February 2008

Approximation Algorithms for Wavelet Transform Coding of Data Streams

Sudipto Guha
University of Pennsylvania, sudipto@cis.upenn.edu

Boulos Harb
University of Pennsylvania

Approximation Algorithms for Wavelet Transform Coding of Data Streams

Abstract

This paper addresses the problem of finding a B-term wavelet representation of a given discrete function $f \in R^n$ whose distance from f is minimized. The problem is well understood when we seek to minimize the Euclidean distance between f and its representation. The first-known algorithms for finding provably approximate representations minimizing general l_p distances (including $l\infty$) under a wide variety of compactly supported wavelet bases are presented in this paper. For the Haar basis, a polynomial time approximation scheme is demonstrated. These algorithms are applicable in the one-pass sublinear-space data stream model of computation. They generalize naturally to multiple dimensions and weighted norms. A universal representation that provides a provable approximation guarantee under all p-norms simultaneously; and the first approximation algorithms for bit-budget versions of the problem, known as adaptive quantization, are also presented. Further, it is shown that the algorithms presented here can be used to select a basis from a tree-structured dictionary of bases and find a B-term representation of the given function that provably approximates its best dictionary-basis representation.

Keywords

adaptive quantization, best basis selection, compacted supported wavelets, nonlinear approximation, sparse representation, streaming algorithms, transform coding, universal representation

Approximation Algorithms for Wavelet Transform Coding of Data Streams

Sudipto Guha and Boulos Harb, *Student Member, IEEE*

***Abstract*—This paper addresses the problem of finding a B-term wavelet representation of a given discrete function $f \in \mathcal{R}^n$ whose distance from f is minimized. The problem is well understood when we seek to minimize the Euclidean distance between f and its representation. The first-known algorithms for finding provably approximate representations minimizing general ℓ_p distances (including ℓ_∞) under a wide variety of compactly supported wavelet bases are presented in this paper. For the Haar basis, a polynomial time approximation scheme is demonstrated. These algorithms are applicable in the one-pass sublinear-space data stream model of computation. They generalize naturally to multiple dimensions and weighted norms. A universal representation that provides a provable approximation guarantee under all p-norms simultaneously; and the first approximation algorithms for bit-budget versions of the problem, known as adaptive quantization, are also presented. Further, it is shown that the algorithms presented here can be used to select a basis from a tree-structured dictionary of bases and find a B-term representation of the given function that provably approximates its best dictionary-basis representation.**

***Index Terms*—Adaptive quantization, best basis selection, compactly supported wavelets, nonlinear approximation, sparse representation, streaming algorithms, transform coding, universal representation.**

I. Introduction

A CENTRAL problem in approximation theory is to represent a function concisely. Given a function or a signal as input, the goal is to construct a representation as a linear combination of several predefined functions, under a constraint which limits the space used by the representation. The set of predefined functions are denoted as the dictionary. One of the most celebrated approaches in this context has been that of *nonlinear approximation*. In this approach, the dictionary elements that are used to represent a function are allowed to depend on the input signal itself.

Nonlinear approximations has a rich history starting from the work of Schmidt [2]; however, more recently these have come to fore in the context of wavelet dictionaries [3], [4]. Wavelets were first analyzed by DeVore *et al.* [5] in nonlinear approximation. Wavelets and multifractals have since found extensive use in image representation, see Jacobs [6]. In fact, the success of wavelets in nonlinear approximation has been hailed by many researchers as "the 'true' reason of the usefulness of wavelets in signal compression" (Cohen *et al.* [7]). Due to lack of space we would not be able to review the extremely rich body of work that has emerged in this context; see the surveys by DeVore [8] and Temlyakov [9] for substantial reviews.

However, with the rise in the number of domains for which wavelets have been found useful, several interesting problems have arisen. Classically, the error in terms of representation has been measured by the Euclidean or ℓ_2 error. This choice is natural for analysis of functions, but not necessarily for representation of data and distributions. Even in image compression, Mallat [3, p. 528] and Daubechies [4, p. 286] point out that while the ℓ_2 measure does not adequately quantify perceptual errors, it is used, nonetheless, since other norms are difficult to optimize. However, non-ℓ_2 measures have been widely used in the literature. Matias, Vitter and Wang [10], suggested using the ℓ_1 metric and showed that wavelets could be used in creating succinct synopses of data allowing us to answer queries approximately. The ℓ_1 distance is a statistical distance and is well suited for measuring distributions. Interestingly, Chapelle, Haffner and Vapnik [11] show that the ℓ_1 norm significantly outperforms the ℓ_2 norm in image recognition on images in the Corel data set using SVM's. From a completely different standpoint, we may be interested in approximating a signal in the ℓ_∞ norm thus seeking a high fidelity approximation throughput rather than an 'average' measure such as other norms. This is particularly of interest if we are trying to process noisy data (we consider ℓ_1, ℓ_∞ approximations in Section IV-C). While we have developed a reasonable understanding of ℓ_2 error, problems involving non-ℓ_2 error are still poorly understood. This paper takes the first steps toward filling this gap.

One of the most basic problems in nonlinear approximation is the following: Given a wavelet basis $\{\psi_i\}$ and a target function (or signal, vector) $f \in \mathcal{R}^n$, construct a representation $\hat{f}$ as a linear combination of at most B basis vectors so as to minimize some normed distance between f and $\hat{f}$. The *B-term representation* $\hat{f}$ belongs to the space $\mathcal{F}_B = \{\sum_{i=1}^n z_i\psi_i : z_i \in \mathcal{R}, \|z\|_0 \leq B\}$, where $\|z\|_0$ is the number of nonzero coefficients in $z \in \mathcal{R}^n$. The problem is well-understood if the error of the representation is measured using the Euclidean or ℓ_2 distance. Since the ℓ_2 distance is preserved under rotations, by Parseval's theorem, we have

$$\|f - \hat{f}\|_2^2 = \sum_i (f_i - \sum_j z_j \psi_j[i])^2 = \sum_i (\langle f, \psi_i \rangle - z_i)^2.$$

It is clear then that the solution under this error measure is to retain the largest B inner products $\langle f, \psi_i \rangle$, which are also the

Manuscript received April 25, 2006; revised May 15, 2007. This work was supported in part by an Alfred P. Sloan Research Fellowship and by an NSF Awards CCF-0430376 and CCF-0644119. The work of B. Harb was done while with the University of Pennsylvania. The material in this paper was presented in part as an extended abstract at SODA '06: Proceedings of the Seventeenth Annual ACM Symposium, Miami, FL, January 2006.

S. Guha is with the Department of Computer Information Science, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: sudipto@cis.upenn.edu).

B. Harb is with Google Inc., New York, NY 10011 USA (e-mail: harb@google.com).

Communicated by V. A. Vaishampayan, Associate Editor At Large.

Digital Object Identifier 10.1109/TIT.2007.913569

coefficients of the wavelet expansion of f. *Note:* the fact that we have to store the inner products or the wavelet coefficients is a natural consequence of the proof of optimality.

The common strategy for the B-term representation problem in the literature has been "to retain the $[B]$ terms in the wavelet expansion of the target function which are largest relative to the norm in which error of approximation is to be measured" [8, p. 4]. This strategy is reasonable in an extremal setting; i.e., if we are measuring the rate of the error as a function of B. But it is easy to show that the common greedy strategy is sub-optimal, see [12]–[17]. In light of this, several researchers [13]–[15], [17], [18] considered a *restricted* version of the problem under the Haar basis where we may only choose wavelet coefficients of the data. However to date, the only bound on its performance with respect to the target function's best possible representation using B terms from the wavelet basis is given by Temlyakov [19] (see also [9, Sec. 7]. Temlyakov shows that given f in the (infinite dimensional) Banach function space $L_p[0,1], 1 < p < \infty$, if the given basis $\{\psi_i\}_{i\in\mathbb{Z}}$ is L_p*-equivalent* to the Haar basis [20], then the error of the common greedy strategy is an α factor away from that of the optimal B-term representation. The factor α depends on p and properties of $\{\psi_i\}$, but the dependence is unspecified. However, from an optimization point of view in the finite-dimensional setting, the relationship between the factor α and the dimension n of the space spanned is the key problem, which we address here. Three relevant questions arise in this context. First is whether there are universal algorithms/representations that simultaneously approximate all ℓ_p norms. This is important because in many applications, it is difficult to determine the most suitable norm to minimize without looking at the data, and an universal representation would be extremely useful. The second question concerns the complexity of representing the optimal solution. It is not immediate *a priori* that the optimal unrestricted solution minimizing, for example, the ℓ_5 norm for a function that takes only rational values can be specified by B rational numbers. The third related question pertains to the computational complexity of finding the optimum solution. Can the solution be found in time polynomial in the size of the input n? Or better yet, can the solution be found in *strongly* polynomial time where the running time of the algorithm does not depend on the numeric values of the input. We focus on these questions using the lens of *approximation algorithms*, where we seek to find a solution that is close to the optimum—in fast polynomial time. Note that the use of approximation algorithms does not limit us from using additional heuristics from which we may benefit, but gives us a more organized starting point to develop heuristics with provable bounds.

A natural generalization of the problem above is known as *Adaptive Quantization*. The B-term representation requires storing $2B$ numbers, the coefficient and the index of the corresponding basis vector to be retained. The actual cost (in bits) of storing the real numbers z_i is, however, nonuniform. Depending on the scenario, it may be beneficial to represent a function with a large number of low-support vectors with low precision z_i's or a few vectors with more detailed precision z_i's. Hence, a B-term representation algorithm does not translate directly into a practical compression algorithm. A natural generalization, and a more practical model as noted in [7], is to minimize the error subject to the constraint that the stored values and indices cannot exceed a given bit-budget. Note that, again, we are not constrained here to storing wavelet expansion coefficients. This bit-budget version of the problem is known as adaptive quantization, which we will also consider. To the best of our knowledge, there are no known approximation algorithms for this problem.

One other natural generalization incorporates a choice of basis into the optimization problem [8]. We are given a dictionary $\mathcal{D}$ of bases and our objective is to choose a best basis in $\mathcal{D}$ for representing f using B terms. This bicriteria optimization problem is a form of *highly nonlinear approximation* [8]. In a seminal work, Coiffman and Wickerhauser [21] construct a binary tree-structured dictionary composed of $O(n\log n)$ vectors and containing $2^{O(\frac{n}{2})}$ orthonormal bases. They present a dynamic programming algorithm that in $O(n\log n)$ time finds a best basis minimizing the entropy of its inner products with the given function f. Mallat [3] discusses generalizations based on their algorithm for finding a basis from the tree dictionary that minimizes an arbitrary concave function of its expansion coefficients. However, finding a basis in $\mathcal{D}$ that minimizes a concave function of its inner products with the given f is not necessarily one with which we can best represent f (in an ℓ_p sense) using B terms. Combining our approximation algorithms for the original B-term representation problem with the algorithm of Coiffman and Wickerhauser, we show how one can construct provably approximate B-term representations in tree-structured wavelet dictionaries. Several of these results also extend to arbitrary dictionaries with low coherence [22], [23].

Along with the development of richer representation structures, in recent years there has been significant increase in the data sets we are faced with. At these massive scales, the data is not expected to fit the available memory of even fairly powerful computers. One of the emergent paradigms to cope with this challenge is the idea of *data stream algorithms*. In a data stream model the input is provided one at a time, and any input item not explicitly stored is inaccessible to the computation, i.e., it is lost. The challenge is to perform the relevant computation in space that is sublinear in the input size; for example, computing the best representation of a discrete signal $f(i)$ for $i \in [n]$ that is presented in increasing order of i, in only $o(n)$ space. This is a classic model of time-series data, where the function is presented one value at a time. It is immediate that under this space restriction we may not be able to optimize our function. This harks back to the issue raised earlier about the precision of the solution. Thus, the question of approximation algorithms is doubly interesting in this context. The only known results on this topic [24], [25] crucially depend on Parseval's Identity and do not extend to norms other than ℓ_2.

In summary, even for the simplest possible transform coding problem, namely the B-term representation problem, we can identify the following issues.

- There are no analysis techniques for ℓ_p norms. In fact this is the bottleneck in analyzing any generalization of the B-term representation problem; e.g., the adaptive quantization problem.

- All of the (limited) analyzes in the optimization setting have been done on the Haar system, which although important, is not the wavelet of choice in some applications. Further, in this setting, the bounds on the performance of the algorithms used in practice which retain wavelet coefficients are unclear.
- Signals that require transform coding are often presented as a streaming input—no algorithms are known except for ℓ_2 norms.
- The computational complexity of transform coding problems for structured dictionaries, or even for wavelet bases, is unresolved.

A. Our Results

We ameliorate the above by showing the following.

1) For the B-*term representation problem* we show that,
   a) The restricted solution that retains at most B *wavelet coefficients* is a $O(\log n)$ approximation to the unrestricted solution under all ℓ_p distances for general compact systems (e.g., Haar, Daubechies, Symmlets, Coiflets, among others).[1] We provide a $O(B+\log n)$ space and $O(n)$ time one-pass algorithm in the data stream model. We give a modified greedy strategy, which is not normalization, but is similar to some scaling strategies used in practice. Our strategy demonstrates why several scaling based algorithms used in practice work well.
   b) A surprising consequence of the above is an universal representation using $O(B\log n)$ coefficients that *simultaneously* approximate the signal for all ℓ_p distances up to $O(\log n)$.
   c) The unrestricted optimization problem has a fully polynomial-time approximation scheme (FPTAS) for all ℓ_p distances in the Haar system, that is, the algorithm runs in time polynomial in B, ϵ, n. The algorithm is one-pass, $n^{\frac{1}{p}}$ space and $n^{1+\frac{1}{p}}$ time for ℓ_p distances. Therefore, the algorithm is a streaming algorithm with sublinear space for $p > 1$. For ℓ_∞, the algorithm runs in polylog space and linear time.[2]
   d) For more general compactly supported systems we display how our ideas yield a quasi-polynomial time approximation scheme (QPTAS).[3] This result is in contrast to the case of an arbitrary dictionary which, as we already mentioned, is hard to approximate to within any constant factor *even allowing* quasi-polynomial time.[4]
   e) The results extend to fixed dimensions and workloads with increases in running time and space.
2) In terms of techniques, we introduce a new lower bounding technique using the basis vectors $\{\psi_i\}$, which gives us the above result regarding the gap between the restricted and unrestricted versions of the problem. We also show that bounds using the scaling vectors $\{\phi_i\}$ are useful for these optimization problems and, along with the lower bounds using $\{\psi_i\}$, give us the approximation schemes. To the best of our knowledge, this is the first use of both the scaling and basis vectors to achieve such guarantees.
3) We show that the lower bound for general compact systems can be extended to an approximation algorithm for adaptive quantization. This is the first approximation algorithm for this problem.
4) For tree-structured dictionaries composed of the type of compactly supported wavelets we consider, our algorithms can be combined with the dynamic programming algorithm of Coiffman and Wickerhauser [21] to find a B-term representation of the given f. The ℓ_p error of the representation we construct provably approximates the error of a best representation of f using B terms from a basis in the dictionary.

The key technique used in this paper is to lower bound the solution based on a system of linear equations but with one nonlinear constraint. This lower bound is used to set the "scale" or "precision" of the solution, and we show that the best solution respecting this precision is a near optimal solution by "rounding" the components of the optimal solution to this precision. Finally, the best solution in this class is found by a suitable dynamic program adapted to the data stream setting.

