Wavelet Synopsis for Data Streams: Minimizing Non-Euclidean Error

Sudipto Guha[*]
Department of Computer Science
University of Pennsylvania
Philadelphia, PA 19104.
sudipto@cis.upenn.edu

Boulos Harb[†]
Department of Computer Science
University of Pennsylvania
Philadelphia, PA 19104.
boulos@cis.upenn.edu

ABSTRACT

We consider the wavelet synopsis construction problem for data streams where given n numbers we wish to estimate the data by constructing a synopsis, whose size, say B is much smaller than n. The B numbers are chosen to minimize a suitable error between the original data and the estimate derived from the synopsis.

Several good one-pass wavelet construction streaming algorithms minimizing the ℓ_2 error exist. For other error measures, the problem is less understood. We provide the first one-pass small space streaming algorithms with provable error guarantees (additive approximation) for minimizing a variety of non-Euclidean error measures including all weighted ℓ_p (including ℓ_∞) and relative error ℓ_p metrics.

In several previous works solutions (for weighted ℓ_2, ℓ_∞ and maximum relative error) where the B synopsis coefficients are restricted to be wavelet coefficients of the data were proposed. This restriction yields suboptimal solutions on even fairly simple examples. Other lines of research, such as probabilistic synopsis, imposed restrictions on how the synopsis was arrived at. To the best of our knowledge this paper is the first paper to address the general problem, without any restriction on how the synopsis is arrived at, as well as provide the first streaming algorithms with guaranteed performance for these classes of error measures.

Categories and Subject Descriptors: F.2 [Analysis of Algorithms and Complexity] : Miscellaneous; G.2 [Discrete Mathematics] : Miscellaneous; H.3 [Information Storage and Retrieval] : Miscellaneous

General Terms: Algorithms, Theory

Keywords: Wavelet Synopses, Streaming Algorithms

1. INTRODUCTION

Wavelets are localized, orthogonal transforms that are extremely versatile for representing discrete signals [3, 21]. They allow a multi-resolution view of the data and easily extend to more than one dimension. Among these, the Haar wavelets have been used extensively in synopsis construction with a variety of uses in image analysis, signal processing, and databases to name a few. The primary attraction of the Haar Wavelets is the existence of linear time forward and inverse transforms. The non-normalized Haar basis for $n = 4$ is:

$$\psi_1 = \{1, 1, 1, 1\} \qquad \psi_2 = \{1, 1, -1, -1\}$$
$$\psi_3 = \{1, -1, 0, 0\} \qquad \psi_4 = \{0, 0, 1, -1\}$$

The vectors generalize to larger powers of 2 quite easily. The intervals over which the basis vectors are defined are powers of 2. The interval corresponding to the basis vector ψ_i is denoted by $\textsc{Supp}(\psi_i)$, e.g. $\textsc{Supp}(\psi_1) = n$ and $\textsc{Supp}(\psi_3) = n/2$ always. Moreover every value of the inverse transform $\mathcal{W}^{-1}(Z) = \sum_i \psi_i Z_i$ in the Haar setting is the sum of only logarithmically (in the length of the signal) many values in the transformed representation Z. This allows fast "on demand" computation for a broad spectrum of analysis tasks. Furthermore, there is a significant intuitive meaning to the Haar wavelet coefficients. The (non-normalized) coefficients denote half of the difference between the averages of the left and right halves of the entire interval (the support).

Most applications of Wavelets consider representing the input in terms of the high level coefficients and broader characteristics of the data, typically referred to as a synopsis or signature. These synopses or signatures are used subsequently in learning, classification, event detection, among many other applications. The synopsis is typically constructed to minimize some desired error criterion. One of the most common error criteria is the sum-of-squares criterion which is also the square of the ℓ_2 distance between the original signal and its representation. However with the emerging mining applications such as time series analysis other error measures (e.g. ℓ_∞, weighted ℓ_2 etc.) have been looked at recently. It would be impossible to conduct a thorough review, however [13, 19, 1, 18, 7, 6, 4, 11] and pointers therein serve as excellent starting points.

