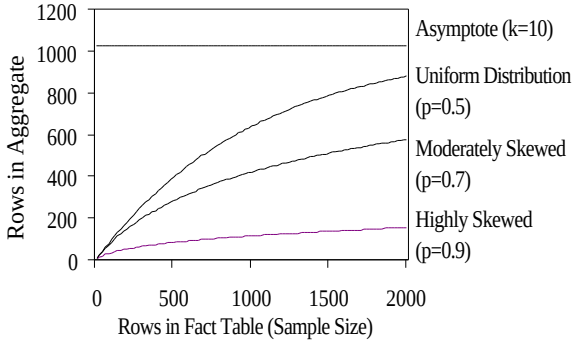

Fig. 3. Three curves calculated using the binomial multi-fractal distribution model

In general, skewness lowers the number of rows in an aggregate. Greater skewness (clustering) tends to compress the curve downward. This is true of real world data as well. Figure 4 shows curves for nine aggregates from our real world data. The aggregates were selected to represent the full range of skewness present in the real data. We will discuss the live data set in full detail in Section 5.

Note there is a wide range of skewness present in the aggregates of the real data. The aggregate represented by the top curve has a very large space of possible keys (1.4×10^{11} possible rows) and follows Cardenas' formula with no measurable error. The result of this combination is a nearly straight line. Skew can have a large effect on the number of rows in an aggregate. The curve at the bottom is the most skewed of all the aggregates. Using Cardenas' formula, the number of expected rows is 5,478 for this aggregate. The actual number of rows is 131. The error resulting from assuming a uniform distribution is over 4000%. Skew must be taken into account when estimating the storage requirements of pre-aggregation. Despite the variety of skewness in the real data, there is regularity present in the trends. It should be possible to project trends based on sampling

and achieve a reasonable approximation for the full fact table. We have developed and tested curve-fitting algorithms for this purpose.

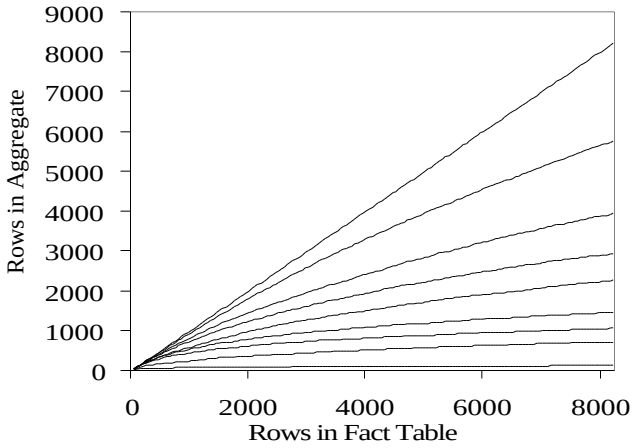

Fig. 4. Example curves illustrating the wide variation of skew present in a real world database. Each curve represents trends in a single a view as base data grows.

4 ALGORITHMS TESTED

We test four different algorithms. Cardenas' formula [1] is a fast calculation based on the assumption of uniform data distribution. The sample frequency algorithm [5] analyzes distribution information detected in a sampling of the data, and adjusts the estimate of Cardenas' formula to account for data clustering. The third approach, published in [3] is based on the binomial multi-fractal distribution model, and data sampling. We will refer to this approach as the FMS binomial multi-fractal approach. The fourth algorithm we test is an original algorithm we will refer to as binomial multi-fractal curve-fitting.

We will now examine the binomial multi-fractal distribution model in closer detail. Then we will compare and contrast the FMS approach with our curve-fitting approach.

