Wavelet Synopses for General Error Metrics

MINOS GAROFALAKIS
Intel Research Berkeley
and
AMIT KUMAR
Indian Institute of Technology

Several studies have demonstrated the effectiveness of the wavelet decomposition as a tool for reducing large amounts of data down to compact *wavelet synopses* that can be used to obtain fast, accurate approximate query answers. Conventional wavelet synopses that greedily minimize the overall root-mean-squared (i.e., L_2-norm) error in the data approximation can suffer from important problems, including severe bias and wide variance in the quality of the data reconstruction, and lack of nontrivial guarantees for individual approximate answers. Thus, *probabilistic thresholding schemes* have been recently proposed as a means of building wavelet synopses that try to *probabilistically* control *maximum* approximation-error metrics (e.g., maximum relative error).

A key open problem is whether it is possible to design efficient *deterministic* wavelet-thresholding algorithms for minimizing *general, non-L_2 error metrics* that are relevant to approximate query processing systems, such as maximum relative or maximum absolute error. Obviously, such algorithms can guarantee better maximum-error wavelet synopses and avoid the pitfalls of probabilistic techniques (e.g., "bad" coin-flip sequences) leading to poor solutions; in addition, they can be used to directly optimize the synopsis construction process for other useful error metrics, such as the *mean relative error* in data-value reconstruction. In this article, we propose novel, computationally efficient schemes for deterministic wavelet thresholding with the objective of optimizing *general approximation-error metrics*. We first consider the problem of constructing wavelet synopses optimized for *maximum error*, and introduce an *optimal* low polynomial-time algorithm for *one-dimensional* wavelet thresholding—our algorithm is based on a new Dynamic-Programming (DP) formulation, and can be employed to minimize the maximum relative or absolute error in the data reconstruction. Unfortunately, directly extending our one-dimensional DP algorithm to *multidimensional* wavelets results in a super-exponential increase in time complexity with the data dimensionality. Thus, we also introduce novel, polynomial-time *approximation schemes* (with tunable approximation guarantees) for deterministic wavelet thresholding in multiple dimensions. We then demonstrate how our optimal and approximate thresholding algorithms for maximum

A preliminary version of this article appeared in the *Proceedings of the 23rd Annual ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems* (Paris, France, June), ACM, New York, 2004.
Most of this work was done while the authors were with Bell Labs, Lucent Technologies.
Authors' addresses: M. Garofalakis, Intel Research Berkeley, Penthouse Suite, 2150 Shattuck Avenue, Berkeley, CA 94704; email: minos.garofalakis@intel.com; A. Kumar, Department of Computer Science and Engineering, Indian Institute of Technology, Hauz Khas, New Delhi, India 110016; email: amitk@cse.iitd.ernet.in.

error can be extended to handle a broad, natural class of *distributive error metrics*, which includes several important error measures, such as mean weighted relative error and weighted L_p-norm error. Experimental results on real-world and synthetic data sets evaluate our novel optimization algorithms, and demonstrate their effectiveness against earlier wavelet-thresholding schemes.

Categories and Subject Descriptors: G.2.1 [**Discrete Mathematics**]: Combinatorics—*Combinatorial algorithms*; H.2.4 [**Database Management**]: Systems—*Query processing*

General Terms: Algorithms, Performance, Theory

Additional Key Words and Phrases: Data synopses, Haar wavelets, approximate query processing

1. INTRODUCTION

Approximate query processing over precomputed data synopses has emerged as a cost-effective approach for dealing with the huge data volumes, the high query complexities, and the increasingly stringent response-time requirements that characterize today's data-analysis applications. Typically, users pose very complex queries to the underlying Database Management System (DBMS) that require complex operations over large amounts of disk-resident data and, thus, take a very long time to execute to completion and produce exact answers. Due to the *exploratory nature* of many data-analysis applications, there are a number of scenarios in which an exact answer may not be required, and a user may in fact prefer a fast, approximate answer. For example, during a drill-down query sequence in ad-hoc data mining, initial queries in the sequence frequently have the sole purpose of determining the truly interesting queries and regions of the database [Hellerstein et al. 1997]. Providing (reasonably accurate) approximate answers to these initial queries gives users the ability to focus their explorations quickly and effectively, without consuming inordinate amounts of valuable system resources. An approximate answer can also provide useful feedback on how wellposed a query is, allowing users to make an informed decision on whether they would like to invest more time and resources to execute their query to completion. Moreover, approximate answers obtained from appropriate *synopses* of the data may be the only available option when the base data is remote and unavailable [Amsaleg et al. 1997]. Finally, for data-analysis queries requesting a numerical answer (e.g., total revenues or annual percentage), it is often the case that the full precision of the exact answer is not needed and the first few digits of precision will suffice (e.g., the leading few digits of a total in the millions or the nearest percentile of a percentage) [Acharya et al. 1999].

Wavelets provide a mathematical tool for the hierarchical decomposition of functions, with a long history of successful applications in signal and image processing [Jawerth and Sweldens 1994; Natsev et al. 1999; Stollnitz et al. 1996]. Recent studies have also demonstrated the applicability of wavelets to selectivity estimation [Matias et al. 1998] and to approximate query processing over massive relational tables [Chakrabarti et al. 2000, 2001; Vitter and Wang 1999] and data streams [Matias et al. 2000; Gilbert et al. 2001]. Briefly, the idea is to apply wavelet decomposition to the input relation (attribute column(s) or OLAP cube) to obtain a compact data synopsis that comprises

a select small collection of *wavelet coefficients*. The results of Matias, Vitter, and Wang [Matias et al. 1998; Vitter and Wang 1999] and Chakrabarti et al. [2000, 2001] have demonstrated that fast and accurate approximate query processing engines can be designed to operate solely over such compact *wavelet synopses*.

A major shortcoming of conventional wavelet-based techniques for approximate query processing (including all the above-cited studies) is the fact that the quality of approximate answers can vary widely and no meaningful error guarantees can be provided to the users of the approximate query engine. Coefficients in conventional wavelet synopses are typically chosen to optimize the overall root-mean-squared (i.e., L_2-norm average) error in the data approximation which, as demonstrated in recent studies by Garofalakis and Gibbons [2002, 2004], can result in wide variance as well as severe bias in the quality of the approximation over the underlying domain of data values. Their proposed solution, termed *probabilistic wavelet synopses*, relies on probabilistic thresholding schemes (based on randomized rounding [Motwani and Raghavan 1995]) for synopsis construction that try to *probabilistically* control *maximum* approximation-error metrics, such as the *maximum relative error* in the data reconstruction [Garofalakis and Gibbons 2002, 2004]. Such maximum relative-error metrics represent an important quality measure for effective approximate query processing and can provide meaningful error guarantees for individual approximate answers. In order to probabilistically control maximum relative error, the algorithms proposed in Garofalakis and Gibbons [2002, 2004] explicitly try to minimize appropriate probabilistic metrics (such as normalized standard error or normalized bias) for the randomized synopsis construction process [Garofalakis and Gibbons 2002, 2004]. (Similar schemes are also given for controlling maximum *absolute* error.)

1.1 Our Contributions

A potential problem with the probabilistic thresholding techniques of Garofalakis and Gibbons [2002, 2004] is that, exactly due to their probabilistic nature, there is always a possibility of a "bad" sequence of coin flips resulting in a poor synopsis; furthermore, they are based on a *quantization* of the possible synopsis-space allotments, whose impact on the quality of the final synopsis is not entirely clear. Clearly, a *deterministic* thresholding algorithm that explicitly minimizes the relevant maximum-error metric (e.g., maximum relative error) in the synopsis, is always guaranteed to give better results. Unfortunately, as already pointed out by Garofalakis and Gibbons [2002, 2004], their thresholding algorithms depend critically on the probabilistic nature of their solution, and are inapplicable in a deterministic setting. An additional shortcoming of the probabilistic thresholding schemes of Garofalakis and Gibbons [2002, 2004] is that they explicitly target *only maximum-error metrics*; it is not at all clear if or how they can be extended to other useful error measures for approximate query answers (such as, e.g., *mean* relative error). In fact, one of the main open problems cited in Garofalakis and Gibbons [2002, 2004] is whether it is possible to design efficient deterministic thresholding for minimizing *non-*L_2

error metrics that are relevant for approximate query answering systems, such as the maximum or mean relative error in the data approximation.

In this article, we propose novel, computationally efficient schemes for deterministic wavelet thresholding with the objective of optimizing a *general class of error metrics* (e.g., maximum or mean relative error). We first consider the problem of constructing wavelet synopses optimized for *maximum error metrics*, and present an *optimal* low polynomial-time algorithm for *one-dimensional* wavelet thresholding. Our algorithm is based on a novel Dynamic-Programming (DP) formulation and can be employed to minimize either maximum relative error or maximum absolute error in the data reconstruction—its running time and working-space requirements are only $O(N^2)$ and $O(N \min\{B, \log N\})$, respectively, where N denotes the size of the data domain and B is the desired size of the synopsis (i.e., number of retained coefficients). Unfortunately, directly extending our optimal DP algorithm to *multidimensional* wavelets results in a superexponential increase in time complexity with the data dimensionality, rendering such a solution unusable even for the relatively small dimensionalities where wavelets are typically used (e.g., 2–5 dimensions). Thus, we also introduce two efficient, polynomial-time *approximation schemes* (with tunable ϵ-approximation guarantees for the target maximum-error metric) for deterministic wavelet thresholding in multiple dimensions. Both our approximation schemes are based on approximate dynamic programs that tabulate a much smaller number of subproblems than the optimal DP solution, while guaranteeing a small deviation from the optimal objective value. More specifically, our first approximation algorithm can give ϵ-additive-error guarantees for maximum relative or absolute error, whereas our second algorithm is a $(1 + \epsilon)$-approximation scheme for maximum absolute error—the running time for both our approximation schemes is roughly proportional to $O(\frac{1}{\epsilon} N \log^2 N B \log B)$. We then demonstrate how our optimal and approximate thresholding algorithms for maximum error can be extended to handle a broad, natural class of *distributive error metrics*; this class includes several useful error measures for approximate query answers, such as mean weighted relative error and weighted L_p-norm error. Finally, we present the results of an empirical study that evaluates our novel synopsis-construction algorithms over real-world and synthetic data sets, demonstrating their effectiveness against known wavelet-thresholding schemes [Garofalakis and Gibbons 2002, 2004]. To the best of our knowledge, our work is the *first* to propose efficient optimal and near-optimal algorithms for building wavelet synopses optimized for general error metrics in one or multiple dimensions.

1.2 Organization

The remainder of this article is organized as follows; Section 2 discusses background material on the wavelet decomposition and wavelet data synopses. In Section 3, we develop our deterministic maximum-error thresholding algorithms for both one- and multidimensional wavelets. Section 4 discusses the extension of our maximum-error algorithms and results to a general class of distributive error metrics. Experimental results on real-world and synthetic

data are presented in Section 5. Section 6 gives an overview of related work and, finally, Section 7 outlines our conclusions along with some interesting directions for future research in this area.

2. WAVELET BASICS

Wavelets are a useful mathematical tool for hierarchically decomposing functions in ways that are both efficient and theoretically sound. Broadly speaking, the wavelet decomposition of a function consists of a coarse overall approximation together with detail coefficients that influence the function at various scales [Stollnitz et al. 1996]. The wavelet decomposition has excellent energy compaction and de-correlation properties, which can be used to effectively generate compact representations that exploit the structure of data. Furthermore, wavelet transforms can generally be computed in linear time.

2.1 One-Dimensional Haar Wavelets

Suppose we are given the one-dimensional data vector A containing the $N = 8$ data values $A = [2, 2, 0, 2, 3, 5, 4, 4]$. The Haar wavelet transform of A can be computed as follows. We first average the values together pairwise to get a new "lower-resolution" representation of the data with the following average values $[2, 1, 4, 4]$. In other words, the average of the first two values (i.e., 2 and 2) is 2, that of the next two values (i.e., 0 and 2) is 1, and so on. Obviously, some information has been lost in this averaging process. To be able to restore the original values of the data array, we need to store some *detail coefficients*, that capture the missing information. In Haar wavelets, these detail coefficients are simply the differences of the (second of the) averaged values from the computed pairwise average. Thus, in our simple example, for the first pair of averaged values, the detail coefficient is 0 since $2 - 2 = 0$, for the second we again need to store -1 since $1 - 2 = -1$. Note that no information has been lost in this process—it is fairly simple to reconstruct the eight values of the original data array from the lower-resolution array containing the four averages and the four detail coefficients. Recursively applying the above pairwise averaging and differencing process on the lower-resolution array containing the averages, we get the following full decomposition:

Resolution	Averages	Detail Coefficients
3	[2, 2, 0, 2, 3, 5, 4, 4]	—
2	[2, 1, 4, 4]	[0, −1, −1, 0]
1	[3/2, 4]	[1/2, 0]
0	[11/4]	[−5/4]

The *wavelet transform* (also known as the *wavelet decomposition*) of A is the single coefficient representing the overall average of the data values followed by the detail coefficients in the order of increasing resolution. Thus, the one-dimensional Haar wavelet transform of A is given by $W_A = [11/4, -5/4, 1/2, 0, 0, -1, -1, 0]$. Each entry in W_A is called a *wavelet coefficient*. The main advantage of using W_A instead of the original data vector A is that for vectors containing similar values most of the detail coefficients tend to

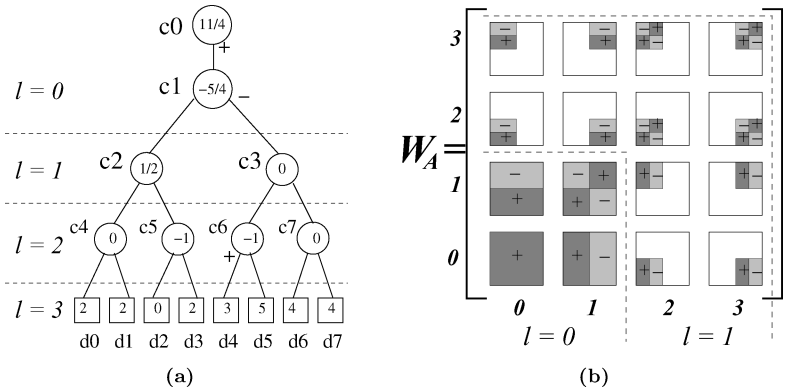

Fig. 1. (a) Error-tree structure for our example data array A ($N = 8$). (b) Support regions and signs for the sixteen nonstandard two-dimensional Haar basis functions. The coefficient magnitudes are multiplied by $+1$ (-1) where a sign of $+$ (respectively, $-$) appears, and 0 in blank areas.

have very small values. Thus, eliminating such small coefficients from the wavelet transform (i.e., treating them as zeros) introduces only small errors when reconstructing the original data, resulting in a very effective form of lossy data compression [Stollnitz et al. 1996].

Note that, intuitively, wavelet coefficients carry different weights with respect to their importance in rebuilding the original data values. For example, the overall average is obviously more important than any detail coefficient since it affects the reconstruction of all entries in the data array. In order to equalize the importance of all wavelet coefficients, we need to *normalize* the final entries of W_A appropriately. A common normalization scheme [Stollnitz et al. 1996] is to divide each wavelet coefficient by $\sqrt{2^l}$, where l denotes the *level of resolution* at which the coefficient appears (with $l = 0$ corresponding to the "coarsest" resolution level). Thus, the normalized coefficient, c_i^*, is $c_i/\sqrt{2^{\texttt{level}(c_i)}}$.

2.1.1 *Basic Haar Wavelet Properties and Notational Conventions.* A helpful tool for exploring and understanding the key properties of the Haar wavelet decomposition is the *error tree* structure [Matias et al. 1998]. The error tree is a hierarchical structure built based on the wavelet transform process (even though it is primarily used as a conceptual tool, an error tree can be easily constructed in linear $O(N)$ time). Figure 1(a) depicts the error tree for our simple example data vector A. Each internal node c_i ($i = 0, \ldots, 7$) is associated with a wavelet coefficient value, and each leaf d_i ($i = 0, \ldots, 7$) is associated with a value in the original data array; in both cases, the index i denotes the positions in the (data or wavelet transform) array. For example, c_0 corresponds to the overall average of A. Note that the values associated with the error tree nodes c_j are the *unnormalized* coefficient values; the resolution levels l for the coefficients (corresponding to levels in the tree) are also depicted. We use the terms "node", "coefficient", and "node/coefficient value" interchangeably in what follows. For ease of reference, Table I summarizes some of the key notation used in this article with a brief description of its semantics. Detailed definitions of all these parameters are provided at the appropriate locations in the text.

Table I. Notation

Symbol ($i \in \{0..N-1\}$)	Description
N	Number of data-array cells
D	Data-array dimensionality
B	Space budget for synopsis
A, W_A	Input data and wavelet transform arrays
d_i	Data value for ith data-array cell
$\hat{d}_i$	Reconstructed data value for ith array cell
c_i	Haar coefficient at index/coordinate i
T_i	Error subtree rooted at node c_i
$\texttt{coeff}(T_i)$, $\texttt{data}(T_i)$	Coefficient/data values in error subtree T_i
$\texttt{path}(u)$	All non-zero proper ancestors of node u in the error tree
s	Sanity bound for relative-error metric
relErr_i, absErr_i	Relative and absolute error for data value d_i

For simplicity, the notation assumes one-dimensional wavelets—extensions to multidimensional wavelets are straightforward. Additional notation will be introduced when necessary.