We believe that approximation algorithms give us the correct standpoint for construction of approximate representations. The goal of approximation theory is to approximate representation; the goal of approximation algorithms is to approximate optimization. Data stream algorithms are inherently approximate (and often randomized) because the space restrictions force us to retain approximate information about the input. These goals, of the various uses of the approximation, are ultimately convergent.

Organization: We begin by reviewing some preliminaries of wavelets. In Section III we present our greedy approximation which also relates the restricted to the unrestricted versions of the problem. Section IV presents applications of the greedy algorithm; namely, an approximate universal representation, approximation algorithms for adaptive quantization, and examples illustrating the use of non-ℓ_2 norms for image representations. Section V is the main section of the paper wherein we present our approximation schemes. We detail the FPTAS for the Haar system and show its extensions to multiple dimensions and workloads. We subsequently demonstrate in Section VI how the same ideas translate to a FPTAS for multidimensional signals and workloads, and a QPTAS under more general compactly supported wavelets. In Section VII we present the tree-structured best-basis selection algorithm. Finally, in Section VIII we display some experimental results contrasting the performance of an optimal algorithm that is restricted to choosing Haar expansion coefficients with our Haar FPTAS.

[1]This statement differs from the statement in the extremal setting that says that discarding all coefficients below τ introduces $O(\tau \log n)$ error, since the latter does not account for the number of terms.

[2]For clarity here, we are suppressing terms based on $\log n$, B, and ϵ. The exact statements appear in Theorems 16 and 18.

[3]This implies that the running time is $2^{O(\log^c n)}$ for some constant c ($c = 1$ gives polynomial time).

[4]Follows from the result of Feige [26].

II. Preliminaries

The problem on which we mainly concentrate is the following:

Problem 1(B-Term Representation: Given $f \in \mathcal{R}^n, p \in [1,\infty]$, a compactly-supported wavelet basis for $\mathcal{R}^n$ $\{\psi_i\}_{i=1}^n$, and an integer B, find a solution $\{z_i\}_{i=1}^n, z_i \in \mathcal{R}$, with at most B nonzero components such that $\|f - \sum_i z_i\psi_i\|_p$ is minimized.

We will often refer to this problem as the *unrestricted* B-term representation problem in order to contrast it with a *restricted* version where the nonzero components of the solution can only take on values from the set $\{\langle f, \psi_i\rangle, i \in [n]\}$. That is, in the restricted version, each z_i can only be set to a coefficient from the wavelet expansion of f, or zero.

A. Data Streams

For the purpose of this paper, a data stream computation is a space bounded algorithm, where the space is sublinear in the input. Input items are accessed sequentially and any item not explicitly stored cannot be accessed again in the same pass. In this paper we focus on *one pass* data streams. We will assume that we are given numbers $f = f(1), \ldots, f(i), \ldots, f(n)$ which correspond to the signal f to be summarized in the increasing order of i. This model is often referred to as the *aggregated model* and has been used widely [24], [27], [28]. It is specially suited to model streams of time series data [29], [30] and is natural for transcoding a single channel. Since we focus on dyadic wavelets (that are dilated by powers of 2), assuming n is a power of 2 will be convenient, but not necessary. As is standard in literature on streaming [25], [31], [32], we also assume that the numbers are polynomially bounded, i.e., all $|f(i)|$'s are in the range $[n^{-c}, n^c]$ for some constant c.

B. Compactly Supported Wavelets

We include here some definitions and notation that we use in the main text. Readers familiar with wavelets can easily skip this section. For thorough expositions on wavelets, we refer the interested reader to the authoritative texts by Daubechies [4] and Mallat [3]. For a brief introduction to wavelets, see [33, Ch. 2.3].

A wavelet basis $\{\psi_k\}_{k=1}^n$ for $\mathcal{R}^n$ is a basis where each vector is constructed by dilating and translating a single function referred to as the *mother wavelet* ψ. For example the Haar mother wavelet, due to Haar [34], is given by

$$\psi_H(t) = \begin{cases} 1, & \text{if } 0 \le t < 1/2 \\ -1, & \text{if } 1/2 \le t < 1 \\ 0, & \text{otherwise.} \end{cases}$$

The Haar basis for $\mathcal{R}^n$ is composed of the vectors $\psi_{j,s}[i] = 2^{-j/2}\psi_H(\frac{i-2^j s}{2^j})$ where $i \in [n], j = 1, \ldots, \log n$, and $s = 0, \ldots, n/2^j - 1$, plus their orthonormal complement $\frac{1}{\sqrt{n}}1^n$. This last basis vector is closely related to the Haar *multiresolution scaling function* $\phi_H(t) = 1$ if $0 \le t < 1$ and 0, otherwise. In fact, there is an explicit recipe for constructing the mother wavelet function ψ from ϕ using a *conjugate mirror filter* [35], [36] (see also Daubechies [3], and Mallat [4]). Notice that the Haar mother wavelet is compactly supported on the interval $[0,1)$. This wavelet, which was discovered in 1910, was the only known wavelet of compact support until Daubechies constructed a family of compactly supported wavelet bases [37] in 1988 (see also [4, Ch. 6]).

The vector $\psi_{j,s}$ is said to be centered at $2^j s$ and of scale j and is defined on at most $(2q-1)2^j - 2(q-1)$ points. For ease of notation, we will use both ψ_i and $\psi_{j,s}$ depending on the context and assume there is a consistent map between them.

The Cascade Algorithm for computing $\langle f, \psi_{j,s}\rangle, \langle f, \phi_{j,s}\rangle$: Assume that we have the conjugate mirror filter h with support $\{0, \ldots, 2q-1\}$. Given a function $f \in \mathcal{R}^n$, we set $a_0[i] = f[i]$, and repeatedly compute $a_{j+1}[t] = \sum_s h[s-2t]a_j[s]$ and $d_{j+1}[t] = \sum_s g[s-2t]a_j[s]$ (where $g[k] = (-1)^k h[1-k]$ is also a conjugate mirror filter). Notice that if the filter h has support $\{0, \ldots, 2q-1\}$, then we have $0 \le s - 2t \le 2q-1$. This procedure gives $a_j[t] = \langle f, \phi_{j,t}\rangle$ and $d_j[t] = \langle f, \psi_{j,t}\rangle$.

In order to compute the inverse transform, we evaluate $a_j[t] = \sum_s h[t-2s]a_{j+1}[s] + \sum_s g[t-2s]d_{j+1}[s]$. Observe that by setting a single $a_j[s]$ or $d_j[s]$ to 1 and the rest to 0, the inverse transform gives us $\phi_{j,s}$ or $\psi_{j,s}$. Indeed, this is the algorithm usually used to compute $\phi_{j,s}$ and $\psi_{j,s}$.

We will utilize the following proposition which is a consequence of the dyadic structure of compactly supported wavelet bases.

Proposition 1: A compactly supported wavelet whose filter has two q nonzero coefficients generates a basis for $\mathcal{R}^n$ that has $O(q \log n)$ basis vectors with a nonzero value at any point $i \in [n]$.

III. Greedy Approximation Algorithms for General Compact Systems and Data Streams

Recall our optimization problem. Given a compactly supported wavelet basis $\{\psi_i\}$ and a target vector f, we wish to find $\{z_i\}$ with at most B nonzero numbers to minimize $\|f - \sum_i z_i\psi_i\|_p$.

We present two analyzes below corresponding to ℓ_∞ and ℓ_p errors when $p \in [1,\infty)$. In each case, we begin by analyzing the sufficient conditions that guarantee the error. A (modified) greedy coefficient retention algorithm will naturally fall out of both analyzes. The proof shows that several of the algorithms that are used in practice have bounded approximation guarantee. Note that the optimum solution can choose any values in the representation $\hat{f}$.

In what follows the pair (p, p') are the usual conjugates; i.e., $\frac{1}{p} + \frac{1}{p'} = 1$ when $1 < p < \infty$, and when $p = 1$ we simply set $p' = \infty$. For simplicity, we start with the $p = \infty$ case.

1) An ℓ_∞ Algorithm and Analysis: The main lemma, which gives us a lower bound on the optimal error, is:

Lemma 2: Let $\mathcal{E}$ be the minimum error under the ℓ_∞ norm and $\{z_i^*\}$ be the optimal solution, then

$$-\|\psi_i\|_1|\mathcal{E}| \le \langle f, \psi_i\rangle - z_i^* \le \|\psi_i\|_1|\mathcal{E}|.$$

Proof: For all j we have $-|\mathcal{E}| \le f(j) - \sum_i z_i^*\psi_i(j) \le |\mathcal{E}|$. Since the equation is symmetric multiplying it by $\psi_k(j)$ we get

$$-|\mathcal{E}||\psi_k(j)| \le f(j)\psi_k(j) - \psi_k(j)\sum_i z_i^*\psi_i(j) \le |\mathcal{E}||\psi_k(j)|$$

Adding the above equation for all j, since $-|\mathcal{E}|\sum_j |\psi_k(j)| = -|\mathcal{E}|\|\psi_k\|_1$ we obtain (consider only the left side)

$$\begin{aligned}-|\mathcal{E}|\|\psi_k\|_1 &\le \sum_j f(j)\psi_k(j) - \sum_j \psi_k(j)\sum_i z_i^*\psi_i(j)\\ &= \langle f,\psi_k\rangle - \sum_i z_i^*\sum_j \psi_k(j)\psi_i(j)\\ &= \langle f,\psi_k\rangle - \sum_i z_i^*\delta_{ik} = \langle f,\psi_k\rangle - z_k^*.\end{aligned}$$

The upper bound follows analogously. □

A Relaxation: Consider the following program:

$$\begin{array}{l}\text{minimize } \tau\\ \begin{array}{lll}-\tau\|\psi_1\|_1 & \le \langle f,\psi_1\rangle - z_1 & \le \tau\|\psi_1\|_1\\ \vdots & \vdots & \vdots\\ -\tau\|\psi_n\|_1 & \le \langle f,\psi_n\rangle - z_n & \le \tau\|\psi_n\|_1\end{array}\\ \text{At most } B \text{ of the } z_i\text{'s are nonzero.}\end{array} \quad (1)$$

Observe that $\mathcal{E}$ is a feasible solution for the above program and $\mathcal{E} \ge \tau^*$ where τ^* is the optimum value of the program. Also, Lemma 2 is not specific to wavelet bases, and indeed we have $\mathcal{E} = \tau^*$ when $\{\psi_i\}$ is the standard basis, i.e., ψ_i is the vector with 1 in the ith coordinate and 0, elsewhere. The next lemma is straightforward.

Lemma 3: The minimum τ of program (1) is the $(B+1)^{th}$ largest value $\frac{|\langle f,\psi_i\rangle|}{\|\psi_i\|_1}$.

The Algorithm: We choose the largest B coefficients based on $|\langle f,\psi_i\rangle|/\|\psi_i\|_1$. This can be done over a one pass stream, and in $O(B+\log n)$ space for any compact wavelet basis. Note that we need not choose $z_i = \langle f,\psi_i\rangle$ but any z_i such that $|z_i - \langle f,\psi_i\rangle|/\|\psi_i\|_1 \le \tau^*$. But in particular, we may choose to retain coefficients and set $z_i = \langle f,\psi_i\rangle$. The alternate choices may (and often will) be better. Also note that the above is only a necessary condition; we *still* need to analyze the guarantee provided by the algorithm.

Lemma 4: For all basis vectors ψ_i of a compact system there exists a constant C s.t., $\|\psi_i\|_p\|\psi_i\|_{p'} \le \sqrt{q}C$.

Proof: Suppose first that $p < 2$. Consider a basis vector $\psi_i[] = \psi_{j,s}[]$ of sufficiently large scale j that has converged to within a constant r (point-wise) of its continuous analog $\psi_{j,s}()$ [3, pp. 264-265]. That is, $|\psi_{j,s}[k] - \psi_{j,s}(k)| \le r$ for all k such that $\psi_{j,s}[k] \ne 0$. The continuous function $\psi_{j,s}()$ is given by $\psi_{j,s}(t) = 2^{-j/2}\psi(2^{-j}t - s)$, which implies $\psi_{j,s}[k] = O(2^{-j/2}\psi(2^{-j}k - s)) = O(2^{-j/2})$. Note that we are assuming $\|\psi\|_\infty$ itself is some constant since it is independent of n and B. Combining the above with the fact that $\psi_{j,s}[]$ has at most $(2q)2^j$ nonzero coefficients, we have $\|\psi_{j,s}\|'_p = O(2^{-j/2}((2q)2^j)^{1/p'}) = O(2^{j(\frac{1}{p'}-\frac{1}{2})}(2q)^{\frac{1}{p'}})$.

By Hölder's inequality, $\|\psi_{j,s}\|_p \le ((2q)2^j)^{\frac{1}{p}} - \frac{1}{2}\|\psi_{j,s}\|_2 = 2^{j(\frac{1}{p}-\frac{1}{2})}(2q)^{\frac{1}{p}-\frac{1}{2}}$. Therefore, for sufficiently large scales j, $\|\psi_{j,s}\|_p\|\psi_{j,s}\|'_p = O(2^{j(\frac{1}{p}+\frac{1}{p'}-1)}(2q)^{\frac{1}{p}+\frac{1}{p'}-\frac{1}{2}}) = O(\sqrt{q})$, and the lemma holds. For basis vectors at smaller (constant) scales, since the number of nonzero entries is constant, the ℓ_p norm and the ℓ'_p norm are both constant.

Finally, for $p > 2$, the argument holds by symmetry. □

Theorem 5: The ℓ_∞ error of the final approximation is at most $O(q^{3/2}\log n)$ times $\mathcal{E}$ for any compactly supported wavelet.

Proof: Let $\{z_i\}$ be the solution of the system (1), and let the set of the inner products chosen be $\mathcal{S}$. Let τ^* is the minimum solution of the system (1). The ℓ_∞ error seen at a point j is $|\sum_{i\notin\mathcal{S}}\langle f,\psi_i\rangle\psi_i(j)| \le \sum_{i\notin\mathcal{S}}|\langle f,\psi_i\rangle|\|\psi_i(j)|$. By Lemma 3, this sum is at most $\sum_{i\notin\mathcal{S}}\tau^*\|\psi_i\|_1|\psi_i(j)|$, which is at most $\tau^*\max_{i\notin\mathcal{S}}\|\psi_i\|_1\|\psi_i\|_\infty$ times the number of vectors that are nonzero at j. By Proposition 1 the number of nonzero vectors at j is $O(q\log n)$. By Lemma 4, $\|\psi_i\|_1\|\psi_i\|_\infty \le \sqrt{q}C$ for all i, and since $\tau^* \le \mathcal{E}$ we have that the ℓ_∞ error is bounded by $O(q^{3/2}\log n)\mathcal{E}$. □

2) An ℓ_p Algorithm and Analysis for $P \in [1,\infty)$: Under the ℓ_p norm, a slight modification to the algorithm above also gives an $O(q^{3/2}\log n)$ approximation guarantee.

