Exploiting Duality in Summarization with Deterministic Guarantees*

Panagiotis Karras
University of Hong Kong
Pokfulam Road, Hong Kong
China
pkarras@cs.hku.hk

Dimitris Sacharidis
NTUA
Zographou, Athens
Greece
dsachar@dblab.ntua.gr

Nikos Mamoulis
University of Hong Kong
Pokfulam Road, Hong Kong
China
nikos@cs.hku.hk

ABSTRACT

Summarization is an important task in data mining. A major challenge over the past years has been the efficient construction of fixed-space synopses that provide a deterministic quality guarantee, often expressed in terms of a maximum-error metric. Histograms and several hierarchical techniques have been proposed for this problem. However, their time and/or space complexities remain impractically high and depend not only on the data set size n, but also on the space budget B. These handicaps stem from a requirement to tabulate all allocations of synopsis space to different regions of the data. In this paper we develop an alternative methodology that dispels these deficiencies, thanks to a fruitful application of the solution to the dual problem: given a maximum allowed error, determine the minimum-space synopsis that achieves it. Compared to the state-of-the-art, our histogram construction algorithm reduces time complexity by (at least) a $\frac{B \log^2 n}{\log \epsilon^*}$ factor and our hierarchical synopsis algorithm reduces the complexity by (at least) a factor of $\frac{\log^2 B}{\log \epsilon^* + \log n}$ in time and $B(1 - \frac{\log B}{\log n})$ in space, where ϵ^* is the optimal error. These complexity advantages offer both a space-efficiency and a scalability that previous approaches lacked. We verify the benefits of our approach in practice by experimentation.

Categories and Subject Descriptors

F.2 [**Analysis of Algorithms and Complexity**]: Miscellaneous; H.3 [**Information Storage and Retrieval**]: Miscellaneous; H.2.4 [**Database Management**]: Systems—*Query processing*

General Terms

Algorithms, Experimentation, Theory, Performance

Keywords

efficiency, histograms, synopses, wavelets

*Work supported by grant 7160/05E from Hong Kong RGC and by project PENED 2003 in the 3rd Community Support Programme.

KDD '07, August 12-15, 2007, San Jose, California, USA

1. INTRODUCTION

The need to reduce a very large data set into a compact representation or *synopsis* that captures its basic characteristics arises often; it finds application in OLAP/DSS systems [28], approximate query answering [26, 3], cost-based query optimization [24], time-series indexing [4], data mining [23], data stream approximation [2, 5] and the efficient handling of multi-measure [6] and multidimensional data sets [18]. Diverse synopsis data structures have been proposed [9]; the goal with all of them is to minimize an appropriate error metric over the original data in a given space budget. Past research has led the way from conventional synopsis techniques such as histograms [15, 17, 26, 14, 11, 2] and Haar wavelets [24, 28, 3, 7, 8, 20, 11, 12, 13, 6] to more sophisticated ones such as compact hierarchical histograms [27] and the Haar$^+$ tree [21].

A general optimal-histogram construction algorithm was presented by Jagadish et al. [17] and later specialized by Guha et al. [14] for the case of *maximum-error* metrics. The practical usefulness of this class of error metrics has spawned focused attention to them in recent studies, all based on *hierarchical* synopsis structures. A dynamic programming algorithm that derives the optimal Haar wavelet synopsis for a maximum-error metric (as opposed to the computationally easier Euclidean error) was developed by Garofalakis and Kumar [8] and optimized in terms of space and time by Guha [11]. Later, Guha and Harb [12, 13] improved on the robustness of the *restricted* Haar wavelet synopsis model of [8, 11]. In their approach, wavelet coefficient values in the synopsis are arbitrary, and differ from those in the wavelet transform of the data; their approximation scheme for *unrestricted* Haar wavelet synopses achieves both higher accuracy of approximation and better asymptotic behavior in time than the *restricted* model. Recent research has created less restrictive hierarchical synopsis data structures in two independent routes [27, 21]. The Haar$^+$ tree [21] goes further, in terms of flexibility, than the unrestricted Haar wavelet model, by enhancing the structure itself. The Compact Hierarchical Histogram (CHH) was independently introduced in [27]; as we observe in this paper, it is a special case of a Haar$^+$ tree. In other words, a Haar$^+$ structure merges the unrestricted Haar wavelet and the CHH models. Both [27] and [21] experimentally demonstrate that the structures they propose can, in certain circumstances, achieve higher quality of approximation than the optimal histogram of [17, 14].

Despite this progress, the complexities of all summarization algorithms for maximum-error metrics are still inefficiently high. The main reason for this defect is their dependence on the given synopsis space budget B, due to a requirement to tabulate possible allocations of space to different data intervals; the problem is gravest in the models of [12, 13, 21], due to their two-dimensional tabulation over both space and candidate approximation values. An effort to tame these space complexities [11] did not manage to erad-

icate their dependence on B; besides, as we show, it creates an unwieldy tradeoff between time- and space-efficiency in the case of maximum-error metrics. In this paper we eliminate these shortcomings with an alternative approach, based on a lucrative application of the solution to the *dual*, *error-bounded* problem: detect a space-optimal synopsis under an error bound. Our solutions do not tabulate over B and do not present performance tradeoffs. Compared to the state of the art, in histogram construction we reduce the time complexity from (at least) $O(nB\log^2 n)$ to $O(n\log\epsilon^*)$; in hierarchical synopsis construction, we reduce the complexity from (at least) $O(R^2 n\log^2 B)$ to $O(R^2 n(\log\epsilon^* + \log n))$ in time and from (at least) $O(RB\log\frac{n}{B})$ to $O(R\log n + n)$ in space, where ϵ^* is the optimal error and R the cardinality of an examined value set. We experimentally verify the practical implications of this reduction.

2. BACKGROUND AND RELATED WORK

In this section we briefly present previous approaches to offline data reduction with a maximum-error deterministic guarantee. We consider the principal synopsis structures employed, namely plain histograms and hierarchical representations. Under both approaches, given an n-size data vector $\mathbf{D} = \langle d_0, d_1, \ldots, d_{n-1}\rangle$, the problem is to devise an approximate representation $\hat{\mathbf{D}}$ of $\mathbf{D}$ using at most B space, so that a given error metric in the approximation is minimized. Maximum-error metrics are most generally expressed in their *weighted* version:

$$\mathcal{L}_\infty^w\left(\hat{\mathbf{D}}, \mathbf{D}\right) = \max_i\left\{w_i|\hat{d}_i - d_i|\right\},$$

where $\hat{d}_i$ denotes the reconstructed value for d_i and w_i denotes a weight for the corresponding error value; in the case of the *maximum relative error* (MRE), it is $w_i = \frac{1}{\max\{|d_i|,S\}}$, where $S > 0$ is a sanity bound that prevents small values from unnaturally dominating the error result [8]. In the case that $\forall i, w_i = 1$, the error metric at hand is the *maximum absolute error* (MAE). Previous studies [17, 8, 12, 13, 27, 21] have generalized their results into wider classes of *distributive* and *Minkowski-distance* metrics. Still, the sub-class of *maximum-error* metrics remains more practically interesting than the esoteric metrics of those classes [7].

2.1 Histogram-based Data Reduction

A *histogram synopsis* (also called segmentation or partitioning) divides $\mathbf{D}$ into $B \ll n$ successive disjoint intervals $[b_i, e_i]$, $1 \le i \le B$ called *buckets* or *segments*, and attributes a single value v_i to each of them that approximates all consecutive values therein, d_j, $j \in [b_i, e_i]$. A single bucket (segment) can be expressed by the triad $s_i = \{b_i, e_i, v_i\}$. Given a target error metric, the best value for v_i is defined as a function of the data values in $[b_i, e_i]$.[1] $2B-1$ numbers suffice to represent a B-bucket histogram (since $\forall i, 1 < i \le B, b_i = e_{i-1}+1$ and the edges are fixed). Initial work on histograms focused on heuristics [16]. An $O(n^2B)$ dynamic programming algorithm that assigns optimal bucket boundaries for the Euclidean ($\mathcal{L}_2$) error metric ($O(n^3B)$ for other metrics) was presented[2] in [17]. The basic idea behind it is that the b-optimal histogram for $\mathbf{D}$ can be recursively derived from the space of (b–1)-optimal partitionings of prefix vectors of $\mathbf{D}$. For a maximum-error metric, the minimal error $E(i,b)$ of a b-bucket histogram of the prefix vector $\langle d_0, d_1, \ldots, d_i\rangle$ is recursively expressed as:

$$E(i,b) = \min_{1\le j<i}\{\max\{E(j,b-1), \mathcal{E}(j+1,i)\}\} \qquad (1)$$

where $\mathcal{E}(j+1,i)$ measures the minimal maximum error for a bucket that contains the items $\langle d_{j+1}, \ldots, d_i\rangle$. The resulting algorithm requires an $O(nB)$ tabulation of minimized error values $E(i,b)$ and chosen last-bucket boundaries j corresponding to those optimal error values. Guha et al. [14] proposed a specialization of the general-purpose algorithm of [17] for (among others) MRE (applicable to any maximum-error metric). The crucial observation is that, in order to determine the j that minimizes the max function in Equation 1, it suffices to perform a binary search, since $E(j,b-1)$ and $\mathcal{E}(j+1,i)$ are monotonic functions of j. [14] employs an interval tree to determine the minimum error for a bucket in logarithmic time. Hence this algorithm requires $O(nB\log^2 n)$ time and $O(nB)$ space. Table 1 summarizes the complexity results of previous work on the offline one-dimensional histogram construction problem for maximum-error metrics and introduces the complexity of the solution[3] we propose; B is the space-bound expressed as the number of buckets and ϵ^* is the optimal error. The space-efficient variant of the algorithm in [14] is discussed in Section 3.