Given a node u in an error tree T, let $\texttt{path}(u)$ denote the set of all proper ancestors of u in T (i.e., the nodes on the path from u to the root of T, including the root but not u) with nonzero coefficients. A key property of the Haar wavelet decomposition is that the reconstruction of any data value d_i depends only on the values of coefficients on $\texttt{path}(d_i)$; more specifically, we have

$$d_i = \sum_{c_j \in \texttt{path}(d_i)} \mathsf{sign}_{i,j} \cdot c_j, \tag{1}$$

where $\mathsf{sign}_{i,j} = +1$ if d_i is in the left child subtree of c_j or $j = 0$, and $\mathsf{sign}_{i,j} = -1$ otherwise. Thus, reconstructing any data value involves summing at most $\log N + 1$ coefficients. For example, in Figure 1(a), $d_4 = c_0 - c_1 + c_6 = \frac{11}{4} - (-\frac{5}{4}) + (-1) = 3$. The *support region* for a coefficient c_i is defined as the set of (contiguous) data values that c_i is used to reconstruct; the support region for a coefficient c_i is uniquely identified by its coordinate i.

2.2 Multidimensional Haar Wavelets

The Haar wavelet decomposition can be extended to *multidimensional* data arrays using two distinct methods, namely the *standard* and *nonstandard* Haar decomposition [Stollnitz et al. 1996]. Each of these transforms results from a natural generalization of the one-dimensional decomposition process described above, and both have been used in a wide variety of applications, including approximate query answering over high-dimensional data sets [Chakrabarti et al. 2000, 2001; Vitter and Wang 1999].

As in the one-dimensional case, the Haar decomposition of a D-dimensional data array A results in a D-dimensional wavelet-coefficient array W_A with the same dimension ranges and number of entries. (The full details as well as efficient decomposition algorithms can be found in Chakrabarti et al. [2000, 2001] and Vitter and Wang [1999].) Consider a D-dimensional wavelet coefficient W in the (standard or nonstandard) wavelet-coefficient array W_A.

W contributes to the reconstruction of a D-dimensional rectangular region of cells in the original data array A (i.e., W's *support region*). Further, the sign of W's contribution ($+W$ or $-W$) can vary along the quadrants of W's support region in A. As an example, Figure 1(b) depicts the support regions and signs of the sixteen nonstandard, two-dimensional Haar coefficients in the corresponding locations of a 4×4 wavelet-coefficient array W_A. The blank areas for each coefficient correspond to regions of A whose reconstruction is independent of the coefficient, that is, the coefficient's contribution is 0. Thus, $W_A[0, 0]$ is the overall average that contributes positively (i.e.,"$+W_A[0, 0]$") to the reconstruction of all values in A, whereas $W_A[3, 3]$ is a detail coefficient that contributes (with the signs shown in Figure 1(b)) only to values in A's upper right quadrant. Each data cell in A can be accurately reconstructed by adding up the contributions (with the appropriate signs) of those coefficients whose support regions include the cell. Figure 1(b) also depicts the two *levels of resolution* ($l = 0, 1$) for our example two-dimensional Haar coefficients; as in the one-dimensional case, these levels define the appropriate constants for normalizing coefficient values [Chakrabarti et al. 2000, 2001; Stollnitz et al. 1996].

Error-tree structures for multidimensional Haar wavelets can be constructed (once again in linear $O(N)$ time) in a manner similar to those for the one-dimensional case, but their semantics and structure are somewhat more complex. A major difference is that, in a D-dimensional error tree, each node (except for the root, i.e., the overall average) actually corresponds to a *set* of $2^D - 1$ wavelet coefficients that have the same support region but different quadrant signs and magnitudes for their contribution. Furthermore, each (nonroot) node t in a D-dimensional error tree has 2^D children corresponding to the quadrants of the (common) support region of all coefficients in t.[1] (Note that the sign of each coefficient's contribution to the leaf (data) values residing at each of its children in the tree is determined by the coefficient's quadrant sign information.) As an example, Figure 2 depicts the error-tree structure for the two-dimensional 4×4 Haar coefficient array in Figure 1(b). Thus, the (single) child t of the root node contains the coefficients $W_A[0, 1]$, $W_A[1, 0]$, and $W_A[1, 1]$, and has four children corresponding to the four 2×2 quadrants of the array; the child corresponding to the lower-left quadrant contains the coefficients $W_A[0, 2]$, $W_A[2, 0]$, and $W_A[2, 2]$, and all coefficients in t contribute with a "+" sign to all values in this quadrant.

Based on the above generalization of the error-tree structure to multiple dimensions, we can naturally extend the formula for data-value reconstruction (Eq. (1)) to multi-dimensional Haar wavelets. Once again, the reconstruction of d_i depends only on the *coefficient sets* for all error-tree nodes in $\texttt{path}(d_i)$, where the sign of the contribution for each coefficient W in node $t \in \texttt{path}(d_i)$ is determined by the quadrant sign information for W.

[1]The number of children (coefficients) for an internal error-tree node can actually be less than 2^D (respectively, $2^D - 1$) when the sizes of the data dimensions are not all equal. In these situations, the exponent for 2 is determined by the number of dimensions that are *"active"* at the current level of the decomposition (i.e., those dimensions that are still being recursively split by averaging/differencing).

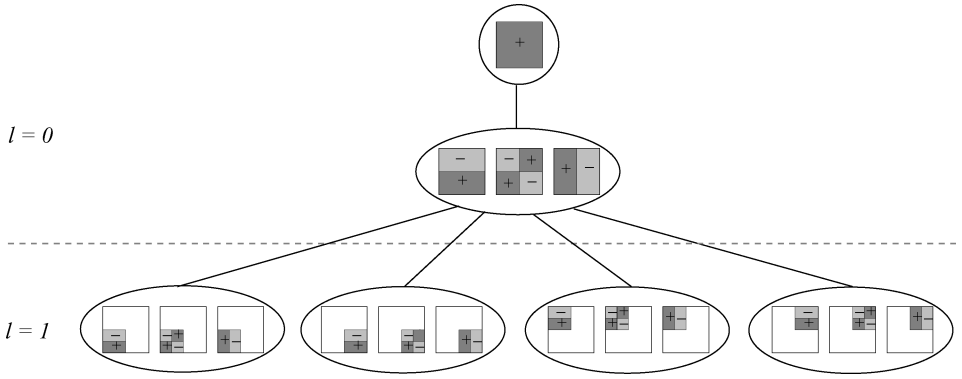

Fig. 2. Error-tree structure for the sixteen nonstandard two-dimensional Haar coefficients for a 4×4 data array (data values omitted for clarity).

2.3 Wavelet-Based Data Reduction: Coefficient Thresholding

Given a limited amount of storage for building a *wavelet synopsis* of the input data array A, a thresholding procedure retains a certain number $B \ll N$ of the coefficients in W_A as a highly compressed approximate representation of the original data (the remaining coefficients are implicitly set to 0). The goal of coefficient thresholding is to determine the "best" subset of B coefficients to retain, so that some overall error measure in the approximation is minimized. The method of choice for the vast majority of earlier studies on wavelet-based data reduction and approximation [Chakrabarti et al. 2000, 2001; Matias et al. 1998, 2000; Vitter and Wang 1999] is *conventional coefficient thresholding* that greedily retains the B largest Haar-wavelet coefficients in *absolute normalized value*. It is a well-known fact that this thresholding method is in fact *provably optimal* with respect to minimizing the overall root-mean-squared error (i.e., L_2*-norm average error*) in the data compression [Stollnitz et al. 1996]. More formally, letting $\hat{d}_i$ denote the (approximate) reconstructed data value for cell i, retaining the B largest normalized coefficients implies that the resulting synopsis minimizes the quantity $\sqrt{\frac{1}{N}\sum_i (\hat{d}_i - d_i)^2}$ (for the given amount of space B).

Unfortunately, wavelet synopses optimized for overall L_2 error may not always be the best choice for approximate query processing systems. As observed in the recent study of Garofalakis and Gibbons [2002, 2004], such conventional wavelet synopses suffer from several important problems, including the introduction of severe bias in the data reconstruction and wide variance in the quality of the data approximation, as well as the lack of nontrivial guarantees for individual approximate answers. To address these shortcomings, their work introduces novel, *probabilistic* thresholding schemes based on randomized rounding [Motwani and Raghavan 1995], that probabilistically round coefficients either up to a larger rounding value (to be retained in the synopsis) or down to zero. Intuitively, their probabilistic schemes assign each non-zero coefficient *fractional storage* $y \in (0, 1]$ equal to its retention probability, and then flip independent, appropriatelybiased coins to construct the synopsis. Their winning

strategies (termed MinRelVar and MinRelBias in Garofalakis and Gibbons [2002, 2004]) are based on trying to *probabilistically* control the *maximum relative error* (with an appropriate *sanity bound* s)[2] in the approximation of individual data values based on the synopsis; that is, they attempt to minimize the quantity

$$\max_i \left\{ \frac{|\hat{d}_i - d_i|}{\max\{|d_i|, s\}} \right\},$$

(where $\hat{d}_i$ denotes the data value reconstructed based on the synopsis) with sufficiently high probability. More specifically, their MinRelVar and MinRelBias algorithms are based on *Dynamic-Programming (DP)* formulations that explicitly minimize appropriate probabilistic metrics, such as the maximum normalized standard error (MinRelVar) or the maximum normalized bias (MinRelBias), in the randomized synopsis construction; these formulations are then combined with a *quantization* of the potential fractional-storage allotments to give combinatorial thresholding algorithms [Garofalakis and Gibbons 2002, 2004]. (Similar probabilistic schemes are also given for probabilistically controlling maximum *absolute* error.)

2.3.1 *Building and Using Wavelet Synopses in a DBMS.* The construction of wavelet synopses typically takes place during the *statistics collection process*, whose goal is to create concise statistical approximations for the value distributions of either individual attributes or combinations of attributes in the relations of a DBMS [Acharya et al. 1999; Garofalakis and Gibbons 2001; Ioannidis 2003a]. (Statistics collection is usually an off-line process, carried out during night-time or other system-idling periods.) Once created, such statistical summaries are typically stored as part of the *DBMS-catalog information*. More specifically, a wavelet synopsis comprising B wavelet coefficients is stored as a collection of B pairs $\{< i_j, c_{i_j} >: j = 1, \ldots, B\}$, where i_j and c_{i_j} denote the index and value (respectively) of the jth retained synopsis coefficient.

Wavelet-synopsis information in the DBMS catalog can be exploited for several different purposes. The primary (and, more conventional) use of such summaries is as a tool for enabling effective (compile-time) estimates of the result sizes of relational operators for the purpose of *cost-based query optimization* [Matias et al. 1998]. For instance, estimating the selectivity of a range predicate like $l \leq X \leq h$ is equivalent to estimating the range summation $f_X(l : h) = \sum_{i=l}^{h} f_X[i]$, where f_X is the frequency-distribution array for attribute X. It is not difficult to see that, given a B-coefficient synopsis of the f_X array, computing $f_X(l : h)$ only involves coefficients in $\texttt{path}(f_X[l]) \cup \texttt{path}(f_X[h])$ and, thus, can be estimated by summing only $\min\{B, 2\log N + 1\}$ synopsis coefficients [Chakrabarti et al. 2000, 2001; Garofalakis and Gibbons 2004; Vitter and Wang 1999. A B-coefficient wavelet synopsis can also be easily expanded (in $O(B)$ time) into an $O(B)$-bucket *histogram* (i.e., piecewise-constant) approximation of the underlying data distribution with several possible

[2]The role of the sanity bound is to ensure that relative-error numbers are not unduly dominated by small data values [Haas and Swami 1992; Vitter and Wang 1999].

uses (e.g., as a data visualization/approximation tool [Ioannidis 2003b]). Finally, wavelet-coefficient synopses can enable very fast and accurate *approximate query answers* [Garofalakis and Gibbons 2001] during interactive data-exploration sessions (e.g., initial drill-down query sequences in data-mining tasks [Hellerstein et al. 1997]); in fact, as demonstrated in Chakrabarti et al. [2000, 2001], efficient approximate query processing algorithms for complex Select-Project-Join relational queries can be designed to operate *entirely in the wavelet-coefficient domain*.

3. DETERMINISTIC WAVELET THRESHOLDING FOR MAXIMUM ERROR

Rather than trying to probabilistically control maximum relative error through the optimization of probabilistic measures (like normalized standard error or bias [Garofalakis and Gibbons 2002, 2004]), a more direct solution would be to design *deterministic* thresholding schemes that explicitly minimize maximum-error metrics. Obviously, such schemes can not only guarantee better synopses, but they can also avoid the potential pitfalls of randomized techniques, as well as the space-quantization requirement of Garofalakis and Gibbons [2002, 2004] whose impact on the quality of the final solution is not entirely clear. Unfortunately, as already pointed out by Garofalakis and Gibbons [2004], their DP formulations and algorithms depend crucially on the ability to assign *fractional storage* (i.e., retention probability) values in $(0, 1]$ to individual coefficients, which renders their schemes inapplicable for *deterministic* wavelet thresholding (where the storage assigned to each coefficient is either 0 or 1). In fact, one of the main open problems in Garofalakis and Gibbons [2002, 2004] is whether it is possible to design efficient algorithms for *deterministic* Haar-coefficient thresholding for minimizing maximum-error metrics that are relevant for approximate query answers, such as maximum relative or absolute error in the data approximation. We now attack this problem for both one- and multidimensional Haar wavelets; later in the article (Section 4), we demonstrate that, unlike the probabilistic schemes of Garofalakis and Gibbons, our algorithmic techniques are in fact applicable to a *general class* of approximation-error metrics (which includes, for example, mean weighted relative error).

3.1 One-Dimensional Wavelet Thresholding

In this section, we propose a novel, simple, low polynomial-time scheme based on Dynamic-Programming (DP) for building *deterministic* Haar-wavelet synopses that minimize the *maximum relative or absolute error* in the data-value approximation. More formally, let $\mathsf{relErr}_i = \frac{|\hat{d}_i - d_i|}{\max\{|d_i|, \mathbf{s}\}}$ be an error metric that combines relative error with an appropriate sanity bound $\mathbf{s}$ (as in Garofalakis and Gibbons [2002, 2004]) and, similarly, let $\mathsf{absErr}_i = |\hat{d}_i - d_i|$ denote the absolute approximation error for the ith data value. Our deterministic wavelet thresholding problem can be formally defined as follows:

Deterministic Maximum Relative/Absolute Error Minimization. Given a synopsis space budget B, determine a subset of at most B Haar-wavelet coefficients that minimize the maximum relative (or, absolute) error in the data-value approximation; that is, for $\mathsf{err} \in \{\mathsf{relErr}, \mathsf{absErr}\}$, *minimize* $\max_{i \in \{0,\ldots,N-1\}}\{\mathsf{err}_i\}$.

The important thing to note here is that, unlike the probabilistic normalized standard error or normalized bias metrics employed in Garofalakis and Gibbons [2002, 2004], our relErr and absErr metrics do not have a simple monotonic/additive structure over the Haar-coefficient error tree. The key problem, of course, is that, even though data values are simple linear combinations of coefficients, each coefficient contributes with a *different sign* on different parts of the underlying data domain. Unfortunately, the above facts also imply that the DP formulations in Garofalakis and Gibbons [2002, 2004] are no longer applicable, since the assumed "principle of optimality" for error subtrees is no longer valid; in other words, due to the absence of additive/monotonic structure for the objective, the optimal solution for the subtree rooted at node c_j is not necessarily a combination of the optimal (partial) solutions for c_j's child subtrees.

We now formulate a novel DP recurrence and algorithm for our deterministic maximum-error optimization problem. In a nutshell, the basic idea in our new DP formulation is to condition the optimal error value for an error subtree not only on the root node c_j of the subtree and the amount B of synopsis storage allotted, but also on *the error that "enters" that subtree* through the coefficient selections made on the path from the root to node c_j (excluding c_j itself), that is, coefficient selections on $\texttt{path}(c_j)$. The key observation here is that, since the depth of the error tree is $O(\log N)$, we can afford to tabulate all such possible selections while keeping the running-time of our algorithm in the low-polynomial range.