Lemma 6: Let $\mathcal{E}$ be the minimum error under the ℓ_p norm and $\{z_i^*\}$ be the optimal solution, then for some constant c_0

$$\left(\sum_k \frac{1}{\|\psi_k\|_{p'}^p}|\langle f,\psi_k\rangle - z_k^*|^p\right)^{\frac{1}{p}} \le (c_0 q\log n)^{\frac{1}{p}}\mathcal{E}.$$

Proof: An argument similar to that of Lemma 2 gives

$$\begin{aligned}&\sum_i |f_i\psi_k(i) - \sum_j z_j^*\psi_j(i)\psi_k(i)|\\ &\quad= \sum_i \xi_i|\psi_k(i)|\\ &\quad\le \left(\sum_{i\in \text{ support of } \psi_k}\xi_i^p\right)^{1/p}\|\psi_k\|_{p'}\end{aligned}$$

which implies that

$$\begin{aligned}&\frac{1}{\|\psi_k\|_{p'}^p}|\langle f,\psi_k\rangle - z_k^*|^p \le \sum_{i\in \text{ support of } \psi_k}\xi_i^p\\ \Rightarrow &\sum_k \frac{1}{\|\psi_k\|_{p'}^p}|\langle f,\psi_k\rangle - z_k^*|^p \le c_0 q\log n\sum_i \xi_i^p\end{aligned}$$

where the last inequality follows from Proposition 1, that each i belongs to $O(q\log n)$ basis vectors (c_0 is the constant hidden by the this O-term). □

A Relaxation: Consider the following system of equations:

$$\begin{array}{l}\text{minimize}\tau\\ \left(\sum_{i=1}^n \frac{|\langle f,\psi_i\rangle - z_i|^p}{\|\psi_i\|_{p'}^p}\right)^{\frac{1}{p}} \le (c_0 q\log n)^{\frac{1}{p}}\tau\\ \text{At most } B \text{ of the } z_i\text{'s are nonzero.}\end{array} \quad (2)$$

The Algorithm: We choose the largest B coefficients based on $|\langle f,\psi_k\rangle|/\|\psi_k\|'_p$, which minimizes the system (2). This computation can be done over a one pass stream, and in $O(B+\log n)$ space.

Theorem 7: Choosing the B coefficients $\langle f,\psi_k\rangle$ that are largest based on the ordering $|\langle f,\psi_k\rangle|/\|\psi_k\|'_p$ is a streaming $O(q^{3/2}\log n)$ approximation algorithm for the unrestricted optimization problem under the ℓ_p norm.

Note this matches the ℓ_∞ bounds, but stores a (possibly) different set of coefficients.

Proof: Let the value of the minimum solution to the above system of (2) be τ^*. Since $\{z_i^*\}$ is feasible for system (2), $\tau^* \leq \mathcal{E}$. Assume $\mathcal{S}$ is the set of coefficients chosen, the resulting error $\mathcal{E}_\mathcal{S}$ is

$$\begin{aligned}\mathcal{E}_\mathcal{S}^p &= \sum_i \left| \sum_{k \notin \mathcal{S}} \langle f, \psi_k \rangle \psi_k(i) \right|^p \\ &\leq \sum_i (c_0 q \log n)^{p-1} \sum_{k \notin \mathcal{S}} |\langle f, \psi_k \rangle|^p |\psi_k(i)|^p \\ &= (c_0 q \log n)^{p-1} \sum_{k \notin \mathcal{S}} |\langle f, \psi_k \rangle|^p \|\psi_k\|_p^p \\ &\leq (c_0 q \log n)^{p-1} \sum_{k \notin \mathcal{S}} \frac{C^p q^{\frac{p}{2}}}{\|\psi_k\|_{p\prime}^p} |\langle f, \psi_k \rangle|^p \\ &= C^p q^{\frac{p}{2}} (\tau^* c_0 q \log n)^p.\end{aligned}$$

Here, the first inequality is Hölder's inequality combined with Proposition 1 and the fact that $p/p\prime = p - 1$; the second inequality follows from Lemma 4; and the final equality follows from the optimality of our choice of coefficients for the system (2). Now since $\tau^* \leq \mathcal{E}$, we have that $\mathcal{E}_\mathcal{S} \leq c_0 C q^{\frac{3}{2}} \mathcal{E} \log n$. □

3) Summary and a Tight Example: In the two preceding sections, we showed the following:

Theorem 8: Let $\frac{1}{p} + \frac{1}{p\prime} = 1$. Choosing the largest B coefficients based on the ordering $|\langle f, \psi_i \rangle|/\|\psi_i\|_p'$, which is possible by a streaming $O(B + \log n)$ algorithm, gives a $O(q^{\frac{3}{2}} \log n)$ approximation algorithm for the unrestricted optimization problem (Problem 1) under the given ℓ_p norm. The argument naturally extends to multiple dimensions.

As is well known, this choice of coefficients is optimal when $p = 2$ (since $p\prime = 2$ and $\|\psi_i\|_2 = 1$).

Note that the above theorem bounds the gap between the restricted (where we can only choose wavelet coefficients of the input in the representation) and unrestricted optimizations.

A tight example for the ℓ_∞ measure. Suppose we are given the Haar basis $\{\psi_i\}$ and the vector f with the top coefficient $\langle f, \psi_1 \rangle = 0$ and with $\langle f, \psi_i \rangle / \|\psi_i\|_1 = 1 - \epsilon$ for $i \leq n/2$, and $\langle f, \psi_i \rangle / \|\psi_i\|_1 = 1$ for $i > n/2$ (where $\psi_i, i > n/2$, are the basis with smallest support). Let $B = n/c - 1$ where $c \geq 2$ is a constant that is a power of 2. The optimal solution can choose the B coefficients which are in the top $\log n - \log c$ levels resulting in an error bounded by $\log c$. The ℓ_∞ error of the greedy strategy on the other hand will be at least $\log n - 1$ because it will store coefficients only at the bottom of the tree. Hence it's error is at least $\log n / \log c - o(1)$ of the optimal.

IV. Applications of the Greedy Algorithm

Our greedy algorithm extends to a variety of scenarios, which illustrate the scope and the applicability of the techniques presented above.

A. A Universal Representation

In this section, we present a strategy that stores $B(\log n)^2$ coefficients and simultaneously approximates the optimal representations for all p-norms. Notice that in Problem 1 we know the p-norm we are trying to approximate. Here, we do *not* know p and we wish to come up with a representation such that for all $p \in [1, \infty]$, its error measured with $\|f - \hat{f}_u\|_p$ is $O(\log n)$ times the optimal error $\min_z \|f - \sum_i z_i \psi_i\|_p$ where x has at most B nonzero components. Notice that we allow our universal representation to store a factor $(\log n)^2$ more components than any one optimal representation; however, it has to approximate all of them concurrently.

We run our algorithm as before computing the wavelet coefficients of the target vector f; however, we need to determine which coefficients to store for our universal representation. To this end, define the set:

$$\mathcal{N} = \left\{ p_t : p_t = 1 + \frac{t}{\log n}, t = 0, \ldots, \log n (\log n - 1) \right\}. \tag{3}$$

For every $p_t \in \mathcal{N}$, we will store the B coefficients that are largest based on the ordering $|\langle f, \psi_k \rangle|/\|\psi_k\|_{p\prime_t}$ where $p\prime_t$ is the dual norm to p_t. Hence, the number of coefficients we store is no more than $B(\log n)^2$ since $|\mathcal{N}| = (\log n)^2$. Note that our dual programs show that for a given p, storing more than B coefficients does not increase the error of the representation. Now let $\hat{f}_u$ be our resultant representation; i.e., if $\mathcal{S}$ contains the coefficients we chose, then $\hat{f}_u = \sum_{i \in \mathcal{S}} \langle f, \psi_i \rangle \psi_i$; and let $f_{(p)}^*$ be the optimal representation under the norm ℓ_p. Consider first the case when $p \in (p_t, p_{t+1})$ where $p_t, p_{t+1} \in \mathcal{N}$

$$\begin{aligned}\|f - \hat{f}_u\|_p &\leq \|f - \hat{f}_u\|_{p_t} \\ &\leq c q^{\frac{3}{2}} (\log n) \|f - f_{(p_t)}^*\|_{p_t} \\ &\leq c q^{\frac{3}{2}} (\log n) \|f - f_{(p)}^*\|_{p_t} \\ &\leq c q^{\frac{3}{2}} (\log n) n^{\frac{1}{p_t} - \frac{1}{p}} \|f - f_{(p)}^*\|_p \end{aligned} \tag{4}$$

where the first inequality follows since $p > p_t$; the second follows from Theorem 8; the third follows from the optimality of $f_{p_t}^*$ for ℓ_{p_t}; and the final inequality is an application of Hölder's inequality. However $1/p_t - 1/p \leq 1/p_t - 1/p_{t+1}$ since $p < p_{t+1}$; and by their definition

$$\frac{1}{p_t} - \frac{1}{p_{t+1}} = \frac{\log n}{(\log n + t)(\log n + t + 1)} \leq \frac{1}{\log n}.$$

Hence, $n^{\frac{1}{p_t} - \frac{1}{p}} \leq n^{1/(\log n)} = 2$; and from expression (4) we have that $\|f - \hat{f}\|_{u_p} = O(q^{\frac{3}{2}} \log n) \|f - f^*\|_{(p)}$ as required. When $p > p_t$ for $t = \log n (\log n - 1)$, we immediately have $n^{\frac{1}{p_t} - \frac{1}{p}} \leq n^{1/(\log n)}$ and the result follows.

B. Adaptive Quantization

Wavelets are extensively used in the compression of images and audio signals. In these applications a small percent saving of space is considered important and attention is paid to the bits being stored. The techniques employed are heavily engineered and typically designed by some domain expert. The complexity is usually twofold. First, the numbers z_i do not all cost the same to represent. In some strategies; e.g., strategies used for audio signals, the number of bits of precision to represent a coefficient z_i corresponding to the basis vector $\psi_i = \psi_{j,s}$ is fixed, and it typically depends only on the scale j. (Recall that there is a mapping from ψ_i to $\psi_{j,s}$.) Further the $a_j[]$'s are computed with

a higher precision than the $d_j[]$'s. This affects the space needed by the top-most coefficients. In yet another strategy, which is standard to a broad compression literature, it is assumed that $\log_2 z$ bits are required to represent a number z. All of these bit-counting techniques need to assume that the signal is bounded and there is some reference unit of precision.

Second, in several systems, e.g., in JPEG2000 [38], a bitmap is used to indicate the nonzero entries. However the bitmap requires $O(n)$ space and it is often preferred that we store only the status of the nonzero values instead of the status of all values in the transform. In a setting where we are restricted to $o(n)$ space, as in the streaming setting, the space efficiency of the map between nonzero coefficients and locations becomes important. For example, we can represent $\psi_i = \psi_{j,s}$ using $\log\log n + \log(n/2^j) + O(1)$ bits instead of $\log n$ bits to specify i. Supposing that only the vectors with support of $\sqrt{n}$ or larger are important for a particular signal, we will then end up using half the number of bits. Notice that this encoding method *increases* the number of bits required for storing a coefficient at a small scale j to more than $\log n$. This increase is (hopefully) mitigated by savings at larger scales. Note also that the wavelet coefficients at the same level are treated similarly.

The techniques we presented in Section III naturally extend to these variants of the bit-budget problem. In what follows, we consider three specific cases.

1) *Spectrum Representations:* The cost c_i of storing a coefficient corresponding to i is fixed. This case includes the suggested strategy of using $\log\log n + \log(n/2^j) + O(1)$ bits.
2) *Bit Complexity Representations:* The cost of storing the ith coefficient with value z_i is $c_i + b(z_i)$ for some (concave) function $b()$. A natural candidate for $b()$ is $b(z) = O(1) - \log z_i^{\text{frac}}$ where z_i^{frac} is the fractional part of z_i and is less than 1 (thus $-\log z_i^{\text{frac}}$ is positive). This encodes the idea that we can store a higher "resolution" at a greater cost.
3) *Multiplane Representations:* Here the data conceptually consists of several "planes", and the cost of storing the ith coefficient in one plane depends on whether the ith coefficient in another plane is retained. For example, suppose we are trying to represent a RGBA image which has four attributes per pixel. Instead of regarding the data as 4×2 dimensional, it may be more useful, for example if the variations in color are nonuniform, to treat the data as being composed of several separate planes, and to construct an optimization that allocates the bits across them.

The fundamental method by which we obtain our approximate solutions to the above three problems is to use a greedy rule to lower bound the errors of the optimal solutions using systems of constraints as we did in Section III. We focus only on the ℓ_∞ error for ease of presentation. As before, the techniques we use imply analogous results for ℓ_p norms.

1) Spectrum Representations: In the case where the cost of storing a number for i is a fixed quantity c_i we obtain a lower bound via a quadratic program that is similar to (1) using Lemma 2. That is, minimize τ with the constraints $x_i \in \{0,1\}$ and $\sum_i x_i c_i \leq B$, and for all i

$$-\tau||\psi_i||_1 \;\leq\; \langle f, \psi_i\rangle - x_i z_i \;\leq\; \tau||\psi_i||_1. \tag{5}$$

The program above can be solved optimally since the c_i's are polynomially bounded. We sort the coefficients in nonincreasing order of $y_i := |\langle f, \psi_i\rangle|/||\psi_i||_1$. If $y_{i_1} \geq y_{i_2} \geq \cdots \geq y_{i_n}$, then we include coefficients $i_1, \ldots, i_k$ where $\sum_{j=1}^{k} c_{i_j} \leq B < \sum_{j=1}^{k+1} c_{i_j}$. The value $y_{i_{k+1}}$ is then a lower bound on the error $\mathcal{E}$ of the optimal representation z^*. Note that z^* is a feasible solution to program (5). Hence, either z^* includes coefficients $i_1, \ldots, i_k$ in which case it cannot choose coefficient i_{k+1} for it will exceed the space bound B, and we have that $\mathcal{E} \geq y_{i_{k+1}}$ (the optimal does not necessarily set $z_i = |\langle f, \psi_i\rangle|$); or, z^* does not include one of $i_1, \ldots, i_k$, thus $\mathcal{E}$ is again greater then or equal to $y_{i_{k+1}}$. A proof similar to that of Theorem 5 shows that the error of our solution is $O(\log n)\mathcal{E}$.

2) Bit Complexity Representations: In the case where the cost is dependent on z_i we cannot write an explicit system of equations as we did in the case of spectrum representations. However, we can guess τ up to a factor of 2 and verify if the guess is correct.

In order to verify the guess, we need to be able to solve equations of the form $\min_z b(z)$ s.t. $|a - z| \leq t$ (since this is the format of our constraints). This minimization is solvable for most reasonable cost models; e.g., if $b(z)$ is monotonically increasing. As the coefficients are generated, we compute $c_i + b(z_i)$ if $z_i \neq 0$, where $z_i = \text{argmin}_z b(z)$ s.t. $|\langle f, \psi_i\rangle - z| \leq t||\psi_i||_1$ for our guess t of the error. If we exceed the allotted space B at any point during the computation, we know that our guess t is too small, and we start the execution over with the guess $2t$. Note that the optimal representation is a feasible solution with value $\mathcal{E}$ and bit complexity B. Applying the analysis of Section III,1) shows that the first solution we obtain that respects our guess is a $O(\log n)$ approximation to the optimal representation.

Since we assume that the error $\mathcal{E}$ is polynomially bounded, the above strategy can be made to stream by running $O(\log n)$ greedy algorithms in parallel each with a different guess of τ as above.

3) Multiplane Representations: In this case we are seeking to represent data that is conceptually in several "planes" simultaneously; e.g., RGBA in images. We could also conceptualize images of the same object at various frequencies or technologies. The goal of the optimization is to allocate the bits across them. However, notice that if we choose the ith coefficient for say the Red and the Blue planes (assuming that we are indicating the presence or absence of a coefficient explicitly which is the case for a sparse representation), then we can save space by storing the fact that "coefficient i is chosen" only once. This is easily achieved by keeping a vector of four bits corresponding to each chosen coefficient. The values of the entries in the bit vector inform us if the respective coefficient value is present. Therefore, the bit vector 1010 would indicate that the next two values in the data correspond to Red and Blue values of a chosen coefficient. Similarly, a vector 1011 would suggest that three values corresponding to Red, Blue and Alpha are to be expected.