In this paper we consider the following problem:

Problem 1. *Given a set of n numbers $X = x_1, \ldots, x_n$, find a synopsis vector Z with at most B non-zero entries,*

[*] Supported in part by an Alfred P. Sloan Research Fellowship and by an NSF Award CCF-0430376.

[†] Sponsored by NIH Training Grant T32HG0046

KDD'05, August 21–24, 2005, Chicago, Illinois, USA.
Copyright 2005 ACM 1-59593-135-X/05/0008 ...$5.00.

such that the inverse wavelet transform of Z *(denoted by* $\mathcal{W}^{-1}(Z)$*) gives a good estimate of the data, i.e., minimizes* $\|X - \mathcal{W}^{-1}(Z)\|_p$ *for some integer* p *(*$p = \infty$ *corresponds to the maximum error). We will assume* n *is a power of* 2*.*

The problem also generalizes to weighted-ℓ_p *error metrics where given weights* $\pi_i \geq 0$ *we seek to minimize*

$$\|X - \mathcal{W}^{-1}(Z)\|_{\pi,p} = \left(\sum_i \pi_i^p \left|(x_i - \mathcal{W}^{-1}(Z)_i)\right|^p \right)^{\frac{1}{p}}$$

For the standard ℓ_k *norm all* π_i *are set to* 1*. The Relative Error metrics have* $\pi_i = \frac{1}{\max\{|x_i|,c\}}$ *for some sanity constant* $c > 0$ *which avoids division by* 0*. The relative error metrics will be denoted as* $\| \cdot \|_p^{rel}$*.*

In absence of any qualifier such as 'relative' or 'weighted', the term error will imply error in the ℓ_p *norm. For the weighted-*ℓ_p *norm we can multiply all the* π_i*'s by a constant and leave the problem unchanged. Therefore for weighted* ℓ_p *we will assume* $\pi_{max} = \max_i \pi_i = 1$*.*

For the Euclidean error, i.e., minimizing unweighted ℓ_2 error $\|X - \mathcal{W}^{-1}(Z)\|_2$, Observe that the set of vectors vectors $\left(1/\sqrt{\textsc{Supp}(\psi_i)}\right) \psi_i$ form an orthonormal basis. In any orthonormal basis the Euclidean length or the ℓ_2 norm of any vector (including $X - \mathcal{W}^{-1}(Z)$) is preserved (Parseval's Theorem). Thus the problem of minimizing $\|X - \mathcal{W}^{-1}(Z)\|_2$ is equivalent to minimizing $\sum_i \textsc{Supp}(\psi_i)(\mathcal{W}(X)_i - Z_i)^2$ and the best choice of Z is to store the largest B (ignoring signs) normalized, i.e. multiplied by $\sqrt{\textsc{Supp}(\psi_i)}$, coefficients. Several algorithms have been proposed for this in the streaming context [18, 7, 8, 10]. For the streaming model considered in this paper the optimum synopsis under unweighted ℓ_2 error can be found in $O(n)$ time and $O(B + \log n)$ space.

The simplicity of the unweighted ℓ_2 solution breaks down in case of non-Euclidean error measures. In an early paper Matias, Vitter and Wang [19], demonstrated a number of different applications for wavelet synopsis for non-Euclidean error measures and proposed greedy algorithms. In fact, several researchers have shown greedy heuristics perform quite well, but no theoretical analysis about the quality of the synopsis exists. The problem is quite non-trivial, because the Wavelet basis vectors overlap and two coefficients can cancel out each other leaving a significantly (exponentially) smaller contribution. In fact, for ℓ_k ($k > 2$, even weighted ℓ_2) there is no known guarantee that the solution will be over rationals since the optimization minimizes an algebraic equation of degree greater than 1. This is the biggest stumbling block in the synopsis construction, and has likely been one of the reasons for considering the restrictions on how the synopses are arrived at. We discuss some of the previous works next.