More formally, let B denote the total space budget for the synopsis, and let T_j be the subtree of the error-tree rooted at node c_j, with $\texttt{coeff}(T_j)$ ($\texttt{data}(T_j)$) denoting the set of coefficient (respectively, data) values in T_j. Finally, let $M[j, b, S]$ denote the optimal (i.e., minimum) value of the maximum error (relative or absolute) among all data values in T_j assuming a synopsis space budget of b coefficients for the T_j subtree, *and* that a subset $S \subseteq \texttt{path}(c_j)$ (of size at most $\min\{B - b, \log N + 1\}$) of proper ancestors of c_j have been selected for the synopsis;[3] that is, assuming a relative-error metric (i.e., err = relErr),

$$M[j, b, S] = \min_{S_j \subseteq \texttt{coeff}(T_j), |S_j| \leq b} \left\{ \max_{d_i \in \texttt{data}(T_j)} \mathsf{relErr}_i \right\},$$

where

$$\mathsf{relErr}_i = \frac{|d_i - \sum_{c_k \in \texttt{path}(d_i) \cap (S_j \cup S)} \mathsf{sign}_{i,k} \cdot c_k|}{\max\{|d_i|, \mathrm{s}\}}.$$

(The case for absolute error (i.e., err = absErr) is defined similarly.) We now formulate a DP recurrence for computing the $M[j, b, S]$ entries; clearly, $M[0, B, \phi]$ gives us the desired optimal error value at the root node of the error tree (the corresponding error-optimal wavelet synopsis can then be built by simply retracing the choices of our DP computation using standard techniques).

[3]Note that all distinct ancestor-coefficient subsets S for a given node c_j can be indexed in our DP array by simply using a bit-string of length $\log N + 1$, or, equivalently, an integer index in the range $0, \ldots, 2^{\log N + 1} - 1 = 2N - 1$.

The *base case* for our recurrence occurs for data (i.e., leaf) nodes in the Haar error tree; that is, for $c_j = d_{j-N}$ with $j \geq N$ (see Figure 1(a)). In this case, $M[j, b, S]$ is not defined for $b > 0$, whereas for $b = 0$ and for each subset $S \subseteq \texttt{path}(d_{j-N})$ (of size $\leq \min\{B, \log N + 1\}$) we define

$$M[j, 0, S] = \frac{|d_{j-N} - \sum_{c_k \in S} \mathsf{sign}_{j-N,k} \cdot c_k|}{r},$$

where $r = \max\{|d_{j-N}|, \mathsf{s}\}$ for $\mathsf{err} = \mathsf{relErr}$, and $r = 1$ for $\mathsf{err} = \mathsf{absErr}$.

In the case of an *internal error-tree node* c_j with $j < N$, our DP algorithm has two distinct choices when computing $M[j, b, S]$, namely either drop coefficient c_j or keep it in the final synopsis. If we choose to *drop* c_j from the synopsis, then it is easy to see that the maximum error from c_j's two child subtrees (i.e., c_{2j} and c_{2j+1}) will be propagated upward; thus, the minimum possible maximum error $M[j, b, S]$ for T_j in this case is simply

$$M_{\text{drop}}[j, b, S] = \min_{0 \leq b' \leq b} \max\{M[2j, b', S], M[2j + 1, b - b', S]\}. \tag{2}$$

Note that, in the above recurrence, S contains proper ancestors of c_j which are clearly proper ancestors of c_{2j} and c_{2j+1} as well; thus, the right-hand side of the recurrence is well defined as the DP computation proceeds bottom-up to reach c_j. If, on the other hand, we choose to *keep* c_j in the synopsis (assuming, of course, $b \geq 1$), the least possible error $M[j, b, S]$ for T_j is computed as

$$M_{\text{keep}}[j, b, S] = \min_{0 \leq b' \leq b-1} \max\{M[2j, b', S \cup \{c_j\}], M[2j + 1, b - b' - 1, S \cup \{c_j\}]\}. \tag{3}$$

(Again, note that the right-hand side of the recurrence is well defined.) The final value for $M[j, b, S]$ is defined as the *minimum* of the two possible choices for coefficient c_j (Eqs. (2) and (3) above); that is, $M[j, b, S] = \min\{M_{\text{drop}}[j, b, S], M_{\text{keep}}[j, b, S]\}$. A pseudo-code description of our optimal DP algorithm for deterministic maximum-error thresholding in one dimension (termed MinMaxErr) is depicted in Figure 3.

3.1.1 *Time and Space Complexity.* Given a node/coefficient c_j at level l of the error tree, our MinMaxErr algorithm considers at most $B+1$ space allotments to the T_j subtree (where the "+1" accounts for the possibility of zero space), and at most 2^{l+1} subsets of ancestors of c_j. Thus, the number of entries in our DP array $M[]$ that need to be computed for c_j is $O(B2^{l+1})$. Furthermore, the time needed to compute each such entry is $O(\log B)$. To see this, note that for any fixed node k and ancestor subset S, $M[k, b', S]$ is a decreasing function of the space allotment b'. Thus, the optimal distribution point b' in Eqs. (2)–(3) (Steps 11–28 in MinMaxErr) can be computed using an $O(\log B)$-time *binary search* procedure, looking for the space allotment where the error values for the two child subtrees are equal or the adjacent pair of cross-over allotments. (To simplify the exposition, this binary-search procedure has been omitted from the pseudo-code description in Figure 3.) Since the total number of error-tree nodes at level l is 2^l, this straightforward analysis directly gives the following

procedure MinMaxErr(W_A , B , root , S, err)
Input: Array $W_A = [c_0, \ldots, c_{N-1}]$ of N Haar wavelet coefficients, space budget B (number of retained coefficients), error-subtree root-node index root, subset of retained ancestors of root node $S \subseteq \texttt{path(root)}$, target maximum error metric err.
Output: Value of $M[\texttt{root}, B, S]$ according to our optimal dynamic program ($M[\texttt{root}, B, S]$.value), decision made for the root node ($M[\texttt{root}, B, S]$.retained), and space allotted to left child subtree ($M[\texttt{root}, B, S]$.leftAllot). (The last two are used for re-tracing the optimal solution to build the synopsis.)
begin
1. **if** ($M[\texttt{root}, B, S]$.computed = **true**) **then**
2. **return** $M[\texttt{root}, B, S]$.value // optimal value already in $M[]$
3. **if** ($N \leq \texttt{root} < 2N$) **then** // leaf/data node
4. **if** ($B = 0$) **then**
5. $M[\texttt{root}, B, S]$.value := $|d_{\texttt{root}-N} - \sum_{c_k \in S} \mathsf{sign}_{\texttt{root}-N,k} \cdot c_k|$
6. **if** (err = relErr) **then**
7. $M[\texttt{root}, B, S]$.value := $\frac{M[\texttt{root},B,S].\text{value}}{\max\{|d_{\texttt{root}-N}|, \mathbf{s}\}}$
8. **endif**
9. **else**
10. $M[\texttt{root}, B, S]$.value := ∞
11. **for** $b := 0$ **to** B **step** 1 **do** // first choice: drop root
12. left := MinMaxErr(W_A , b , $2 * \texttt{root}$, S, err)
13. right := MinMaxErr(W_A , $B - b$, $2 * \texttt{root} + 1$, S, err)
14. **if** (max{ left, right } < $M[\texttt{root}, B, S]$.value) **then**
15. $M[\texttt{root}, B, S]$.value := max{ left, right }
16. $M[\texttt{root}, B, S]$.retained := **false**
17. $M[\texttt{root}, B, S]$.leftAllot := b
18. **endif**
19. **endfor**
20. **for** $b := 0$ **to** $B - 1$ **step** 1 **do** // second choice: keep root
21. left := MinMaxErr(W_A , b , $2 * \texttt{root}$, $S \cup \{\texttt{root}\}$, err)
22. right := MinMaxErr(W_A , $B - b - 1$, $2 * \texttt{root} + 1$, $S \cup \{\texttt{root}\}$, err)
23. **if** (max{ left, right } < $M[\texttt{root}, B, S]$.value) **then**
24. $M[\texttt{root}, B, S]$.value := max{ left, right }
25. $M[\texttt{root}, B, S]$.retained := **true**
26. $M[\texttt{root}, B, S]$.leftAllot := b
27. **endif**
28. **endfor**
29. **endif**
30. $M[\texttt{root}, B, S]$.computed := **true**
31. **return** ($M[\texttt{root}, B, S]$.value)
end

Fig. 3. The MinMaxErr algorithm: Optimal deterministic thresholding for maximum error in one dimension.

upper bound on the overall time complexity of our DP algorithm

$$O\left(\sum_{l=0}^{\log N} 2^l 2^{l+1} B \log B\right) = O\left(B \log B \sum_{l=0}^{\log N} 2^{2l+1}\right) = O(N^2 B \log B).$$

Following our original conference paper [Garofalakis and Kumar 2004], Guha [2004] observed that the above (loose) worst-case bound on the running-time

complexity of our DP scheme can be tightened by a more thorough accounting for the number of entries in our DP array. In a nutshell, his key observation is that, given an internal node c_j at level l of the error tree and a synopsis budget B, the number of coefficients that can be retained inside the T_j subtree is actually upper bounded by $\min\{B, 2^{\log N - l} - 1\}$ (since T_j comprises at most $2^{\log N - l} - 1$ coefficient nodes). Thus, the number of entries for c_j in our $M[]$ array is at most $O(2^{l+1} \min\{B+1, 2^{\log N - l}\}) = O(\min\{B2^{l+1}, 2N\})$, with each entry requiring $O(\min\{\log B, \log N - l\})$ computation time. Summing across all error tree nodes (as above) using these tighter bounds gives an overall time complexity of only $O(N^2)$ for our DP algorithm; the full details can be found in Guha [2004].

With respect to the space requirements of our scheme, employing Guha's observation once again, it is easy to see that the overall size of our DP array $M[]$ is

$$O\left(\sum_{l=0}^{\log N} 2^l \min\{B2^{l+1}, 2N\}\right) \le O\left(N \sum_{l=0}^{\log N} 2^l\right) = O(N^2).$$

It is important to note, however, that our MinMaxErr algorithm does not actually require the entire DP array to be memory-resident at all times. For instance, the results for all descendants of a node c_j are no longer needed and can be swapped out of memory once the results for node c_j have been computed. With this small optimization, it is easy to see that our bottom-up DP computation never requires more than one active "line" (i.e., entries corresponding to a single tree node) of the $M[]$ array *per error-tree level*, where the size of such a line for a node at level l is $O(\min\{B2^{l+1}, 2N\})$ (as discussed earlier). Thus, the size of the memory-resident working set for our DP algorithm drops to only $O(\sum_{l=0}^{\log N} \min\{B2^{l+1}, 2N\}) = O(N \min\{B, \log N\})$. We summarize our discussion for the one-dimensional case in the following theorem.

THEOREM 3.1. *Our* MinMaxErr *algorithm is an optimal deterministic thresholding scheme for building one-dimensional wavelet synopses that minimize the maximum relative error* (*or, maximum absolute error*) *in the data approximation.* MinMaxErr *runs in time* $O(N^2)$ *and has a total-space* (*working-space*) *requirement of* $O(N^2)$ (*respectively,* $O(N \min\{B, \log N\})$).

3.2 Multidimensional Wavelet Thresholding

Our deterministic wavelet thresholding problem becomes significantly more complex for multi-dimensional wavelets, and directly extending our optimal one-dimensional DP formulation to the case of multiple dimensions fails to give a practical solution. Remember that, in the D-dimensional error-tree structure (Section 2.2), even though the tree depth remains $O(\log N)$, each node in the tree now contains up to $2^D - 1$ wavelet coefficients with the same support region and different quadrant signs (Figure 2).[4] This implies that the total number of possible ancestor subsets S for a multidimensional coefficient at a level

[4]Note that, since N here denotes the total number of cells in the multi-dimensional data array, the error tree depth is actually $O(\log N^{1/D}) = \frac{1}{D} O(\log N)$. Similarly, the total number of error-tree

$l = \Theta(\log N)$ is $O(2^{\log N \cdot (2^D - 1)}) = O(N^{2^D - 1})$, rendering the exhaustive-enumeration DP scheme of Section 3.1 completely impractical, even for the relatively small data dimensionalities (i.e., $D = 2$–5) where wavelet-based data reduction is typically employed. (It is well known that, due to the "dimensionality curse", wavelets and other space-partitioning schemes become ineffective above 5–6 dimensions [Chakrabarti et al. 2001; Deshpande et al. 2001; Garofalakis and Gibbons 2004; Gunopulos et al. 2000].)

In this section, we introduce two efficient, polynomial-time *approximation schemes* for deterministic multi-dimensional wavelet thresholding for maximum-error metrics. Both our approximation schemes are based on *approximate dynamic programs* that explore a much smaller number of options than the optimal DP formulation, while offering tunable ϵ-approximation guarantees for the final target maximum-error metric. More specifically, our first scheme can give ϵ-additive-error guarantees for maximum relative or absolute error, whereas our second scheme is a $(1+\epsilon)$-approximation algorithm for maximum absolute error.

3.2.1 *An ϵ-Additive-Error Approximation Scheme for Maximum Absolute/Relative Error Minimization.* Intuitively, our optimal one-dimensional DP scheme is based on exhaustively enumerating, for each error subtree, all the possible error values "entering" the subtree through the choices made on the path to the subtree's root node. The key technical idea in our first approximation scheme is to avoid this exhaustive enumeration and, instead, try to *approximately "cover"* the range of all possible error contributions for paths up to the root node of an error subtree using a much smaller number of (approximate) error values. Our approximation scheme is then going to be based on a much "sparser" DP formulation, that only tabulates this smaller set of error values.

Specifically, let R denote the maximum absolute coefficient value in the error tree, and let $\epsilon < 1$ denote the desired approximation factor. Clearly, the additive contribution to the absolute data-value reconstruction error from *any* possible path in the error tree is guaranteed to lie in the range $\mathcal{R} = [-R2^D \log N, +R2^D \log N]$. Our approximation scheme covers the entire $\mathcal{R}$ range using *error-value breakpoints* of the form $\pm(1+\epsilon)^k$, for a range of (contiguous) integer values for the exponent k. Note that the number of such breakpoints needed is essentially

$$O(\log_{1+\epsilon}(2R2^D \log N)) \approx O\left(\frac{D + \log R + \log\log N}{\epsilon}\right),$$

for small values of $\epsilon < 1$; in other words, $k \in \mathcal{K} = \{0, 1, \ldots, O(\frac{D+\log R+\log\log N}{\epsilon})\}$. Now, let $\mathsf{round}_\epsilon(v)$ be a function that rounds any value $v \in \mathcal{R}$ *down to the closest value* in the set $\mathcal{E} = \{0\} \cup \{\pm(1+\epsilon)^k, k \in \mathcal{K}\}$; that is, letting $l = \log_{1+\epsilon}|v|$, we have $\mathsf{round}_\epsilon(v) = (1+\epsilon)^{\lfloor l \rfloor}$ if $v \geq 1$, $-(1+\epsilon)^{\lceil l \rceil}$ if $v \leq -1$, and 0 otherwise. The

nodes is not N, but rather $O(\frac{N}{2^D-1}) = \frac{1}{2^D-1}O(N)$. Since, for most wavelet applications, the dimensionality D is typically a small constant (e.g., between 2–5), and to simplify the exposition, we tend to ignore these constant multiplicative factors in the development that follows.

DP array $M^a[]$ for our approximation scheme tabulates the values $M^a[j, b, e]$ capturing the approximate maximum error (relative or absolute) in the T_j error subtree (rooted at node j), assuming a space budget of b coefficients allotted to T_j and an approximate/rounded additive error contribution of $e \in \mathcal{E}$ due to proper ancestors of node j being discarded from the synopsis.

The *base case* for the computation of $M^a[]$, that is, the case of a leaf/data node $j \geq N$ in the error tree is fairly straightforward: once again, $M^a[j, b, e]$ is only defined for $b = 0$, and $M^a[j, b, e] = \frac{|e|}{r}$, where r is either $\max\{|d_{j-N}|, \mathrm{s}\}$ or 1 depending on whether we are targeting a relative or absolute error metric.

In the case of an *internal error-tree node* j, remember that each node now corresponds to a *set* $S(j)$ of at most $2^D - 1$ (nonzero) coefficients, and has at most 2^D child subtrees (with indices, say, $j_1, \ldots, j_m$). Assume that we choose to maintain a subset $s \subseteq S(j)$ of node j's coefficients in the synopsis and, for each coefficient $c \in S(j)$, let $\mathsf{sign}(c, j_i)$ denote the sign of c's contribution to the j_i child subtree (Section 2.2); then, we can estimate the least possible maximum error entries $M^a[j, b, e]$, $e \in \mathcal{E}$, for T_j (assuming, of course, that $b \geq |s|$) as

$$\min_{0 \leq b_1 + \ldots + b_m \leq b - |s|} \max_{1 \leq i \leq m} \left\{ M^a \left[j_i, b_i, \mathsf{round}_\epsilon(e + \sum_{c \in S(j) - s} \mathsf{sign}(c, j_i) \cdot c) \right] \right\}.$$

In other words, for a given selected coefficient subset s, we consider all possible allotments of the remaining $b - |s|$ space to children of node j, with the rounded cumulative error that enters those children taking into account the contribution from the dropped coefficients in $S(j) - s$. To avoid the $O(B^{2^D})$ factor in run-time complexity implied by the search of all possible space allotments $b_1, \ldots, b_m$ to child subtrees in the above recurrence, we can simply *order* the search among a node's children (in a manner similar to Garofalakis and Gibbons [2004]). The basic idea is to generalize our approximate DP-array entries to $M^a[\mathbf{j}, b, \mathbf{e}]$, where $\mathbf{j} = (j_1, \ldots, j_k)$ is a *list* of error-tree nodes and $\mathbf{e} = (e_1, \ldots, e_k)$ is a *list* of "incoming" additive errors corresponding to the nodes in $\mathbf{j}$. The DP recurrence for computing $M^a[(j), b, (e)]$ then becomes simply

$$M^a[(j), b, (e)] = \min_{s \subseteq S(j)} \{M^a[(j_1, \ldots, j_m), b - |s|, (e_1, \ldots, e_m)]\},$$

where $e_k = \mathsf{round}_\epsilon(e + \sum_{c \in S(j) - s} \mathsf{sign}(c, j_k) \cdot c)$, for $k = 1, \ldots, m$, and

$$M^a[(j_1, \ldots, j_m), b', (e_1, \ldots, e_m)] = \min_{0 \leq b'' \leq b'} \max\{M^a[(j_1), b'', (e_1)], M^a[(j_2, \ldots, j_m), b' - b'', (e_2, \ldots, e_m)]\}.$$

Intuitively, the first equation states that the approximate error value $M^a[(j), b, (e)]$ at node j is computed as the minimum (over all possible choices for the subset $s \subseteq S(j)$ of retained coefficients at j) of the corresponding approximate error values for the *list* of node j's children. The second equation then computes the approximate error values for this list $(j_1, \ldots, j_m)$ by exploring all possible allotments of a given space budget b' between the *first* child (j_1) and the *list suffix* of all remaining children $(j_2, \ldots, j_m)$ of node j.