In what follows, we assume that the data is D dimensional and it is comprised of t planes (in the RGBA example, $D = 2$ and $t = 4$). We are constrained to storing at most B bits total for the bit vectors, the indices of the chosen coefficients, and

the values of these coefficients. For simplicity we assume that we are using the ℓ_∞ error across all the planes. Otherwise, we would also have to consider how the errors across the different planes are combined.

We construct our approximate solution by first sorting the coefficients of the t planes in a single nonincreasing order while keeping track of the plane to which each coefficient belongs. As before, we add the coefficients that are largest in this ordering to our solution, and stop immediately before the coefficient whose addition results in exceeding the alloted space B. Note that if we had added the ith coefficient of the Red plane first, and thereafter wanted to include the Blue plane's ith coefficient, then we need only account for the space of storing the index i and the associated bit vector when we add the coefficient for the first (in this case Red) plane. The subsequent ith coefficients only contribute to the cost of storing their values to the solution. (We can think of the cost of storing each coefficient as fixed *after* the ordering of the coefficients is determined.) This strategy is reminiscent of the strategy used by Guha, Kim and Shim [39] to lower bound the optimum error for a similar problem in the ℓ_2 setting.

The first coefficient that we did not choose using this greedy selection process is a lower bound on the optimal representation error. Now, an argument similar to that of Theorem 5 shows that the error of the resulting solution is a $O(\log n)$ factor away from the error of the optimal solution.

C. Sparse Image Representation Under Non-ℓ_2 Error Measures

In this section, we give three examples that demonstrate uses for our greedy algorithm in compressing images. A nonstreaming version of the algorithm for Haar and Daubechies wavelets was implemented in MATLAB using the Uvi_Wave·300 toolbox[5] [40]. Pseudocode of the implementation is provided below in Fig. 1. The algorithm takes four parameters as input: the image X, the number of coefficients to retain B, the p-norm to minimize, and the type of Daubechies wavelet to use. The last parameter, q, determines the number of nonzero coefficients in the wavelet filter. Recall that the Haar wavelet is the Daubechies wavelet with smallest support; i.e., it has $q = 1$.

The first example illustrates a use of the ℓ_∞ measure for sparse representation using wavelets. Minimizing the maximum error at any point in the reconstructed image implies we should retain the wavelet coefficients that correspond to sharp changes in intensity; i.e., the coefficients that correspond to the "details" in the image. The image we used, shown in Fig. 2(a), is composed of a gradient background and both Japanese and English texts.[6] The number of nonzero wavelet coefficients in the original image is 65524. We set $B = 3840$ and ran Algorithm *daubGreedy* with $p = 1, 2$, and ∞ under the Haar wavelet (with $q = 1$). When $p = 2$, the algorithm outputs the optimal

Algorithm *DaubGreedy*(X, B, p, q)
1. $(*$ X is a grayscale image (intensity matrix) $*)$
2. Perform a 2D wavelet transform of X using the Daubechies D_q wavelet
3. Let w be the wavelet coefficients of the transform
4. $p' \leftarrow p/(p-1)$
5. $y_i \leftarrow |w_i|/\|\psi_i\|_{p'}$
6. Let $\mathcal{I}$ be the indices of the B largest y_i's
7. $w_i \leftarrow 0$ if $i \notin \mathcal{I}$
8. Perform a 2D inverse wavelet transform on w
9. Let X' be the resulting image representation
10. **return** X'

Fig. 1. Pseudocode of the greedy algorithm's implementation.

B-term representation that minimizes the ℓ_2 error measure. That is, the algorithm simply retains the largest B wavelet coefficients (since $p' = 2$ and $\|\psi_i\|_{p'} = 1$ for all i). When $p = 1$, or $p = \infty$, the algorithm outputs a $O(\log n)$-approximate B-term representation as will be explained in Section III. The results are shown in Fig. 2. Notice that the ℓ_∞ representation essentially ignores the gradient in the background, and it retains the wavelet coefficients that correspond to the text in the image. The ℓ_1 representation also does better than the ℓ_2 representation in terms of rendering the Japanese text; however, the English translation in the former is not as clear. The attribution in the ℓ_2 representation, on the other hand, is completely lost. Although the differences between the three representations are not stark, this example shows that under such high compression ratios using the ℓ_∞ norm is more suitable for capturing signal details than other norms.

The second example illustrates a use of the ℓ_1 error measure. Since the ℓ_1 norm is robust in the sense that it is indifferent to outliers, the allocation of wavelet coefficients when minimizing the ℓ_1 norm will be less sensitive to large changes in intensity than the allocation under the ℓ_2 norm. In other words, it implies that under the ℓ_1 norm the wavelet coefficients will be allocated more evenly across the image. The image we used, shown in Fig. 3(a), is a framed black and white matte photograph. The number of nonzero wavelet coefficients in the original image is 65536. We set $B = 4096$ and ran Algorithm *daubGreedy* with $p = 1, 2$, and ∞ under the Daubechies D_2 wavelet. The results are shown in Fig. 3. Notice that the face of the subject is rendered in the ℓ_1 representation more "smoothly" than in the ℓ_2 representation. Further, the subject's mouth is not portrayed completely in the ℓ_2 representation. As explained earlier, these differences between the two representations are due to the fact that the ℓ_1 norm is not as affected as the ℓ_2 norm by other conspicuous details in the image; e.g., the frame. The ℓ_∞ representation, on the other hand, focuses on the details of the image displaying parts of the frame and the eyes well, but misses the rest of the subject entirely. This example foregrounds some advantages of the ℓ_1 norm over the customary ℓ_2 norm for compressing images.

The last example highlights the advantage of representing an image sparsely using a nonlinear wavelet approximation versus using a rank-k approximation of the image. Recall that if X is our image then the best rank-k approximation is given by $U_k \Sigma_k V_k^T$ where $X = U \Sigma V^T$ is the SVD decomposition of X, and U_k is comprised of the k singular vectors corresponding to

[5]For compatibility with our version of MATLAB, slight modifications on the toolbox were performed. The toolbox can be obtained from http://www.gts.tsc.uvigo.es/~wavelets/.

[6]The Japanese text is poem number 89 of the *Kokinshu* anthology [41]. The translation is by Helen Craig McCullough.

Fig. 2. Representing an image with embedded text using the optimal strategy that minimizes the ℓ_2 error, and our greedy approximation algorithm under the ℓ_∞ and ℓ_1 error measures. The Haar wavelet is used in all three representations, and the number of retained coefficients is $B = 3840$ (a) The original image. (b) Output of the optimal ℓ_2 algorithm (which retains the largest B wavelet coefficients). (c) Output of our greedy algorithm under ℓ_∞. (d) Output of our greedy algorithm under ℓ_1.

the largest k singular values of X (see, e.g., [42]). The original image is shown in Fig. 4(a)[7] and the number of nonzero coefficients in its Haar wavelet expansion is 65536. Fig. 4(c) shows the best rank-12 approximation of the image; i.e., it displays $X_{12} = U_{12}\Sigma_{12}V_{12}^T$. This representation stores 6144 values corresponding to the number of elements in $U_{12}\Sigma_{12}$ plus V_{12}. We set $B = 3072$ and ran Algorithm *daubGreedy* with $p = 1, 2$ under the Haar wavelet (Fig. 4(d) and (b)). (The B-term representation problem implicitly requires storing two B numbers: the B *values* of the solution components that we compute, and the B indices of these components.) It is clear that the nonlinear approximations offer perceptually better representations that the approximation offered by the SVD. Also, as in the previous example, the ℓ_1 representation is again "smoother" than the ℓ_2 with less visible artifacts.

[7]The image is taken from a water painting by Shozo Matsuhashi. It is untitled.

V. A Streaming $(1+\epsilon)$ Approximation for Haar Wavelets

In this section, we will provide a FPTAS for the Haar system. The algorithm will be bottom up, which is convenient from a streaming point of view. Observe that in case of general ℓ_p norm error, we cannot disprove that the optimum solution cannot have an irrational value, which is detrimental from a computational point of view. In a sense, we will seek to narrow down our search space, but we will need to preserve near optimality. We will show that *there exists* sets R_i such that if the solution coefficient z_i was drawn from R_i, then *there exists* one solution which is close to the optimum unrestricted solution (where we search over all reals). In a sense the sets R_i "rescue" us from the search. Alternately we can view those sets as a "rounding" of the optimal solution. Obviously such sets exist if we did not care about the error, e.g., take the all zero solution. We would

Fig. 3. Representing an image using the optimal strategy that minimizes the ℓ_2 error, and our greedy approximation algorithm under the ℓ_∞ and ℓ_1 error measures. The Daubechies D_2 wavelet is used in all three representations, and the number of retained coefficients is $B = 4096$. (a) The original image. (b) Output of the optimal ℓ_2 algorithm (which retains the largest B wavelet coefficients). (c)Output of our greedy algorithm under ℓ_∞. (d) Output of our greedy algorithm under ℓ_1.

expect a dependence between the sets R_i and the error bound we seek. We will use a type of "dual" wavelet bases; i.e., where we use one basis to construct the coefficients and another to reconstruct the function. Our bases will differ by scaling factors. We will solve the problem in the scaled bases and translate the solution to the original basis. This overall approach is similar to that in [43], however, it is different in several details critical to the proofs of running time, space complexity and approximation guarantee.

Definition 1: Define $\psi_{j,s} = 2^{-j/2}\psi_{j,s}$ and $\psi_{j,s} = 2^{j/2}\psi_{j,s}$. Likewise define $\phi_{j,s} = 2^{-j/2}\phi_{j,s}$.

Proposition 9: The Cascade algorithm used with $\frac{1}{\sqrt{2}}h[]$ computes $\langle f, \psi_i^n \rangle$ and $\langle f, \phi_i^n \rangle$.

We now use the change of basis. The next proposition is clear from the definition of $\{\psi_i^b\}$.

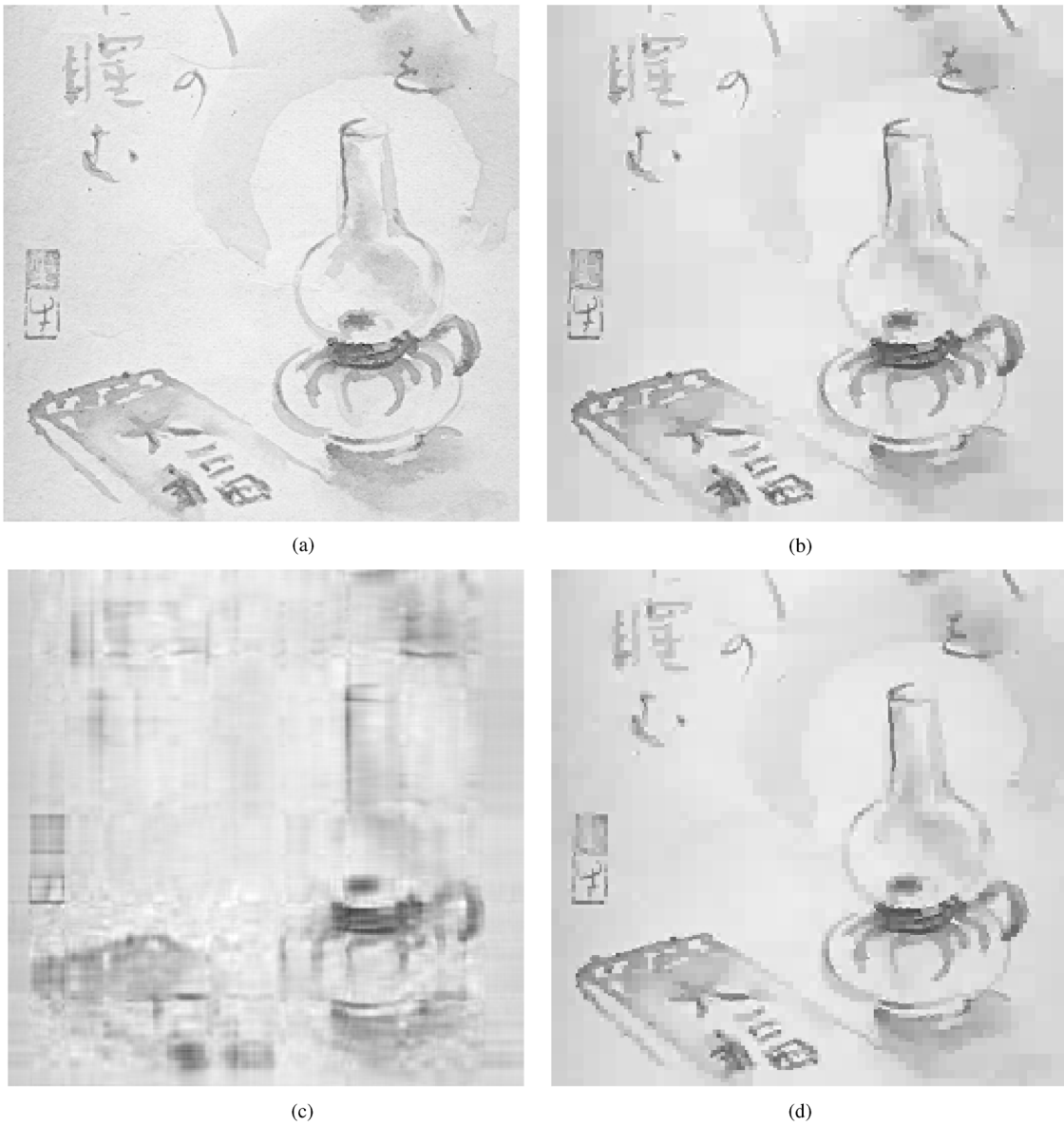
Proposition 10: The problem of finding a representation $\hat{f}$ with $\{z_i\}$ and basis $\{\psi_i\}$ is equivalent to finding the same representation $\hat{f}$ using the coefficients $\{y_i\}$ and the basis $\{\psi_i\}$. The correspondence is $y_i = y_{j,s} = 2^{-j/2}z_{j,s}$.

Lemma 11: Let $\{y_i^*\}$ be the optimal solution using the basis set $\{\psi_i\}$ for the reconstruction, i.e., $\hat{f} = \sum_i y_i^* \psi_i$ and $\|f - \hat{f}\|_p = \mathcal{E}$. Let $\{y_i^\rho\}$ be the set where each y_i^* is rounded to the nearest multiple of ρ. If $f^\rho = \sum_i y_i^\rho \psi_i^b$ then $\|f - f^\rho\|_p \leq \mathcal{E} + O(qn^{1/p}\rho \log n)$.

Proof: Let $\rho_i = y_i^* - y_i^\rho$. By the triangle inequality

$$\|f - f^\rho\|_p \leq \mathcal{E} + \|\sum_i \rho_i \psi_i^b\|_p.$$

Proposition 1 and the fact that $|\rho_i| \leq \rho$ imply $|\sum_k \rho_i \psi_i^b(k) \leq c\rho q \log n \max_i |\psi_i^b(k)$ for a small constant c. This bound gives



(a) (b) (c) (d)

Fig. 4. Representing an image using the optimal strategy that minimizes the ℓ_2 error and using our greedy approximation algorithm under the ℓ_1 error measure versus its best rank-k approximation. Here $k = 12$, and the number of values stored in all three representations is 6144. The Haar wavelet is used in the two nonlinear representations (the number of retained wavelet coefficients is $B = 3072$). (a) The original image. (b) Output of the optimal ℓ_2 algorithm (which retains the largest B wavelet coefficients). (c)Output of the best rank-12 approximation. (d) Output of our greedy algorithm under ℓ_1.