Reference	Time	Space	Algorithm Type
[17]	$O(n^3B)$	$O(nB)$	
[14]	$O(nB\log^2 n)$	$O(nB)$	time efficient
[14, 11]	$O(nB\log^3 n)$	$O(n)$	space efficient
This work	$O(n\log\epsilon^*)$	$O(n)$	

Table 1: Summary of results for *optimal* offline one-dimensional histogram construction (maximum-error metrics)

2.2 Hierarchical Data Reduction

Another stream of research has been based on index structures that represent the data in consecutive hierarchical levels of detail. This approach started with the application of the Haar wavelet decomposition, long used in signal processing [19]. Most recently, two independent, yet interrelated structures employing a hierarchy have been introduced [21, 27]. We now review this research.

The Haar Wavelet Hierarchy.

The Haar wavelet hierarchy can be visualized through a complete binary tree, the *Haar tree*. The coefficient in the Haar tree root node contains the overall average value and each other coefficient value c_i contributes the value $+c_i$ to all data values (leaves) in its *left* sub-tree and $-c_i$ to those in its *right* sub-tree. Hence each original data value is reconstructed by adding/subtracting the coefficients in the path towards its position. Figure 1a depicts the Haar decomposition of an example data vector $\mathbf{D}$ of 8 values (shown at the leaves of the tree). Value $d_3 = -6$ can be reconstructed as $+c_0 + c_1 - c_2 + c_5$. A *Haar wavelet synopsis* of $\mathbf{D}$ is a vector $\hat{\mathbf{Z}}$ of $B \ll n$ non-zero $\langle i, c_i\rangle$ terms, such that its inverse wavelet transform $\hat{\mathbf{D}} = \mathcal{W}^{-1}(\hat{\mathbf{Z}})$ approximates the data vector $\mathbf{D}$. Figure 1b shows a $\{\langle 0,4\rangle, \langle 3,-2\rangle\ \langle 4,6\rangle\ \langle 5,-7\rangle\}$ synopsis for the data array of Figure 1a, with maximum absolute error 4. This is the optimal MAE synopsis with $B = 4$. For the Euclidean error ($\mathcal{L}_2$), the optimal Haar wavelet synopsis consists of the top-B *normalized* coefficients of the complete Haar wavelet transform [19]; the normalized value of a coefficient c is $\frac{|c|}{\sqrt{2^\ell}}$, where ℓ is the level where c resides in the Haar tree. For example, the $\mathcal{L}_2$-optimal synopsis, with $B = 2$, for the data vector in Figure 1a is $\{\langle 0,4\rangle, \langle 5,-7\rangle\}$. This

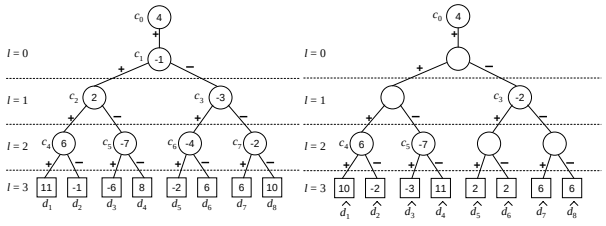
[1] For the Euclidean error, the optimal v_i is the mean of the values in $[b_i, e_i]$ [17]; for MAE it is the mean of maximum and minimum values in $[b_i, e_i]$, while for MRE a case analysis is given in [14].

[2] Since this problem is a special case of the problem of approximating a curve by line segments, the solution of [17] is a special case of the algorithm introduced in [1].

[3] After this work was submitted for publication, [2] proposed an $O(n + n\frac{\log U}{\log\frac{n}{B}})$ algorithm for offline histogram summarization, where U is the size of the domain for data values; as U can be arbitrarily large, our solution retains its competitiveness towards that algorithm too.

computational convenience has allowed for the extension of the $\mathcal{L}_2$-synopsis methodology to various settings [10, 18, 5, 6]. On the other hand, the problem is computationally harder for *maximum-error* metrics.

Restricted Synopses for Maximum-Error Metrics. After its identification in [24], the first systematic treatment of the *space-bounded* Haar wavelet synopsis problem for maximum-error metrics was based on a randomized rounding scheme [7]. However, as shown in [14] and [8], this scheme does not produce results of high quality. Garofalakis and Kumar [8] suggested a dynamic programming (DP) scheme that deterministically retains the optimal coefficient subset of a dataset's Haar wavelet transform. [20] proposed a streaming-capable and reliable greedy counterpart to this solution. Muthukrishnan [25] suggested that an algorithm solving the dual, *error-bounded* problem[4] can provide a shortcut to the solution of the space-bounded problem, gaining a $\frac{\log n}{\log \epsilon^*}$-factor time complexity advantage. Still, these solutions are all confined to the *restricted* variant of problem, in which a coefficient may be only assigned a fixed value in the complete Haar tree (candidate assigned values are also fixed in advance in the low-quality probabilistic model).



(a) Haar tree ($n = 8$) (b) Unrestricted synopsis

Figure 1: A Haar tree and unrestricted synopsis ($n = 8$)

Unrestricted Synopses for Maximum-Error Metrics. Guha and Harb [12, 13] discerned that the values assigned to the coefficients retained in a wavelet synopsis can be arbitrary and provided a fully polynomial-time approximation scheme (FPAS) for the resulting *unrestricted* space-bounded Haar wavelet synopsis problem. The solution of [12, 13] is a DP algorithm guided by a two-dimensional tabulation per Haar tree node. Each node c_i calculates the minimum attainable error $E(i, v, b)$ over *both* every possible incoming value[5] v *and* every possible amount of space b allocated to the subtree rooted at c_i; possible incoming values are discretized by a resolution step δ. For each $E(i, v, b)$ entry, both the δ-optimal assigned value z (also quantized as a multiple of δ) and the δ-optimal distribution of b units of space among the left i_L and right i_R subtrees of c_i are detected. This DP recursion can be summarized as:

$$E(i,v,b) = \min\left\{ \begin{array}{c} \min\limits_{0\le b'\le b}\left\{\max\left\{\begin{array}{c} E(i_L,v,b'),\\ E(i_R,v,b-b')\end{array}\right\}\right\},\\ \min\limits_{z,0\le b'\le b-1}\left\{\max\left\{\begin{array}{c} E(i_L,v+z,b'),\\ E(i_R,v-z,b-1-b')\end{array}\right\}\right\}\end{array}\right.$$

Computing $E(0, 0, B)$ determines the best B nodes to keep in the synopsis and the best values z to be assigned to each of these nodes for a given value of δ. The ranges of incoming values v and assigned values z to be tested per node can be restricted using the maximum absolute value M in $\mathbf{D}$ [12], or, more efficiently, by a guessed upper-bound $\mathcal{E}$ for the target minimized error [13]. In both cases, the resulting cardinality $R = O(\frac{M}{\delta})$ or $R = O(\frac{\mathcal{E}}{\delta})$ of the set of examined values enters the complexity expressions.

[4]That is, find a minimal-space synopsis achieving error bound ϵ.

[5]The incoming value of a node c_i is the value constructed by the path from the root of the sparse Haar tree up to c_i. For example, the incoming value of node c_7 in the tree of Figure 1b is $c_0 - c_3 = 6$.

The Haar$^+$ Tree The Haar$^+$ tree [21] extends the Haar wavelet hierarchy by allowing extra coefficient values which contribute their (signed) value to a single dyadic interval alone. In the example Haar$^+$ tree of Figure 2, node c_0 (root coefficient) contributes its value to all approximated data values $\{d_0, d_1, d_2, d_3\}$. The root is followed by a binary tree of triads (C_1, C_2 and C_3), which substitute the single non-root coefficients of the classical Haar tree. In each such triad (e.g., C_1), the *head coefficient* (e.g., c_1) contributes its value positively to its left sub-tree and the same value negatively to its right sub-tree. The left (e.g., c_2) and right (e.g., c_3) *supplementary coefficients* contribute their values positively only in the single subinterval that they affect (e.g., c_2 contributes positively to d_0 and d_1 only). An *optimal* synopsis of space budget B for a given error metric $\mathcal{E}$ places B non-zero coefficient values at any positions in the Haar$^+$ tree so that $\mathcal{E}$ is minimized. For example, for the four-element data set $\{5, 3, 12, 4\}$ the 2-term Haar$^+$ synopsis that minimizes the MAE consists of the coefficients $\{c_0 = 4, c_8 = 8\}$. The Haar$^+$ structure outperforms its predecessors in both accuracy of approximation and synopsis construction time [21].

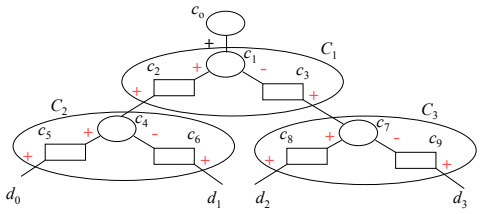


Figure 2: An One-Dimensional Haar$^+$ Tree

Compact Hierarchical Histograms.
The Compact Hierarchical Histogram (CHH) [27] is a related data approximation structure, which defines a (binary by default) hierarchy of (dyadic) intervals and selects an optimal subset of nodes to represent the approximated data set. In fact, the CHH structure is equivalent to a Haar$^+$ tree, in which only supplementary coefficients are allowed. With the benefit of hindsight, a Haar$^+$ tree can be seen as a merging of a CHH and a Haar tree. [27] proposed heuristic CHH construction techniques, after observing that the calculation of the optimal value to retain on a node is computationally hard, due to the interdependence between nodes in the hierarchy. On the other hand, [21] eschews this problem by an approximation technique, similar to that in [12, 13], which provably approximates the theoretically optimal solution by a small margin of error. Hence, the Haar$^+$ technique can achieve at least as high accuracy as an heuristically derived CHH over a binary hierarchy due to both its structural and algorithmic advantages.