Binomial Multi-Fractal Distribution Model (FMS)
Large-scale structure resembles small-scale structure in multi-fractal models. We will illustrate with a binomial multi-fractal distribution example momentarily. The concept that large-scale structure resembles small-scale structure is observed by [3] to be similar to the 80-20 rule in databases. The theory behind the approach presented in [3] is that by calculating the parameters of a multi-fractal distribution based on a small sample, the number of distinct members can be predicted for a larger set of data. Formulas (2) and (3) are presented in [3] for this purpose.

$$\text{Expected distinct values} = \sum_{a=0}^{k} C_a^k \left(1 - (1 - P_a)^n\right). \quad (2)$$

$$P_a = P^{k-a} (1 - P)^a. \quad (3)$$

Figure 5 illustrates with an example. Order *k* is the decision tree depth. C^k_a is the number of bins in the set reachable by taking some combination of *a* left hand edges and *k - a* right hand edges in the decision tree. P_a is the probability of reaching a given bin whose path contains *a* left hand edges. Bias *P* is the probability of selecting the right hand edge at a choice point in the tree.

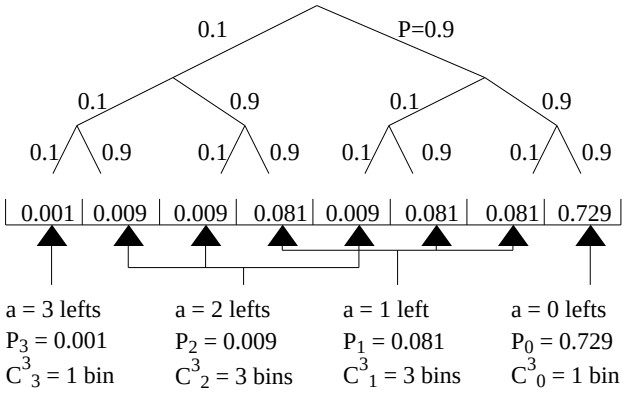

Fig. 5. Example of a binomial multi-fractal distribution tree. This small example is intended to illustrate the binomial multi-fractal model. The decision tree depth is k = 3. The probability of a right edge is the bias parameter P = 0.9. Small scale structure resembles large scale structure in multi-fractal models. The probability of a right branch remains the same regardless of the depth in the tree. Note the bins group as sets. There is a relationship between the number of bins in each set, and the elements in Pascal's triangle. The super-script of C is the depth into Pascal's triangle, starting with row 0 at the top of Pascal's triangle. The sub-script of C is the position into the row of Pascal's triangle, beginning with the left-most item as element 0

We illustrate the calculations of formula (2) with a small example. An actual database would yield much larger numbers, but the concepts and the formulas are the same. These calculations can be done with logarithms, resulting in very good scalability. Based on figure (5), given 5 rows calculate the expected distinct values:

$$\begin{aligned}\text{Expected distinct values} = {} & 1 \ (1 - (1 - 0.729)^5 \\ & + 3 \ (1 - (1 - 0.081)^5 \\ & + 3 \ (1 - (1 - 0.009)^5 \\ & + 1 \ (1 - (1 - 0.001)^5 \approx 1.965\end{aligned}$$

The algorithm in [3] for estimating the values of P and k is based on one sample. The algorithm has three inputs: The number of rows in the sample, the frequency of the most commonly occurring value, and the number of distinct aggregate rows in the sample. The value of P is calculated based on the frequency of the most commonly occurring value. They begin with

$$k = \lceil \text{Log}_{-}(\text{Distinct rows in sample}) \rceil. \quad (4)$$

and then adjust k upwards, recalculating P until a good fit to the number of distinct rows in the sample is found.

Binomial Multi-Fractal Curve Fitting
This approach is also based on Eq. (2). Our approach differs from [3] in the method of finding a good fit for the parameters P and k: our approach uses two samples instead of one. This is sufficient to determine the two missing parameters for each aggregate. There are five inputs to our algorithm for computing a good fit for P and k. The inputs are the number of possible

aggregate rows, the number of rows in each sample size, and the number of distinct aggregate rows in each sample. The general idea is to use one sample point to calculate the bias parameter for a specific order parameter k1. The same process is carried out for k2 = k1 – 1. The second sample point is used in calculating an error measurement for evaluating the accuracy of the parameter settings. The error measurements for the two parameter settings are compared. The algorithm decrements k1 and repeats until accuracy would be hindered. Then the best fit parameters are used to calculate the expected number of rows in the aggregate for the full fact table. A summary of the algorithm follows. Note the check p1 = 0.5 in line 12, it does not make sense to continue decrementing k1 if p1 is already at 0.5 because it is not possible to have a distribution less skewed than uniform.