$\|f - f^\rho\|_p \leq \mathcal{E} + O(qn^{1/p}\rho \log n \max_i \|\psi_i^b\|_\infty)$. Now $\psi_i^b = \psi_{j,s}^b = 2^{j/2}\psi_{j,s}$, and from the Proof of Lemma 4 we know that for large j, $\|\psi_{j,s}\|_\infty$ is at most $2^{-j/2}$ times a constant. For smaller j, $\|\psi_{j,s}^b\|_\infty$ is a constant. □

We will provide a dynamic programming formulation using the new basis. But we still need to show two results; the first concerning the y_i^*'s and the second concerning the $a_j[]$'s. The next lemma is very similar to Lemma 2 and follows from the fact that $\|\psi_{j,s}^a\|_1 = 2^{-j/2}\|\psi_{j,s}\|_1 \leq \sqrt{2q}$.

Lemma 12: $-C_0\sqrt{q}\mathcal{E} \leq \langle f, \psi_i^a \rangle - y_i^* \leq C_0\sqrt{q}\mathcal{E}$ for some constant C_0.

Now suppose we know the optimal solution $\hat{f}$, and suppose we are computing the coefficients $a_j[]$ and $d_j[]$ for both f and $\hat{f}$ at each step j of the Cascade algorithm. We wish to know by how much their coefficients differ since bounding this gap would shed more light on the solution $\hat{f}$.

Proposition 13: Let $a_j[s](F)$ be $a_j[s]$ computed from $a_0[s] = F(s)$, then $a_j[s](f) - a_j[s](\hat{f}) = a_j[s](f - \hat{f})$.

Lemma 14: If $\|f - \hat{f}\|_p \leq \mathcal{E}$ then $|a_j[s](f - \hat{f})| \leq C_1\sqrt{q}\mathcal{E}$ for some constant C_1. (We are using $\frac{1}{\sqrt{2}}h[]$.)

Proof: The proof is similar to that of Lemma 2. Let $F = f - \hat{f}$. We know $-\mathcal{E} \leq F(i) \leq \mathcal{E}$. Multiplying by $|\phi_{j,s}^a(i)|$ and summing over all i we get $-\mathcal{E}\|\phi_{j,s}^a\|_1 \leq \langle F, \phi_{j,s}^a \rangle = a_j[s](F) \leq \mathcal{E}\|\phi_{j,s}^a\|_1$. By definition, $\phi a_{j,s} = 2^{-j/2}\phi_{j,s}$. Further, $\|\phi_{j,s}\|_2 = 1$ and has at most $(2q)2^j$ nonzero values. Hence, $\|\phi_{j,s}^a\|_1 \leq \sqrt{2q}$. The lemma follows. □

At this point we have all the pieces. Summarizing:

Lemma 15: Let $\{z_i\}$ be a solution with B nonzero coefficients and with representation $\hat{f} = \sum_i z_i \psi_i$. If $\|f - \hat{f}\|_p \leq \mathcal{E}$, then there is a solution $\{y_i\}$ with B nonzero coefficients and representation $f' = \sum_i y_i \psi_i^b$ such that for all i we have the following:

i) y_i is a multiple of ρ;
ii) $|y_i - \langle f, \psi_i^a \rangle| \leq C_0 \sqrt{q}\mathcal{E} + \rho$; and
iii) $|\langle f, \phi_i^a \rangle - \langle f', \phi_i^a \rangle| \leq C_1 \sqrt{q}\mathcal{E} + O(q\rho \log n)$;

and $\|f - f'\|_p \leq \mathcal{E} + O(qn^{1/p}\rho \log n)$.

Proof: Rewrite $\hat{f} = \sum_i z_i \psi_i = \sum_i z_i^* \psi_i^b$ where $z_i^* = z_{j,s}^* = 2^{-j/2} z_{j,s}$. Let $\{y_i\}$ be the solution where each y_i equals z_i^* rounded to the nearest multiple of ρ. Lemmas 12 and 14 bound the z_i^*'s thus providing properties ii) and iii). Finally, Lemma 11 gives the approximation guarantee of $\{y_i\}$. □

The above lemma ensures the existence of a solution $\{y_i\}$ that is $O(qn^{1/p}\rho \log n)$ away from the optimal solution and that possesses some useful properties which we shall exploit for designing our algorithms. Each coefficient y_i in this solution is a multiple of a parameter ρ that we are free to choose, and it is a constant multiple of $\mathcal{E}$ away from the ith wavelet coefficient of f. Further, without knowing the values of those coefficients $y_{j,s}$ contributing to the reconstruction of a certain point $f'(i)$, we are guaranteed that during the incremental reconstruction of $f'(i)$ using the cascade algorithm, every $a_j[s](f')$ in the support of $f'(i)$ is a constant multiple of $\mathcal{E}$ away from $a_j[s](f) = \langle f, \phi_{j,s}^a \rangle$. This last property allows us to design our algorithms in a bottom-up fashion making them suitable for data streams. Finally, since we may choose ρ, setting it appropriately results in true factor approximation algorithms. Details of our algorithms follow.

A. The Algorithm: A Simple Version

We will assume here that we know the optimal error $\mathcal{E}$. This assumption can be circumvented by running $O(\log n)$ instances of the algorithm presented below "in parallel," each with a different guess of the error. This will increase the time and space requirements of the algorithm by a $O(\log n)$ factor, which is accounted for in Theorem 16 (and also in Theorem 18). We detail the guessing procedure in Section V-A1. Our algorithm will be given $\mathcal{E}$ and the desired approximation parameter ϵ as inputs (see Fig. 6).

The Haar wavelet basis naturally form a complete binary tree, termed the *coefficient tree*, since their support sets are nested and are of size powers of 2 (with one additional node as a parent of the tree). The data elements correspond to the leaves, and the coefficients correspond to the nonleaf nodes of the tree. Assigning a value y to the coefficient corresponds to assigning $+y$ to all the leaves that are *left descendants* (descendants of the left child) and $-y$ to all right descendants (recall the definition of $\{\psi_i^b\}$). The leaves that are descendants of a node in the coefficient tree are termed the *support* of the coefficient.

Definition 2: Let $E[i, v, b]$ be the minimum possible contribution to the overall error from all descendants of node i using exactly b coefficients, under the assumption that ancestor coefficients of i will add up to the value v at i (taking account of the signs) in the final solution.

The value v will be set later for a subtree as more data arrive. Note that the definition is bottom up and after we compute the table, we do not need to remember the data items in the subtree. As the reader would have guessed, this second property will be significant for streaming.

The overall answer is $\min_b E[\text{root}, 0, b]$—by the time we are at the root, we have looked at all the data and no ancestors exist to set a nonzero v. A natural dynamic program arises whose idea is as follows. Let i_L and i_R be node i's left and right children respectively. In order to compute $E[i, v, b]$, we guess the coefficient of node i and minimize over the error produced by i_L and i_R that results from our choice. Specifically, the computation is as follows.

1) A nonroot node computes $E[i, v, b]$ as follows:

$$\min \begin{cases} \min_{r,b'} E[i_L, v + r, b'] + E[i_R, v - r, b - b' - 1] \\ \min_{b'} E[i_L, v, b'] + E[i_R, v, b - b'] \end{cases}$$

where the upper term computes the error if the ith coefficient is chosen and its value is $r \in R_i$ where R_i is the set of multiples of ρ between $\langle f, \psi_i^a \rangle - C_0 \sqrt{q}\mathcal{E}$ and $\langle f, \psi_i^a \rangle + C_0 \sqrt{q}\mathcal{E}$; and the lower term computes the error if the ith coefficient is not chosen.

2) Then the root node computes

$$\min \begin{cases} \min_{r,b'} E[i_C, r, b' - 1], & \text{root coefficient is } r \\ \min_{b'} E[i_C, 0, b'], & \text{root not chosen} \end{cases}$$

where i_C is the root's only child.

The streaming algorithm will borrow from the paradigm of reduce-merge. The high level idea will be to construct and maintain a small table of possibilities for each resolution of the data. On seeing each item $f(i)$, we will first find out the best choices of the wavelets of length one (over all future inputs) and then, if appropriate, construct/update a table for wavelets of length $2, 4, \ldots$, etc.

The idea of subdividing the data, computing some information and merging results from adjacent divisions were used in [27] for stream clustering. The stream computation of wavelets in [24] can be viewed as a similar idea—where the divisions corresponds to the support of the wavelet basis vectors.

Our streaming algorithm will compute the error arrays $E[i, \cdot, \cdot]$ associated with the internal nodes of the coefficient tree in a post-order fashion. Recall that the wavelet basis vectors, which are described in Section II, form a complete binary tree. For example, the scaled basis vectors for nodes 4, 3, 1, and 2 in the tree of Fig. 5(a) are $[1, 1, 1, 1]$, $[1, 1, -1, -1]$, $[1, -1, 0, 0]$ and $[0, 0, 1, -1]$, respectively. The data elements correspond to the leaves of the tree and the coefficients of the synopsis correspond to its internal nodes.

We need not store the error array for every internal node since, in order to compute $E[i, v, b]$ our algorithm only requires that $E[i_L, \cdot, \cdot]$ and $E[i_R, \cdot, \cdot]$ be known. Therefore, it is natural to perform the computation of the error arrays in a post-order fashion. An example best illustrates the procedure. Suppose $f = \langle x_1, x_2, x_3, x_4 \rangle$. In Fig. 5(a), when element x_1 arrives, the algorithm computes the error array associated with x_1, call it E_{x_1}. When element x_2 arrives E_{x_2} is computed. The array $E[1, \cdot, \cdot]$ is then computed and E_{x_1} and E_{x_2} are discarded. Array E_{x_3} is computed when x_3 arrives. Finally the arrival of x_4 triggers the

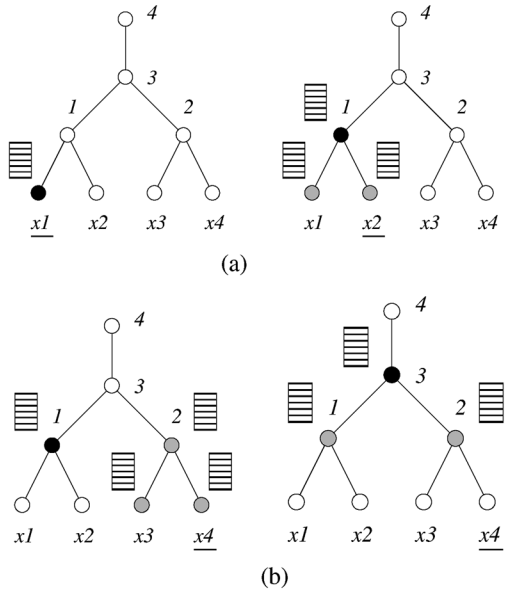


Fig. 5. Upon seeing x_2 node 1 computes $E[1,\cdot,\cdot]$ and the two error arrays associated with x_1 and x_2 are discarded. Element x_4 triggers the computation of $E[2,\cdot,\cdot]$ and the two error arrays associated with x_3 and x_4 are discarded. Subsequently, $E[3,\cdot,\cdot]$ is computed from $E[1,\cdot,\cdot]$ and $E[2,\cdot,\cdot]$ and both the latter arrays are discarded. If x_4 is the last element on the stream, the root's error array, $E[3,\cdot,\cdot]$, is computed from $E[2,\cdot,\cdot]$ (a) The arrival of the first three elements. (b) The arrival of x_4.

computations of the rest of the arrays as in Fig. 5(b). Note that at any point in time, there is only one error array stored at each *level* of the tree. In fact, the computation of the error arrays resembles a binary counter. We start with an empty queue Q of error arrays. When x_1 arrives, E_{q_0} is added to Q and the error associated with x_1 is stored in it. When x_2 arrives, a temporary node is created to store the error array associated with x_2. It is immediately used to compute an error array that is added to Q as E_{q_1}. Node E_{q_0} is emptied, and it is filled again upon the arrival of x_3. When x_4 arrives: 1) a temporary E_{t_1} is created to store the error associated with x_4; 2) E_{t_1} and E_{q_0} are used to create E_{t_2}; E_{t_1} is discarded and E_{q_0} is emptied; 3) E_{t_2} and E_{q_1} are used to create E_{q_2} which in turn is added to the queue; E_{t_2} is discarded and E_{q_1} is emptied. The algorithm for ℓ_∞ is shown in Fig. 6.

1) Guessing the Optimal Error: We have so far assumed that we know the optimal error $\mathcal{E}$. As mentioned at the beginning of Section V-A, we will avoid this assumption by running multiple instances of our algorithm and supplying each instance a different guess G_k of the error. We will also provide every instance A_k of the algorithm with $\epsilon' = \frac{\sqrt{1+4\epsilon}-1}{2}$ as the approximation parameter. The reason for this will be apparent shortly. Our final answer will be that of the instance with the minimum representation error.

Theorem 16 shows that the running time and space requirements of our algorithm do not depend on the supplied error parameter. However, the algorithm's search ranges *do* depend on the given error. Hence, as long as $G_k \geq \mathcal{E}$ the ranges searched by the kth instance will include the ranges specified by Lemma 15. Lemma 15 also tells us that if we search these ranges in multiples of ρ, then we will find a solution whose approximation guarantee is $\mathcal{E} + cqn^{1/p}\rho \log n$. Our algorithm chooses ρ so that its running time does not depend on the supplied error parameter. Hence, given G_k and ϵ', algorithm A_k sets $\rho = \epsilon' G_k/(cqn^{1/p}\log n)$. Consequently, its approximation guarantee is $\mathcal{E} + \epsilon' G_k$.

Now if guess G_k is much larger than the optimal error $\mathcal{E}$, then instance A_k will not provide a good approximation of the optimal representation. However, if $G_k \leq (1+\epsilon')\mathcal{E}$, then A_k's guarantee will be $\mathcal{E} + \epsilon'(1+\epsilon')\mathcal{E} = (1+\epsilon)\mathcal{E}$ because of our choice of ϵ'. To summarize, in order to obtain the desired $(1+\epsilon)$ approximation, we simply need to ensure that one of our guesses (call it G_{k^*}) satisfies

$$\mathcal{E} \leq G_{k^*} \leq (1+\epsilon')\mathcal{E}$$

Setting $G_k = (1+\epsilon')^k$, the above bounds will be satisfied when $k = k^* \in [\log_{1+\epsilon'}(\mathcal{E}), \log_{1+\epsilon'}(\mathcal{E}) + 1]$.

Number of guesses: Note that the optimal error $\mathcal{E} = 0$ if and only if f has at most B nonzero expansion coefficients $\langle f, \psi_i \rangle$. We can find these coefficients easily in a streaming fashion.

Since we assume that the entries in the given f are polynomially bounded, by the system of (1) we know that the optimum error is at least as much as the $(B+1)^{\text{st}}$ largest coefficient. Now any coefficient $(\langle f, \psi_k^a \rangle)$ is the sum of the left half minus the sum of the right half of the f_i's that are in the support of the basis and the total is divided by the length of the support. Thus if the smallest nonzero number in the input is n^{-c} then the smallest nonzero wavelet coefficient is at least $n^{-(c+1)}$. By the same logic the largest nonzero coefficient is n^c. Hence, it suffices to make $O(\log n)$ guesses.