2.3 A Space-Efficiency Technique

Guha [11] identified space as the most significant resource for an offline summarization problem and furnished a space-efficiency paradigm for synopsis construction. His main idea is to avoid storing all tabulated results throughout the DP; part of them can be dropped and re-computed later. In histogram construction, the tabulation (Equation 1) on $\{i, b\}$ should progress with increasing b, $1 \le b \le B$ (i.e., the loop of b is the outer loop). Since the values $E(*, b)$ are fully determined by $E(*, b-1)$, after a b-column has been used to calculate the $(b+1)$-column, it is dropped. Hence the space is $O(n)$. Besides, the tabulation also detects and stores the single bucket $M(i, b)$ in the optimal b-partitioning of $\langle d_0, d_1, \ldots, d_i\rangle$ that contains the *middle* data item $\lfloor \frac{n}{2} \rfloor$ of the summarized vector. After the optimal error $E(n, B)$ and middle-item bucket $\mathcal{M} = M(n, B)$ have been established, the two $O(\frac{n}{2})$ independent sub-

Reference	Time	Space	Synopsis Model
[7]	$O(n\mathtt{q}^2 B\log(\mathtt{q}B))$	$O(n+\mathtt{q}B\log^2 n)$	probabilistic restricted Haar
[8]	$O(n^2 B\log B)$	$O(n^2 B)$	optimal restricted Haar
[20]	$O(n\log^3 n)$	$O(n\log n)$	greedy restricted Haar
[11]	$O(n^2)$	$O(n)$	optimal restricted Haar
[25]	$O(n^2\frac{\log\epsilon^*}{\log n})$	$O(n)$	optimal restricted Haar
[27]	$O(nB\log n\log B)$	$O(nB\log n)$	Compact Hierarchical Histogram (time efficient)
[27]	$O(nB\log^2 n\log B)$	$O(B\log^2 n+n)$	Compact Hierarchical Histogram (space efficient)
[12, 13, 21]	$O(R^2 n\log^2 B)$	$O(R\min\{B^2\log\frac{n}{B}, n\log B\})$	unrestricted Haar and Haar$^+$ (time efficient)
[12, 13, 21]	$O(R^2 n\log n\log^2 B)$	$O(RB\log\frac{n}{B}+n)$	unrestricted Haar and Haar$^+$ (space efficient)
This work	$O(R^2 n(\log\epsilon^*+\log n))$	$O(R\log n+n)$	unrestricted Haar and Haar$^+$

Table 2: Summary of results for offline one-dimensional hierarchical synopsis construction (maximum-error metrics)

problems for the intervals on the left and right of $\mathcal{M}$ are re-solved recursively. Hence, the total time for the general-error histogram construction algorithm [17] becomes $O\left(\sum_{\ell=1}^{\log n} 2^\ell \left(\frac{n}{2^\ell}\right)^2 B\right) = O(n^2 B)$, i.e., the re-computation cost is amortized. [11] applies the same methodology to the restricted Haar wavelet synopsis algorithm of [8]. In this case, the required tabulation progresses in a bottom-up fashion in the Haar tree; all table entries on a parent node are computed from the tables of its children nodes, which can then be dropped. Accordingly, at most $\log n + 1$ tables need be concurrently stored, covering one path through the Haar tree. After the solution is established at the top level of the Haar tree, the two half-size sub-problems in the two sub-trees of c_1 are re-solved [11]. Restricted Haar wavelet synopsis construction requires time quadratic to n, because each of n Haar tree nodes has to consider $O(2^{\log n}) = O(n)$ possible choices of values in its ancestor-set [8]. Hence, the re-computation cost is amortized in this case as well. Table 2 summarizes the complexity results of previous work on the offline one-dimensional space-bounded hierarchical synopsis construction problem for maximum-error metrics and introduces the complexity of the solution we propose; $\mathtt{q}$ is a probability quantization parameter, R is the cardinality of the examined set of incoming or assigned values per coefficient, and ϵ^* is the optimal error. We explain the space- and time-efficient variants in the sequel.

3. MOTIVATION

The state-of-the-art for all examined methods features a demanding tabulation over space allocations [14, 12, 13, 27, 21]. Guha strived to tame these demands [11]; the result was good, but not sufficient: the burden of space tabulation remains. This burden is heaviest for the unrestricted Haar and Haar$^+$ methods: their two-dimensional tabulation renders their memory requirement impractical for large data sizes. Besides, the amortization achieved by the paradigm of [11] does *not* hold for the algorithms of time *linear* (or near-linear) to n reviewed in Section 2. Applied on them, the paradigm creates a tradeoff between space- and time-efficiency, as [21] presented for the Haar$^+$ case. Hence, applied on the MRE algorithm of Section 2.1, this technique decreases its space requirements to $O(n)$, but increases its time complexity to $O(nB\log^3 n)$ (Table 1). The same holds for the *unrestricted* Haar and the Haar$^+$ cases of Section 2.2 (Table 2). In the space-efficient variant, after the arrays $E(i_L,*,*)$ and $E(i_R,*,*)$ have been used to calculate the entries of $E(i,*,*)$, they are dropped. Again, at most $\log n + 1$ arrays need to be concurrently stored. The price for this space-efficiency is an extra $\log n$ time complexity factor due to re-computation. For the time-efficient variant two different approaches are possible: If $B \gg \sqrt{n}$, then it is advantageous to maintain all $E(i,*,*)$ arrays in memory. The size of the array at node c_i, residing in level ℓ_i in the tree, is $O\left(R\min\{B, 2^{\ell_i}\}\right)$, which, after summation, gives a space complexity of $O\left(Rn\log B\right)$. Still, if $B \ll \sqrt{n}$, then it is preferable to keep only the at most $\log n+1$ necessary arrays, with the full solutions corresponding to each of their entries appended on them as lists, as suggested in [13]. The size of a solution maintained with each entry of an array at level ℓ_i is at most $\min\{B, 2^{\ell_i}\}$, therefore the space required for an array at level ℓ_i is $O\left(R(\min\{B, 2^{\ell_i}\})^2\right)$. This sums up to a space complexity of $O\left(RB^2\log\frac{n}{B}\right)$. The two expressions are equal when $n\log B = B^2\log(\frac{n}{B}) \Leftrightarrow B = \sqrt{n}$. Values of B both higher and lower than $\sqrt{n}$ are likely to occur, thus the preferable method depends on the application at hand. Table 2 shows both. A similar performance tradeoff applies to the winner greedy heuristic of [27] (Table 2). Overall, the complexity question on summarization with deterministic guarantees remains unsatisfactorily resolved. In this paper, we provide an alternative methodology that addresses[6] these shortcomings. We show how the *space-bounded* summarization problems can be solved more efficiently by exploiting their duality to the corresponding *error-bounded* problems through binary search; in those dual problems, the goal is to minimize the space of a synopsis that achieves error no larger than an error bound ϵ. In the sequel, we formulate and solve the error-bounded histogram and hierarchical synopses problems. Then we define and analyze *Indirect* synopsis construction algorithms for the corresponding space-bounded problems.

4. INDIRECT HISTOGRAMS

This section introduces our solution to the space-bounded histogram construction problem for weighted maximum-error metrics. Our technique utilizes the solution to the complementary error-bounded problem. Section 4.1 presents a linear algorithm[7] for this auxiliary problem, which achieves the minimal space B^* under an error bound ϵ; we discuss how the solution can be tested on whether it achieves, secondarily, the minimal error ϵ^* in the required space B^*, providing a *strong* optimization. In Section 4.2 we exploit this solution in order to efficiently solve the space-bounded histogram construction problem, which is our main interest and contribution.

4.1 Error-bounded Histogram Construction

We formulate the $\mathcal{L}_\infty^w$-bounded histogram construction problem:

Problem 1 *Given a data vector* $\mathbf{D}$ *and an* $\mathcal{L}_\infty^w$*-error bound* ϵ*, construct a histogram* $\mathbf{H}$ *of* $\mathbf{D}$ *with the minimum number of buckets* B^**, such that* $\mathcal{L}_\infty^w(\mathbf{D},\mathbf{H}) \leq \epsilon$.

[6]The basic idea behind this methodology was applied for the restricted Haar wavelet synopsis problem in [25], but yielded only a marginal benefit (see Table 2) that did not reveal its full potential.

[7]A similar algorithm was proposed in [22] for the effective summarization of data streams, albeit it treated the MAE metric only.

Our algorithm for this problem establishes a minimal-space histogram $\mathbf{H}$ of B^* buckets that satisfies ϵ in one linear pass, drawing from the following Lemma.

LEMMA 1. *Let B^* be the minimum number of buckets required to satisfy the bound ϵ for data vector $\mathbf{D}$ and $\mathbf{H} = \{\{b_i, e_i, v_i\}\}$, $1 \leq i \leq B^*$ be a B^*-bucket histogram such that the achieved error is $\mathcal{L}_\infty^w(\mathbf{H}, \mathbf{D}) \leq \epsilon$. Furthermore, let $\mathcal{L}_{\infty,i}^w$ be the error in bucket (segment) $s_i = \{b_i, e_i, v_i\} \in \mathbf{H}$, $i < B^*$. Then, if after we advance the right bucket boundary e_i by one position, so that the bucket becomes $\tilde{s}_i = \{b_i, e_i + 1, \tilde{v}_i\}$, the new bucket error value remains $\tilde{\mathcal{L}}_{\infty,i}^w \leq \epsilon$, then the new segmentation $\tilde{\mathbf{H}}$ as a whole also achieves the error bound $\mathcal{L}_\infty^w\left(\tilde{\mathbf{H}}, \mathbf{D}\right) \leq \epsilon$.*

Based on Lemma 1, the MinHistSpace algorithm of Figure 3 performs a linear scan of the data. During this scan, it extends the right boundary of the running segment s_i as long as $\mathcal{L}_{\infty,i}^w \leq \epsilon$. An encountered data item d_i with error weight w_i defines a tolerance interval $[d_i - \frac{\epsilon}{w_i}, d_i + \frac{\epsilon}{w_i}]$; bucket values within this interval satisfy ϵ for d_i. The algorithm only needs to calculate the intersection $\mathcal{I}$ of such intervals for arriving data items. When $\mathcal{I}$ becomes null, s_i cannot be magnified any more. Then a new bucket boundary is inserted before the last read data item. The value assigned to the formed bucket is defined as $v = \frac{w_j d_j + w_k d_k}{w_j + w_k}$, where d_j, d_k are the data items responsible for the limits[8] a, b of the last non-null value of $\mathcal{I} = [a, b]$. Hence MinHistSpace needs $O(n)$ time and space.