To see how this generalization affects the space complexity of our DP formulation note that, assuming a given node j in the error tree: (1) there are at most 2^D possible different values for the list suffix $\mathbf{j}$ of child error-tree nodes of j; and, (2) for each such list suffix $\mathbf{j}$ and each possible incoming error value $e \in \mathcal{E}$ at node j, there are at most 2^{2^D-1} possible different values for the corresponding list of child error values $\mathbf{e}$ (i.e., one for each possible subset $s \subseteq S(j)$ of retained coefficients at j). Thus, given that we obviously have $O(B)$ choices for the b parameter, it is easy to see that the overall space requirements of our generalized DP array are $O(|\mathcal{E}|2^{2^D+D}NB)$. (Again, remember that D is a small constant, typically between 2–5.) In terms of time complexity, note that our generalization also allows for an $O(\log B)$-time search for the breakup of a node's allotment to its children (using binary search in the above recurrence, as described earlier in this article); thus, the overall time complexity of our dynamic program is $O(|\mathcal{E}|2^{2^D+D}NB\log B)$. The following theorem summarizes the results of our analysis for our approximate deterministic thresholding scheme:

THEOREM 3.2. *The previously described approximation scheme for deterministic multidimensional wavelet thresholding discovers an approximate solution that is guaranteed to be within a worst-case additive error of ϵR (respectively, $\epsilon\frac{R}{s}$) of the optimal (i.e., minimum) maximum absolute (respectively, relative) error in time $O(\frac{D+\log R+\log\log N}{\epsilon}2^{2^D+2D}NB\log^2 N\log B)$ and with a total space requirement of $O(\frac{D+\log R+\log\log N}{\epsilon}2^{2^D+2D}NB\log^2 N)$.*

PROOF. Our goal is to demonstrate that our approximation scheme produces a solution that is within a small additive error of the optimal solution. We only consider the case of *absolute error*, as the argument for relative error is similar. Let $M[\mathbf{j}, b, \mathbf{e}] = M[(j_1, \ldots, j_m), b, (e_1, \ldots, e_m)]$ denote the *optimal* absolute error value when we are allowed a coefficient synopsis of size b in the list of error subtrees $(T_{j_1}, \ldots, T_{j_m})$, assuming an incoming additive error of e_{j_k} at subtree T_{j_k} (due to its discarded ancestor coefficients), where $k = 1, \ldots, m$. Thus, our goal is to upper-bound the absolute difference $|M^a[(\texttt{root}), B, (0)] - M[(\texttt{root}), B, (0)]|$.

Let the *height* of an error-tree node denote its distance from the error-tree leaf (i.e., data) nodes (i.e., the leaves are at height 0). We use induction to demonstrate the following claim.

CLAIM 3.3. *If the tree nodes in list $\mathbf{j} = (j_1, \ldots, j_m)$ are at height i, then, for any space allotment b and error-value list $\mathbf{e}$,*

$$|M^a[\mathbf{j}, b, \mathbf{e}] - M[\mathbf{j}, b, \mathbf{e}]| \leq (i+1)\cdot\epsilon 2^D R\log N.$$

Let $N(\mathbf{j})$ denote the number of nodes in the subtrees rooted at all nodes in $\mathbf{j}$; that is, $N(\mathbf{j}) = |T_{j_1} \cup \cdots \cup T_{j_m}|$. Our proof of Claim 3.2.1 is based on an inductive argument on $N(\mathbf{j})$. For the base case, note that if $\mathbf{j}$ comprises only a leaf node, then our claim is clearly true by the construction of our approximate breakpoint set $\mathcal{E}$. Now, assume that the claim is true for all lists $\mathbf{j}'$ with $N(\mathbf{j}') < n$, and let $\mathbf{j}$ be a node list such that $N(\mathbf{j}) = n$; also, assume that the nodes in j are at height i in the error tree.

Consider the simpler case where $\mathbf{j}$ comprises a single node j, that is, $\mathbf{j} = (j)$, and let $(j_1, \ldots, j_m)$ denote the list of children of node j. Recall that

$$M^a[(j), b, (e)] = \min_{s \subseteq S(j)} \{M^a[(j_1, \ldots, j_m), b - |s|, (e_1, \ldots, e_m)]\}, \tag{4}$$

where $e_k = \mathsf{round}_\epsilon(e + \sum_{c \in S(j)-s} \mathsf{sign}(c, j_k) \cdot c)$, for $k = 1, \ldots, m$. Now, if we let $e'_k = e + \sum_{c \in S(j)-s} \mathsf{sign}(c, j_k) \cdot c$, for $k = 1, \ldots, m$, it is easy to see that the corresponding *optimal* error value at node j is defined similarly as

$$M[(j), b, (e)] = \min_{s \subseteq S(j)} \{M[(j_1, \ldots, j_m), b - |s|, (e'_1, \ldots, e'_m)]\}. \tag{5}$$

For the sake of brevity, let $\mathbf{j}_c = (j_1, \ldots, j_m)$, $\mathbf{e} = (e_1, \ldots, e_m)$, and $\mathbf{e}' = (e'_1, \ldots, e'_m)$. By our inductive hypothesis, we have that, for any choice of the s subset,

$$|M^a[\mathbf{j}_c, b - |s|, \mathbf{e}] - M[\mathbf{j}_c, b - |s|, \mathbf{e}]| \leq i \cdot \epsilon 2^D R \log N. \tag{6}$$

Now, assume that we want to compute the optimal error value for $\mathbf{j}_c$ with the *unrounded* incoming additive errors, that is, $M[\mathbf{j}_c, b - |s|, \mathbf{e}']$. Clearly, an upper bound for this error can be obtained by retaining those coefficients corresponding to the optimal value with the *rounded* incoming errors (i.e., $M[\mathbf{j}_c, b - |s|, \mathbf{e}]$), and then adding the rounding error $|e_k - e'_k|$ to all the leaf nodes in the subtree rooted at node j_k ($k = 1, \ldots, m$). In other words, we have

$$|M[\mathbf{j}_c, b - |s|, \mathbf{e}] - M[\mathbf{j}_c, b - |s|, \mathbf{e}']| \leq \max_{k=1,\ldots,m} |e_k - e'_k|.$$

But, since $e_k = \mathsf{round}_\epsilon(e'_k)$, we obviously have $|e_k - e'_k| \leq \epsilon |e'_k| \leq \epsilon 2^D R \log N$. Thus, the above inequality gives

$$|M[\mathbf{j}_c, b - |s|, \mathbf{e}] - M[\mathbf{j}_c, b - |s|, \mathbf{e}']| \leq \epsilon 2^D R \log N. \tag{7}$$

Now, a simple application of the triangle inequality over Inequalities (6) and (7) gives

$$|M^a[\mathbf{j}_c, b - |s|, \mathbf{e}] - M[\mathbf{j}_c, b - |s|, \mathbf{e}']| \leq (i + 1) \cdot \epsilon 2^D R \log N,$$

which, combined with Inequalities (4) and (5), guarantees that $|M^a[(j), b, (e)] - M[(j), b, (e)]| \leq (i + 1) \cdot \epsilon 2^D R \log N$. This completes the proof of Claim 3.2.1 for the case of a single-node list $\mathbf{j} = (j)$.

For the case of a multinode list $\mathbf{j} = (j_1, \ldots, j_l)$, where $l > 1$, recall that

$$M^a[\mathbf{j}, b, \mathbf{e}] = \min_{0 \leq b' \leq b} \max\{M^a[(j_1), b', (e_1)], M^a[(j_2, \ldots, j_l), b - b', (e_2, \ldots, e_l)]\},$$

with the exact same relation holding for the *optimal* error values (using $M[]$ instead of $M^a[]$ above). Thus, once again, applying the inductive hypothesis and the properties of our error-rounding procedure to the right-hand side of the above expression, we have that $|M^a[\mathbf{j}, b, \mathbf{e}] - M[\mathbf{j}, b, \mathbf{e}]| \leq (i + 1) \cdot \epsilon 2^D R \log N$. Thus, our claim holds for this case as well.

Now, a simple application of Claim 3.2.1 with $\mathbf{j} = (\texttt{root})$, gives

$$\begin{aligned} |M^a[(\texttt{root}), B, (0)] - M[(\texttt{root}), B, (0)]| &\leq (1 + \log N)\epsilon 2^D R \log N \\ &= O(\epsilon 2^D R \log^2 N). \end{aligned}$$

Thus, simply setting $\epsilon' = \frac{\epsilon}{O(2^D \log^2 N)}$, gives a worst-case additive deviation of ϵR from the optimal maximum absolute error value. The running-time and space complexity of our approximation scheme for the above value of ϵ' follow immediately from our earlier discussion. □

We should stress here that the bounds in Theorem 3.2 represent a truly pathological worst-case scenario for our scheme, where all coefficients on a root-to-leaf path are of maximum absolute value R. In real-life applications, most of the energy of a data signal (i.e., array) is typically concentrated in a few large wavelet coefficients [Chakrabarti et al. 2001; Gilbert et al. 2001; Matias et al. 1998; Vitter and Wang 1999], which implies that most coefficient values in the error tree will be (much) smaller than R. Thus, for real-life practical scenarios, we would typically expect our approximation scheme to be much closer to the optimal solution than the worst-case deviation bounds asserted in Theorem 3.2.

3.2.2 *A $(1 + \epsilon)$ Approximation Scheme for Maximum Absolute Error Minimization.* We now consider the special case of minimizing maximum absolute error for deterministic multidimensional wavelet thresholding, and propose a novel, polynomial-time $(1 + \epsilon)$-approximation scheme. Our discussion here assumes that all wavelet coefficients are *integers*—we can always satisfy this assumption by appropriately *scaling* the coefficient values. For example, for integer data values d_i (e.g., a multidimensional frequency count array), scaling by a factor of $O(2^{D \log N}) = O(N^D)$ is always guaranteed to give integer Haar coefficients. Let R_Z denote the maximum (scaled) coefficient value in the error tree; as previously, it is easy to see that the additive (integer) contribution to the absolute reconstruction error from any possible path in the Haar error tree is guaranteed to lie in an integer range $\mathcal{R}_Z = [-E_{\max} \log N, +E_{\max} \log N]$, where, of course, $E_{\max} \leq R_Z 2^D$. This observation directly leads to an *optimal pseudopolynomial time algorithm* for our maximum absolute-error minimization problem. (In fact, this pseudopolynomial time scheme directly extends to maximum relative-error minimization as well.) The key idea of our $(1 + \epsilon)$-approximation scheme for absolute error is then to intelligently *scale-down* the coefficients in the error tree so that the possible range of integer additive-error values entering a subtree is *polynomially bounded*.

We start by describing our optimal pseudo-polynomial time scheme. Briefly, our scheme is again based on dynamic-programming over the error tree of integer coefficient values, and follows along similar lines as our additive-error approximation algorithm presented in Section 3.2.1, but without doing any "rounding" of incoming error values. The algorithm starts by precomputing an *error vector* $E(j, s)$ for each internal node j in the error tree and each subset s of the coefficient values in $S(j)$. The $E(j, s)$ vector basically stores the total additive error propagated to each of the children of node j if we decide

to drop the coefficients in $s \subseteq S(j)$ from the synopsis; thus, the $E(j,s)$ vector is indexed by the children of node j, and, for the ith child node j_i of j, we define the corresponding ith error-vector entry as $E(j,s)[i] = \sum_{c \in s} \mathsf{sign}(c, j_i) \cdot c$, where $\mathsf{sign}(c, j_i)$ is the sign of c's contribution to the j_i child subtree. It is easy to see that computing these error vectors $E(j,s)$ for every j and every $s \subseteq S(j)$ takes $O(N 2^{2^D-1} 2^D) = O(N 2^{2^D+D})$ time. (Also, note that the $E_{\max}$ boundary in the integer error range $\mathcal{R}_Z$ can now be defined more precisely as $E_{\max} = \max_{j,s,i}\{E(j,s)[i]\} \leq R_Z 2^D$.)

Our pseudopolynomial DP scheme then constructs a table $M[j,b,e]$ for every node j, space budget $b \leq B$ and error value $e, e \in \mathcal{R}_Z$, where $M[j,b,e]$ denotes the minimum possible maximum absolute error in the T_j subtree, assuming a space budget of b coefficients in T_j and an error contribution of e due to ancestors of node j. As earlier, we define $M[j,0,e] = |e|$ for a leaf/data node j, whereas for an internal node j (with children $j_1, \ldots, j_m$) and assuming that a subset $s \subseteq S(j)$ of coefficients is *dropped* from the synopsis, we compute $M[j,b,e]$ as

$$\min_{0 \leq b_1 + \cdots + b_m \leq b - (2^D - 1 - |s|)} \max_{1 \leq i \leq m} \{M[j_i, b_i, E(j,s)[i] + e]\}.$$

Thus, the final (optimal) value of $M[j,b,e]$ is computed by dropping the coefficient subset $s \subseteq S(j)$ that minimizes the above term. Once again, the $O(B^{2^D})$ factor needed to cycle through all the $b_1, \ldots, b_m$ allotments can be avoided using the generalization described earlier; thus, we have an optimal DP algorithm that runs in time $O(E_{\max} 2^{2^D+D} N \log N B \log B)$. The key observation here is that, if $E_{\max}$ is *polynomially bounded*, then the described DP scheme above is also a polynomial-time algorithm. We use this idea to devise a polynomial-time $(1+\epsilon)$-approximation scheme.

Given a threshold parameter $\tau > 0$, we define a *truncated DP algorithm* as follows. Let $V_{\leq \tau}$ denote the set of error vectors $E(j,s)$ for which the absolute value of each of their entries $E(j,s)[i]$ is at most τ; that is, $V_{\leq \tau} = \{E(j,s) : |E(j,s)[i]| \leq \tau \text{ for all } i\}$. Similarly, let $V_{>\tau}$ denote the set of error vectors $E(j,s)[i]$ not in $V_{\leq \tau}$. Finally, define K_τ as the quantity $K_\tau = \frac{\epsilon \tau}{\log N}$. Our truncated DP algorithm replaces each error vector $E(j,s)$ in the error tree with a *scaled-down* error vector $E^\tau(j,s) = \lfloor \frac{E(j,s)}{K_\tau} \rfloor$ (i.e., $E^\tau(j,s)[i] = \lfloor \frac{E(j,s)[i]}{K_\tau} \rfloor$ for all i), and works with these scaled (integer) error vectors; furthermore, at any node j, our algorithm only considers dropping subsets $s \subseteq S(j)$ such that $E(j,s) \in V_{\leq \tau}$ from the synopsis. More formally, we build a DP array $M^\tau[j,b,e]$ using the scaled error vectors as follows. As previously, for a leaf node j, we define $M^\tau[j,0,e] = |e|$. For an internal node j, our algorithm cycles through *only those subsets* $s \subseteq S(j)$ such that $E(j,s) \in V_{\leq \tau}$ (the $M^\tau[j,b,e]$ entry is undefined if $b < 2^D - 1 - |s|$ for all s such that $E(j,s) \in V_\tau$). The DP recurrence for computing $M^\tau[j,b,e]$ is identical to the one for our pseudo-polynomial scheme above, except for the fact that $E(j,s)$ is replaced by its scaled version $E^\tau(j,s)$.