B. Analysis of the Simple Algorithm

The size of the error table $E[i,\cdot,\cdot]$ at node i is $R_\phi \min\{B, 2^{t_i}\}$ where $R_\phi = 2C_1\mathcal{E}/\rho + \log n$ and t_i is the height of node i in the Haar coefficient tree (the leaves have height 0). Note that $q = 1$ in the Haar case. Computing each entry of $E[i,\cdot,\cdot]$ takes $O(R_\psi \min\{B, 2^{t_i}\})$ time where $R_\psi = 2C_0\mathcal{E}/\rho + 2$. Hence, letting $R = \max\{R_\phi, R_\psi\}$, the total running time is $O(R^2B^2)$ for computing the root table plus $O(\sum_{i=1}^{n}(R\min\{2^{t_i}, B\})^2)$ for computing all the other error tables. Now

$$\begin{aligned}\sum_{i=1}^{n}(R\min\{2^{t_i}, B\})^2 &= R^2\sum_{t=1}^{\log n}\frac{n}{2^t}\min\{2^{2t}, B^2\} \\ &= nR^2\left(\sum_{t=1}^{\log B}2^t + \sum_{t=\log B+1}^{\log n}\frac{B^2}{2^t}\right) \\ &= O(R^2nB),\end{aligned}$$

where the first equality follows from the fact that the number of nodes at level t is $\frac{n}{2^t}$. For ℓ_∞, when computing $E[i,v,b]$ we do not need to range over all values of B. For a specific $r \in R_i$, we can find the value of b' that minimizes $\max\{E[i_L, v+r, b'], E[i_R, v-r, b-b'-1]\}$ using binary search. The running time thus becomes

$$\sum_t R^2\frac{n}{2^t}\min\{t2^t, B\log B\} = O(nR^2\log^2 B).$$

Algorithm *HaarPTAS*$(B, \mathcal{E}, \epsilon)$

1. Let $\rho = \epsilon\mathcal{E}/(cq \log n)$ for some suitably large constant c. Note that $q = 1$ in the Haar case.
2. Initialize a queue Q with one node q_0

($*$ Each q_i contains an array E_{q_i} of size at most $R \min\{B, 2^i\}$ and a flag `isEmpty` $*$)

3. **repeat** Until there are no elements in the stream
4. Get the next element from the stream, call it e
5. **if** q_0 is empty
6. **then** Set $q_0.a = e$. For all values r s.t. $|r - e| \le c_1\mathcal{E}$ where c_1 is a large enough constant and r is a multiple of ρ, initialize the table $E_{q_0}[r, 0] = |r - e|$
7. **else** Create t_1 and Initialize $E_{t_1}[r, 0] = |r - e|$ *as in Step 6.*
8. **for** $i = 1$ until the 1^{st} empty q_i or end of Q
9. **do** Create a temporary node t_2.
10. Compute $t_2.a = \langle f, \phi_i^a \rangle$ and the wavelet coefficient $t_2.o = \langle f, \psi_i^a \rangle$. This involves using the a values of t_1 and q_{i-1} (t_2's two children in the coefficient tree) and taking their average to compute $t_2.u$ and their difference divided by 2 to compute $t_2.o$. (Recall that we are using $\frac{1}{\sqrt{2}}h[]$).
11. For all values r that are multiples of ρ with $|r - t_2.a| \le c_1(\mathcal{E} + \rho \log n)$, compute the table $E_{t_2}[r, b]$ for all $0 \le b \le B$. This uses the tables of the two children t_1 and q_{i-1}. The size of the table is $O(\epsilon^{-1} B n^{1/p} \log n)$. (Note that the value of a chosen coefficient at node t_2 is at most a constant multiple of $\mathcal{E}$ away from $t_2.o$. Keeping track of the chosen coefficients (the answer) costs $O(B)$ factor space more.)
12. Set $t_1 \leftarrow t_2$ and Discard t_2
13. Set q_i.`isEmtpy` = true
14. **if** we reached the end of Q
15. **then** Create the node q_i
16. Compute $E_{q_i}[r, b \in B]$ from t_1 and q_{i-1} *as in Step 11.*
17. Set q_i.`isEmpty` = false and Discard t_1

Fig. 6. The Haar streaming FPTAS for ℓ_∞.

The bottom-up dynamic programming will require us to store the error tables along at most two leaf to root paths. Thus the required space is

$$2\sum_t R \min\{2^t, B\} = O(RB(1 + \log \frac{n}{B})).$$

Finally, $R = O((n^{1/p} \log n)/\epsilon)$ since we have set $\rho = \epsilon\mathcal{E}/(cn^{1/p} \log n)$.

Theorem 16: Algorithm *HaarPTAS* computes a $(1 + \epsilon)$ approximation to the best B-term unrestricted representation of a signal in the Haar system using $O(\epsilon^{-1} B^2 n^{1/p} \log^3 n)$ space. Under the ℓ_p norm, the algorithm runs in time $O(\epsilon^{-2} n^{1+2/p} B \log^3 n)$. Under ℓ_∞ the running time becomes $O(\epsilon^{-2} n \log^2 B \log^3 n)$.

The extra B factor in the space required by the algorithm accounts for keeping track of the chosen coefficients.

C. An Improved Algorithm and Analysis

For large n (compared to B), we gain in running time if we change the rounding scheme given by Lemma 11. The granularity at which we search for the value of a coefficient will be fine if the coefficient lies toward the top of the tree, and it will be coarse if the coefficient lies toward the bottom. The idea is that, for small ℓ_p norms, a mistake in a coefficient high in the tree affects everyone, whereas mistakes at the bottom are more localized. This idea utilizes the strong locality property of the Haar basis. We start with the lemma analogous to Lemma 11.

Lemma 17: Let $\{y_i^*\}, i = (t_i, s)$ be the optimal solution using the basis set $\{\psi_i^b\}$ for the reconstruction, i.e., $\hat{f} = \sum_i y_i^* \psi_i^b$ and $\|f - \hat{f}\|_p = \mathcal{E}$. Here t_i is the height of node i in the Haar coefficient tree. Let $\{y_i^\rho\}$ be the set where each y_i^* is first rounded to the nearest multiple of $\rho_{t_i} = \epsilon\mathcal{E}/(2B2^{t_i/p})$ then the resulting value is rounded to the nearest multiple of $\rho_{t_{\text{root}}} = \epsilon\mathcal{E}/(2Bn^{1/p})$. If $f^\rho = \sum_i y_i^\rho \psi_i^b$ then $\|f - f^\rho\|_p \le (1 + \epsilon)\mathcal{E}$.

Proof: As in Lemma 11, we need to estimate $\|\sum_i (y_i^\rho - y_i^*)\psi_i^b\|_p$ but using the new rounding scheme. Let $\mathcal{S}$ be the set of indices i such that $y_i \neq 0$

$$\begin{aligned}\|\sum_{i\in\mathcal{S}} (y_i^\rho - y_i^*)\psi_i^b\|_p &\le \sum_{i\in\mathcal{S}} \|(y_i^\rho - y_i^*)\psi_i^b\|_p \\ &\le \sum_{i\in\mathcal{S}} (\rho_{t_i} + \rho_{t_{\text{root}}})\|\psi_i^b\|_p \\ &\le 2\sum_{i\in\mathcal{S}} \rho_{t_i} 2^{t_i/p}.\end{aligned}$$

The last inequality follows from the fact that 2^{t_i} components of ψ_i^b are equal to one and the rest are zero. The approximation hence follows from $|\mathcal{S}| \le B$ and our choices of ρ_{t_i}. □

The granularity of the dynamic programming tables $E[i, \cdot, \cdot]$ is set according to the smallest ρ_{t_i}, which is $\rho_{t_{\text{root}}} = \epsilon\mathcal{E}/(2Bn^{1/p})$. This allows their values to align correctly. More specifically, when a coefficient is not chosen we compute (see Section V-A)

$$E[i, v, b] = \min_{b'} E[i_L, v, b'] + E[i_R, v, b - b'].$$

A value v will that is not outside the range of $E[i_L,\cdot,\cdot]$ and $E[i_R,\cdot,\cdot]$ will be a correct index into these two arrays. We gain from this rounding scheme, however, when we are searching for a value to assign to node i. If i is chosen, we can search for its value in the range $\langle f,\psi_i^a\rangle \pm 2C_0\mathcal{E}/\rho$ in multiples of ρ_{t_i}. Hence, as mentioned earlier, the granularity of our search will be fine for nodes at top levels and coarse for nodes at lower levels. More formally, if i is chosen, we compute

$$E[i,v,b] = \min_{r,b'} E[i_L, v+r, b'] + E[i_R, v-r, b-b'-1]$$

where we search for the best r in multiples of ρ_{t_i}. The value $v+r$ (respectively, $v-r$) may not index correctly into $E[i_L,\cdot,\cdot]$ (respectively, $E[i_R,\cdot,\cdot]$) since $\rho_{t_i} = 2^{d/p}\rho_{t_{\text{root}}}$ where $d = t_{\text{root}} - t_i$. Hence, we need to round each value of r we wish to check to the nearest multiple of $\rho_{t_{\text{root}}}$. This extra rounding is accounted for in Lemma 17.

Letting R be the number of values each table holds and $R_{t_i} = 2C_0\mathcal{E}/\rho_{t_i} + 2$ be the number of entries we search at node i, and using an analysis similar to that of Section V-B, the running time (ignoring constant factors) becomes

$$\begin{aligned} &O\left(\sum_{i=1}^{n} RR_{t_i}\min\{2^{2t}, B^2\}\right)\\ &= O\left(R\sum_{t=1}^{\log n}\frac{n}{2^t}\frac{B2^{t/p}}{\epsilon}\min\{2^{2t},B^2\}\right)\\ &= O\left(\frac{nRB}{\epsilon}\left(\sum_{t=1}^{\log B} 2^{t/p+t} + B^2\sum_{t=\log B+1}^{\log n} 2^{t/p-t}\right)\right)\\ &= O\left(\frac{nRB}{\epsilon}B^{1+1/p}\right). \end{aligned}$$

Hence, since $R = O(n^{1/p}B/\epsilon)$ based on the granularity $\rho_{t_{\text{root}}}$, the running time for each instance of the algorithm is $O((nB)^{1+1/p}B^2/\epsilon^2)$. The space requirement is the same as that of the simpler algorithm; namely, $O(RB\log n)$.

Theorem 18: The above algorithm (with the new rounding scheme) is a $O(\epsilon^{-1}B^3n^{1/p}\log^2 n)$ space algorithm that computes a $(1+\epsilon)$ approximation to the best B-term unrestricted representation of a signal in the Haar system under the ℓ_p norm. The algorithm runs in time $O(\epsilon^{-2}(nB)^{1+1/p}B^2\log n)$.

Again, and as in Theorem 16, the extra B factor in the space requirement accounts for keeping track of the chosen coefficients, and the extra $\log n$ factor in both the space and time requirements accounts for the guessing of the error.

We choose the better of the two algorithms (or rounding schemes) whose approximation and time and space requirements are guaranteed by Theorems 16 and 18.

VI. Extensions

A. PTAS for Multidimensional Haar Systems

Our algorithm and analysis from Section V extend to multidimensional Haar wavelets when the dimension D is a given constant. For $D \geq 2$ define $2^D - 1$ mother wavelets (see also [12], [18]). For all integers $0 \leq d < 2^D$ let

$$\psi^d(x) = \theta^{d_1}(x_1)\theta^{d_2}(x_2)\cdots\theta^{d_D}(x_D)$$

where $d_1d_2\cdots d_D$ is the binary representation of d and $\theta^0 = \phi, \theta^1 = \psi$. For $d = 0$ we obtain the D-dimensional scaling function $\psi^0(x) = \phi(x_1)\phi(x_2)\cdots\phi(x_D)$. At scale 2^j and for $s = (s_1, s_2, \ldots, s_D)$ define

$$\psi_{j,s}^d(x) = 2^{-Dj/2}\psi^d\left(\frac{x_1 - 2^j s_1}{2^j}, \ldots, \frac{x_D - 2^j s_D}{2^j}\right).$$

The family $\{\psi_{j,s}^d\}_{1\leq d<2^D,(j,n)\in\mathbb{Z}^{D+1}}$ is an orthonormal basis of $L^2(\mathbb{R}^D)$ [3, Th. 7.25]. Note that in multidimensions, we define $\psi_{j,s}^{a,d} = 2^{-Dj/2}\psi_{j,s}^d, \psi_{j,s}^{b,d} = 2^{Dj/2}\psi_{j,s}^d$ and $\phi_{j,s}^{a,d} = 2^{-Dj/2}\psi_{j,s}^0$ which is analogous to Definition 1. Thus $\|\psi_{j,s}^{a,d}\|_1 = \|\phi_{j,s}^{a,d}\|_1 = 1$ since $\|\psi_{j,s}^d\|_1 = 2^{Dj}2^{-Dj/2} = 2^{Dj/2}$. Also $\|\psi_{j,s}^{b,d}\|_\infty = 1$. Each node in the coefficient tree has 2^D children and corresponds to $2^D - 1$ coefficients (assuming the input is a hypercube). The structure of the coefficient tree will result in a $O(R^{2^D-1})$ increase in running time over the one-dimensional case where $R = O(\epsilon^{-1}n^{1/p}\log n)$.

As in Section V-A, we associate an error array $E[i,b,v]$ with each node i in the tree where v is the result of the choices of i's ancestors and $b \leq B$ is the number of coefficients used by the subtree rooted at i. The size of each table is thus $O(\min\{2^{Dj}, B\}R)$ where j is the level of the tree to which i belongs. When computing an entry $E[i,b,v]$ in the table, we need to choose the best nonzero subset S of the $2^D - 1$ coefficients that belong to the node and the best assignment of values to these $|S|$ coefficients. These choices contribute a factor $O((2R)^{2^D-1})$ to the time complexity. We also have to choose the best partition of the remaining $b - |S|$ coefficients into 2^D parts adding another $O(B^{2^D})$ factor to the running time. We can avoid the latter factor by ordering the search among the node's children as in [12], [18]. Each node is broken into $2^D - 1$ *subnodes*: Suppose node i has children $c_1, \ldots, c_{2^D}$ ordered in some manner. Then subnode i_t, will have c_t as its left child and subnode i_{t-1} as its right child. Subnode i_{2^D-1} will have c_{2^D-1} and c_{2^D} as its children. Now all subnode i_t needs to do is search for the best partition of b into two parts as usual. Specifically, fix S and the values given to the coefficients in S. For each v, b' with $0 \leq b' \leq \min\{2^{Dj}, b - |S|\}$, each subnode starting from i_{2^D-1} computes the best allotment of b' coefficients to its children. This process takes $O(R(\min\{2^{Dj}, B\})^2)$ time per subnode. For ℓ_∞ the bounds are better. All the error arrays for the subnodes are discarded before considering the next choice of S and values assigned to its elements. Hence, assuming the input is of size N, and since there are $N/2^{Dj}$ nodes per level of the coefficient tree, the total running time is

$$O\left(\sum_{j=1}^{\frac{\log N}{D}} \frac{N}{2^{Dj}}(2R)^{2^D-1}2^D R(\min\{2^{Dj}, B\})^2\right) = O(NBR^{2^D})$$

where we dropped the constant factors involving D in the final expression. Finally, recall from Section V-A, that we need to make $O(\log N)$ guesses for the error $\mathcal{E}$.

B. QPTAS for General Compact Systems

We show a simple dynamic programming algorithm that finds a $(1+\epsilon)$-approximation to the wavelet synopsis construction problem under the ℓ_∞ norm. The algorithm uses $g(q,n) = n^{O(q(\log q + \log\log n))}$ time and space. Under the ℓ_p norm, the algorithm uses $n^{O(q(\log q + \frac{\log n}{p}))}$ time and space. We will describe the algorithm for the Daubechies wavelet under the ℓ_∞ norm. Recall that the Daubechies filters have $2q$ nonzero coefficients.