As an example, assume that we want to find a histogram with $\mathcal{L}_\infty$ error at most $\epsilon = 5$ approximating the data vector $\mathbf{D} = \{11, -1, -6, 8, -2, 6, 6, 10\}$. MinHistSpace scans $\mathbf{D}$ and computes $\mathbf{H}$ incrementally. $d_0 = 11$ combined with $d_1 = -1$ violate the ϵ bound (the absolute error of such a bucket is $\frac{11-(-1)}{2} = 6$). Thus, the first bucket has $b_1 = e_1 = 0$ and value $v_1 = 11$. The algorithm continues by putting d_1 in the next bucket which is terminated when $d_3 = 8$ is found (d_3 violates ϵ). Continuing this way, MinHistSpace eventually computes the histogram $\mathbf{H} = \{\{0, 0, 11\}, \{1, 2, -3.5\}, \{3, 6, 3\}, \{7, 7, 10\}\}$.

Algorithm MinHistSpace(ϵ)
Input: error bound ϵ, n-data vector $[d_0, \ldots, d_{n-1}]$
Output: histogram partitioning $\mathbf{H}$ that satisfies ϵ
1. $i = 0$; $r = 1$;
2. **while** $(i < n)$
3. read d_i;
4. compute v_r, $\mathcal{L}_{\infty,r}^w$ for the r-th bucket from data read so far;
5. **if** $\left(\mathcal{L}_{\infty,r}^w > \epsilon\right)$;
7. $e_r = i - 1$; $v_r = prev$; $r := r + 1$; // fix r-th bucket
8. re-compute v_r from d_i; //initialize new bucket
9. $prev = v_r$; $i := i + 1$;
10. $e_r = n$; // fix last bucket
11. **return** created partitioning $\mathbf{H}$;

Figure 3: Minimum Space Histogram construction algorithm

We now prove that MinHistSpace achieves space-optimality.

THEOREM 1. *The histogram $\mathbf{H}$ returned by* MinHistSpace *has achieved the minimal space B^* subject to the $\mathcal{L}_\infty^w$-error bound ϵ.*

PROOF. Let B be the number of buckets in $\mathbf{H}$. Assume there exists a histogram segmentation of $\mathbf{D}$ in $B' < B$ segments $\mathbf{H}' = \{\{b_i', e_i', v_i'\}\}$, $1 \leq i \leq B'$, such that $\mathcal{L}_\infty^w(\mathbf{H}', \mathbf{D}) < \epsilon$. Then there will be at least one segment $\{b_i', e_i', v_i'\} \in \mathbf{H}'$ such that $e_i' > e_i^1$, where e_i^1 is the right boundary of the i-th segment in $\mathbf{H}$, otherwise $\mathbf{H}'$ would not have less segments than $\mathbf{H}$. Let $\{s_i', e_i', v_i'\} \in \mathbf{H}'$ be the first such segment encountered from left; then $s_i' = s_i^1$, hence $[s_i^1, e_i^1] \subset [s_i', e_i']$. Since $[s_i', e_i']$ satisfies ϵ, its subdivision $[s_i^1, e_i^1 + 1]$ can also satisfy this bound as a bucket. However, if algorithm MinHistSpace has fixed the i-th segment as $[s_i^1, e_i^1]$, then the interval $[s_i^1, e_i^1 + 1]$ could not make a segment satisfying ϵ. By reductio ad absurdum, it follows that there is no histogram segmentation $\mathbf{H}'$ as we assumed. Hence the B is the ϵ-optimal space B^*. □

[8]The item d_i most distant from the limit at hand, hence of smallest weight w_i, is chosen in case more than one items are responsible for the same limit.

The following lemma defines an error-optimality test for the histogram returned by MinHistSpace. Given the result $\mathbf{H}$ of an execution of MinHistSpace, the objective of the test is to determine, with one more call of MinHistSpace (i.e., in linear time), whether the actual $\mathcal{L}_\infty^w$-error of $\mathbf{H}$ is the minimum possible for the space B^* occupied by $\mathbf{H}$.

LEMMA 2. *Let $\mathbf{H}$ be the B^*-bucket histogram segmentation of $\mathbf{D}$ for error bound ϵ returned by* MinHistSpace *and $\bar{\epsilon} \leq \epsilon$ be the actual $\mathcal{L}_\infty^w$-error of $\mathbf{H}$. Let $\tilde{\mathbf{H}}$ be the $\tilde{B}$-bucket histogram segmentation of $\mathbf{D}$ returned by* MinHistSpace *running under the constraint $\mathcal{L}_{\infty,r}^w < \bar{\epsilon}$, allowing error values less than but not equal to $\bar{\epsilon}$. Let ϵ^* be the minimum $\mathcal{L}_\infty^w$-error of a histogram segmentation of $\mathbf{D}$ in B^* buckets. Then $\bar{\epsilon} = \epsilon^*$ if and only if $\tilde{B} > B^*$.*

PROOF. B^* is the least number of buckets required to satisfy error bound $\epsilon \geq \bar{\epsilon}$, hence $\tilde{B} \geq B^*$. If $\tilde{B} = B^*$, then there exists a B^*-bucket histogram partitioning of $\mathbf{D}$ with $\mathcal{L}_\infty^w$-error less than $\bar{\epsilon}$, hence $\mathbf{H}$ *has not* achieved the optimal error ϵ^* in B^* buckets. Therefore $\bar{\epsilon} = \epsilon^* \Rightarrow \tilde{B} > B^*$. In reverse, if $\tilde{B} > B^*$ then any histogram partitioning of $\mathbf{D}$ with $\mathcal{L}_\infty^w$-error less than $\bar{\epsilon}$ requires more than B^* buckets, hence $\mathbf{H}$ *has* achieved error optimality. Thus, $\tilde{B} > B^* \Rightarrow \bar{\epsilon} = \epsilon^*$. In conclusion, $\bar{\epsilon} = \epsilon^* \Leftrightarrow \tilde{B} > B^*$. □

Algorithm IndirectHist(B)
Input: space bound B, n-data vector $[d_0, \ldots, d_{n-1}]$
Output: $\mathcal{L}_\infty^w$-error optimal histogram partitioning $\mathbf{H}$
1. $\epsilon_u = \mathcal{L}_\infty^w$-error of equi-width B-histogram;
2. $e_{low} = 0$; $e_{high} = \epsilon_u$;
3. **while** not finished
4. $e_{mid} = (e_{high} + e_{low})/2$;
5. $\mathbf{H}$=MinHistSpace(e_{mid}); $\bar{B}$ = size of $\mathbf{H}$;
6. $\bar{\epsilon}$ = actual $\mathcal{L}_\infty^w$-error of $\mathbf{H}$; /* $\bar{\epsilon} \leq \epsilon$ */
7. **if** $(\bar{B} \leq B)$
8. $\tilde{\mathbf{H}}$=MinHistSpace($< \bar{\epsilon}$); $\tilde{B}$ = size of $\tilde{\mathbf{H}}$;
9. **if** $(\tilde{B} > B)$
10. finished := 1; /* optimal result found */
11. **else** $e_{high} = \bar{\epsilon}$;
12. **else if** $(\bar{B} > B)$ $e_{low} = e_{mid}$
13. **return** $\mathbf{H}$;

Figure 4: Indirect histogram construction algorithm

4.2 Application to the Space-Bounded Problem

We now define an efficient algorithm for space-bounded histogram construction under a maximum-error metric that exploits the solution to the dual error-bounded problem. Formally, given a data vector $\mathbf{D}$ and a space bound B, we seek a histogram with at most B buckets that has minimal $\mathcal{L}_\infty^w$-error. The crucial observation is that the $\mathcal{L}_\infty^w$ -error of the optimal B-histogram is monotonically non-decreasing with B. Therefore we can apply binary search with guesses of ϵ in the space of error-bounded problems. This idea is materialized by our IndirectHist algorithm shown in Figure 4. In our implementation, the seed value of ϵ is obtained by linearly measuring the $\mathcal{L}_\infty^w$-error of an equi-width B-bucket histogram of $\mathbf{D}$, which provides an upper bound for the B-optimal $\mathcal{L}_\infty^w$-error. Thereafter, the MinHistSpace procedure is repeatedly

invoked with binary search on the error bound value ϵ; it performs an *optimality test*, as defined in Lemma 2, for each guessed error bound value ϵ that does not require more than B space. The search terminates when the guessed error bound reaches a value that requires a histogram of $\bar{B} \leq B$ space and actual error $\bar{\epsilon}$, while the optimality test indicates than any error bound $\epsilon < \bar{\epsilon}$ requires $\tilde{B} > B$ space; then an optimal histogram of minimum error $\epsilon^* = \bar{\epsilon}$ in the space budget B has been created. At line 8 of Figure 4, the call MinHistSpace($< \bar{\epsilon}$) corresponds to a variation of MinHistSpace($\bar{\epsilon}$), in which the condition at line 5 of Figure 3 is replaced by ($\mathcal{L}^w_{\infty,r} \geq \bar{\epsilon}$). This search process brings an $O(\log \epsilon^*)$ runtime factor[9], hence the time complexity of the Indirect algorithm is $O(n \log \epsilon^*)$. Section 6.2 verifies the time advantage of this algorithm in practice.

5. INDIRECT SPACE-BOUNDED HIERARCHICAL SYNOPSES

In this section we introduce our solution to the space-bounded hierarchical synopsis problem for maximum-error metrics. We study both the unrestricted Haar and Haar$^+$ models. Again our technique exploits the solution to the dual error-bounded problem.