We claim that the above truncated dynamic program is a *polynomial-time algorithm* for any value of the threshold parameter τ. Indeed, since, at each node j, we only consider dropping coefficient subsets s with error vectors $E(j,s)$ in $V_{\leq \tau}$, the absolute additive error propagated to each child of j is guaranteed

to be at most $\max_i |E^\tau(j,s)[i]| \leq \frac{\tau}{K_\tau} = \frac{\log N}{\epsilon}$. This, of course, implies that the absolute additive error that can enter any subtree in our truncated DP algorithm is at most $\log^2 N/\epsilon$; in other words, the range of possible (integer) incoming error values e for our truncated DP array $M^\tau[]$ is guaranteed to be only $\mathcal{R}_Z{}^\tau = [-\frac{1}{\epsilon}\log^2 N, +\frac{1}{\epsilon}\log^2 N]$. Thus, based on our earlier analysis, the running time of our truncated DP algorithm for a fixed parameter τ is only $O(\frac{1}{\epsilon}2^{2^D+D}N\log^2 N B\log B)$.

Given a threshold parameter τ, our truncated DP algorithm selects a subset $\mathcal{C}_\tau$ of coefficients to retain in the synopsis. Our absolute-error approximation scheme employs the truncated DP algorithm for each value $\tau \in \{2^k : k = 0, \ldots, \lceil \log(R_Z 2^D)\rceil\}$, and finally selects the synopsis $\mathcal{C}_\tau$ that minimizes the maximum absolute error in the data-value reconstruction. Clearly, since we only try $O(D + \log R_Z)$ different values for τ, the running time of our approximation algorithm remains polynomial.

We now demonstrate that the above-described scheme gives a $(1+\epsilon)$-approximation algorithm for maximum absolute error minimization. Consider the optimal maximum absolute error synopsis $\mathcal{C}_{\text{OPT}}$, and let $\mathsf{absErr}(\mathcal{C}_{\text{OPT}})$ denote the corresponding maximum absolute error value. Also, for each internal node j in the error tree, let $s_j^* \subseteq S(j)$ denote the subset of coefficients *dropped* from the optimal synopsis $\mathcal{C}_{\text{OPT}}$ at node j. Finally, let C denote the *maximum absolute value* across all entries in the collection of error vectors $E(j, s_j^*)$ for all j; that is, $C = \max_{j,i}\{E(j, s_j^*)[i]\}$. Clearly, our approximation algorithm is going to try a threshold parameter, say τ', such that $\tau' \in [C, 2C)$. Our goal is to show that the maximum absolute error achieved by $\mathcal{C}_{\tau'}$ (i.e., $\mathsf{absErr}(\mathcal{C}_{\tau'})$) is very close to that achieved by the optimal solution $\mathcal{C}_{\text{OPT}}$.

First, note that, by the definition of C and τ', the optimal solution may drop a subset of coefficients $s \subseteq S(j)$ at node j *only if* $E(j,s) \in V_{\leq\tau'}$. Thus, $\mathcal{C}_{\text{OPT}}$ is obviously a feasible solution to our truncated DP instance (with threshold $= \tau'$). Now, let $\mathsf{absErr}_{\tau'}(\mathcal{C}_{\tau'})$, $\mathsf{absErr}_{\tau'}(\mathcal{C}_{\text{OPT}})$ denote the maximum absolute errors in the $K_{\tau'}$*-scaled instance* for the $\mathcal{C}_{\tau'}$ synopsis (obtained by our truncated DP scheme) and the optimal $\mathcal{C}_{\text{OPT}}$ synopsis, respectively. Given the optimality of our truncated dynamic program for the scaled instance, clearly

$$\mathsf{absErr}_{\tau'}(\mathcal{C}_{\tau'}) \;\leq\; \mathsf{absErr}_{\tau'}(\mathcal{C}_{\text{OPT}}). \tag{8}$$

Let $\mathcal{C}$ be any wavelet synopsis (i.e., subset of Haar coefficients). Obviously, in a $K_{\tau'}$-scaled instance, any error-vector value for $\mathcal{C}$ is represented by $E^{\tau'}(j,s)[i] = \lfloor \frac{E(j,s)[i]}{K_{\tau'}} \rfloor$ which differs by at most 1 from $\frac{E(j,s)[i]}{K_{\tau'}}$; thus, it is easy to see that the scaled and nonscaled maximum absolute errors for the $\mathcal{C}$ synopsis are related as follows

$$\mathsf{absErr}(\mathcal{C}) \;\in\; (K_{\tau'}\mathsf{absErr}_{\tau'}(\mathcal{C}) \pm K_{\tau'}\log N). \tag{9}$$

Applying the above formula for $\mathcal{C} = \mathcal{C}_{\text{OPT}}$ and combining with Eq. (8), we have

$$\begin{aligned}\mathsf{absErr}(\mathcal{C}_{\text{OPT}}) &\geq K_{\tau'}\mathsf{absErr}_{\tau'}(\mathcal{C}_{\text{OPT}}) - K_{\tau'}\log N \\ &\geq K_{\tau'}\mathsf{absErr}_{\tau'}(\mathcal{C}_{\tau'}) - K_{\tau'}\log N,\end{aligned}$$

and, using Eq. (9), once again with $\mathcal{C} = \mathcal{C}_{\tau'}$, we get

$$\mathsf{absErr}(\mathcal{C}_{\tau'}) \;\leq\; K_{\tau'}\mathsf{absErr}_{\tau'}(\mathcal{C}_{\tau'}) + K_{\tau'}\log N.$$

Now, simply combining the last two formulas and substituting $K_{\tau'} = \frac{\epsilon\tau'}{\log N}$, we have

$$\begin{aligned}\mathsf{absErr}(\mathcal{C}_{\tau'}) &\leq \mathsf{absErr}(\mathcal{C}_{\mathrm{OPT}}) + 2K_{\tau'}\log N \\ &\leq \mathsf{absErr}(\mathcal{C}_{\mathrm{OPT}}) + 2\epsilon\tau'. \end{aligned} \tag{10}$$

Our goal now is to demonstrate that $\mathsf{absErr}(\mathcal{C}_{\mathrm{OPT}})$ is at least $\Omega(\tau')$. Our proof relies on the following ancillary claim.

CLAIM 3.4. *Let $\mathcal{C}$ be any Haar-coefficient synopsis and, for any internal node j, let $e_j^{\mathcal{C}}$ denote the absolute value of the additive error coming into the T_j subtree due to ancestors of node j dropped from the $\mathcal{C}$ synopsis. Then,* $\mathsf{absErr}(\mathcal{C}) \geq e_j^{\mathcal{C}}$.

PROOF. We make use of the following key observation. Let T denote the (multi-dimensional) Haar error tree (Figure 2) where *all* coefficient values are retained (to allow for exact data reconstruction). Given an internal node j, let $v(T_j)$ denote the total additive contribution from ancestors of node j during the data-reconstruction process for the data values at the leaves of the T_j subtree (rooted at j). Clearly, this additive contribution $v(T_j)$ is the same for all leaves of T_j, and, letting $\mathtt{path}(j)$ denote the set of all proper ancestor nodes of j in T, $v(T_j)$ is simply computed as $v(T_j) = \sum_{c_i \in \mathtt{path}(j)} \mathsf{sign}_{i,j} \cdot c_i$, where $\mathsf{sign}_{i,j}$ is either $+1$ or -1 depending on how the c_i coefficient affects the data values in the T_j subtree (Section 2.2). One of the basic properties of the Haar-wavelet decomposition is that $v(T_j)$ is exactly the *average* of all the data values at the leaf nodes of the T_j subtree. (This is obviously quite intuitive, as the Haar-wavelet coefficients are essentially trying to summarize the data at successively coarser levels of resolution.)

Now, let $T^{\mathcal{C}}$ denote the error tree where only the coefficients in $\mathcal{C}$ are retained (with all other coefficients set to 0). Note that, by our definition of $e_j^{\mathcal{C}}$, we have $e_j^{\mathcal{C}} = |v(T_j) - v(T_j^{\mathcal{C}})|$. Also, let $j_1, \ldots, j_m$ be the children of node j in T and $T^{\mathcal{C}}$. Our observation above directly implies that $v(T_j)$ is the average of the child-subtree values $v(T_{j_i})$, that is, $v(T_j) = \frac{1}{m}\sum_{i=1}^{m} v(T_{j_i})$ (and, of course, the same also holds for the corresponding approximate values $v(T_j^{\mathcal{C}})$ and $v(T_{j_1}^{\mathcal{C}}), \ldots, v(T_{j_m}^{\mathcal{C}})$). Consider the absolute incoming additive errors $e_{j_i}^{\mathcal{C}}$ at child nodes j_i of j. Our discussion easily implies the following inequality:

$$e_j^{\mathcal{C}} \;=\; \left|v(T_j) - v\left(T_j^{\mathcal{C}}\right)\right| \;=\; \frac{1}{m}\left|\sum_{i=1}^{m} v\left(T_{j_i}\right) - \sum_{i=1}^{m} v\left(T_{j_i}^{\mathcal{C}}\right)\right| \;\leq\; \frac{1}{m}\sum_{i=1}^{m} e_{j_i}^{\mathcal{C}}.$$

Thus, there must exist at least one child node, say j_k, of j such that $e_{j_k}^{\mathcal{C}} \geq e_j^{\mathcal{C}}$. Continuing this argument, we obtain a path from node j to a leaf/data value d_l in the $T_j^{\mathcal{C}}$ subtree such that the absolute additive error entering that leaf, that is, $|\hat{d}_l - d_l|$, is at least $e_j^{\mathcal{C}}$. But then, clearly, $\mathsf{absErr}(\mathcal{C}) \geq |\hat{d}_l - d_l| \geq e_j^{\mathcal{C}}$. This completes the proof for our claim. □

LEMMA 3.5. *Let $\mathcal{C}_{OPT}$ be the optimal maximum absolute error synopsis, and let τ' be as defined above. Then,* $\mathsf{absErr}(\mathcal{C}_{OPT}) > \frac{\tau'}{4}$.

PROOF. By our definition of τ', there exists an error-tree node j such that the subset of dropped coefficients $s_j^* \subseteq S(j)$ for j in the optimal solution $\mathcal{C}_{\text{OPT}}$ satisfies $|E(j, s_j^*)[k]| \geq \frac{\tau'}{2}$ for some child j_k of j. We now consider two possible cases for $e_j^{\mathcal{C}_{\text{OPT}}}$, the absolute additive error entering the subtree rooted at node j in the optimal solution: (1) If $e_j^{\mathcal{C}_{\text{OPT}}} \geq \frac{\tau'}{4}$, then, by Claim 3.4, we immediately have that $\mathsf{absErr}(\mathcal{C}_{\text{OPT}}) \geq \frac{\tau'}{4}$. (2) If $e_j^{\mathcal{C}_{\text{OPT}}} < \frac{\tau'}{4}$, then the absolute value of the additive error entering the subtree rooted at child j_k of j is at least $\frac{\tau'}{2} - \frac{\tau'}{4} = \frac{\tau'}{4}$; thus, Claim 3.4 again implies that $\mathsf{absErr}(\mathcal{C}_{\text{OPT}}) \geq \frac{\tau'}{4}$. This completes the proof. □

Combining Inequality (10) with Lemma 3.5, we have

$$\mathsf{absErr}(\mathcal{C}_{\tau'}) \leq (1 + 8\epsilon)\mathsf{absErr}(\mathcal{C}_{\text{OPT}}).$$

Thus, simply setting $\epsilon' = \epsilon/8$, we have a $(1+\epsilon)$-approximation scheme for maximum absolute error minization in multiple dimensions. The following theorem summarizes our analysis.

THEOREM 3.6. *The above-described approximation scheme for deterministic multidimensional wavelet thresholding discovers an approximate solution that is guaranteed to be within $(1 + \epsilon)$ of the optimal (i.e., minimum) maximum absolute error in time $O(\frac{\log R_Z}{\epsilon} 2^{2^D+D} N \log^2 NB \log B)$ and with a total space requirement of $O(\frac{1}{\epsilon} 2^D N \log^2 NB)$.*

4. EXTENSION TO GENERAL ERROR METRICS

Our discussion thus far has focused on the minimization of maximum-error metrics (like, maximum relative error) in the approximate data-value reconstruction. However, as we demonstrate in this section, our algorithmic solutions have much more general applicability; once again, this is in sharp contrast with earlier probabilistic thresholding schemes [Garofalakis and Gibbons 2002, 2004] that can handle *only* maximum-error metrics. Consider the natural class of *distributive error metrics* defined formally below.

Definition 4.1 (*Distributive Error Metrics*). Consider an approximation of a (one- or multidimensional) data array A, and let $f(R)$ denote the error in the data-value approximation over the (one- or multidimensional) range of values R in A. We say that the error metric $f()$ is *distributive* if and only if, for any collection of disjoint ranges $R_1, \ldots, R_k$, there exists some combining function $g()$ such that the error over the entire region $\cup_{i=1}^k R_i$ can be defined as

$$f\left(\cup_{i=1}^k R_i\right) = g(f(R_1), \ldots, f(R_k)).$$

The class of distributive error metrics defined above encompasses several important approximation-error functions. For instance, the maximum-error metrics defined in Sections 2 and 3 are clearly distributive (with the combining function $g()$ being simply the max{} of its input arguments). Furthermore, it is not hard to see that most important cumulative approximation-error

metrics are also naturally distributive, including the *mean relative error* $\frac{1}{N}\sum_i \text{relErr}_i$ and the *L_p-norm error* $[\frac{1}{N}\sum_i |\hat{d}_i - d_i|^p]^{\frac{1}{p}}$ (for any $p \geq 0$) in the data reconstruction, as well as the *weighted variants* of these metrics ($\frac{1}{N}\sum_i w_i \cdot \text{relErr}_i$ and $[\frac{1}{N}\sum_i w_i \cdot |\hat{d}_i - d_i|^p]^{\frac{1}{p}}$, respectively), where different weights w_i (typically normalized, such that $\sum_i w_i = 1$) are associated with the errors for different values in the underlying data domain. (Such weights are an important tool for capturing the importance of individual data values, e.g., based on the nonuniformities of the observed query workload [Muthukrishnan 2004].)

The key observation here is that our optimal DP formulation of Section 3.1 can, in fact, be easily adapted to work with *any distributive error metric*. Given such an error metric $f()$, the basic idea is to define the DP array $M[j, b, S]$ as well as the DP recurrences for computing its entries using the general form of the $f()$-metric and the corresponding combining function $g()$ that allows us to "distribute" the computation of $f()$ over sub-ranges of the data domain (Definition 4.1). More specifically, following along similar lines as in Section 3.1, we define the base case for our DP recurrence (i.e., for leaf nodes $c_j = d_{j-N}$, $j \geq N$) as $M[j, 0, S] = f(\{d_{j-N}\}|S)$ for each subset $S \subseteq \texttt{path}(d_{j-N})$, where $f(\{d_{j-N}\}|S)$ denotes the value of the $f()$ error metric at data value d_{j-N} assuming the coefficients in S are kept in the synopsis. Now, in the case of an internal node c_j with $j < N$, we define the optimal error $M[j, b, S]$ when coefficient c_j is either dropped from or kept in the synopsis in a manner similar to Eqs. (2)-(3) with the key difference that max{} is now replaced by the combining function $g()$ for our distributive error metric $f()$. More formally, our general DP recurrence for $f()$ simply defines $M[j, b, S] = \min\{M_{\text{drop}}[j, b, S], M_{\text{keep}}[j, b, S]\}$, where

$$M_{\text{drop}}[j, b, S] = \min_{0 \leq b' \leq b} g(M[2j, b', S], M[2j+1, b-b', S]), \qquad \text{and}$$

$$M_{\text{keep}}[j, b, S] = \min_{0 \leq b' \leq b-1} g(M[2j, b', S \cup \{c_j\}], M[2j+1, b-b'-1, S \cup \{c_j\}]).$$

Example 4.1. As a more concrete problem setting, consider the adaptation of our optimal DP formulation for the case of the *mean weighted relative error* metric $\frac{1}{N}\sum_i w_i \cdot \text{relErr}_i$. Since the averaging factor N is basically constant (i.e., the size of the data domain), it is obviously sufficient to optimize the cumulative weighted relative error; that is, we seek to minimize

$$f(\{d_1, \ldots, d_N\}) = \sum_i w_i \cdot \text{relErr}_i = \sum_i \frac{w_i \cdot |\hat{d}_i - d_i|}{\max\{|d_i|, \texttt{s}\}},$$

(for a given sanity bound value $\texttt{s}$).