For a given subproblem, call an edge an *interface edge* if exactly one of its endpoints is in the subproblem. Each interface edge has a value associated with it which is eventually determined at a later stage. We will maintain that each subproblem has at most $4q\log n$ interface edges. A subproblem has a table E associated with it where for each $b \le B$ and each configuration I of values on interface edges, $E[b,I]$ stores the minimum contribution to the overall error when the subproblem uses b coefficients and the interface configuration is I. From Lemma 15, setting $\rho = \epsilon\mathcal{E}/(c_1 q\log n)$ for some suitably large constant c_1, each interface edge can have one of $V = O(\frac{q^{3/2}\log n}{\epsilon})$ values under the ℓ_∞ norm. Hence, the size of E is bounded by $BV^{4q\log n} = g(q,n)$.

The algorithm starts with an initialization phase that creates the first subproblem. This phase essentially flattens the cone-shape of the coefficient graph, and the only difference between it and later steps is that it results in one subproblem as opposed to two. We select any $2q$ consecutive leaves in the coefficient graph and their ancestors. This is at most $2q\log n$ nodes. We will guess the coefficients of the optimal solution associated with this set of nodes. Again, from Lemma 15, each coefficient can take one of $W = O(\frac{q^{3/2}\log n}{\epsilon})$ values under the ℓ_∞ norm. For each of the $(2W)^{2q\log n} = g(q,n)$ guesses, we will run the second phase of the algorithm.

In the second phase, given a subproblem A, we first select the $2q$ 'middle' leaves and their ancestors. Call this strip of nodes S. Note that $|S| \le 2q\log n$. The nodes in S break A into two smaller subproblems L and R (see Fig. 7). Suppose we have E_L and E_R, the two error arrays associated with L and R respectively. We compute each entry $E_A[b,I]$ as follows. First, we guess the b' nonzero coefficients of the optimal solution associated with the nodes in S and their values. Combined with the configuration I, these values define a configuration I_L (respectively, I_R) for the interface edges of L (respectively, R) in the obvious way. Furthermore, they result in an error e associated with the leaf nodes in S. Hence

$$E[b,I] = e + \min_{b''} \max\{E_L[b'', I_L], E_R[b-b'-b'', I_R]\}.$$

Therefore, computing each entry in E takes at most $B(2W)^{2q\log n} = g(q,n)$ time. The running time of the algorithm follows.

Theorem 19: We can compute a $(1+\epsilon)$ approximation to the best B-term unrestricted representation of a compact system under the ℓ_∞ norm in time $n^{O(q(\log q + \log\log n))}$.

The result also extends to ℓ_p norms, but remains a quasipolynomial time algorithm. The main point of the above theorem is that the representation problem is not MAX-SNP-HARD.

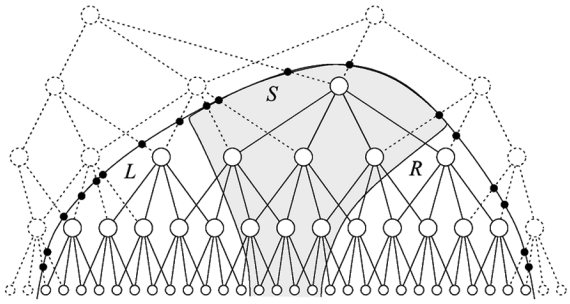


Fig. 7. An example subproblem. The shaded nodes belong to the strip S. The edges crossing the 'frontier' are interface edges.

C. Workloads

The algorithm and analysis from Section V also extend to weighted cases/workloads under the same assumptions as in [16]. Namely, given f and $\{w_i\}$ where $\sum_{i=1}^n w_i = 1$ and $0 < w_i \le 1$, we wish to find a solution $\{z_i\}$ with at most B nonzero coefficients that minimizes

$$\|f - \sum_i z_i\psi_i\|_{p,\mathbf{w}} = \left(\sum_j w_i^p |f(j) - \sum_i z_i\psi_i(j)|^p\right)^{1/p}.$$

Letting $w = \min_i w_i$ and $W = \max_i w_i$, we will show how our approximation algorithm extends to this case with a factor $\frac{W}{w}$ increase in its space requirement and a factor $(\frac{W}{w})^2$ increase in running time.

The following three lemmas are analogs of Lemmas 11, 14, and 12, respectively. The first two are straightforward, but note the factor W in the additive approximation.

Lemma 20: Let $\{y_i^*\}$ be the optimal solution using the basis set $\{\psi_i^b\}$ for the reconstruction, i.e., $\hat{f} = \sum_i y_i^*\psi_i^b$ and $\|f - \hat{f}\|_{p,\mathbf{w}} = \mathcal{E}$. Let $\{y_i^\rho\}$ be the set where each y_i^* is rounded to the nearest multiple of ρ. If $f^\rho = \sum_i y_i^\rho\psi_i^b$ then $\|f - f^\rho\|_{p,\mathbf{w}} \le \mathcal{E} + O(qWn^{1/p}\rho\log n)$.

Lemma 21: $-C_1\sqrt{q}\frac{\mathcal{E}}{w} \le \langle f,\phi_{j,s}\rangle - \langle\hat{f},\phi_{j,s}\rangle \le C_1\sqrt{q}\frac{\mathcal{E}}{w}$ for some constant C_1.

Lemma 22: $-C_0\sqrt{q}\frac{\mathcal{E}}{w} \le \langle f,\psi_{j,s}^a\rangle - y_i^* \le C_0\sqrt{q}\frac{\mathcal{E}}{w}$ for some constant C_0.

Proof: For all j we have $w_j|f(j) - \sum_i y_i^*\psi_i^b(j)| \le \mathcal{E}$. Multiplying by $|\psi_k^a(j)|$ and summing over all j we get

$$\begin{aligned}
&\sum_j w_j|f(j)\psi_k^a(j) - \sum_i y_i^*\psi_i^b(j)\psi_k^a(j)| \le \|\psi_k\|_1\mathcal{E}\\
\Rightarrow& w|\sum_j f(j)\psi_k^a(j) - \sum_i y_i^*\sum_j \psi_i(j)\psi_k(j)| \le \|\psi_k^a\|_1\mathcal{E}\\
\Rightarrow& w|\langle f,\psi_k^a\rangle - y_k^*| \le \sqrt{2q}\mathcal{E}
\end{aligned}$$

completing the proof. □

Hence, setting $\rho = \epsilon\mathcal{E}/(cqWn^{1/p}\log n)$ for some suitably large constant c, we get the desired approximation with R from the analysis above equal to $O(\mathcal{E}/(w\rho)) = O(\frac{W}{w}q\epsilon^{-1}n^{1/p}\log n)$.

D. Quality Versus Time

A natural question arises, if we were interested in the restricted synopses only, can we develop streaming algorithms for them? The answer reveals a rich tradeoff between synopsis quality and running time.

Observe that if at each node we only consider either storing the coefficient or 0, then we can limit the search significantly. Instead of searching over all $v + r$ to the left and $v - r$ to the right in the dynamic program (which we repeat below)

$$\min \begin{cases} \min_{r,b'} E[i_L, v+r, b'] + E[i_R, v-r, b-b'-1] \\ \min_{b'} E[i_L, v, b'] + E[i_R, v, b-b'] \end{cases}$$

we only need to search for $r = \langle f, \psi_i^a \rangle$—observe that a streaming algorithm can compute $\langle f, \psi_i^a \rangle$ (See [24]). However we have to "round" $\langle f, \psi_i^a \rangle$ to a multiple of ρ since we are storing the table corresponding to the multiples of ρ between $\langle f, \phi_i^a \rangle - C_1\sqrt{q}\mathcal{E}$ and $\langle f, \phi_i^a \rangle + C_1\sqrt{q}\mathcal{E}$. We consider *the better of rounding up or rounding down $\langle f, \psi_i^a \rangle$ to the nearest multiple of ρ*. The running time improves by a factor of R in this case since in order to compute each entry we are now considering only two values of R_i (round up/down) instead of the entire set. The overall running time is $O(RnB)$ in the general case and $O(Rn \log^2 B)$ for the ℓ_∞ variants. The space bound and the approximation guarantees remain unchanged. However the guarantee is now against the synopsis which is restricted to storing wavelet coefficients.

The above discussion sets the ground for investigating a variety of *Hybrid algorithms* where we choose different search strategies for each coefficient. We introduced this idea in [43] but in the context of a weaker approximation strategy. One strategy we explore in Section VIII is to allow the root node to range over the set R_1 while considering the better of rounding up or rounding down $\langle f, \psi_i^a \rangle$ to the nearest multiple of ρ for all other coefficients ($i > 1$). We show that this simple modification improves on the quality of the restricted synopsis and on the running time of the unrestricted algorithm.

VII. Best Basis Selection From a Dictionary

In this section, we show how our algorithms can be extended to find representations in certain types of tree-structured dictionaries. Specifically, the dictionaries we consider are full binary tree-structured dictionaries composed of compactly supported wavelets. Given $f \in \mathbb{R}^n, B \in \mathbb{Z}$ and such a dictionary $\mathcal{D}$, we now wish to find the best B-term representation of f in a basis from $\mathcal{D}$. Notice that we seek both the best basis in $\mathcal{D}$ for representing f using B terms and the best B-term representation of f in this basis. The error of the representation is its ℓ_p distance from f. We show in Theorem 25 how our algorithms from the previous sections can be used to find provable approximate answers to this bicriteria optimization problem.

We start with the description of our tree-structured dictionaries. Similar to Coiffman and Wickerhauser [21], our dictionaries will be composed of $O(n \log n)$ vectors, and will contain $2^{O(\frac{n}{2})}$ bases: equal to the number of cuts in a complete binary tree.

Let $a_{(j,p)} = 2^j p$ and let $g_{(j,p)}[t] = \mathbf{1}_{[a_{(j,p)}, a_{(j,p+1)}-1]}[t]$ be the discrete dyadic window that is 1 in $[a_{(j,p)}, a_{(j,p+1)} - 1]$ and zero elsewhere. Each node in $\mathcal{D}$ is labeled by $(j,p), 0 \le j \le \log n, 0 \le p \le n2^{-j} - 1$, where j is the height of the node in the tree (the root is at height $\log n$), and p is the number of nodes to its left that are at the same height in a complete binary tree. With each node (j,p) we associate the subspace $\mathcal{W}_{(j,p)}$ of $\mathbb{R}^n$ that exactly includes all functions $f \in \mathbb{R}^n$ whose support lies in $g_{(j,p)}$. Clearly, $\mathcal{W}_{(\log n, 0)} = \mathbb{R}^n$.

Now suppose $\{e_{k,l}\}_{0 \le k < l}, l > 0$ is an orthonormal basis for $\mathbb{R}^l$. Then

$$\mathcal{B}_{(j,p)} = \{\psi_{k,(j,p)}[t] = g_{(j,p)}[t] e_{k,2^j}[t - a_{(j,p)}]\}_{0 \le k < 2^j}$$

is an orthonormal basis for $\mathcal{W}_{(j,p)}$.

Proposition 23: For any internal node (j,p) in the dictionary $\mathcal{D}, \mathcal{W}_{(j-1,2p)}$ and $\mathcal{W}_{(j-1,2p+1)}$ are orthogonal, and

$$\mathcal{W}_{(j,p)} = \mathcal{W}_{(j-1,2p)} \oplus \mathcal{W}_{(j-1,2p+1)}.$$

We can thus construct an orthonormal basis of $\mathcal{W}_{(j,p)}$ via a union of orthonormal bases of $\mathcal{W}_{(j-1,2p)}$ and $\mathcal{W}_{(j-1,2p+1)}$.

Corollary 24: Let $\{(j_i, p_i)\}$ be the set of nodes corresponding to a cut in the dictionary tree. We have

$$\bigoplus_i \mathcal{W}_{(j_i,p_i)} = \mathbb{R}^n.$$

Hence, there are $O(2^{n/2})$ bases in our dictionary.

The main result of this section follows. We prove it under the ℓ_∞ error measure. The argument is extended to general ℓ_p error measures in a straightforward manner.

Theorem 25: If A is an (streaming) algorithm that achieves a C-approximation for the B-term representation problem under ℓ_∞ (for any wavelet included in the dictionary $\mathcal{D}$), then A is a (streaming) C-approximation for the bicriteria representation problem.

Proof: Let $E_{(j,p)}[b]$ be the minimum contribution to the overall error (as computed by A) from representing the block $g_{(j,p)}[t]f[t]$ using b vectors from a basis of $\mathcal{W}_{(j,p)}$. Call the basis that achieves this error the *best basis for* $\mathcal{W}_{(j,p)}$ and denote it by $\mathcal{O}_{(j,p)}[b]$. By Proposition 23 there are $O(2^{2^{j-1}})$ possible bases for the space $\mathcal{W}_{(j,p)}$ in $\mathcal{D}$. Now if (j,p) is a leaf node, then $E_{(j,p)}[b] = A(g_{(j,p)} \odot f, \mathcal{B}_{(j,p)}, b)$, which is the error resulting from representing the block $g_{(j,p)}[t]f[t]$ using b vectors from the basis $\mathcal{B}_{(j,p)}$. Otherwise, if (j,p) is an internal node, $E_{(j,p)}[b]$ equals

$$\min \begin{cases} \min_{0 \le b' \le b} \max\{E_{(j-1,2p)}[b'], E_{(j-1,2p+1)}[b-b']\} \\ A(g_{(j,p)} \odot f, \mathcal{B}_{(j,p)}, b) \end{cases}$$

and

$$\mathcal{O}_{(j,p)}[b] = \begin{cases} \mathcal{B}_{(j,p)} & \text{if } E_{(j,p)}[b] = A(g_{(j,p)} \odot f, \mathcal{B}_{(j,p)}, b) \\ \mathcal{O}_{(j-1,2p)}[b_{(j,p)}] \cup \mathcal{O}_{(j-1,2p-1)}[b - b_{(j,p)}] & \text{else} \end{cases}$$

where $b_{(j,p)}$ is the argument that minimizes the top expression in $E_{(j,p)}[b]$.