5.1 Error-bounded Hierarchical Synopses

We formulate a *strong* version of the $\mathcal{L}^w_\infty$-bounded hierarchical synopsis problem as follows:

Problem 2 *Given a data vector $\mathbf{D}$ and an error bound ϵ, construct a representation $\hat{\mathbf{Z}}$ of $\mathbf{D}$, producing a reconstruction $\hat{\mathbf{D}}$, such that $\mathcal{L}^w_\infty\left(\mathbf{D}, \hat{\mathbf{D}}\right) \leq \epsilon$ and the number of non-zero entries s^* in $\hat{\mathbf{Z}}$ is minimized. Of all representations with s^* non-zero terms satisfying ϵ, select the one with the minimal actual error $\epsilon^* \leq \epsilon$.*

An *incoming* value at node c_i of the Haar tree (or triad C_i of the Haar$^+$ structure) is a value reconstructed in the path of ancestor coefficients from the root node up to c_i in the sparse representation $\hat{\mathbf{Z}}$ of $\mathbf{D}$. In a wavelet decomposition $\mathcal{W}(\mathbf{D})$, this is the average value in the interval I under the scope of c_i, henceforward called *real incoming value* at c_i. Similarly, an *assigned* value at node c_i is a coefficient value retained at that node in $\hat{\mathbf{Z}}$; in $\mathcal{W}(\mathbf{D})$, this is the actual semi-difference of the average values in the two sub-intervals I_L, I_R under the scope of c_i, henceforward called *real assigned* value. For example, in Figure 1a, the real incoming value of node c_6 is 2, while the incoming value constructed for this node in the synopsis of Figure 1b is also 2. On the other hand, the incoming value of node c_3 in Figure 1b is 4, whereas the corresponding real incoming value is 5 (see Figure 1a). Similarly, the real assigned value to node c_3 is -3, whereas the value assigned to this node in the synopsis -2. These concepts are directly extended to the Haar$^+$ tree [21]. In order to construct our solution, we need to explore the space of possible retained coefficients and values assigned to them. We use a dynamic-programming (DP) framework, as in previous hierarchical synopsis algorithms [6, 7, 8, 11, 25, 12, 13, 21]. In a bottom-up process, this algorithm considers all possible incoming values v and, for each v, all possible assigned values z_i^v at each node c_i of the Haar tree and determines the optimal value to assign at c_i for v; in a Haar$^+$ tree, possible head and left/right supplementary coefficients, z_h, z_l, z_r, on a triad C_i are all examined. We quantize the (real-valued) domains of v and z_i^v into multiples of a small resolution step δ. The next section outlines some lemmata that establish upper and lower bounds for these domains.

[9]The log function expresses the dependence of running time on the derived error value; it is to be understood as a *growth* function, as in [25]; the case $\epsilon^* \leq 1$ does *not* imply non-positive time.

5.1.1 Delimiting the Value Domains

Haar Wavelet Synopses.

We study the simple Haar wavelet case first. As we will see, despite its disadvantage in accuracy and, for non-maximum error metrics, complexity, in relation to the Haar$^+$ tree, the classical Haar wavelet structure has an advantage in its potential for delimitation of search space for maximum error metrics.

LEMMA 3. *Let v_i be the* real incoming value *at node c_i. Let v be an incoming value to c_i for which the error bound ϵ under the $\mathcal{L}^w_\infty$ metric can be satisfied, and $\bar{\epsilon} = \frac{\epsilon}{\min_{j \in I}\{|w_j|\}}$, where I is the interval under the scope of node c_i; then $|v_i - v| \leq \bar{\epsilon}$.*

Lemma 3 implies that the finite set $\mathcal{S}_i \subset \mathbb{R}$ of possible incoming values we have to examine at node c_i consists of the multiples of δ in the interval $[v_i - \bar{\epsilon}, v_i + \bar{\epsilon}]$; thus, $|\mathcal{S}_i| \leq \lfloor \frac{2\bar{\epsilon}}{\delta} \rfloor + 1 = O(\frac{\epsilon}{\delta})$. [10] We now demarcate the assigned values.

LEMMA 4. *Let v_i be the real incoming value to node c_i, z_i the real assigned value at c_i, $v \in \mathcal{S}_i$ be a possible incoming value to c_i for which the maximum error bound ϵ can be satisfied, and z_i^v be a value that can be assigned at c_i for incoming value v, satisfying ϵ; then $|z_i - z_i^v| \leq \bar{\epsilon} - |v_i - v|$.*

Lemma 4 implies that the finite set $\mathcal{S}_i^v \subset \mathbb{R}$ of possible assigned values we have to examine at node c_i, for a given incoming value $v \in \mathcal{S}_i$, consists of the multiples of δ in the interval $[z_i - (\bar{\epsilon} - |v_i - v|), z_i + (\bar{\epsilon} - |v_i - v|)]$; hence, $|\mathcal{S}_i^v| \leq \lfloor \frac{2(\bar{\epsilon} - |v_i - v|)}{\delta} \rfloor + 1 = O(\frac{\epsilon}{\delta})$. Lemmata 3 and 4 are most simple in the case of the maximum absolute error metric, when $\forall i, w_i = 1$; in the case of the maximum relative error metric, $\bar{\epsilon} = \epsilon \cdot \max\{S, \max_{j \in I}\{|d_j|\}\}$, where S is the sanity bound. Naturally, the same lemmata hold with any upper bound $\mathcal{E}$ for the optimal $\mathcal{L}^w_\infty$ error of a synopsis, even when that error is not known in advance. This observation will be useful in our implementation of the direct solution to the space-bounded problem (Section 6.3).

Haar$^+$ Synopses.

Delimitation lemmata analogous to Lemmata 3 and 4 also apply to the Haar$^+$ structure. [21] shows how the flexibility of this structure enables an *equally* robust delimitation of the search space, based on minimum and maximum data values, for any target error metric; this comes in contrast to the classical Haar tree, where such target-generic delimitation is not possible. However, due to the same flexibility, the delimitation that exploits a given error bound, particular to the case of a maximum error metric, is less tight with the Haar$^+$ structure. In this case, the delimitation lemmata take the following forms.

LEMMA 5. *Let m_i be the minimum and M_i the maximum individual data value under the scope of triad C_i and $v \in \mathcal{S}_i$ be a possible incoming value at C_i for which the maximum error bound ϵ is satisfied, and $\bar{\epsilon} = \frac{\epsilon}{\min_{j \in I}\{|w_j|\}}$, where I is the interval under the scope of C_i; then $v \in [m_i - \bar{\epsilon}, M_i + \bar{\epsilon}]$.*

Lemma 5 implies that the set $\mathcal{S}_i$ of incoming values we have to examine for triad C_i consists of the multiples of δ in the interval $[m_i - \bar{\epsilon}, M_i + \bar{\epsilon}]$; thus, $|\mathcal{S}_i| \leq \lfloor \frac{M_i - m_i + 2\bar{\epsilon}}{\delta} \rfloor + 1 = O(\frac{\Delta}{\delta})$, where Δ is the difference of the minimum from the maximum value in $\mathbf{D}$. We now demarcate the values assigned to the head coefficient.

LEMMA 6. *Let $v \in \mathcal{S}_i$ be a possible incoming value at C_i and $z_h \in \mathcal{S}_{i,H}^v$ be a value that can be assigned at the head coefficient of C_i for incoming value v, satisfying the individual-data error bound ϵ; then $|z_h| \leq \min\{M_i - v, v - m_i\} + \bar{\epsilon}$.*

[10]The inequality $\leq$ accommodates for the variation in the number of integers in a fixed interval.

Lemma 6 implies that the finite set of possible assigned values we have to examine for the head coefficient at C_i is $\mathcal{S}_{i,H}^v$, where $|\mathcal{S}_{i,H}^v| = O(\frac{\Delta}{\delta})$. The possible assigned values at the left and right supplementary coefficients of triad C_i can be delimited in a similar fashion. Based on this delimitation, we devise our dynamic programming solution. Its essence is the same in both the Haar wavelet and the Haar$^+$ case. We use the former model as our illustrative example. The extension to the latter is straightforward by incorporating provisions for the supplementary coefficients.

5.1.2 Deriving the Answer

In a nutshell, our recursive MinHaarSpace procedure works in a bottom-up left-to-right scan over the Haar (or Haar$^+$) tree. At each visited node c_i it calculates an array A of size $|\mathcal{S}_i|$ from the precalculated arrays L and R of its children nodes c_{i_L}, c_{i_R} (a single array C for the child i_C of the root node). A holds an entry $A[v]$ for each possible *incoming* value v at c_i (a single element A for the root node). Such an entry contains: (i) the minimum number $A[v].s = S(i,v)$ of non-zero coefficients that need to be retained in the sub-tree rooted at c_i with incoming value v, so that the resulting synopsis satisfies the error bound ϵ; (ii) the δ-optimal value $A[v].z$ to assign at c_i, for incoming value v; and (iii) the actual minimized maximum error $A[v].e$ thus obtained in the scope of c_i. $S(i,v)$ is recursively expressed as:

$$S(0,0) = \min_{z \in \mathcal{S}_0^0} \{S(i_C, z) + (z \neq 0)\}$$

$$S(i,v) = \min_{z \in \mathcal{S}_i^v} \{S(i_L, v+z) + S(i_R, v-z) + (z \neq 0)\}$$

The above equations compute the least of (i) the minimum required space if a non-zero coefficient value z is assigned at node c_i; and (ii) the required space if a zero value is assigned at it. The latter case applies only if $0 \in \mathcal{S}_i^v$. For economy in presentation, the $+1$ term that appears in the former case is uniformly expressed by the boolean integer $(z \neq 0)$. This convention is used throughout this section. For a last level node ($i \geq \frac{n}{2}$), the value of $S(i,v)$ is 0 if the coefficient at c_i can be omitted with incoming value v, or $0 \in \mathcal{S}_i^v$, 1 otherwise. The former case occurs if and only if the maximum error yielded by v at the affected data values below c_i satisfies ϵ. The s entry of array A at node c_i for each allowed incoming value v, $A[v].s$, is computed from those of arrays L and R of children nodes c_{i_L} and c_{i_R} (array C for the child c_{i_C} of the root node). Let $\bar{\mathcal{S}}_i^v \subset \mathbb{R}$ denote the set of those assigned values at node c_i for incoming value v that require the minimum space in order to achieve the error bound ϵ:

$$\bar{\mathcal{S}}_0^0 = \operatorname*{argmin}_{z \in \mathcal{S}_0^0} \{S(i_C, z) + (z \neq 0)\}$$

$$\bar{\mathcal{S}}_i^v = \operatorname*{argmin}_{z \in \mathcal{S}_i^v} \{S(i_L, v+z) + S(i_R, v-z) + (z \neq 0)\}$$

