

# Analysis of Accuracy of Data Reduction Techniques

Pedro Furtado and H. Madeira

University of Coimbra  
Portugal  
pnf@dei.uc.pt

**Abstract.** There is a growing interest in the analysis of data in warehouses. Data warehouses can be extremely large and typical queries frequently take too long to answer. Manageable and portable summaries return interactive response times in exploratory data analysis. Obtaining the best estimates for smaller response times and storage needs is the objective of simple data reduction techniques that usually produce coarse approximations. But because the user is exposed to the approximation returned, it is important to determine which queries would not be approximated satisfactorily, in which case either the base data is accessed (if available) or the user is warned. In this paper the accuracy of approximations is determined experimentally for simple data reduction algorithms and several data sets. We show that data cube density and distribution skew are important parameters and large range queries are approximated much more accurately than point or small range queries. We quantify this and other results that should be taken into consideration when incorporating the data reduction techniques into the design.

## 1. Introduction

Data warehouses integrate information from operational databases, legacy systems, worksheets or any external source, to be used for decision support. The data warehouse must have efficient exploration tools, which, regardless of data size, may give fast reasonably approximate answers to users exploring the data interactively and multidimensional models are usual for the interactive exploration of the data in Online Analytical Processing (OLAP). To build the data cube, facts and dimensions must be identified as well as the data granularity. The dimensions can be products, stores and time with granularity of days. The space needed is calculated as in:

*time span = three years*

*# products = 100.000 products (of which only 20% are sold daily)*

*# stores = 100*

*n° of records in the fact table = 3 × 365 × 20.000 × 100 = 2.19Gbytes*

*average record size = 8 attributes × 4 bytes = 32 bytes*

These figures do not include indexes, materialized views and other fact tables. A complete data warehouse such as this one could have 70 GBytes of size. Data reduction techniques can be applied to any data cube derived from the facts or materialized views to reduce parts of the multidimensional space and obtain fast response times for approximate answers. This in turn is very important for exploring

the data. There are several alternative data reduction techniques, such as sampling [2, 7, 8], singular values decomposition (SVD) [6], wavelets [10], histogram-based techniques such as MHIST [5], clustering algorithms such as BIRCH [14] and index trees. These techniques are summarized in [9]. Data analysis needs for approximate answers directly expose the user to the estimates obtained. Although the reduced data is frequently associated with a very coarse initial approximation of the data, accuracy is very important. Even the simplest histogram-based reduction techniques return very small errors for large range queries encompassing whole summary regions. But queries may not encompass whole regions and answers to smaller range queries are also important. Furthermore, the possible absence or slow access of base data stored in tertiary memory or the reduction of summary tables in which points represent aggregated values require higher accuracy. In any case the exploration tool must be able to determine which queries are inaccurate and either access the base data or warn the user. Typical estimation errors are determined experimentally in this paper. The input data set is reduced using alternative techniques and several classes of queries are issued to determine the average estimation error. The experiment involves different data distributions and characteristics such as skew, density and sparseness. The results obtained in these experiments are used to conclude the accuracy that can be expected from data reduction algorithms. The paper is organized as follows. In section 2 alternative generic reduction strategies are discussed. Section 3 presents the data reduction strategies and section 4 the data sets used in the experiments. Section 5 shows the point and range error results that were obtained from the experiments and the conclusions that can be drawn from such results. Section 6 concludes the paper.

## 2. Alternative Reduction Strategies

In this section we address the strategies for histogram-based data reduction and the impact on storage of choosing a given strategy.

Multidimensional data points are represented in relational OLAP (ROLAP) as  $\text{tuple}(a_1, \dots, a_n, v_1, \dots, v_m)$ . The multidimensional view can be obtained from the tuples by using the dimension attributes  $a_i$  as axis and the values  $v_i$  as the data cube contents. The task of the reduction algorithm is to derive approximate values for sets of value attributes  $v_i$  in the data cube regions, reducing the data set size (for simplicity we will consider only one value attribute). The reduced data can be stored as a summary, loaded or maintained in memory for fast answers to queries from tools exploring the data.