For the base case of data (i.e., leaf) nodes $c_j = d_{j-N}$ with $j \geq N$, we define $M[j, 0, S]$ (for each subset $S \subseteq \texttt{path}(d_{j-N})$) as the weighted relative error at value d_{j-N} (assuming the coefficients in S are retained in the synopsis); that is,

$$M[j, 0, S] = f(\{d_{j-N}\}|S) = \frac{w_{j-N} \cdot |d_{j-N} - \sum_{c_k \in S} \text{sign}_{j-N,k} \cdot c_k|}{\max\{|d_{j-N}|, \texttt{s}\}}.$$

For the case of internal nodes c_j with $j < N$, note that the combining function $g()$ for our cumulative weighted relative error metric is simple summation; thus,

we define $M[j, b, S] = \min\{M_{\text{drop}}[j, b, S], M_{\text{keep}}[j, b, S]\}$, where

$$M_{\text{drop}}[j, b, S] = \min_{0 \le b' \le b} \{M[2j, b', S] + M[2j+1, b-b', S]\}, \quad \text{and}$$
$$M_{\text{keep}}[j, b, S] = \min_{0 \le b' \le b-1} \{M[2j, b', S \cup \{c_j\}] + M[2j+1, b-b'-1, S \cup \{c_j\}]\}.$$

Since our optimal DP formulation extends directly to multidimensional data arrays and wavelets, the above-described generalizations are directly applicable to multiple dimensions as well; that is, of course, modulo the super-exponential explosion in space/time complexity with increasing dimensionality (Section 3.2). Furthermore, note that both our efficient approximation schemes for near-optimal thresholding in multipledimensions are essentially based on *sparser versions* of this optimal multidimensional dynamic program. Thus, our maximum-error approximation algorithms can also be adapted to work with other distributive error metrics as well with similar approximation guarantees; for instance, the techniques and analyses in Sections 3.2.1–3.2.2 can easily be extended to (approximately) optimize for the corresponding cumulative error metrics (i.e., mean absolute and mean relative error).

5. EXPERIMENTAL STUDY

In this section, we present the results of an empirical study we have conducted using the algorithmic techniques developed in this article for building deterministic wavelet synopses optimized for general error metrics. The primary objective of our study is to verify the effectiveness of our deterministic synopsis construction techniques in reducing *relative-error metrics* in the data-value reconstruction compared to the probabilistic wavelet synopses of Garofalakis and Garofalakis and Gibbons [2002, 2004], as well as the more conventional, L_2-optimal wavelet summaries (Section 2.3). To this end, we have experimented with a variety of different synthetic and real-world data sets. The major findings of our study can be summarized as follows:

—*More Consistent, Low Relative Error Data Reconstruction.* By directly optimizing for the desired relative-error metric and avoiding the randomness of probabilistic coin flips and randomized value rounding, our deterministic thresholding algorithms enable a more consistent, lower-error approximation of the original data values. The end result is a consistently smaller *mean relative error* in data reconstruction across the entire underlying data domain.

—*More Consistent, Improved Quality Guarantees for Individual Data-Value Approximation.* Again, by avoiding probabilistic coin flips when it comes to minimizing the *maximum relative error* at individual data points, our deterministic optimization strategies can offer tighter quality guarantees for reconstructed data values than probabilistic wavelets.

All experiments reported in this section were performed on a 3.0-GHz Pentium-IV machine with 1-GB of main memory running Red Hat Enterprise Linux 4.

5.1 Testbed and Methodology

5.1.1 *Techniques, Parameter Settings, and Approximation-Error Metrics.* Our experimental study compares the deterministic wavelet thresholding techniques developed in this paper with: (1) the two winning probabilistic wavelet schemes of Garofalakis and Gibbons [2002, 2004], namely the MinRelVar and MinRelBias algorithms, designed to minimize the maximum normalized standard error and the maximum normalized bias, respectively and, (2) conventional, greedy wavelet coefficient thresholding (denoted by L2opt) for minimizing the overall L_2 error in the approximation (Section 2.3). More specifically, we experimented with two adaptations of our optimal deterministic DP algorithm for coefficient thresholding (Sections 3–4), MinMaxRelErr and MinAvgRelErr, designed to minimize the maximum and mean relative error in data-value reconstruction, respectively. Our MinMaxRelErr and MinAvgRelErr implementations employ the optimizations discussed near the end of Section 3.1 to guarantee a worst-case running time of $O(N^2)$ and a working-space requirement of $O(N \min\{B, \log N\})$ (by computing our dynamic program in a bottom-up fashion, swapping out unnecessary lines of the DP array for descendant nodes). For the two probabilistic schemes (MinRelVar and MinRelBias), we utilize the quantization and perturbation parameter settings already suggested in Garofalakis and Gibbons [2002, 2004].

We consider two key metrics to gauge the accuracy of the different wavelet-synopsis techniques. Let d_i ($\hat{d}_i$) denote the ith accurate (respectively, reconstructed) value in the domain, and let s be the specified sanity bound for the approximation. The *maximum relative error* in the data reconstruction is $\max_i\{\frac{|\hat{d}_i - d_i|}{\max\{d_i, s\}}\}$. The *mean relative error* is $\frac{1}{N}\sum_{i=1}^{N}\frac{|\hat{d}_i - d_i|}{\max\{d_i, s\}}$. As suggested in Garofalakis and Gibbons [2002, 2004], the maximum relative error value can be returned to the user as a *guaranteed-error bound* for the reconstruction of any individual data value, and both probabilistic techniques (MinRelVar and MinRelBias) as well as our MinMaxRelErr algorithm are designed to help minimize this error. However, it is based solely on the largest error, and hence it provides a less informative comparison than a mean relative error metric. Thus, following Garofalakis and Gibbons [2002, 2004], we will also primarily use the mean relative error for the comparisons in this section (the corresponding results for maximum relative error are qualitatively similar). Remember that, while our MinAvgRelErr strategy directly optimizes for mean relative error in the data reconstruction, there are no known probabilistic thresholding schemes targeting mean relative error metrics.

5.1.2 *Synthetic Data Generation.* We ran our techniques against several different one-dimensional synthetic data distributions, generated as follows. First, a Zipfian data generator was used to produce Zipfian frequencies for various levels of skew (controlled by the z parameter of the Zipfian), numbers of distinct values N, and total frequency values (i.e., data-tuple counts). We varied the z parameter between 0.3 (low skew) to 2.0 (high skew), the number of distinct values N between $16,384$ and $65,536$, and the tuple count between 2×10^9 and 16×10^9. Next, a permutation step was applied on the generated

Zipfian frequencies to order them over the data domain; we experimented with four different permutation techniques: (1) *"NoPerm"* basically leaves the ordering as specified by the Zipfian data generator, that is, smaller values have higher frequencies; (2) *"Normal"* permutes the frequencies to resemble a bell-shaped normal distribution, with the higher (lower) frequencies at the center (respectively, ends) of the domain; (3) *"PipeOrgan"* permutes the frequencies in a "pipe-organ"-like arrangement, with higher (lower) frequencies at the two ends (respectively, center) of the data domain; and, (4) *"Random"* permutes the frequencies in a completely random manner over the data domain. (Our synthetic data generator emulates that of Garofalakis and Gibbons [2002, 2004], using permutations that try to capture "smooth" Zipfian data distributions of various shapes as well as more random, noncanonical distribution patterns.) Following Garofalakis and Gibbons [2002, 2004], we determined the value of the sanity bound s in our relative-error metrics for each input data set as the 10-percentile value in the data (i.e., 90% of the data points had values greater than s).

5.2 Experimental Results—Accuracy on Synthetic Data Sets

We present some of our experimental results with synthetic data sets for different frequency permutations and settings of Zipfian skew. The numbers shown in this section were obtained using a data domain of $N = 32,768$ distinct values, a tuple count of 4×10^9, and varying the number of retained synopsis coefficients B between 25 and 2,000—we observed similar results for other parameter settings. As suggested by Garofalakis and Gibbons [2002, 2004], once the probabilistic MinRelVar and MinRelBias schemes determined their respective coefficient-retention probabilities, *five trials* of the (randomized) coefficient-selection process using different random seeds were performed, and the synopsis was selected that gave the *best* (i.e., smallest) value for the observed mean/maximum relative error (depending on the specific error measure of interest). The goal, of course, was to avoid worst-case sequences of coin flips that could result in poorly performing wavelet synopses [Garofalakis and Gibbons 2002, 2004]. In our study, however, in order to quantify the impact of such potential bad coin-flip sequences on the quality of the probabilistic wavelet solutions, we also obtained error measurements for the *worst* synopsis out of our five trials (i.e., the one with the largest observed mean/maximum relative error).

5.2.1 *Data-Value Reconstruction Relative Errors.* Figure 4 depicts the mean and maximum relative error numbers in the data-value reconstruction obtained by our deterministic optimization techniques, the probabilistic MinRelVar and MinRelBias algorithms (for both the best- and worst-case randomized coefficient selections), and the conventional L2opt thresholding scheme on a "Normal" Zipfian data set with skew parameters of $z = 1.0$ (a, b), $z = 1.5$ (c, d), and $z = 2.0$ (e, f). It is easy to see that our deterministic MinMaxRelErr and MinAvgRelErr schemes consistently outperform the best-case probabilistic MinRelVar and MinRelBias synopses, offering improvements of up to over 100% in terms of both mean and maximum relative error, with the benefits becoming more evident for small synopsis sizes (e.g., $B \leq 200$) and larger values

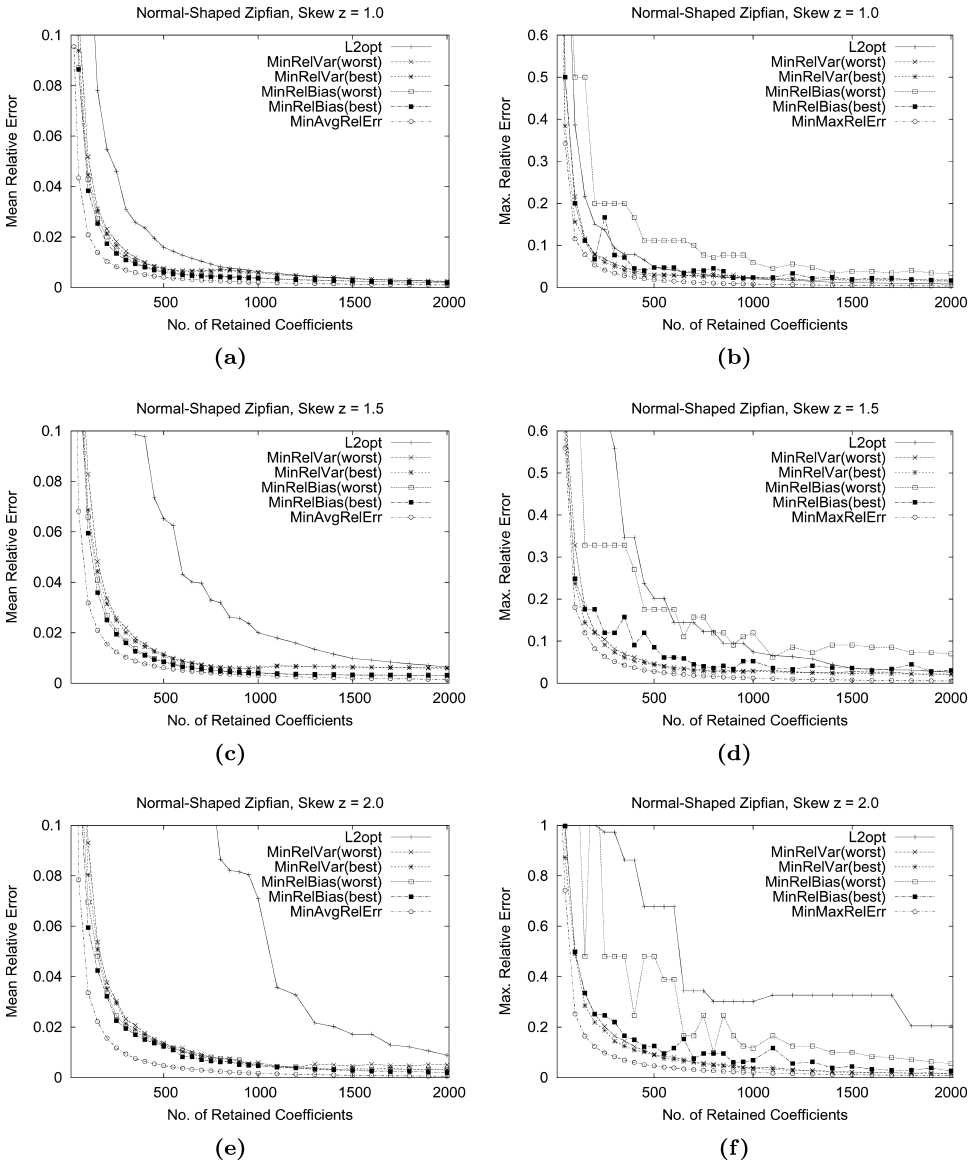

Fig. 4. Mean and maximum data-reconstruction error for "Normal" Zipfian permutations with data skew values of $z = 1.0$ (a, b), $z = 1.5$ (c, d), and $z = 2.0$ (e, f).

of data skew z. Furthermore, the numbers in Figure 4 also demonstrate that a bad sequence of coin flips can, in fact, significantly deteriorate the quality of the probabilistic wavelet synopses constructed by the MinRelVar and MinRelBias schemes. (The impact of such bad coin-flip sequences is particularly evident for the MinRelBias algorithm and larger values of the skew parameter z.) Our numbers also clearly show that all our relative-error schemes significantly outperform conventional L2opt thresholding in terms of both mean and maximum

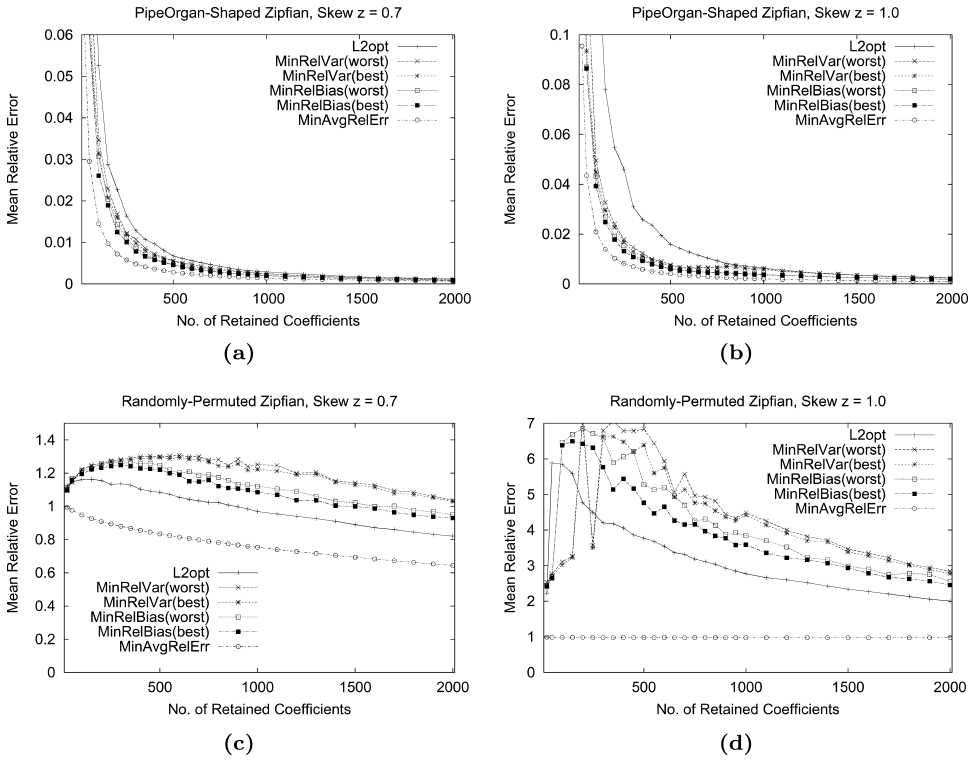

Fig. 5. Mean data-reconstruction error for "PipeOrgan" (a, b) and "Random" (c, d) Zipfian permutations with data skew values of $z = 0.7$ and $z = 1.0$.

relative error on "Normal" Zipfian data (with the benefits, once again, becoming more pronounced for larger data-skew values). This is obviously expected since, by optimizing for overall L_2 error, the L2opt algorithm can result in very large relative errors for nondominant frequency values (see also Garofalakis and Gibbons [2002, 2004]).

Similar trends can be observed in the plots of Figure 5 which depict the mean relative-error numbers for "PipeOrgan" (a, b) and "Random" (c, d) Zipfian data sets with skew $z = 0.7$ and $z = 1.0$. The numbers for our "PipeOrgan" frequency arrangement are essentially identical to those for "Normal" Zipfian data (Figure 4), since both represent fairly smooth data distributions (with somewhat different overall shapes). In the case of randomly-permuted Zipfian data (Figures 5(c, d)), our MinAvgRelErr deterministic scheme offers even more significant improvements (typically ranging between 20% and well over 100%) in terms of mean relative error compared to both best-case probabilistic (MinRelVar and MinRelBias) synopses and conventional L_2-optimal synopses. The impact of coin-flip variability on the quality of the data reconstruction for the two probabilistic schemes is, once again, evident in our "Random" data numbers (especially for small synopsis sizes). Note that, by randomly permuting Zipfian frequencies, "Random" typically results in irregular, non-smooth data distributions which are quite difficult to approximate well by compact Haar

wavelet synopses (hence, the higher error numbers compared to, say, "Normal" and "PipeOrgan" data sets). For such "difficult" data distributions, L2opt synopses can often result in lower mean relative errors compared to probabilistic MinRelVar/MinRelBias synopses (an observation also made by Garofalakis and Gibbons [2002, 2004]). Still, as our results demonstrate, our deterministic MinAvgRelErr synopses can offer even more consistent and significant accuracy benefits over their probabilistic and L_2-optimal counterparts for such "difficult" data sets. Overall, by directly optimizing for the objective error metrics, our deterministic thresholding algorithms enable significantly better and more consistent mean and maximum relative error behavior over a wide range of data sets and parameters.