Symbol	Definition
r	The number of possible rows in the space of the aggregate.
kInitial	The initial value for the order parameter k.
kLimit	kLimit is used to reduce algorithm complexity. It is based on the number of rows in the fact table. 2^{kLimit} should exceed the number of rows in the fact table by several orders of magnitude to assure accuracy.
k1	The larger of two settings under consideration for the order parameter.
k2	The smaller of two settings under consideration for the order parameter.
p1	The bias parameter associated with k1.
p2	The bias parameter associated with k2.
error1	The number of rows error associated with the set of parameters p1 and k1.
error2	The number of rows error associated with the set of parameters p2 and k2.
bestError	Tracks the best error measurement found so far.
tolerance	This variable allows the algorithm to continue past insignificant fluctuations in the error measurements caused by round off errors. We used a setting of 0.3 rows (our estimated row calculations were done in floating point) which was large enough to move past round off errors, yet small compared to our unit of error measurement, a row. We expect tolerance = 0.3 should be a good setting on other platforms as well.

Algorithm

```
1   Let kInitial = é log2(r) ù as a first estimate
2   kLimit = é log2(f) ù + 16, where f = |rows in fact table|
3   If (kInitial > kLimit) kInitial = kLimit
4   For (k1 = kInitial, k1 > 1, k1--) {
5       k2 = k1 – 1
6       Calculate bias p1 for larger sample, based on k1
7       Calculate error1 for smaller sample, based on p1 and k1
8       If (bestError is uninitialized) bestError = error1
9       Calculate bias p2 for larger sample, based on k2
10      Calculate error2 for smaller sample, based on p2 and k2.
11      If (error2>error1 and error2 - bestError >tolerance) break
12      If (p1 = 0.5) break
13      If (error2 < bestError) bestError = error2
14  }
15  Calculate expected rows for full data set, based on p1 and k1
```

5. THE DATA SETS

We tested the algorithms empirically, using both synthetic data and real world data. The synthetic data has been previously used in [5]. Using different samples from the same synthetic data has allowed us to validate the results in [5], and also compare the newer algorithm in the same setting. The real world data was gathered in cooperation with McNaughton & Gunn, Inc., a book manufacturing company.

Synthetic Data

The schema for the synthetic data has two dimensions. Each dimension has a three level hierarchy. There are six data sets, each with a different level of skew. The data was generated with Zipf distributions. The Zipf exponent in our test results refers to the exponent in the Zipf power-law function. A distribution with Zipf exponent 0.0 has a uniform distribution. A distribution with Zipf exponent 1.0 is skewed. The Zipf exponent can be higher than 1.0, and the higher the Zipf exponent, the more skewed the data. The fact tables in each of the synthetic data sets contain 10,000 rows. Details of how the synthetic data sets were generated can be found in [5].

Real World Data

The motivations for testing aggregate storage size estimation algorithms on real world data are many. The usefulness of any algorithm ultimately depends on the effectiveness of its application to real world problems. Real world data presents a challenge in that the skewness of the data is usually unknown. Often the type of distribution present is not clear. Should the data be modeled using Zipf distributions, binomial multifractal distributions, or some other distribution model? The curve fitting approach analyzed here attempts to measure the effect of skewness in a sample, and extrapolate the result to the entire fact table.