### 2.1 Classification of Reduction Techniques

The types of data reduction techniques we evaluate divide the multidimensional space into regions and approximate each region by a summarized description. Fast querying and searching is obtained by accessing the summarized descriptions. A generic summary is a set of regions  $R([a_{1s}, a_{1e}], \dots, [a_{ns}, a_{ne}], \text{coeff}_1, \dots, \text{coeff}_x)$  forming a histogram where a region is usually called bucket or cell. We define some important properties of data reduction techniques regarding the resulting histograms:

- *Reduction strategy* - distinguishes algorithms producing fixed grids (*fixed grid strategy* - FG) or variable sized buckets (*variable grid strategy* - VG).
- *Variable grid strategy adaptability* - measures the degree to which the space partitioning strategy is able to adapt to the data distribution.
- *Approximation function* - used to determine the coefficients that approximate the data in each bucket. Those coefficients will be kept in the bucket.
- *Approximation function adaptability* - measures the degree to which the approximation function is able to adapt to the data distribution.

The *fixed grid strategy* (FG) imposes a fixed grid upon a multidimensional space view of the input data and approximates each grid cell by the approximation function coefficients. The *variable grid partitioning strategies* (VG) determines the best bucket partitioning of the same multidimensional space. A generic bucket produced by the VG strategy is represented by the structure `bucket(MBR(l_point,ur_point), data)`, where MBR denotes the region minimum bounding rectangle. Buckets produced by the fixed grid strategy can be represented more compactly by either the structure `bucket(bucket_ID, data)`, where `bucket_ID` is determined by a computation on the indices, or stored as multidimensional array cells as `cell(data)` where the cell position is also determined by computation on the indices. Variable grid strategies use alternative algorithms to partition the space into buckets dynamically. There are recent proposals for both fixed grid and variable grid algorithms. Regression is used in [1] and wavelets in [11]. These are fixed grid techniques that use the approximation function and occasionally outliers to obtain a higher adaptability. Mhist [5] is a variable grid technique. In this paper we consider mainly fixed grid techniques because, by storing only the reduced values (molap organization), a lot of space is saved in comparison with adaptable techniques (which must store the buckets as explained before) and reduced values are accessed by simple computation of an offset. The extra space can be traded for lower reduction rates to improve approximations.

Data with smooth variations is usually easier to approximate by most algorithms and large peaks often disturb the approximation. For this reason the use of outliers is important in histogram-based techniques whenever there are strong “thin” peaks such as a point completely divergent from the normal trend. Nevertheless, outliers are very expensive in terms of storage space: they require the storage of both the point coordinates and the value and do not provide associative access (must be searched). An outlier is stored as `Outlier(point, value)`.

Summary tables (or materialized views) are frequently computed to speed-up query answering in data warehouses: group-by queries are issued to compute partial sums on alternative combinations of dimension attributes, building the summary or materialized view. It is possible to recover range values from partial sums using the independence assumption or linear regularization [3]. This way, aggregation can be used as a data reduction technique.

## 2.2 Storage Cost Analysis of Fixed and Variable Grid Strategies

Variable grid strategies must store bucket boundaries, while fixed grid strategies store only the approximation coefficients. This subsection compares the storage space for the two alternative strategies to quantify the overhead incurred by the VG strategy. The storage space occupied by both strategies is simply,

$$ss(\text{fixed\_grid}) = \text{sizeofcoeffs} \quad (\text{molap organization}) \quad (1)$$

$$ss(\text{variable grid}) = 2^{\wedge} \text{PRSZ} + \text{sizeofcoeffs} \quad (\text{bucket tuples})$$

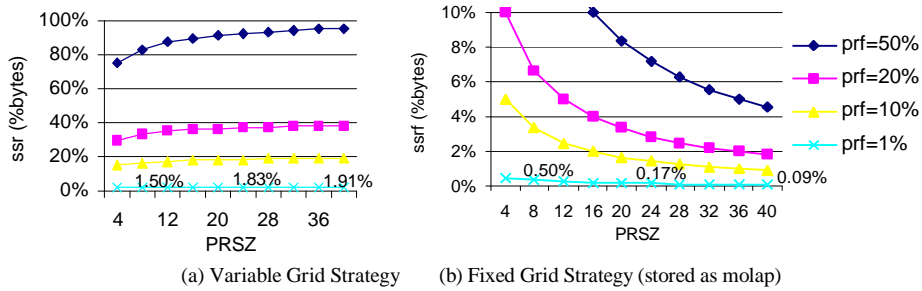
Given the following quantities,

$$\text{Point reduction factor PRF} = \frac{\# \text{ buckets}}{\# \text{ points}} \quad (2)$$