5.2.2 *Effect of Data Skew.* The plots in Figure 6 depict the ratios between the mean relative error values obtained by our deterministic MinAvgRelErr scheme and the (best-case) MinRelVar/MinRelBias probabilistic synopses and conventional L2opt synopses for "Normal" (a, c, e) and "Random" (b, d, f) Zipfian distributions with varying values of the skew parameter z. (The corresponding ratios for the worst-case probabilistic synopses are omitted to avoid cluttering the figures.) It is easy to see that our MinAvgRelErr synopses guarantee consistently lower mean relative errors across the range of Zipfian skew parameters, with the benefits being, once again, more evident in the case of the more difficult to summarize "Random" data sets. Furthermore, the relative error improvements offered by our (optimal) deterministic thresholding strategies, in general, tend to increase as the level of skew in the data goes up; in fact, we had to omit the curves for high data skew ($z = 2.0$) from several of our plots since they resulted in error ratios well outside our plot ranges. The impact of data skew on relative-error benefits is again more pronounced for the "difficult", randomly-permuted Zipfian data distributions (Figures 6(b, d, f)).

5.3 Experimental Results—Accuracy on Real-World Data Sets

To explore how our techniques performed on real-world data distributions, we employed two distinct real-life data collections:

—The `Weather` data set (`www-k12.atmos.washington.edu/k12/grayskies/`) comprises different meteorological measurements obtained from a station at the University of Washington Department of Atmospheric Sciences. For our experiments here, we extracted $N = 524,288$ real-valued data points for six measured quantities (wind speed, wind peak, solar irradiance, relative humidity, air temperature, and dewpoint temperature) over the 2002 calendar year.

—The `Corel` Image Features data set (`kdd.ics.uci.edu/databases/CorelFeatures/CorelFeatures.html`) contains several different image features extracted from a Corel collection of $N = 68,040$ photo images from various categories. For our experiments, we primarily made use of individual real-valued attributes from the color histogram and color-histogram layout features. In a nutshell, the color histogram data describes each Corel image in terms of 32 real-valued features corresponding to individual color

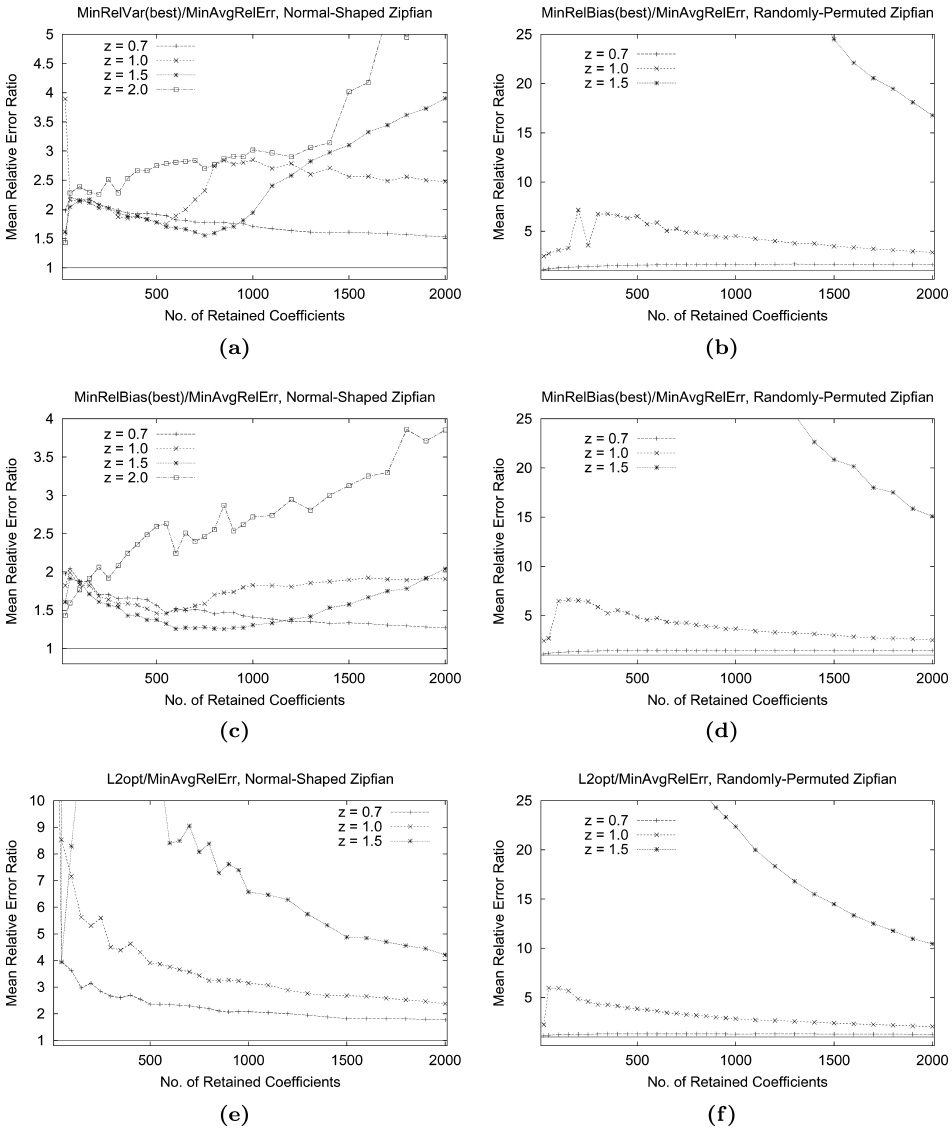

Fig. 6. Effect of Skew on Data Reconstruction: Ratio of mean relative reconstruction errors between MinAvgRelErr and (best-case) MinRelVar/MinRelBias and L2opt synopses for "Normal" (a, c, e) and "Random" (b, d, f) Zipfian data distributions.

densities (in the entire image) for an 8×4 partitioning of the HSV color space. Similarly, the color-histogram layout table comprises 32-dimensional, real-valued tuples, each giving a 4×2 HSV color histogram for 4 subimages (one horizontal + one vertical split) of each original Corel image.

In all cases, we treat the sequences of data points (or, subsequences thereof) as our input data array (with a specified number of entries N), which we compress using the different wavelet-based techniques explored in our study.

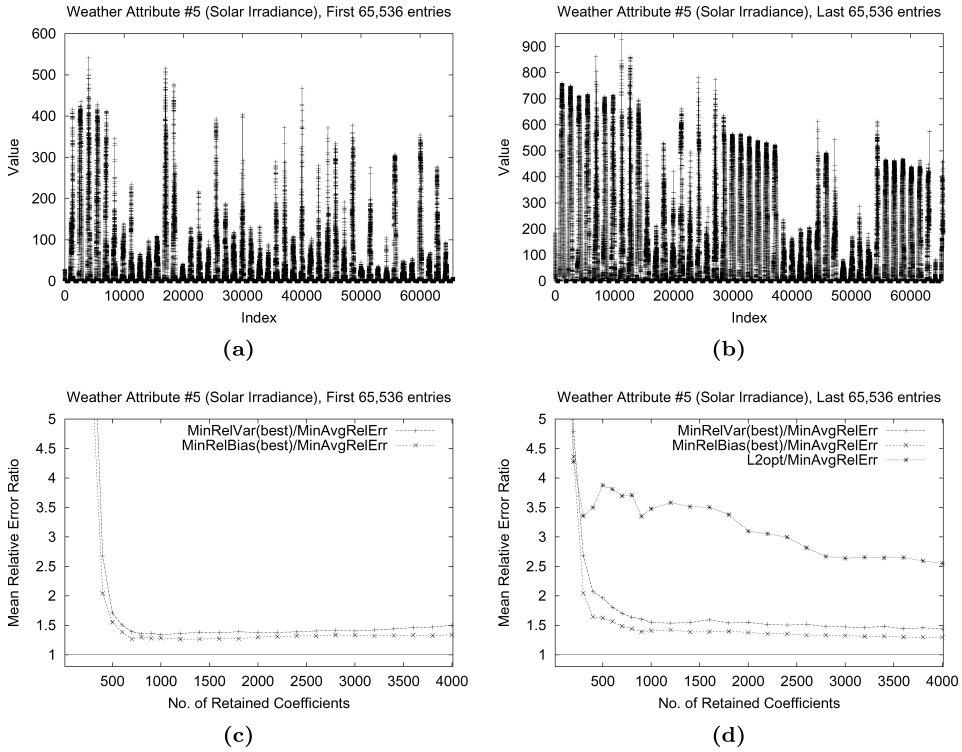

Fig. 7. (a, b) First and last segment of the `Weather` solar-irradiance measurements. (c, d) Ratio of mean relative errors between MinAvgRelErr and (best-case) MinRelVar/MinRelBias and L2opt synopses for the two `Weather` solar-irradiance segments.

We report results from a representative subset of the data sequences for each of our three real-life data sets—qualitatively similar results were obtained for other attributes in our data collections.

Figures 7(a, b) depict the histograms of data-array values (i.e., the input to our synopsis techniques) for two distinct segments of $N = 65,536$ entries of the *solar-irradiance* attribute in the `Weather` data set. Figures 7(c, d) show the corresponding mean relative error ratios between our optimal deterministic MinAvgRelErr scheme and the (best-case) MinRelBias/MinRelVar probabilistic synopses and conventional L2opt synopses as the number of retained coefficients B is varied between 25 and $4,000$. Clearly, our MinAvgRelErr synopses offer consistent, significant accuracy benefits over both L_2-optimal and (best-case) probabilistic wavelet summaries, especially in the case of space-constrained synopses (e.g., $B \leq 500$). Furthermore, all three relative error strategies (MinAvgRelErr, MinRelVar, and MinRelBias) perform consistently much better than L2opt—in fact, the L2opt curve for the first, more "spiky" segment of our solar-irradiance data had to be omitted since its error ratios were well outside our plot range.

Figure 8 depicts the corresponding data-array histograms and mean relative error ratio numbers for two distinct segments of $N = 65,536$ entries of the

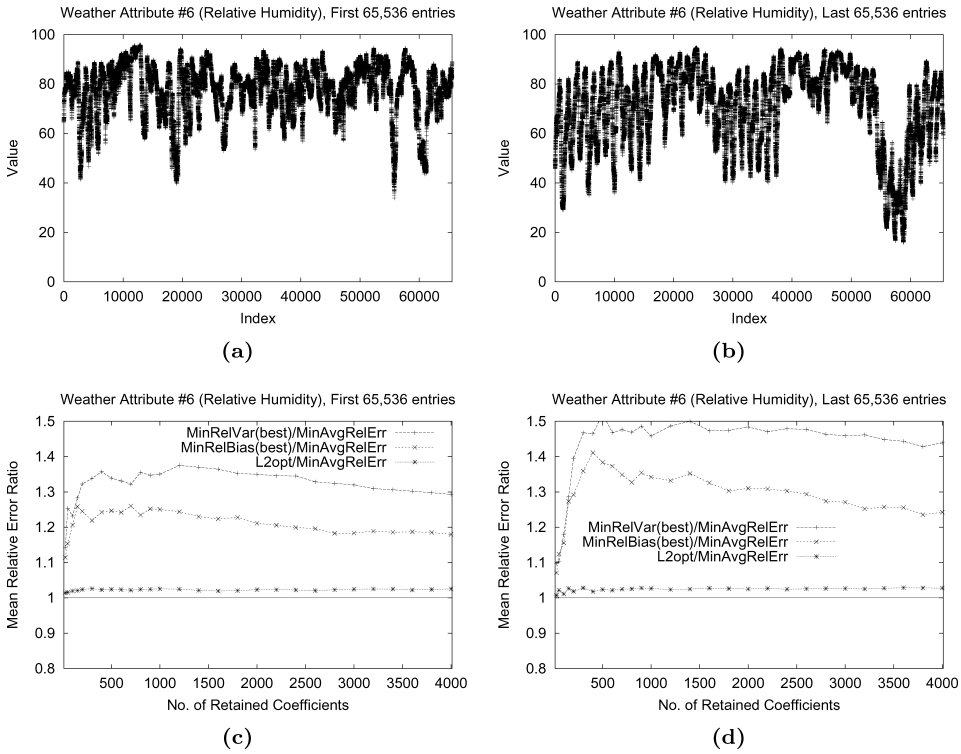

Fig. 8. (a, b) First and last segment of the `Weather` relative-humidity measurements. (c, d) Ratio of mean relative errors between MinAvgRelErr and (best-case) MinRelVar/MinRelBias and L2opt synopses for the two `Weather` relative-humidity segments.

relative-humidity measurements in the `Weather` data set. Clearly, the relative-humidity attribute exhibits a significantly smoother distribution than solar irradiance, thus resulting in smaller relative errors (and relative-error ratios) for all schemes. It is also interesting to note that, in this case, the relative ordering of the conventional L2opt strategy and the probabilistic MinRelVar/MinRelBias strategies is reversed, with L2opt giving consistently better accuracy numbers. Still, our MinAvgRelErr synopses continue to outperform all the other strategies in our study, offering consistent and substantial relative-error benefits throughout the range of synopsis sizes.

Finally, Figure 9 shows the histograms and corresponding mean relative error ratio numbers (for synopsis sizes $B = 25$ to $4,000$) for two distinct color-histogram image features (attribute numbers 1 and 5 out of a total of 32 features) in the `Corel` data set. (We obtained qualitatively similar results with other color histogram and color-histogram layout attributes.) Due to their high variability and numerous large "spikes", these data arrays turned out to be particularly difficult for the conventional L2opt strategy, resulting in large relative-error ratios that were well outside the range of our plots. Compared to the (best-case) probabilistic MinRelVar/MinRelBias algorithms, our proposed MinAvgRelErr strategy once again emerges as a clear winner, giving consistent and very significant accuracy benefits across the range of synopsis sizes.

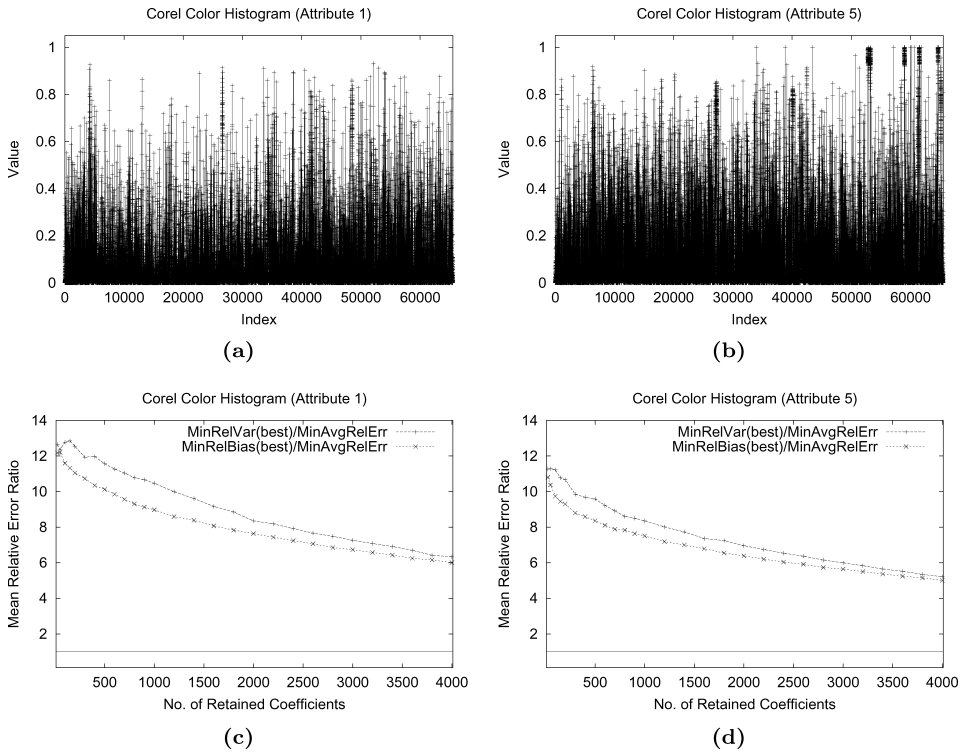

Fig. 9. (a, b) First and fifth color-histogram attributes in the `Corel` image features data set. (c, d) Ratio of mean relative errors between MinAvgRelErr and (best-case) MinRelVar/MinRelBias synopses for the two `Corel` data arrays.

5.4 Experimental Results—Running Time and Memory Requirements

In order to test the scalability of our wavelet-thresholding techniques to large domain sizes N and synopsis sizes B, we measure the running times and memory requirements for the different relative-error minimization schemes employed in our experimental study. (The L2opt thresholding algorithm is obviously quite trivial and, thus, excluded from our discussion here.) We use MinMaxRelErr as a representative of the optimal deterministic-thresholding schemes proposed in this paper, and MinRelVar as a representative of the probabilistic-thresholding schemes [Garofalakis and Gibbons 2002, 2004]. (Note that MinAvgRelErr and MinRelBias are based on essentially identical dynamic programs as MinMaxRelErr and MinRelVar (respectively), thus giving very similar running-time and memory-requirement numbers.) We also focus primarily on the `Weather` relative-humidity data arrays, since we observed very little variation in the running times and memory requirements of the algorithms across different data sets (for the same values of N and B).