The δ-optimal value to select is the one among these candidates that also minimizes, in a secondary priority, the obtained $\mathcal{L}_\infty^w$ error in the scope of c_i. Let $E(i,v)$ be the minimum $\mathcal{L}_\infty^w$ error obtained in the scope of c_i with incoming value v and an assigned value z, with $S(i,v)$ coefficients retained in the sub-tree rooted at c_i:

$$E(0,0) = \min_{z \in \bar{\mathcal{S}}_0^0} \{E(i_C, z)\}$$

$$E(i,v) = \min_{z \in \bar{\mathcal{S}}_i^v} \{\max\{E(i_L, v+z), E(i_R, v-z)\}\}$$

This error value is assigned to $A[v].e$; the value $A[v].z$ is the assigned value that minimizes the error expression above. For a last level node ($i \geq \frac{n}{2}$), if $0 \notin \mathcal{S}_i^v$, then the best non-zero value z^* to assign at c_i is the one that minimizes the $\mathcal{L}_\infty^w$ error yielded at the two affected data values: $w_{i_L}|d_{i_L} - (v+z^*)|$ and $w_{i_R}|d_{i_R} - (v-z^*)|$. This maximum error is minimized when the two are equal: $w_{i_L}|d_{i_L} - (v+z^*)| = w_{i_R}|d_{i_R} - (v-z^*)| \Leftrightarrow z^* = \frac{w_{i_L} d_{i_L} - w_{i_R} d_{i_R} + v(w_{i_R} - w_{i_L})}{w_{i_L} + w_{i_R}}$. In the case of the maximum absolute error, this is the actual Haar wavelet decomposition value at node c_i. Hence, for last-level nodes, we do not need to consider multiples of δ; the value of $E(i,v)$ for a last-level node is:

$$E(i,v) = \begin{cases} \max\{w_{i_L}|d_{i_L} - v|, w_{i_R}|d_{i_R} - v|\}, & 0 \in \mathcal{S}_i^v \\ \frac{w_{i_L} w_{i_R}}{w_{i_L} + w_{i_R}}|d_{i_L} + d_{i_R} - 2v|, & 0 \notin \mathcal{S}_i^v \end{cases}$$

This error value has to be assigned to $A[v].e$ in this case; $A[v].z$ is either 0 or z^*, respectively. A pseudo-code for the proposed recursive MinHaarSpace DP procedure is shown in Figure 5. Following the generic space-efficiency paradigm of [11], the memory occupied by the arrays C, L and R needs to be reserved only when their entries are first computed and is freed after they have in turn been used for the creation of A. Therefore, for a data set of size n, the maximum number of arrays that need to be concurrently stored is $\log n + 1$: one array for each level of resolution plus the currently computed ones. This maximum is necessitated when the right-bound post-order recursion reaches the right-most Haar tree node. This basic bottom-up process computes the wavelet transform's incoming and assigned values on-the-fly in order to define the sets $\mathcal{S}_i$ and $\mathcal{S}_i^v$ as it needs. Hence a recursive procedure that derives the δ-optimal space result answer without constructing the synopsis itself is defined.

Algorithm MinHaarSpace(i, ϵ)
Input: index i, error bound ϵ, n-data vector $\mathbf{D} = [d_0, \ldots, d_{n-1}]$
Output: array A with retained value z for c_i, minimum space s occupied in sub-tree and error e for each $v \in \mathcal{S}_i$

```
1.  if (i = 0) // root node
2.      C = MinHaarSpace(1, ε);
3.      compute s, z ∈ S_0^0, e of A from C;
4.  else if (i < N/2) then // internal node
5.      L = MinHaarSpace(i_L, ε); R = MinHaarSpace(i_R, ε);
6.      for each v ∈ S_i
7.          compute s, z ∈ S_i^v, e of A[v] from L, R;
8.  else if (i ≥ N/2) then // leaf node
9.      for each v ∈ S_i
10.         compute s ∈ {0,1}, z ∈ {0, c_i}, e of A[v] from D;
11. return A;
```

Figure 5: Recursive Minimum Space procedure

Complexity Analysis The result array A on each node c_i holds $|\mathcal{S}_i|$ entries, one for each possible incoming value, hence its size is $O(\frac{\epsilon}{\delta})$; besides, at each node c_i and for each $v \in \mathcal{S}_i$, the loop through all $|\mathcal{S}_i^v|$ possible assigned values needs $O(\frac{\epsilon}{\delta})$ time. Hence, the runtime of MinHaarSpace$(0,\epsilon)$ is $O\left((\frac{\epsilon}{\delta})^2 n\right)$. Besides, since at most $\log n + 1$ arrays need to be concurrently stored, the space complexity is $O\left(\frac{\epsilon}{\delta} \log n + n\right)$, where n stands for the storage of the data.

5.1.3 Constructing the Synopsis

The construction of the synopsis after the δ-optimal answer has been established by a run of MinHaarSpace$(0,\epsilon)$ presents us with a time-space tradeoff. We outline both alternatives.

The Space-Efficient Solution. After MinHaarSpace returns from the topmost level, so that the values of c_0 and c_1 have been established, we can call a process that reenters the problem in the two branches of c_1 and recomputes the respective solutions thereafter, recursively. The total running time is the sum of the basic running time for all re-entered sub-problems. Setting ℓ as the Haar tree level, this sum becomes $O\left((\frac{\epsilon}{\delta})^2 \sum_{\ell=0}^{\log n} 2^\ell \frac{n}{2^\ell}\right) = O\left((\frac{\epsilon}{\delta})^2 n \log n\right)$. On the other hand, the space complexity remains $O\left(\frac{\epsilon}{\delta} \log n + n\right)$,

as we need to maintain the stored data set (or its wavelet transform) throughout the computation.

The Time-Efficient Solution. Alternatively, we may choose to maintain all necessary computed information throughout the recursion of MinHaarSpace. This maintenance allows us to construct the final solution as soon as the minimum space has been derived. Consider a DP array entry $A[v]$ at node c_i; this entry describes the local part of a candidate solution, for incoming value v, which has already been calculated in the sub-tree rooted at c_i. The rest of this candidate solution is maintained by annexing to entry $A[v]$ the set of all coefficient values retained in it. Therewith the sub-problem re-entry is avoided. The total running time remains only $O\left((\frac{\epsilon}{\delta})^2 n\right)$. For each DP array entry $A[v]$ of node c_i at level ℓ, a set of at most $\min\{B_M, 2^\ell\}$ coefficients is retained, where B_M is the maximum size of a candidate solution stored throughout the computation. Thus, the space complexity becomes $O\left(\frac{\epsilon}{\delta}\sum_{\ell=0}^{\log n}\min\{B_M, 2^\ell\}\right) = O\left(\frac{\epsilon}{\delta} B_M \log\frac{n}{B_M}\right)$. In this case, storage of the decomposition is not required.

5.1.4 Verifying Space Optimality

MinHaarSpace approximates the optimal solution in $\mathbb{R}$. In the space-bounded problem, if $\mathcal{E}_B$ is the optimal maximum absolute ($\mathcal{L}_\infty$) error for a B-term synopsis in $\mathbb{R}$, then rounding its values to the closest multiples of δ can increase that error by at most $\frac{\delta}{2}\min\{B, \log n\}$ [12, 21]. For the error-bounded problem, we can formulate the conditions under which the minimum space under resolution δ is the optimal in $\mathbb{R}$, as follows:

THEOREM 2. *Let B be the minimum space, under resolution δ, that satisfies the $\mathcal{L}_\infty$ error bound ϵ, and $\mathcal{E}$ be the minimum $\mathcal{L}_\infty$ error that can be achieved within space $B-1$, under resolution δ. If $\delta < \frac{2(\mathcal{E}-\epsilon)}{\min\{B-1,\log n\}}$ then B is the minimum space required to satisfy error bound ϵ in $\mathbb{R}$.*

PROOF. If $\mathcal{E} \leq \epsilon$, then the approximation algorithm for the error-bounded problem with bound ϵ would find the solution with $B-1$ space; hence $\mathcal{E} > \epsilon$. Let $\mathcal{E}_{B-1}$ be the error achieved by the optimal $(B-1)$-term representation in $\mathbb{R}$. B is the optimal space for bound ϵ if and only if ϵ *cannot* be satisfied with *less* than B non-zero terms; hence it should be $\mathcal{E}_{B-1} > \epsilon$. Since $\mathcal{E} \leq \mathcal{E}_{B-1} + \frac{\delta}{2}\min\{B-1, \log n\}$, a sufficient condition for optimality is $\mathcal{E} - \frac{\delta}{2}\min\{B-1, \log n\} > \epsilon$, or $\delta < \frac{2(\mathcal{E}-\epsilon)}{\min\{B-1,\log n\}}$. □

According to Theorem 2, in order to ascertain that the answer B, derived for an $\mathcal{L}_\infty$-error bound ϵ under resolution δ, is optimal in $\mathbb{R}$, we need to derive the error result $\mathcal{E}$ for the space-bounded problem with space bound $B-1$ under δ. If δ and $\mathcal{E}$ satisfy the condition $\delta < \frac{2(\mathcal{E}-\epsilon)}{\min\{B-1,\log n\}}$, then B is optimal; otherwise, we set a smaller value of δ and repeat the process until we reach space-optimality.