$$\text{Storage reduction factor SSR} = \frac{\text{space occupied by summary}}{\text{initial space}} \quad (3)$$

$$\text{Points representation size PRSZ} = \text{size of coordinates} \times n^{\circ} \text{ of dimensions} \quad (4)$$

Figure 1 compares SSR against PRF for VG and FG strategies considering 2 to 10 dimensions and coordinates with 2 to 4 bytes (PRSZ between 4 and 40). The data values size considered was 4 bytes.



**Fig. 1.** Storage Reduction Factors for Fixed grid and Adaptable Strategies

This figure quantifies the space overhead required to represent the buckets in adaptable strategies and, conversely, the space gains when fixed grid strategies are used. For instance, for a point reduction to 10% of an original five dimensional space (with each dimension represented by 2 bytes: PRSZ = 10), the reduced data occupies 17% of the original data size for the adaptable strategies and 3% for the fixed grid approach. This shows that adaptable partitioning strategies incur in high storage space overhead in comparison to fixed grid strategies, which also offer faster computation. These are strong motivations for the choice of these algorithms instead of variable grid ones, although the lack of partitioning adaptability must be compensated by approximation function adaptability, requiring more coefficients in each cell (e.g. Wavelets) (bucket partitioning vs. coefficients storage overhead).

### 3. Data Reduction Techniques

The data reduction techniques work on a multidimensional view of the input data, which must be cubed by parts (by loading arrays from tables such as in [12] – in production systems - or issuing queries against the database that retrieve the regions successively) to be fed as input to the reduction algorithm. It is preferable to apply the reduction only to non-empty points (those appearing in the rolap table) because the approximation error will be much larger if zeroes are also approximated. Empty positions are indicated by heavily compressed zero bitmap cuboids. The data reduction algorithms used in the experiments are:

- *Average* (g) – this gold experiment simply returns the average value. If the error is large, the data set will be difficult to approximate.
- *Outliers* (ol) – This algorithm simply extracts extremes, storing them as outliers. It can be used with any other technique, smoothing peak data variations. Outliers are stored as outlier(point, value) pair in table tuples.
- *Mhist* (mhist) – This VG technique was proposed in [5] for selectivity estimation. It implements space partitioning by analyzing marginal frequencies. Buckets are stored in table tuples as bucket(bucket\_ll,bucket\_ur, avg).
- *Fixed Grid* (fgrid) – This FG technique divides the space into equal-sized regions and computes the average for each one. It is stored as a multidimensional array as cell[avg].
- *Regression* (regr) – We used the implementation of linear regression described in Quasi-Cubes [1]. This FG technique approximates the values in a column or row bucket by a line described by the parameters  $m$  and  $b$  in  $y = m \times x + b$  and stores (m,b) in the bucket cell in a multidimensional array: cell[(m,b)].
- *Wavelets* (wav) – The wavelets technique (FG) was proposed for selectivity estimation in [11] and for data cube reduction in [10]. Wavelets represent a function in terms of a coarse overall shape, plus details that range from coarse to fine. Wavelet coefficients had to be stored together with a location identifier because smaller coefficients were stripped from the array of coefficients: cell[set(locator,coeff)].

In Figure 2 we further characterize the techniques.

technique	Partition strategy	Adapt algorithm	Approx. function	Approx. adaptability	Storage cost
<b>outliers</b>	FG	-	extract extremes	choice of extremes	very high
<b>fgrid</b>	FG	-	average	-	low
<b>regr</b>	FG	-	linear regression	choice of line	low
<b>wav</b>	FG	-	wavelet	keep larger coeffs	medium
<b>mhist</b>	VG	marg. freq	average	-	high
<b>clustering</b>	VG	clusters	average	-	high
<b>aggregate</b>	-	-	average(sum,..)	-	medium