McNaughton & Gunn periodically analyzes the job mix in their plant. Analyzing a fact table containing job specifications is a realistic application. Specifications for 14,438 jobs were gathered. A fact table was built using ten job attributes as the key. An eleventh field was used for tracking the number of jobs with the given job specifications. Some jobs have the same key values. The job count field is not part of the key, it is a measurement field. Job count is the information to be aggregated. Some jobs have duplicate specifications, so the resulting fact table has fewer rows than the original set of jobs. The fact table has 8,238 rows. The dimensions are as follow:

Attributes	Possible Values	Explaination/Examples
Bind Style	14	Paper back, hard cover, comb bound etc.
Trim Width	13	The width of the book (e.g. 6”)
Trim Length	14	Book length from top to bottom (e.g. 9”)
Pages	31	The number of pages in the book
Quantity	28	The number of books ordered
Stock Color	18	Paper color used in the book (e.g. white)
Stock Weight	5	An industry standard paper weight measurement for a fixed amount of paper (e.g. 50#, 60# etc.)
Stock Width	12	Width of paper run on the press (e.g. 29”)
Stock Length	12	Length of paper run on press (e.g. 42”)
Press	5	Type of press (e.g. Miehle, Planeta etc.)

A quick calculation shows that the total number of possible tuples in the base data is 143,315,827,200. The density of the

base data is 5.7 x 10^{-8}. Even with such sparsity, 43% of the jobs have the same specifications as other jobs. This already gives a clue that the data is very skewed. There are 10 dimensions, and therefore 2^{10} different aggregates. The aggregates will dominate the fact table in the number of rows after cubing.

6. EXPERIMENTAL RESULTS

The algorithms were implemented using Visual Basic with a Microsoft Access database. The computer was a Pentium-II 266, with 32 MB RAM. We tested empirically with two types of data: Synthetic and real.

Measurements on Synthetic Data

Our measurements on the synthetic data examine how two parameters affect the algorithms. We vary the amount of skew, and sample size. The sample size indicates the percent of the fact table included in the sample. For the binomial multi-fractal curve fitting algorithm, this percentage is the size of the larger sample. Samples were selected at random from the fact table. The error measurements are calculated based on cube size.

Error = (Estimated Cube Size – Actual Cube Size) / Actual. (**5**)

The size of a data cube equals the sum of the number of rows in all possible aggregates and the core. Negative error measurements indicate the algorithm under estimated the actual number of rows. We present results in Figures 6, 7 and 8.

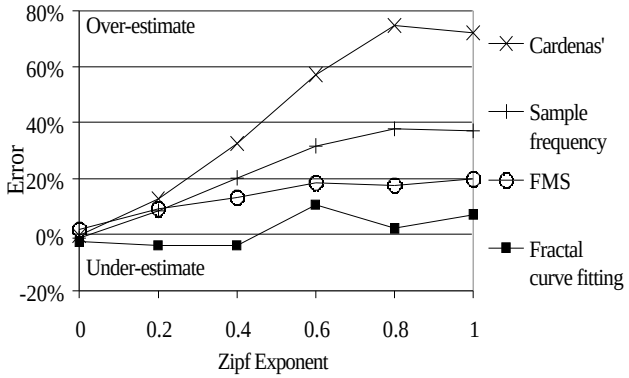

Fig. 6. Error Measurements on Synthetic Data, 10% Sample Size. A *Zipf Exponent* of 0 is a uniform distribution. Increasing *Zipf Exponents* indicate increasing skew