5.2 Application to the Primal Problem

As discussed in Section 2.2, the state-of-the-art solution [12, 13, 21] for space-bounded hierarchical synopsis construction is burdened by a two-dimensional tabulation of $E(i, v, b)$ entries per node. We infer that, as in the histogram case, the space-bounded problem can be more efficiently solved through a binary search invocation of the algorithm for the error-bounded that shuns the tabulation over b. In our implementation, the upper bound of ϵ in the search is the $\mathcal{E}$ corresponding to the synopsis of B largest Haar decomposition coefficients by absolute value, easily computed in $O(n \log B)$ time; the lower bound of ϵ is the $(B+1)$-th highest absolute coefficient value $|z_k|$. Given that the solution to the *strong* error-bounded problem minimizes the error within the δ-optimal space, its application to the space-bounded problem yields the δ-optimal error when the binary search converges to the space budget B. Still, in order to ensure the termination of the search, MinHaarSpace also performs an *optimality test* (as in Section 4.2) for guessed error bound values ϵ that require *less* than B space. Hence, the search terminates when it reaches an error bound that either requires a synopsis of exactly B space, or requires a synopsis of $\bar{B} < B$ space and actual error $\bar{\epsilon}$, while any error bound $\epsilon < \bar{\epsilon}$ requires $\tilde{B} > B$ space. When the tested error bound is decreased during the binary search, the minimum error derived for the previous bound is used for determining the new bound. Figure 6 shows a pseudocode for this IndirectHaar algorithm.

```
Algorithm IndirectHaar(B)
Input:   space bound B, n-data vector [d_0, ..., d_{n-1}]
Output:  L_inf^w-error optimal B-sized unrestricted synopsis
1.  e_u = L_inf^w-error of B-largest-term synopsis;
2.  e_l = (B+1)-th largest coefficient;
3.  e_low = e_l; e_high = e_u;
4.  while (not finished)
5.      e_mid = (e_high + e_low)/2;
6.      Ẑ=MinHaarSpace(e_mid); B̄ = size of Ẑ;
7.      ε̄ = actual L_inf^w-error of Ẑ; /* ε̄ ≤ ε */
8.      if (B̄ < B)
9.          Z̃=MinHaarSpace(< ε̄); B̃ = size of Z̃;
10.         if (B̃ > B)
11.             finished := 1; /* optimal result found */
12.         else e_high = ε̄;
13.     else if (B̄ > B) e_low = e_mid
14.     else finished := 1; /* B̄ = B */
15. return Ẑ;
```

Figure 6: Indirect hierarchical synopsis construction

Complexity Analysis. As in the histogram case, the binary search increases the time requirements of the error-bounded problem by an $O(\log \epsilon^*)$ worst-case factor. We present a space-efficient solution without dependence on B. The advantage of the alternative time-efficient solution in time is negligible, since the $O(\log \epsilon^*)$ factor is comparable to the $\log n$ factor which is paid only once for synopsis construction. Since the highest value of the changing bound ϵ is $\mathcal{E}$, the runtime of this Indirect algorithm is $O\left((\frac{\mathcal{E}}{\delta})^2 n(\log \epsilon^* + \log n)\right)$. The former log term expresses the cost of the binary search, while the latter expresses the cost of constructing the B-term synopsis in a space-efficient manner after the optimal error value ϵ^* has been established. This complexity absorbs the $O(n \log B)$ term for determining the seeds of the search. The $\log \epsilon^*$ factor does not grow with n, hence this runtime is decisively lower than the $O\left((\frac{\mathcal{E}}{\delta})^2 n \log n \log^2 B\right)$ runtime of the space-efficient Direct algorithm and, unless[11] $n \gg B^{\log B}$, lower than its $O\left((\frac{\mathcal{E}}{\delta})^2 n \log^2 B\right)$ *basic* runtime too. Besides, the Indirect algorithm requires $O\left(\frac{\mathcal{E}}{\delta}\log n + n\right)$ space, which is lower than the $O\left(\frac{\mathcal{E}}{\delta} B \log\frac{n}{B} + n\right)$ space of the *space-efficient* Direct algorithm in cases where $\log n \ll B\log\frac{n}{B} \Leftrightarrow n \gg B^{\frac{B}{B-1}}$. This inequality holds in reasonable summarization scenaria, assuming $B \leq \frac{n}{2}$. In addition, the respective $O(n^2)$-time *restricted* algorithm uses $O(B\log\frac{n}{B} + n) = O(n)$ space [11], which becomes larger than $O\left(\frac{\mathcal{E}}{\delta}\log n + n\right)$ when $B\log\frac{n}{B} \gg \frac{\mathcal{E}}{\delta}\log n$. This inequality holds when $B \gg \frac{\mathcal{E}}{\delta}$ and, additionally, $(\frac{n}{B})^B \gg n^{\frac{\mathcal{E}}{\delta}} \Leftrightarrow n \gg B^{\frac{B}{B-\frac{\mathcal{E}}{\delta}}}$; hence, it holds for large enough summarization problems. In conclusion, this Indirect algorithm has better asymptotic behavior than both direct counterparts in time and space. Section 6.3 verifies the runtime benefit of this Indirect algorithm in practice.

[11] The constraint is verified for reasonable $\frac{B}{n}$ ratios; e.g. for $B = 16$, $B^{\log B} = 65536$.

5.3 Comparison to the Restricted Haar Strategy

Our focus is the Indirect solution to the space-bounded problem. We have devised an algorithm for the error-bounded problem in order to serve this goal. Still, this algorithm can present an independent interest of its own. In this context, it is comparable to the $O(\frac{n^2}{\log n})$-time *restricted Haar* algorithm for the maximum-error-bounded problem that was proposed in [25]. This algorithm tries the incoming values yielded by all 2^ℓ ancestor subsets of a node c_i at level ℓ in the Haar tree; it stops recursing and resorts to *local search* within each of the $\lceil \frac{n}{\log n} \rceil$ sub-trees in the bottom $\lfloor \log \log n \rfloor$ Haar tree levels; the examined assigned values for c_i are z_i and 0. The application of Lemmata 3 and 4 prunes ancestor subsets that add up to prohibited incoming values in this algorithm, yet does not annul its near-quadratic time complexity. This complexity is due to the fact that the restricted strategy explicitly enumerates and examines different ancestor-subsets whose coefficients may add up to *nearby* incoming values. In contrast, the unrestricted Haar and Haar$^+$ strategies precalculate a set of equally-spaced allowed incoming values that anticipate all possible contributions ancestor coefficients can add up to. The algorithm in [25] determines the minimum space without constructing the synopsis itself, using $O(n)$ space for storing the decomposition. A sub-problem recomputation in this case costs $O\left(\sum_{\ell=0}^{\log n-1} 2^\ell (\frac{n}{2^\ell})^2 \frac{1}{\log \frac{n}{2^\ell}}\right)$. Setting $k = \log \frac{n}{2^\ell}$, the complexity becomes $O\left(n \sum_{k=1}^{\log n} \frac{2^k}{k}\right) = O(\frac{n^2}{\log n})$. Hence, constructing the synopsis does not present a time-space tradeoff. On the other hand, the algorithm of Section 5.1 is linear to n; hence, for sufficiently large n, it outpaces the near-quadratic restricted Haar algorithm. For an appropriate value of δ, it produces better synopses too (as in [12, 13, 21]). Hence, our algorithm for the error-bounded problem not only provides the basis for a more efficient solution to the dual space-bounded problem, but treats the error-bounded problem itself more efficiently and accurately than previous approaches too.

6. EXPERIMENTAL EVALUATION

In this section we present experimental results demonstrating the advantage of our Indirect solutions vs. the respective Direct for all considered summarization methods. Both solutions compute synopses of equal error (in the hierarchical cases, for equal resolution δ); hence our comparison pertains to runtime (with MAE as the target metric); besides, the space advantage for hierarchical synopses is clear, due to its connection with B. All algorithms were implemented with the g++ 3.4.3 compiler and run on a 4 CPU Opteron 2.2GHz machine with 4GB of main memory running Solaris.

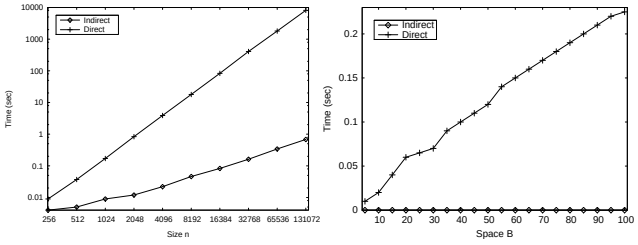
6.1 Description of Data

We used two real data sets. The first data set (TM) is a sequence of 178,080 sea surface temperature measures extracted from drifting buoys positioned throughout the equatorial Pacific. The average value in TM is 26.75 and the set has a standard deviation of 1.91. The second data set (FC) is extracted from a relation of 581,012 tuples describing the forest cover type for 30 x 30 meter cells, obtained from US Forest Service. FC contains the frequencies of the distinct values of attribute `aspect` in the relation. The frequencies average at 1613 (standard deviation: 730) and feature spikes of large values (min value: 499, max value: 6308). FC and TM were downloaded from the UCI KDD Archive.[12]

[12]Available at http://kdd.ics.uci.edu/

6.2 Histogram-based Summarization

In this experiment we measure the runtime for histogram construction. We compare the Direct solution of [14], in its space-efficient variant, to our Indirect method (Section 4.2). For the Direct method, we measured the basic runtime required to derive the optimal error result only. The Indirect algorithm computes the optimal histogram segmentation per se, which is the same for both. Figure 7a shows their performance as a function of n with a constant summarization ratio $B = n/64$ for various-sized subsets of the TM data set. As expected from our theoretical analysis, Indirect vastly outperforms Direct. Our second experiment measures the running time with respect to the bucket space B for a constant data size n. Figure 7b shows the results for the FC data set with constant $n = 360$. As expected, the Indirect method exhibits its independence of B in both cases; it always terminated after a few repetitions (mean 12.3). In contrast, the runtime of Direct grows with B.