**Fig. 2.** Characterization of Techniques

## 4. Datasets

We have tested synthetic and typical warehouse data sets. The synthetic data sets were random (completely random values), zipf [13], and clustered (n clusters following the normal distribution). Figure 3 shows typical shapes for those distributions. The zipf distribution (figure 3(a) and (b)) is said to be typical of many real data distributions in databases [13]. The skewed zipf distributions (figure 3(b)) contains high thin peaks which are often difficult to approximate by fixed grid techniques but can be extracted as outliers. Skewed clustered distributions (figure 3(c)) also show some peaks, but those peaks are thick and therefore cannot be handled efficiently by outliers. Adaptable techniques handle these skews much better than fixed grid techniques because they adapt the bucket boundaries to the topography.

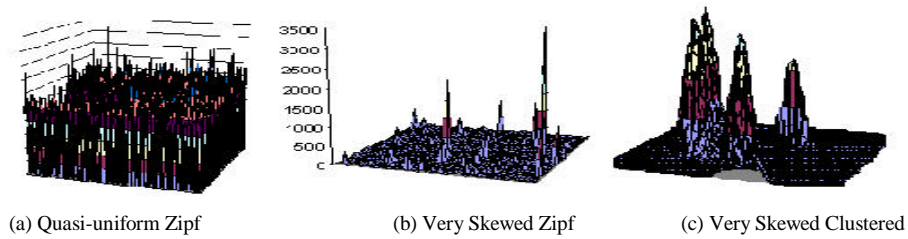


Fig. 3. Typical Synthetic Data Sets

Typical warehouse data sets were taken from several data warehouses in [4], including the sales dataset (product×days×stores = revenue: 60 ×184×20) and summaries resulting from roll-up operations (e.g days to weeks) (figure 4).

Data Set	Distribution	Data Set	Distribution
Zipf	Skewed Zipf	Sales	Sales Data Cube (97% sparse)
Cluster	Normal Cluster	Sales-agreg	Week Roll-Up Sales Data Cube
Cluster-Skew	Skewed Cluster		

Fig. 4. Typical Data Sets

## 5. Experiments

Two error measures were used: the point query error and the range query error.

$$perror_{estimation} = \frac{\sum |v_{estimated} - v_{exact}|}{\sum_{all\_points} v_{exact}} \quad (5)$$

$$Rerror_{estimation} = \frac{|\sum_{range} v_{estimated} - \sum_{range} v_{exact}|}{\sum_{range} v_{exact}} \quad (6)$$

These error measures are plotted against storage space reduction. Query experiments were held for *small ranges* with areas ranging from 1 to 10.000 points and *large ranges*. For each sub-category we made 10,000 queries of the type range-sum query. We first discuss the results for clustered and zipf synthetic data sets

which reveal important characteristics. Then we show the results for the SALES data set.

### 5.1 Clustered Distributions

We use four clustered data sets to show how skew and sparseness influence the results,

Data set	stdev	%cluster centers	Sparseness	Characterization
Random	-----		10%	Uniform
Cl_Skew	$\sigma=0.02$	0.25%	10%	Skewed
Cl_Lskew	$\sigma=0.02$	0.025%	10%	Very Skewed
Cl_Skew_LSp	$\sigma=0.02$	0.25%	80%	Skewed, Sparse

Figure 5 shows the point error for these data sets. The x-axis represents the storage space reduction factor (the output data size as a percentage of the input data size) when the input data is represented either as a data cube (%DC) or using a rolap organization (the data cube is the most compressed representation for non-sparse data sets - dimension attributes are implicit -, while rolap is the most compressed organization for sparse data sets - data cubes must represent empty points as zeroes). From the reduced data, only *fgrid* and *regr* are stored in small data cube representations, while wavelets must store a large number of coefficients in each cell, *ol* must store the points in tuples and *mhist* must store the buckets in tuples as well. Figures 5(a) ,(b) and (c) refer to dense data sets and 5(d) to a sparse version of 5(b).