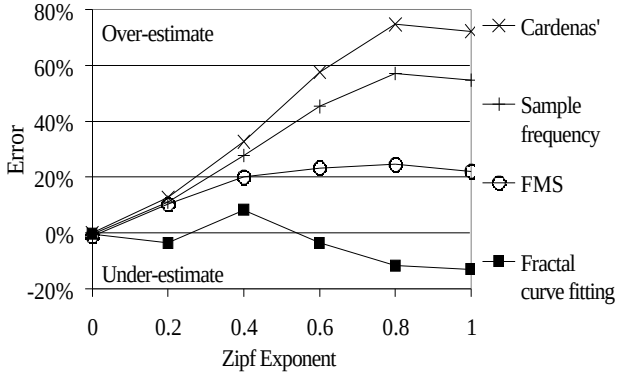

Fig. 7. Error Measurements on Synthetic Data, 3% Sample Size

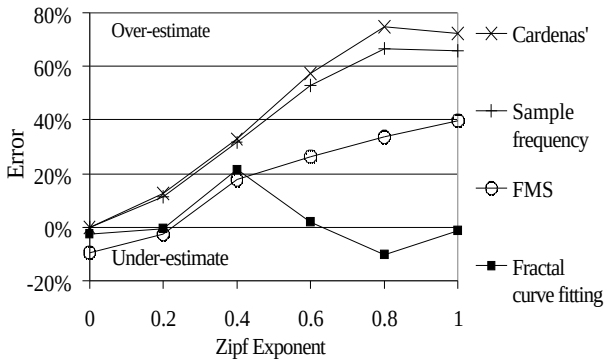

Fig. 8. Error Measurements on Synthetic Data, 1% Sample Size

Measurements on Real World Data

We varied one parameter when testing on the real world data: Sample size. For each algorithm / sample size combination, we measured three runs. The samples for each run were selected independently, at random from the same fact table (i.e. our base data remained the same, the sample set was varied). We were interested in measuring performance in estimating individual aggregate sizes in a real world environment. Towards this end, we measured the error rates on individual aggregates. The real world data set has 1023 aggregates. For each aggregate we calculated the ratio estimate/actual. Then we calculated the mean and the standard deviation of the ratio measurement, over all aggregates. These measurements are presented in table 1. The ideal algorithm would have a "mean of estimate/actual" ratio of 1.0, with a small standard deviation.

		Mean of Est/Act.			Stand. Dev.		
		Sample Size			Sample Size		
Algorithm	Run	1%	3%	10%	1%	3%	10%
Cardenas'	1	5.33	5.33	5.33	5.05	5.05	5.05
	2	5.33	5.33	5.33	5.05	5.05	5.05
	3	5.33	5.33	5.33	5.05	5.05	5.05
Sample Frequency	1	2.80	1.77	0.98	1.73	1.08	0.57
	2	2.92	1.76	1.01	1.84	1.11	0.57
	3	2.97	1.84	0.97	1.90	1.10	0.57
FMS Binomial	1	1.27	1.10	1.00	0.59	0.38	0.20
Multi-Fractal	2	1.39	1.07	1.03	0.69	0.39	0.20
	3	1.37	1.14	1.01	0.68	0.33	0.19
Binomial Multi-Fractal	1	0.96	1.04	0.86	0.70	0.39	0.20
Curve Fitting	2	0.88	0.83	1.04	0.69	0.41	0.23
	3	0.82	0.89	0.78	0.66	0.45	0.19

Table 1. Algorithm performance on real world data

7. ANALYSIS AND CONCLUSIONS

When data is uniformly distributed, Cardenas' formula estimates the number of rows in an aggregate very well. Cardenas' formula fails miserably when the data is skewed. Real world data can be very skewed as evidenced by our test bed.

The sample frequency algorithm does better than Cardenas' formula because the effect of skew is modeled. The performance still leaves much to be desired. The flaw is in the underlying

hypothesis that the error percent in Cardenas' formula due to data skew will remain relatively constant as the rows in the fact table increase. To see the flaw, consider the origin. We demonstrated earlier that the gradient of the actual curve at the origin is one. This is true regardless of the distribution. The sample frequency algorithm compresses the entire curve of Cardenas' formula by a constant factor to fit the sample point and then makes further adjustments based on clustering detected in the sample. The step where the curve of Cardenas' formula is compressed by a constant leads to a gradient less than one at the origin. The error percent in Cardenas' formula due to data skew is not constant. The hypothesis is flawed leading to significant error.

The FMS binomial multi-fractal is the algorithm outlined in [3]. We note one check that needs to be added to their algorithm. As k increments, it is possible for step three of their algorithm to over-estimate the actual F_0 (see [3] for details). If ε is small enough, the algorithm may overshoot the mark, and fail to terminate. This can be easily corrected by terminating when the estimate begins moving away from F_0, and use the best approximation found.

Our binomial multi-fractal curve fitting approach fares better than the FMS approach when the sample size is small. The FMS approach is sensitive to the count of the most common key value in the sample. As the sample size decreases the count of the most common key value tends to decrease linearly. The curve fitting algorithm is sensitive to the number of rows added to the aggregate between the two sample points. As the sample size decreases, the change in the number of rows between the aggregates at the two sample points tends to decrease less than linear because of the increasing gradient. This results in greater stability for the curve fitting approach over the FMS approach at small sample sizes.