(a) Runtime vs. n, $B = n/64$ (b) Runtime vs. B

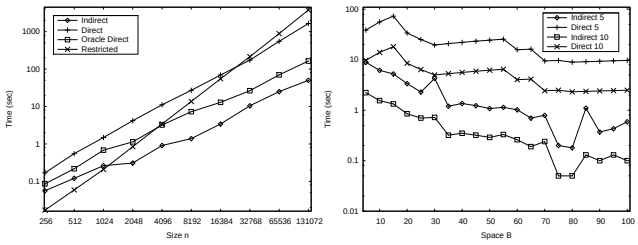
Figure 7: Runtime comparison (Histograms)

6.3 Hierarchical Summarization

Haar Wavelet Synopsis Construction.

In this experiment we measure the runtime of hierarchical summarization algorithms, starting with the classical Haar case. We first compare the Indirect method of Section 5.2 to two versions of the maximum-error unrestricted Haar synopsis algorithm of [12, 13]. The former, Direct, first calculates, in $O(n \log B)$ time, the target error for the synopsis consisting of the top-B Haar wavelet terms by absolute value; then it employs it for bounding the search space. The latter, OracleDirect, is an infeasible algorithm, which represents a conceptual limit for the best-case performance of the guess-based solution of [13]. In OracleDirect, the value of the final optimal error is assumed to be provided in advance by an oracle, hence its search space is optimally delimited. Both direct algorithms compute the same error result as Indirect. After experimentation, we settled for a reasonable, in the given data set, constant value of $\delta = 0.1$ (i.e. the resolution step for delimiting the domains of incoming and assigned values) for all three algorithms. Smaller values burdened the running time without significant quality increase; larger values were undermining the quality of the synopses. We also ran an *enhanced* version of the *restricted* algorithm of [8, 11] (Restricted), which also prunes its search space using a precalculated error bound. For all algorithms, we measured the basic time required to derive (for OracleDirect, to verify) the optimal error result. Figure 8a shows (on a log-log graph) their performance as a function of n with a constant summarization ratio $B = n/64$ for various-sized subsets of the TM data set. As expected, Indirect presents the most affordable runtime growth; not only it outperforms Direct, but it outpaces the OracleDirect too; hence it invariably produces identical quality in shorter time and smaller space. Besides, Restricted, which achieves lower synopsis quality, undergoes the fastest growth, eventually becoming the slowest, due to its quadratic time complexity; this result reconfirms the finding of [12]

in the realm of these pruning-intensive *enhanced* variants and with larger data sizes that reveal the disadvantage more clearly. Figure 8b plots (on a log-lin graph) the runtime for the FC data set with respect to B (for constant $n = 512$, obtained after zero-padding the wavelet decomposition), setting δ at 5 and 10. The results for Direct exhibit the interplay between two factors affecting the running time: One the one hand, the increase of B results into tighter delimitation of the search space based on a smaller pre-calculated error upper bound, with a significant impact on running time. However, B affects the time complexity itself as well. Hence the runtime of Direct presents unstable behavior, with a maximum at the intermediary position $B = 15$. On the other hand, the runtime of Indirect tends to decrease as B (hence the tightness of the error bound) grows; this algorithm terminated after a few repetitions, even fewer than in the histogram case: it converges more robustly thanks to its exploitation of the *strong* version of the error-bounded problem.

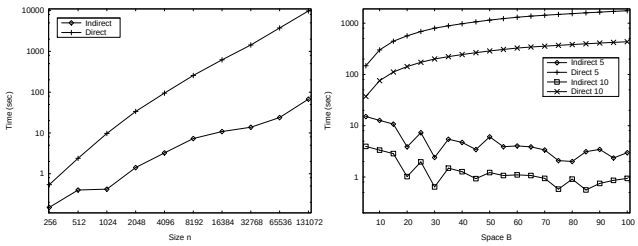


(a) Runtime vs. n, $B = n/64$ (b) Runtime vs. B

Figure 8: Runtime comparison (Haar wavelets)

Haar⁺ Synopsis Construction.

We repeated, in the Haar$^+$ case, with the same data sets and configurations, the comparison between the Indirect method of Section 5.2 and its Direct counterpart [21]; this also prunes its search space as much as possible, using a precalculated error bound and Lemmata 5 and 6. Figure 9 shows the results. The runtime of Direct is larger in this case, due to the less intensive pruning that the Haar$^+$ structure allows. However, the performance of Indirect is equally satisfactory as in the classical Haar case (compare Figures 8a and 9a). The difference is more conspicuous for runtime versus synopsis size B (Figure 9b). In this case, the increasing tightness of the pre-calculated error bound cannot overcome the effect of the increasing B itself on the runtime of Direct; hence, it grows with B. Still, the runtime of Indirect shows a decreasing trend as B grows in this case too. The plots of Figure 9 have identical x-axes to those of Figure 8, but their logarithmic y-axes are scaled differently for the sake of readability.



(a) Runtime vs. n, $B = n/64$ (b) Runtime vs. B

Figure 9: Runtime comparison (Haar$^+$)

7. CONCLUSIONS

In this paper we have examined the problem of summarization with deterministic guarantees from a new perspective, applied on state-of-the-art histogram and hierarchical methods. We demonstrated the advantage gained by solving the computationally heavy and memory-hungry space-bounded problems through their lighter error-bounded counterparts; this advantage consists of complexities which are lower, independent of synopsis space and free of performance tradeoffs; it stems from the removal of a tabulation that hindered previous solutions and, in the hierarchical case, the tight delimitation of the search space. In conclusion, our solutions provide the most recommendable option for the time- and space-efficient offline summarization of very large data sets with a maximum-error guarantee. In the future, we plan to extend our techniques to the summarization of multi-measure and multidimensional data.

8. REFERENCES

[1] R. Bellman. On the approximation of curves by line segments using dynamic programming. *Communications of the ACM*, 4(6):284, 1961.

[2] C. Buragohain, N. Shrivastava, and S. Suri. Space efficient streaming algorithms for the maximum error histogram. In *ICDE*, 2007.

[3] K. Chakrabarti, M. Garofalakis, R. Rastogi, and K. Shim. Approximate query processing using wavelets. *VLDB J.*, 10(2-3):199–223, 2001 (also VLDB 2000).

[4] K. Chakrabarti, E. Keogh, S. Mehrotra, and M. Pazzani. Locally adaptive dimensionality reduction for indexing large time series databases. *TODS*, 27(2):188–228, 2002 (also SIGMOD 2001).

[5] G. Cormode, M. Garofalakis, and D. Sacharidis. Fast approximate wavelet tracking on streams. In *EDBT*, 2006.

[6] A. Deligiannakis, M. Garofalakis, and N. Roussopoulos. Extended wavelets for multiple measures. *TODS*, 32(1), 2007 (also SIGMOD 2003).

[7] M. Garofalakis and P. B. Gibbons. Probabilistic wavelet synopses. *TODS*, 29(1):43–90, 2004 (also SIGMOD 2002).

[8] M. Garofalakis and A. Kumar. Wavelet synopses for general error metrics. *TODS*, 30(4):888–928, 2005 (also PODS 2004).

[9] P. B. Gibbons and Y. Matias. Synopsis data structures for massive data sets. In *SODA*, 1999.

[10] A. C. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. J. Strauss. One-pass wavelet decompositions of data streams. *IEEE TKDE*, 15(3):541–554, 2003.

[11] S. Guha. Space efficiency in synopsis construction algorithms. In *VLDB*, 2005.

[12] S. Guha and B. Harb. Wavelet synopsis for data streams: minimizing non-euclidean error. In *SIGKDD*, 2005.

[13] S. Guha and B. Harb. Approximation algorithms for wavelet transform coding of data streams. In *SODA*, 2006.

[14] S. Guha, K. Shim, and J. Woo. REHIST: Relative error histogram construction algorithms. In *VLDB*, 2004.

[15] Y. E. Ioannidis. Universality of serial histograms. In *VLDB*, 1993.

[16] Y. E. Ioannidis. The history of histograms (abridged). In *VLDB*, 2003.

[17] H. V. Jagadish, N. Koudas, S. Muthukrishnan, V. Poosala, K. C. Sevcik, and T. Suel. Optimal histograms with quality guarantees. In *VLDB*, 1998.

[18] M. Jahangiri, D. Sacharidis, and C. Shahabi. SHIFT-SPLIT: I/O efficient maintenance of wavelet-transformed multidimensional data. In *SIGMOD*, 2005.

[19] B. Jawerth and W. Sweldens. An overview of wavelet based multiresolution analyses. *SIAM Rev.*, 36(3):377–412, 1994.

[20] P. Karras and N. Mamoulis. One-pass wavelet synopses for maximum-error metrics. In *VLDB*, 2005.

[21] P. Karras and N. Mamoulis. The Haar$^+$ tree: a refined synopsis data structure. In *ICDE*, 2007.

[22] I. Lazaridis and S. Mehrotra. Capturing sensor-generated time series with quality guarantees. In *ICDE*, 2003.

[23] T. Li, Q. Li, S. Zhu, and M. Ogihara. A survey on wavelet applications in data mining. *SIGKDD Explorations Newsletter*, 4(2):49–68, 2002.

[24] Y. Matias, J. S. Vitter, and M. Wang. Wavelet-based histograms for selectivity estimation. In *SIGMOD*, 1998.

[25] S. Muthukrishnan. Subquadratic algorithms for workload-aware haar wavelet synopses. In *FSTTCS*, 2005.

[26] V. Poosala, V. Ganti, and Y. E. Ioannidis. Approximate query answering using histograms. *IEEE Data Eng. Bull.*, 22(4):5–14, 1999.

[27] F. Reiss, M. Garofalakis, and J. M. Hellerstein. Compact histograms for hierarchical identifiers. In *VLDB*, 2006.

[28] J. S. Vitter and M. Wang. Approximate computation of multidimensional aggregates of sparse data using wavelets. In *SIGMOD*, 1999.