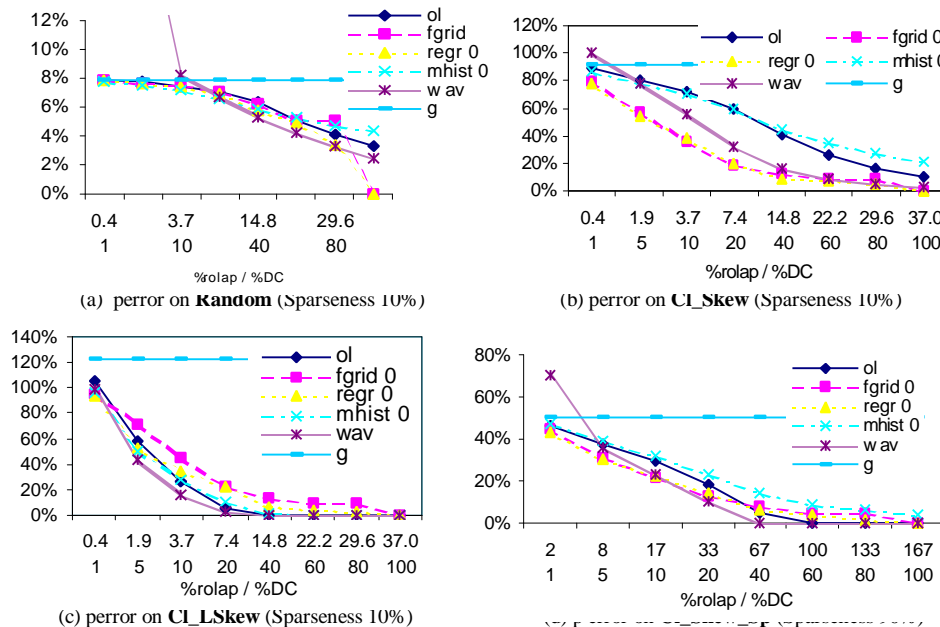
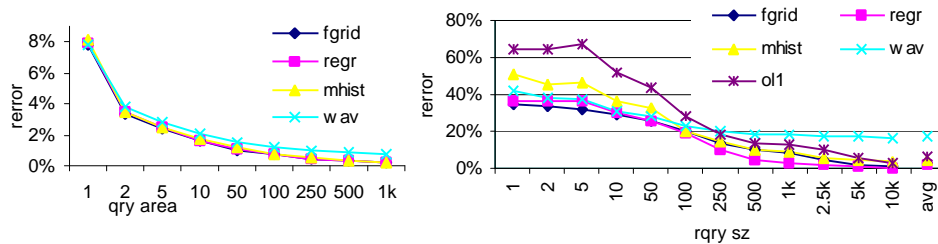