At small sample sizes (i.e. 1%), the binomial multi-fractal curve fitting approach performed the best of the algorithms tested.

8. FUTURE WORK

The two binomial multi-fractal approaches produce more accurate aggregate size estimations under different conditions. Perhaps the two algorithms can be combined into a new algorithm, which exceeds the accuracy of both. There are at least two approaches to combining the two algorithms. We could modify our algorithm to also examine the frequency of the most common member, then average the two calculations for the bias parameter P. Another approach would be to map out the circumstances where each algorithm performs better. Depending on conditions, the best algorithm could then be used.

Multiple independent runs would likely decrease the variance in the estimates. The algorithm could build up a cloud of estimate points, and find a best fit for those points as the final estimate. There is a trade-off between processing and accuracy here. A good approach would be to do as much processing as is possible within an acceptable amount of time.

The binomial multi-fractal distribution model was chosen for approximating the actual curve for two reasons. The available parameters in the formula allow the curve to be adjusted to fit closely to the actual curves, and the formulas are easily programmed. There may be other distribution models that would yield good results if applied to a curve fitting approach.

The scalability in the number of rows in the fact table is currently being tested empirically using databases with up to 5 million rows. Scalability in the number of dimensions is a problem for further research.

We are working on implementing the probabilistic counting approach [2]. We did not include this algorithm with the original testing since probabilistic counting requires a full scan of the fact table, and we were focusing on sampling approaches. However, probabilistic counting deserves testing along with these other approaches. Probabilistic counting is extremely memory efficient It may perform well in terms of resources if implemented to process all aggregates in one pass through the fact table.

The amount of skew found in the different aggregates varies greatly. The skew itself may be an indicator of the interest value to humans. When humans look for trends in data, generally we look for correlations. Skew may be a good heuristic for data mining. The algorithm could bring highly skewed aggregates to the attention of the user.

We need to move toward a practical system. Statistical methods need to be applied to determine the proper sample size automatically. The aggregate size estimation needs to be incorporated into the larger problem of materialized view selection. For example, the number of rows used in the cost calculations of [7] could be obtained by estimating from sampling.

9. REFERENCES

[1] A. F. Cardenas. "Analysis and Performance of Inverted Database Structures". *Comm. ACM 13*, (May 5, 1975), pp.253 - 264.

[2] P. Flajolet, G. N. Martin. "Probabilistic Counting Algorithms for Database Applications". *Journal of Computer and System Sciences 31*, 1985, pp. 182 - 209.

[3] C. Faloutsos, Y. Matias, A. Silberschatz. "Modeling skewed distributions using multifractal and the '80-20 law'". *Proceedings of the 22nd VLDB Conference*, Mumbai (Bombay), India, 1996, pp. 307 - 317.

[4] V. Harinarayan, A. Rajaraman, J. D. Ullman. "Implementing Data Cubes Efficiently". *Proceedings of 1996 ACM-SIGMOD Conf.*, Montreal, Canada, pp. 205 - 216.

[5] K. Runapongsa, T. P. Nadeau, T. J. Teorey. "Storage Estimation for Multidimensional Aggregates in OLAP". *Proceedings of 10th CASCON Conference*, Toronto, Nov. 8 – 11, 1999, pp. 40 – 54.

[6] A. Shukla, P. M Deshpande, J. F. Naughton, K. Ramasamy. "Storage estimation for multidimensional aggregates in the presence of hierarchies". *Proceedings of the 22nd VLDB Conference*, Mumbai (Bombay), India, 1996, pp. 522 - 531.

[7] H. Uchiyama, K. Runapongsa, T. J. Teorey. "Progressive View Materialization Algorithm", *Proc. 2nd Int'l Data Warehousing and OLAP Workshop*, Kansas City, Nov. 6, 1999, pp. 36 - 41.