Figures 10(a, b) depict the observed running times for the MinMaxRelErr and MinRelVar algorithms as a function of the domain size N (for fixed $B = 2,000$) and the synopsis size B (for fixed $N = 32,768$). The quadratic dependence of the

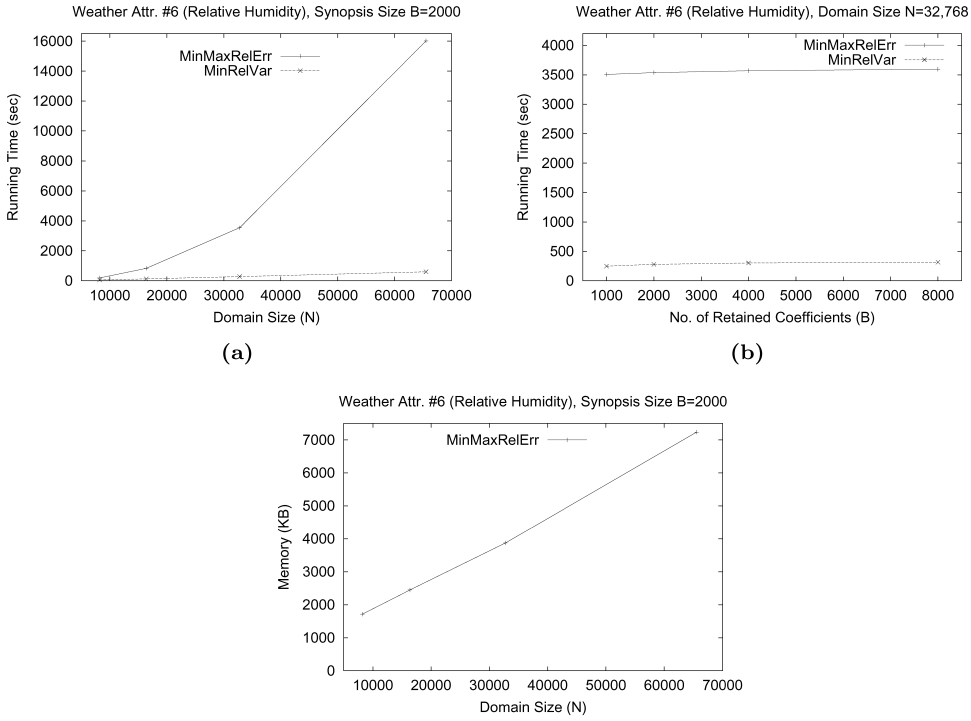

Fig. 10. Running times (a, b) and memory requirements (c) for our relative-error thresholding schemes as a function of the domain size (N) and the synopsis size (B).

running times of our dynamic-programming scheme on the data-domain size N is immediately apparent—even though the running times of MinMaxRelErr start at the same levels as those of MinRelVar for small N (roughly a couple of minutes for $N = 8,192$), they quickly jump to a little less than an hour for $N = 32,768$ and about four hours for $N = 65,536$. On the other hand, the time complexity of the probabilistic MinRelVar scheme is only linear in N, keeping its running times at about ten minutes even for $N = 65,536$. Still, given that statistics construction is typically an off-line process (e.g., run during night-time or other system-idling periods), as well as the significant accuracy benefits of our optimal deterministic strategies (Sections 5.2–5.3), we believe that our schemes can provide an effective solution for data summarization in decision-support environments. Also, note that, in situations where the quadratic time complexity of our optimal dynamic programs becomes a limiting factor, our provably near-optimal approximation algorithms of Section 3.2 (whose complexity is roughly linear in N) can provide a more efficient alternative. Figure 10(b) shows that, as expected, the observed running times for our optimized MinMaxRelErr algorithm are essentially independent of the synopsis size B, demonstrating only a very slight increase as B is varied from 1,000 to 8,000 coefficients. The same observation roughly holds for the probabilistic MinRelVar algorithm as well (even though its worst-case time complexity is linear in B [Garofalakis and Gibbons

2002, 2004]), mostly due to the large amount of pruning achieved at the lower levels of the coefficient error tree.

With respect to memory requirements, our experimental runs verify that the amount of memory needed by our optimized MinMaxRelErr algorithm is essentially *independent of the synopsis size* B for the range of parameter settings considered in our study. (Remember that the corresponding asymptotic working-space bound is $O(N \min\{B, \log N\})$.) More specifically, with the domain size fixed at $N = 32,768$, the maximum memory requirement for our MinMaxRelErr implementation is approximately only 3.75-MB, regardless of the specified synopsis size B (varied between $1,000$ and $8,000$ coefficients). The corresponding total-space requirements for the dynamic-programming array of the probabilistic MinRelVar algorithm are significantly higher (between 112-MB–125-MB)—this is due to the fact that, even though the size of the MinRelVar DP array is linear in N, B, it also has a linear dependence on other algorithm constants (i.e., the integer quantization parameter). (We should, of course, note here that, using the working-space optimizations discussed in Garofalakis and Gibbons [2002, 2004] typically lowers the memory needs of MinRelVar in the range of a few hundred Kilobytes.) Finally, Figure 10(c) depicts the memory requirements of our MinMaxRelErr implementation as a function of the domain size N (for fixed $B = 2,000$). As expected, the maximum memory needed by our optimized dynamic-programming scheme shows a roughly linear dependence on the data-domain size, and remains at reasonably-small levels even for fairly large values of N (e.g., about 7.2-MB for $N = 65,536$). Overall, our experimental numbers demonstrate that, in terms of memory requirements, our proposed dynamic-programming algorithms can easily scale to large domain sizes N and synopsis sizes B.

6. RELATED WORK

Wavelets have a long history of successes in the signal and image processing arena [Jawerth and Sweldens 1994; Natsev et al. 1999; Stollnitz et al. 1996] and, recently, they have also found their way into data-management applications. Matias et al. [1998] first proposed the use of Haar-wavelet coefficients as synopses for accurately estimating the selectivities of range queries. Vitter and Wang [1999] describe I/O-efficient algorithms for building multidimensional Haar wavelets from large relational data sets and show that a small set of wavelet coefficients can efficiently provide accurate approximate answers to range aggregates over OLAP cubes. Chakrabarti et al. [2000, 2001] demonstrate the effectiveness of Haar wavelets as a general-purpose approximate query processing tool by designing efficient algorithms that can process complex relational queries (with joins, selections, etc.) entirely in the wavelet-coefficient domain. Matias et al. [2000] consider the problem of online maintenance for coefficient synopses and propose a probabilistic-counting technique that approximately maintains the largest normalized-value coefficients in the presence of updates. Gilbert et al. [2001] propose algorithms for building approximate one-dimensional Haar-wavelet synopses over numeric data streams. Deligiannakis and Roussopoulos [2003] introduce

time- and space-efficient techniques for constructing Haar-wavelet synopses for data sets with multiple measures (such as those typically found in OLAP applications).

All the above papers rely on conventional, L_2-error-based thresholding schemes that typically decide the significance of a coefficient based on its absolute normalized value. Garofalakis and Gibbons [2002, 2004] have shown that such conventional wavelet synopses can suffer from several important problems, including the introduction of severe bias in the data reconstruction and wide variance in the quality of the data approximation, as well as the lack of nontrivial guarantees for individual approximate answers. In contrast, their proposed *probabilistic wavelet synopses* rely on a a probabilistic thresholding process based on *randomized rounding* [Motwani and Raghavan 1995], that tries to *probabilistically* control the *maximum relative (or, absolute) error* in the synopsis by minizing appropriate probabilistic metrics (like, normalized standard error or normalized bias). The problem addressed in this article, namely the design of efficient *deterministic* thresholding schemes for maximum error as well as more general, non-L_2 error metrics, is one of the main open problems posed by their study [Garofalakis and Gibbons 2004]. More recent work has also considered the problem of minimizing the maximum absolute error in the data approximation in the context of different applications, such as the indexing of spatio-temporal trajectory data using Chebyshev polynomials [Cai and Ng 2004].

There is a rich mathematics literature on m-term approximations using wavelets (m is the number of coefficients in the synopsis). Some prior work has studied thresholding approaches for meeting a target upper bound for an L_p-error metric [DeVore 1998; Stollnitz et al. 1996]. We are not aware of work addressing the deterministic minimization of relative errors with sanity bounds (arguably the most important scenario for approximate query processing in databases) and, to the best of our knowledge, ours are the first results on computationally efficient (optimal and near-optimal) deterministic thresholding schemes for minimizing general error metrics for one- and multidimensional wavelet summaries.

Following up on the conference version of this work [Garofalakis and Kumar 2004], Muthukrishnan has recently proposed optimization algorithms based on our optimal DP formulation for Haar-coefficient thresholding with the objective of minimizing a *non-uniform* (i.e., weighted) L_2-norm error [Muthukrishnan 2004]. An interesting contribution of that work is the intelligent coupling of our optimal dynamic program with *exhaustive search* at lower levels of the error tree, resulting in a running-time complexity of $O(N^2B/\log B)$ (i.e., a $O(\log^2 B)$ improvement over our time bounds in Section 3.1). As we demonstrate in this paper, our dynamic-programming ideas are, in fact, applicable to a much broader class of "distributive" error metrics, which includes several useful error measures for approximate query processing engines (such as mean weighted relative error and general weighted L_p norms). Even more recently, Guha has given a careful complexity analysis of our optimal dynamic program, showing the $O(N^2)$ bound on the worst-case running-time complexity of our DP algorithm [Guha 2004] (see Section 3.1).

7. CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

In this article, we have proposed novel, computationally efficient schemes for deterministic wavelet thresholding for general error metrics in both one and multiple dimensions. For one-dimensional wavelets, we have introduced an optimal, low polynomial-time thresholding algorithm based on a new Dynamic-Programming (DP) formulation that can be used to efficiently minimize any distributive error measure in the data approximation. For the multidimensional case, we have designed novel, polynomial-time approximation schemes (with tunable ϵ-approximation guarantees for the target error metric) for wavelet thresholding based on approximate dynamic programs. Results from an empirical study of our DP optimization algorithms over real-world and synthetic data sets have clearly demonstrated their effectiveness against earlier-proposed wavelet-thresholding techniques.

There are several interesting directions for future research in this area. As demonstrated in this article, deterministic Haar-wavelet thresholding for general, non-L_2 error metrics appears to become significantly more difficult as the data dimensionality increases (similar observations have also been made for the related problem of *histogram construction* [Muthukrishnan et al. 1999]). Investigating the existence of optimization algorithms for multidimensional wavelet thresholding that potentially avoid the super-exponential explosion in dimensionality inherent in our dynamic programs is certainly a challenging area for future work. The question of designing an efficient $(1 + \epsilon)$-approximation scheme for maximum relative error in multiple dimensions is also left open. Finally, an important question in this realm concerns the general suitability of the Haar-wavelet transform as a data-summarization and approximate query processing tool when it comes to error metrics other than L_2 norms. Could there be other (existing or new) wavelet bases that are better suited for optimizing, for example, relative-error metrics in the data approximation?

ACKNOWLEDGMENTS

We would like to thank Sudipto Guha for bringing his tighter analysis of our dynamic-programming schemes [Guha 2004] to our attention, as well as the anonymous referees for insightful comments on this article.

REFERENCES

ACHARYA, S., GIBBONS, P. B., POOSALA, V., AND RAMASWAMY, S. 1999. Join synopses for approximate query answering. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data* (Philadelphia, PA). ACM, New York, 275–286.

AMSALEG, L., BONNET, P., FRANKLIN, M. J., TOMASIC, A., AND URHAN, T. 1997. Improving responsiveness for wide-area data access. *IEEE Data Eng. Bull. 20*, 3 (Sept.), 3–11 (Special Issue on Improving Query Responsiveness).

CAI, Y. AND NG, R. 2004. Indexing spatio-temporal trajectories with Chebyshev polynomials. In *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data* (Paris, France). ACM, New York.

CHAKRABARTI, K., GAROFALAKIS, M., RASTOGI, R., AND SHIM, K. 2000. Approximate query processing using wavelets. In *Proceedings of the 26th International Conference on Very Large Data Bases* (Cairo, Egypt), 111–122.

CHAKRABARTI, K., GAROFALAKIS, M. N., RASTOGI, R., AND SHIM, K. 2001. Approximate query processing using wavelets. *The VLDB Journal 10*, 2-3 (Sept.), 199–223 ("Best of VLDB'2000" Special Issue).

DELIGIANNAKIS, A. AND ROUSSOPOULOS, N. 2003. Extended wavelets for multiple measures. In *Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data* (San Diego, CA). ACM, New York.

DESHPANDE, A., GAROFALAKIS, M., AND RASTOGI, R. 2001. Independence is Good: Dependency-Based Histogram Synopses for High-Dimensional Data. In *Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data* (Santa Barbara, CA). ACM, New York.

DEVORE, R. A. 1998. Nonlinear Approximation. *Acta Num. 7*, 51–150.

GAROFALAKIS, M. AND GIBBONS, P. B. 2001. Approximate query processing: Taming the terabytes. Tutorial in *Proceedings of the 27th International Confrence on Very Large Data Bases*, Roma, Italy.

GAROFALAKIS, M. AND GIBBONS, P. B. 2002. Wavelet synopses with error guarantees. In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data* (Madison, WI). ACM, New York, 476–487.

GAROFALAKIS, M. AND GIBBONS, P. B. 2004. Probabilistic wavelet synopses. *ACM Trans. Database Systems 29*, 1 (Mar.) (SIGMOD/PODS Special Issue).

GAROFALAKIS, M. AND KUMAR, A. 2004. Deterministic wavelet thresholding for maximum-error metrics. In *Proceedings of the 23rd ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems* (Paris, France). ACM, New York.

GILBERT, A. C., KOTIDIS, Y., MUTHUKRISHNAN, S., AND STRAUSS, M. J. 2001. Surfing wavelets on streams: One-pass summaries for approximate aggregate queries. In *Proceedings of the 27th International Conference on Very Large Data Bases* (Roma, Italy).

GUHA, S. 2004. A note on wavelet optimization. (Manuscript available from: `http://www.cis.upenn.edu/~sudipto/note.html`.)

GUNOPULOS, D., KOLLIOS, G., TSOTRAS, V. J., AND DOMENICONI, C. 2000. Approximating multi-dimensional aggregate range queries over real attributes. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data* (Dallas, Texas). ACM, New York.

HAAS, P. J. AND SWAMI, A. N. 1992. Sequential sampling procedures for query size estimation. In *Proceedings of the 1992 ACM SIGMOD International Conference on Management of Data* (San Diego, CA). ACM, New York, 341–350.

HELLERSTEIN, J. M., HAAS, P. J., AND WANG, H. J. 1997. Online aggregation. In *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data* (Tucson, AZ). ACM, New York.

IOANNIDIS, Y. 2003a. Approximations in database systems. In *Proceedings of the 9th International Conference on Database Theory (ICDT'2003)* (Siena, Italy).

IOANNIDIS, Y. 2003b. The history of histograms (abridged). In *Proceedings of the 29th International Conference on Very Large Data Bases* (Berlin, Germany).

JAWERTH, B. AND SWELDENS, W. 1994. An overview of wavelet based multiresolution analyses. *SIAM Rev. 36*, 3, 377–412.

MATIAS, Y., VITTER, J. S., AND WANG, M. 1998. Wavelet-based histograms for selectivity estimation. In *Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data* (Seattle, WA). ACM, New York, 448–459.

MATIAS, Y., VITTER, J. S., AND WANG, M. 2000. Dynamic maintenance of wavelet-based histograms. In *Proceedings of the 26th International Conference on Very Large Data Bases* (Cairo, Egypt).

MOTWANI, R. AND RAGHAVAN, P. 1995. *Randomized Algorithms*. Cambridge University Press, Cambridge, MA.

MUTHUKRISHNAN, S. 2004. Nonuniform sparse approximation with Haar wavelet basis. Tech. Rep. 2004-42, DIMACS. Sept.

MUTHUKRISHNAN, S., POOSALA, V., AND SUEL, T. 1999. On rectangular partitionings in two dimensions: Algorithms, complexity, and applications. In *Proceedings of the 7th International Conference on Database Theory (ICDT'99)* (Jerusalem, Israel).

NATSEV, A., RASTOGI, R., AND SHIM, K. 1999. WALRUS: A similarity retrieval algorithm for image databases. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data* (Philadelphia, PA). ACM, New York.

STOLLNITZ, E. J., DEROSE, T. D., AND SALESIN, D. H. 1996. *Wavelets for Computer Graphics—Theory and Applications*. Morgan Kaufmann, San Francisco, CA.

VITTER, J. S. AND WANG, M. 1999. Approximate computation of multidimensional aggregates of sparse data using wavelets. In *Proceedings of the 1999 ACM SIGMOD International Conference on Management of Data* (Philadelphia, PA). ACM, New York.

Received October 2004; revised May 2005; accepted August 2005