Fig. 5. Perror for Clustered Data Sets

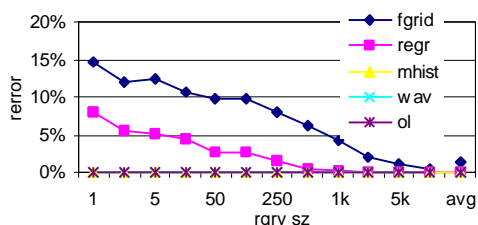
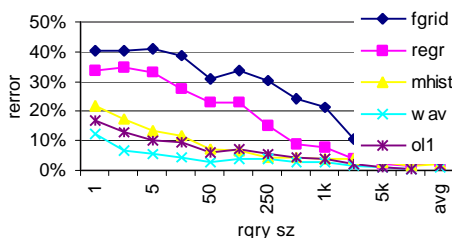
**Discussion of Results:** Figure 5(a) shows a point error of 8% for the gold experiment and a random distribution. Summaries occupying less than 10% of the data cube do not achieve any significant reduction of estimation error. The techniques progressively become more effective as more storage space is allocated for the reduced histogram. With about 80% of the data cube (30% of the rolap input) the error is halved to 4%. Figure 5(b) shows an estimation error of 90% for the gold experiment on skewed cluster distributions. With about 20% of the data cube size (7% of rolap input size) *fgrid* and *regr* techniques reduce the error to 20%, *wavelets* to 30% and *mhst* or *outliers* to 60%. With 80% of the data cube (30% of rolap) both *wavelets* and *regression* techniques reduce the average point error to 4%. These results suggest that very large reduction rates produce high estimation errors. Comparing (a), (b) and (c) we can see that skew is a source of approximation difficulty for very large reduction rates. The gold experiment (g) gives a clue to quantify the degree of this difficulty. The skew increases from the Random distribution to CI\_Skew and to CI\_Lskew and the point error for the gold experiment was 10%, 95% and 125% respectively. *Wavelets* are particularly well succeeded for skewed distributions, but have difficulty approximating low skew distributions for very large reduction rates, because wavelet coefficients occupy a significant space and important coefficients should not be dropped (higher reduction rates are obtained by dropping more coefficients, starting by the least significant ones). The three techniques - *wav*, *ol* and *mhst* - would be superior to *fgrid* and *regr* considering the point reduction factor (PRF) but that superiority is often lost by considering the storage space reduction factor (SSRF) (see figure 5(b)). Still, for very skewed distributions the adaptability (coefficient or point adaptability in the case of wavelets or outliers respectively) allows these techniques to yield better results than *fgrid* or *regr* (see figure 5(c)). Even *regr* achieves better results than *fgrid* because it is slightly more adaptable. Figure 5(b) and (d) were generated similarly but with totally different sparseness (10% on (b) and 80% on (d)). In this case the same reduction to 40% of the data cube size corresponds to completely different reduction if rolap input is considered (15% in (b) against 67% in (d)). This is because the non-reduced data cube can become considerably larger than the rolap representation due to sparseness (non-existent points in the rolap organization must be represented in the data cube). The estimation error is small when compared against the non-reduced data cube size (because a large fraction of the data cube are zeroes which are not included in the approximation) but, if compared against rolap size, it is comparable to those in Figure 5(b). This has two major implications: it is advantageous to compute reduced data cubes instead of normal data cubes when data sets are sparse because the normal data cube wastes a large space with zeroes, while the reduced version is much more compact with a small error. On the other hand, in order to be accurate, these reduced data cubes occupy a space that corresponds to a large portion of the initial rolap space, both for dense or sparse data sets.

Figure 6 shows the range query estimation errors vs query size for clustered data sets using a reduced data cube to (3.7% rolap / 10% DC ) in (a), (b) and (c), and a reduction to (14.8% rolap / 40% DC ) in (d). The query size is the query range volume.



(a) Error on **Random** (3.3% rolap / 10% DC )

(b) Error on **CL\_Skew** (3.3% rolap / 10% DC )



(c) Error on **CL\_LSkew** (3.7% rolap / 10% DC )

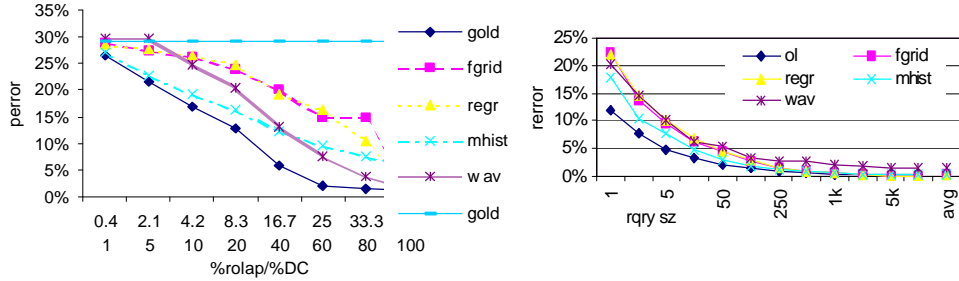
(d) Error on **CL\_LSkew\_Sp** (14.8% rolap / 40% DC )

**Fig. 6.** Range Query Errors for Clustered Datasets and Reductions as Indicated

Results in figure 6 show that large range queries return reasonably accurate results. This result is logical. For instance, *fgrid* buckets lying completely within the query range contribute with error 0 to the result because the bucket average value is used. For the random data set, ranges with areas above 500 points had an error below 0.4%. The results are not so good for skewed distributions (b) and for very skewed distributions (c). Figure 6(c) and (d) show that more adaptable techniques (*mhist*, *ol*, *wav*) are able to approximate very skewed distributions much more accurately than simpler techniques such as *regr*, *fgrid* or aggregation (these should use outliers to adapt to large skews). When a large portion of the data cube is kept (Figure 6(d)) and adaptable techniques are used the approximation is more accurate. Very sparse data sets have a very small number of non-empty points per unit area. To achieve real reduction of the rolap input in this case buckets must be very large and large range queries frequently select only a small number of non-empty points, giving significant estimation errors.