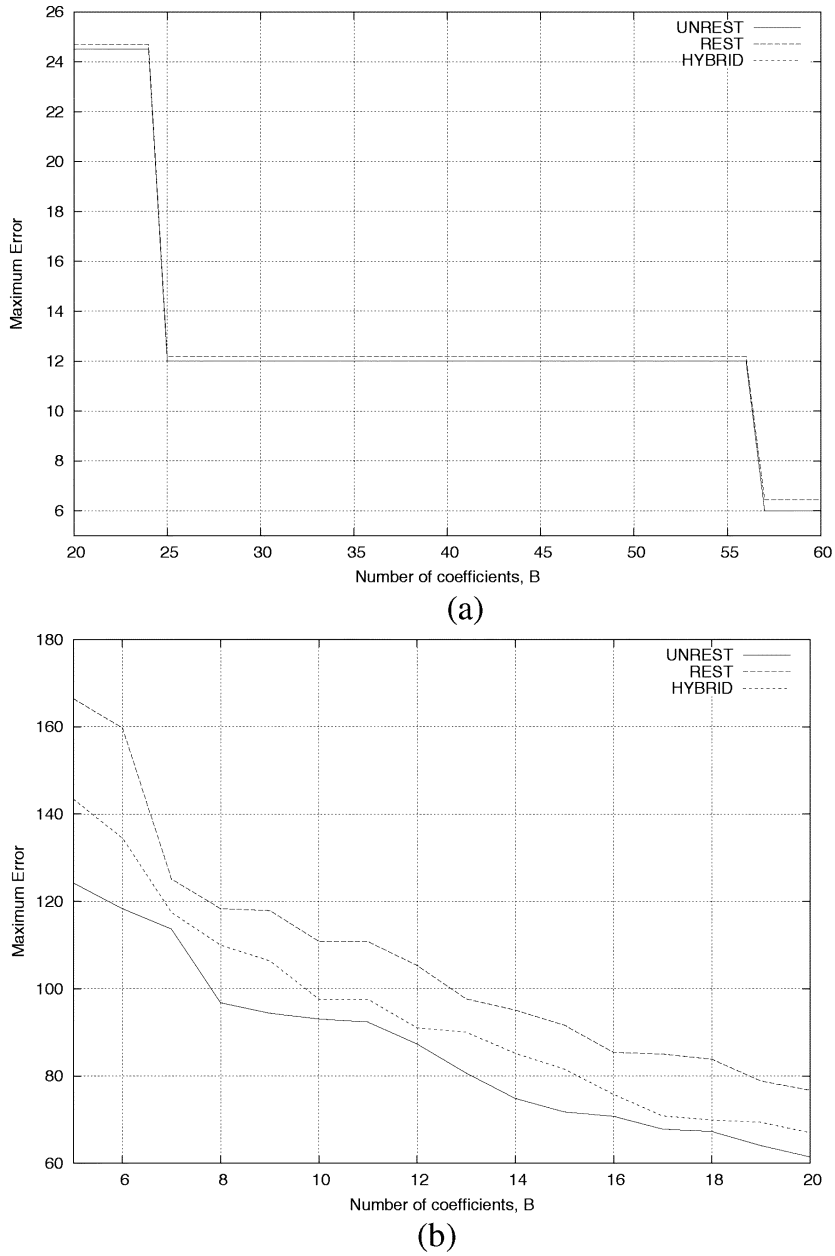


Fig. 8. The ℓ_∞ error of the three algorithms, UNREST, REST, and HYBRID for the two data sets. (a) Error for the Saw data set $(n = 2048)$ (b) Error for the Dow data set $(n = 16\,384)$.

Suppose OPT chooses the cut $\{(j_o, p_o)\}$ with the corresponding partition $\{b_o\}$ of B and we choose the cut $\{(j_i, p_i)\}$ with partition $\{b_i\}$. By the dynamic program above, we have

$$\begin{aligned}
&\max_i A(g_{(j_i,p_i)} \odot f, \mathcal{B}_{(j_i,p_i)}, b_i) \\
&\quad = \max_i E_{(j_i,p_i)}[b_i] && \text{(6a)} \\
&\quad \le \max_o E_{(j_o,p_o)}[b_o] && \text{(6b)} \\
&\quad \le \max_o A(g_{(j_o,p_o)} \odot f, \mathcal{B}_{(j_o,p_o)}, b_o) && \text{(6c)} \\
&\quad \le C \max_o \mathrm{OPT}(g_{(j_o,p_o)} \odot f, \mathcal{B}_{(j_o,p_o)}, b_o) && \text{(6d)} \\
&\quad = C\mathrm{OPT} && \text{(6e)}
\end{aligned}$$

where (6b) follows from the fact that our dynamic program chooses the best cut and corresponding partition of B among all possible cuts and partitions based on the errors computed by algorithm A; (6c) follows from the definition of our dynamic programming table entries $E_{(j,p)}[b]$; (6c) follows from the assumption that A is a C-approximation algorithm; and (6e) follows from the optimal substructure property of our problem. □

VIII. Comparing Restricted and Unrestricted Optimizations

We consider two issues in this section, namely, 1) the quality of the unrestricted version vis-a-vis the restricted optimum solution and 2) the running times of the algorithms. We will restrict our experiments to the ℓ_∞ norm.

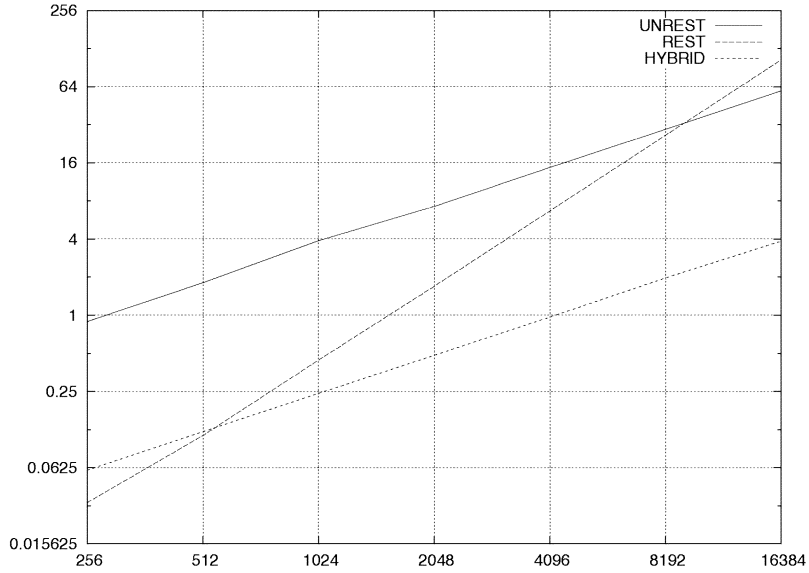


Fig. 9. Running times for prefixes of the Dow data set.

A. The Algorithms

All experiments reported in this section were performed on a 2-CPU Pentium-III 1.4 GHz with 2GB of main memory, running Linux. All algorithms were implemented using version 3.3.4 of the gcc compiler.

We show the performance figures of the following schemes:

REST This characterizes the algorithms for the *restricted* version of the problem. This is the $O(n^2)$ time $O(n)$ space algorithm in [17] (see also [14], [15], and [18]).

UNREST This is the streaming algorithm for the *full general version* described in Algorithm *HaarPTAS* based on the discussion in Section V.[8]

HYBRID This is the streaming hybrid algorithm proposed in Section VI-D.

Note that the UNREST and HYBRID algorithms are not the additive approximation algorithms in [43] (although we kept the same names).

B. The Data Sets

We chose a synthetic data set to showcase the point made in the introduction about the suboptimality of the restricted versions. Otherwise we use a publicly available real life data set for our experiment.

- **Saw:** This is a periodic data set with a line repeated eight times, with 2048 values total.
- **DJIA data set:** We used the Dow-Jones Industrial Average (DJIA) data set available at StatLib[9] that contains Dow-Jones Industrial Average (DJIA) closing values from 1900 to 1993. There were a few negative values (e.g., -9), which we removed. We focused on prefixes of the data set of sizes up to $16\,384$.

[8]The implementation is available at http://www.cis.upenn.edu/~boulos/publications.

[9]See http://lib.stat.cmu.edu/datasets/djdc0093.

C. Quality of Synopsis

The ℓ_∞ errors as a function of B are shown in Fig. 8(a) and (b). The ϵ in the approximation algorithms UNREST and HYBRID was set to 1. All the algorithms gave very similar synopses for the Saw data and had almost the same errors. In case of the Dow data we show the range $B = 5$ onward since the maximum value is ~ 500 and the large errors for $B < 5$ (for all algorithms) bias the scale making the differences in the more interesting ranges not visible. The algorithm REST has more than 20% worse error compared to UNREST or requires over 35% more coefficients to achieve the same error (for most error values). The HYBRID algorithm performs consistently in the middle.

D. Running Times

Fig. 9 shows the running times of the algorithms as the prefix size n is varied for the Dow data. As mentioned above ϵ was set to 1. The grid in the log-log plot helps us clearly identify the quadratic nature of REST. The algorithms UNREST and HYBRID behave linearly as is expected from streaming algorithms. Given its speed and quality, the HYBRID algorithm seems to be the best choice from a practical perspective.

REFERENCES

[1] S. Guha and B. Harb, "Approximation algorithms for wavelet transform coding of data streams," in *SODA'06: Proc. Seventeenth Annu. ACM-SIAM Symp. Discr. Algor.*, Miami, FL, 2006, pp. 698–707.

[2] E. Schmidt, "Zur theorie der linearen und nichtlinearen integralgleichungen - i," *Math. Annalen*, vol. 63, pp. 433–476, 1907.

[3] S. Mallat, *A Wavelet Tour of Signal Processing*. New York: Academic, 1999.

[4] I. Daubechies, *Ten Lectures on Wavelets*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 1992.

[5] R. DeVore, B. Jawerth, and V. A. Popov, "Compression of wavelet decompositions," *Amer. J. Math.*, vol. 114, pp. 737–785, 1992.

[6] C. E. Jacobs, A. Finkelstein, and D. H. Salesin, "Fast multiresolution image querying," *Comput. Graph.*, vol. 29, pp. 277–286, 1995.

[7] A. Cohen, I. Daubechies, O. Guleryuz, and M. Orchard, "On the importance of combining wavelet-based nonlinear approximation in coding strategies," *IEEE Trans. Inf. Theory* vol. 48, no. 7, pp. 1895–1921, 2002 [Online]. Available: citeseer.ist.psu.edu/article/cohen97importance.html, [Online]. Available

[8] R. DeVore, "Nonlinear approximation," *Acta Numerica*, pp. 1–99, 1998.

[9] V. N. Temlyakov, "Nonlinear methods of approximation," *Found. Comput. Math.*, vol. 3, pp. 33–107, 2003.

[10] Y. Matias, J. S. Vitter, and M. Wang, "Wavelet-based histograms for selectivity estimation," in *Proc. ACM SIGMOD*, Seattle, WA, 1998, pp. 448–459.

[11] O. Chapelle, P. Haffner, and V. Vapnik, "Support vector machines for histogram-based image classification," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 1055–1064, 1999.

[12] M. N. Garofalakis and P. B. Gibbons, "Probabilistic wavelet synopses," *ACM TODS*, vol. 29, pp. 43–90, 2004.

[13] M. Garofalakis and A. Kumar, "Wavelet synopses for general error metrics," *ACM Trans. Database Syst.*, vol. 30, no. 4, pp. 888–928, 2005.

[14] S. Muthukrishnan, "Nonuniform sparse approximation using Haar wavelet basis," DIMACS TR 2004-42, 2004.

[15] Y. Matias and D. Urieli, Personal Communication 2004.

[16] Y. Matias and D. Urieli, "Optimal workload-based weighted wavelet synopses," in *Proc. ICDT*, 2005, pp. 368–382.

[17] S. Guha, "Space efficiency in synopsis construction problems," in *VLDB Conf., Proc. 31st Int. Conf. Very Large Data Bases*, Trondheim, Norway, 2005, pp. 409–420.

[18] M. Garofalakis and A. Kumar, "Deterministic wavelet thresholding for maximum error metric," in *PODS'04, Proc. 23rd ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems*, Paris, France, 2004, pp. 166–176.

[19] V. N. Temlyakov, "The best m-term representation and greedy algorithms," *Adv. Comput. Math.*, vol. 8, pp. 249–265, 1998.

[20] R. A. DeVore, S. V. Konyagin, and V. N. Temlyakov, "Hyperbolic wavelet approximation," *J. Constr. Approx.*, vol. 14, pp. 1–26, 1998.

[21] R. R. Coifman and M. V. Wickerhauser, "Entropy-based algorithms for best basis selection," *IEEE Trans. Inf. Theory*, vol. 38, no. 2, pp. 713–718, 1992.

[22] G. Davis, S. Mallat, and M. Avellaneda, "Adaptive greedy approximation," *J. Constr. Approx.*, vol. 13, pp. 57–98, 1997.

[23] A. C. Gilbert, S. Muthukrishnan, and M. Strauss, "Approximation of functions over redundant dictionaries using coherence," in *Proc. SODA*, 2003, pp. 243–252.

[24] A. C. Gilbert, Y. Kotidis, S. Muthukrishnan, and M. Strauss, "Optimal and approximate computation of summary statistics for range aggregates," in *PODS'01, Proc. 20th ACM SIGMOD-SIGACT-SIGART Symp. Principles of Database Systems*, Santa Barbara, CA, 2001, pp. 227–236.

[25] A. C. Gilbert, S. Guha, P. Indyk, Y. Kotidis, S. Muthukrishnan, and M. Strauss, "Fast, small-space algorithms for approximate histogram maintenance," in *Proc. ACM STOC*, Montreal, QC, Canada, 2002, pp. 389–398.

[26] U. Feige, "A threshold of $\ln n$ for approximating set cover," *J. ACM*, vol. 45, no. 4, pp. 634–652, 1998.

[27] S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan, "Clustering data streams," in *Proc. Symp. Found. Comput. Sci. (FOCS)*, 2000, pp. 359–366.

[28] S. Guha, P. Indyk, S. Muthukrishnan, and M. Strauss, "Histogramming data streams with fast per-item processing," in *Proc. ICALP*, 2002.

[29] E. Keogh, K. Chakrabati, S. Mehrotra, and M. Pazzani, "Locally adaptive dimensionality reduction for indexing large time series databases," in *Proc. ACM SIGMOD*, Santa Barbara, CA, Mar. 2001, pp. 188–228.

[30] K. Chakrabarti, E. J. Keogh, S. Mehrotra, and M. J. Pazzani, "Locally adaptive dimensionality reduction for indexing large time series databases," *Proc. ACM TODS*, vol. 27, no. 2, pp. 188–228, 2002.

[31] S. Guha, N. Koudas, and K. Shim, "Data streams and histograms," in *STOC'01, Proc. 33rd Annu. ACM Symp. Very Large Data Bases*, Hersonissos, Greece, 2001, pp. 471–475.

[32] Y. E. Ioannidis, "The history of histograms (abridged)," in *VLDB'2003, Proc. 29th Int. Conf. Very Large Data Bases*, Berlin, Germany, 2003, pp. 19–30.

[33] B. Harb, "Algorithms for Linear and Nonlinear Approximation of Large Data," Ph.D. dissertation, University of Pennsylvania, Phildelphia, PA, 2007.

[34] A. Haar, "Zur theorie der orthogonalen funktionen-systeme," *Math. Ann.*, vol. 69, pp. 331–371, 1910.

[35] S. Mallat, "Multiresolution approximations and wavelet orthonormal bases of $l^2(\mathbb{R})$," *Trans. Amer. Math. Soc.*, vol. 315, pp. 69–87, Sept. 1989.

[36] Y. Meyer, *Wavelets and Operators*, ser. Advanced mathematics. Cambridge, U.K.: Cambridge University Press, 1992.

[37] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Comm. Pure Appl. Math.*, vol. 41, pp. 909–996, 1988.

[38] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: An overview," *IEEE Trans. Consumer Electron.*, vol. 46, no. 4, pp. 1103–1127, 2000.

[39] S. Guha, C. Kim, and K. Shim, "XWAVE: Optimal and approximate extended wavelets for streaming data," in *Proc. VLDB Conf.*, 2004.

[40] S. G. Sanchez, N. G. Prelcic, and S. J. G. Galan, Uvi_Wave Version 3.0–Wavelet Toolbox for Use With MATLAB [Online]. Available: citeseer.ist.psu.edu/672431.html

[41] H. McCullough, *Kokin Wakashu: The First Imperial Anthology of Japanese Poetry* Transl.:Japanese. Palo Alto, CA: Stanford University Press, 1984.

[42] G. Golub and C. V. Loan, *Matrix Computations*. : Johns Hopkins University Press, 1989.

[43] S. Guha and B. Harb, "Wavelet synopsis for data streams: minimizing noneuclidean error," in *Proc. KDD'05: Proc. Eleventh ACM SIGKDD Int. Conf. Knowledge Discovery in Data Mining*, New York, NY, 2005, pp. 88–97.