## 5.2 Zipf Distribution

Next we show the point and range error results for very skewed zipf data set.



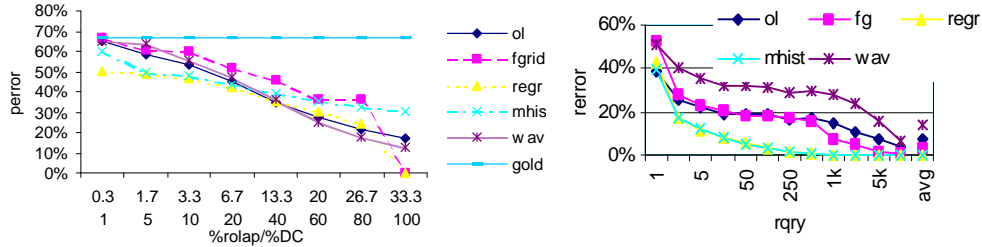
(a) Point Error for Zipf Skewed Data Set (b) Range Error for Zipf Skewed Data Set (20%DC)

**Fig. 7.** Point and Range Errors for Zipf Skewed Distribution

**Discussion of Results:** This is a skewed distribution and the most adaptable techniques (wavelets, mhist and outliers) held the best results for point errors. Outliers held the best result overall because they eliminate the thin peaks of the zipf distribution. This means that outliers should be used together with other techniques for isolated points that are very distant from the approximation.

### 5.3 Warehouse Data Sets

We have chosen a normal (**Sales**) and an aggregated (**Sales-aggreg**) warehouse data set. Figure 8 shows results for **Sales-aggreg**, a very dense data set summarizing sales.

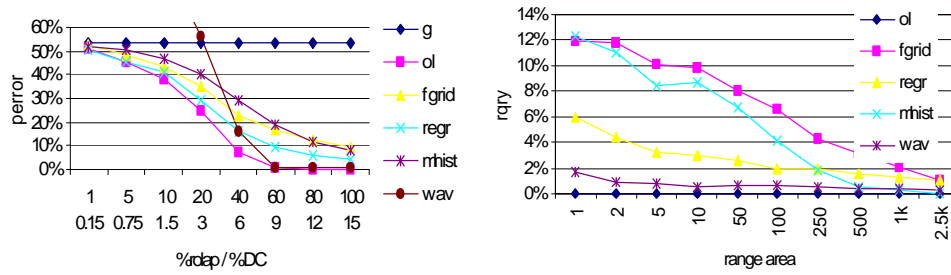


(a) Point Error on **Sales-aggreg** (b) Range Error on **Sales-aggreg** (20%DC)

**Fig. 8.** Point and Range Errors for Sales-aggreg Data Set

The dense Sales roll-up data set was difficult to approximate. Point errors are always large. *regr* was the best technique for large reduction rates and *wav* or *outl* for smaller reduction rates. This is because *regr* is more space efficient and *wav* or *outl* are more adaptable to the data distribution but less space efficient. *mhist* and *regr* achieved better results than the other techniques for range queries.

The multidimensional view of rolap data is frequently very sparse. Most data sets presented before were dense but the Sales data set of Figure 9 is very sparse (97%).



(a) Point Error (b) Range Query Error (80% rolap/ 12%DC)

Fig. 9. Point and Range Errors for the Sales Data set

In Figure 9(a) the estimation error is reduced to very small values with only 9% to 15% of the data cube (*wav* or *outl* are able to reduce the error to an insignificant amount with just 9% of the data cube). But the remarks made concerning sparse data sets apply here as well. 15% of the data cube corresponds to 100% of the rolap input size! The estimation error is so small simply because no storage space reduction was achieved (comparing to the rolap input). If the rolap data occupied 20 GB, the reduced data cube would occupy 20 GB as well. When compared against rolap size, the estimation error is comparable to the dense data set case. The range query results were also obtained for a small reduction rate considering the rolap data size (to 80%). For this size the range query errors are small using the *wav* or *outl* adaptable techniques.

### 5.4 Experiment Conclusions

The estimation error varies a lot with different data sets and distribution skews. Even the simplest techniques can obtain small estimation errors for range queries with large size (with a large number of non-empty points). But when querying typical sparse data sets, small ranges or points, the estimation error is frequently large. Even when a significant fraction of the input data size is allocated to the reduced summary, the error is still significant. For several data sets the estimation error does not decay exponentially as more space is allocated to the approximation. In the case of sparse data cubes, although they can be highly reduced, such reduction does not correspond to a large compression of the base (rolap) data. The tool doing analysis on the reduced summaries should rely on a minimum number of non-zero values to determine if a query can be answered with sufficient accuracy. This threshold could be for instance 1000 non-empty values, but the actual value depends on the data reduction rate.

None of the techniques seems to be substantially better than the other ones for all data sets. Although more sophisticated techniques such as *rhist* or *wavelets* obtain a lower point reduction factor (prf), they incur in higher storage overhead (for storing the buckets and coefficients), such that the approximation is not much better than the one obtained using compact molap reduced data sets from simpler algorithms. Still, adaptability is important to approximate irregular and skewed data sets. It is possible to conclude that the best results can be achieved by using a fixed grid strategy with some adaptability that can be obtained by either using strongly adaptable approximating functions such as wavelets or outliers.

## 6. Conclusions

In this paper we have made an experimental evaluation of histogram-based data reduction techniques focusing on the approximation error for several classes of queries. The data reduction algorithms were classified according to important characteristics and those characteristics were compared through the experiments. Data sets were also analyzed to determine how the distribution, skew and sparseness are relevant to the approximation accuracy. We have derived some guidelines for data reduction tools.

## References

- [1] D. Barbara and M. Sullivan, "Quasi-Cubes: A space-efficient way to support approximate multidimensional databases," Technical Report, ISE Dept., September 1997.
- [2] W. G. Cochran. Sampling Techniques. Wiley, New York, third edition, 1977.
- [3] C. Faloutsos, H. V. Jagadish, Nikolaos Sidiropoulos: Recovering Information from Summary Data. In VLDB'97, Proceedings of 23rd International Conference on Very Large Data Bases, August 25-29, 1997, Athens, Greece.
- [4] R. Kimball. The Datawarehouse Toolkit. John Wiley & Sons, 1996 ISBN 0-471-15337-0.
- [5] V. Poosala, Y. Ioannidis. Selectivity Estimation Without the Attribute Value Independence Assumption. Proceedings of the 23<sup>rd</sup> VLDB Conference, Athens, Greece, 1997.
- [6] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. Numerical Recipes in C, The Art of Scientific Computing. Cambridge University Press, Cambridge, MA, 1996.
- [7] C.-E. Sarndal, B. Swensson, and J. Wretman. Model Assisted Survey Sampling. Springer-Verlag, New York, 1992.
- [8] S. Sudman. Applied Sampling. Academic Press, New York, 1976.
- [9] Special issue on data reduction techniques of the bulletin of the Technical Committee on Data Engineering of the IEEE Computer Society, December 1997, Vol. 20, n 4.
- [10] Jeffrey Scott Vitter, Min Wang, B. Iyer, "Data Cube Approximation and Histograms via Wavelets". 7<sup>th</sup> International Conference on Information and Knowledge Management, Bethesda, Maryland, November 1998.
- [11] Yossi Matias, J. S. Vitter, Min Wang, "Wavelet-Based Histograms for Selectivity Estimation" in Proceedings of the 1998 International Conference on the Management of Data, Seattle, Washington, June 1-4 1998.
- [12] Y. Zhao, P. M. Deshpande, J. F. Naughton, " An Array-Based Algorithm for Simultaneous Multidimensional Aggregates" in Sigmod '97, AZ, USA.
- [13] G. K. Zipf. Human Behaviour and the principle of least effort. Addison-Wesley, Reading, MA, 1949.
- [14] T. Zhang, R. Ramakrishnan and M. Livny. (1996) BIRCH: an Efficient Data Clustering Method for Very Large Databases, Proc. 1996 SIGMOD, pp. 103-114.