1.0.0.1 Related Work.

Garofalakis and Gibbons [6] proposed a strategy that improves upon storing the largest coefficients for non-Euclidean errors. They consider probabilistic synopsis where the i'th coefficient takes the value λ_i with probability $\mathcal{W}(X)_i/\lambda_i$ or is set to 0. They show the estimation using probabilistic synopses is unbiased and provide algorithms for finding the best probabilistic synopsis under different measures. However, we note that,

- The algorithm is inherently offline.
- Space bound is preserved only in expectation and the variance in the space usage (computed analytically) appears to be large.
- There is no immediate connection between the best probabilistic synopsis and the best synopsis. Note that the best synopsis does not have restrictions on how the the synopsis is arrived at. For example consider $X = \{1, 4, 5, 6\}$, whose transform is $\mathcal{W}(X) = \{4, -1.5, -1.5, -0.5\}$. The best solution for ℓ_∞ error and $B = 1$ is $Z = \{3.5, 0, 0, 0\}$. The probabilistic synopses will not even consider this solution since it will restrict $\lambda_1 \geq 4$. Note that the example can generalize to any B, simply repeat with alternate signs, i.e., $\{1, 4, 5, 6, -1, -4, -5, -6, 1, 4, 5, 6, \ldots\}$. The gap between the errors can be made as large in magnitude by considering $\{a, 4a, 5a, 6a\}$ for some large constant a. [6] also consider the maximum relative error ℓ_∞^{rel} which minimizes $\max_i \left| \frac{x_i - \mathcal{W}^{-1}(Z)_i}{\max\{x_i, c\}} \right|$. The optimum solution for X under this error measure and $B = 1$ is $Z = \{12/7, 0, 0, 0\}$; which is again ruled out by the probabilistic synopses.

Garofalakis and Kumar [5] avoid the problem with the space bound and give a deterministic $O(n^2 B \log B)$ time and $O(n^2 B)$ space algorithm for maximum error metrics for the restricted case where the i^{th} entry of Z, Z_i, is restricted to be 0 or $\mathcal{W}(X)_i$, the i^{th} Wavelet coefficient of the input. Muthukrishnan [20] extends the algorithm of [5] to handle weighted ℓ_2 error measures. Matias and Urieli [16] as well as [20] improve the running time of the algorithm in [5]. Guha [9] shows that all weighted ℓ_k error measures, in the restricted version can be solved in $O(n^2 \log B)$ time and $O(n)$ space (constants independent of B) using space efficient dynamic programming techniques. All the algorithms for this restricted version in [5, 20, 16, 9] share the following properties:

- The algorithms are inherently offline.
- The algorithms choose coefficients of the data for the optimization, i.e., solve the restricted problem. The earlier example of $X = \{1, 4, 5, 6\}$ is problematic for this restriction on Z_i, and applies to all of them since the best solution which retains $B = 1$ coefficient of the data is $\{4, 0, 0, 0\}$ for both the ℓ_∞ or ℓ_∞^{rel} errors. As we have already seen, the optimum solutions for these errors are $\{3.5, 0, 0, 0\}$ and $\{12/7, 0, 0, 0\}$ respectively. The same example $X = \{1, 4, 5, 6\}$ carries over to the weighted-ℓ_2 case; consider $\pi = \{1, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}\}$. The best solution is $\{3.4, 0, 0, 0\}$ (follows from the 4 quadratic equations formed) instead of $\{4, 0, 0, 0\}$ which arises from retaining the best single coefficient of the input. Again, the examples generalize to any B using the alternation and to any gap using multiplication.

Matias and Urieli [17] consider the weighted-ℓ_2 error and provide a near linear time optimal algorithm; but for a different wavelet basis that depends on the weights. Their algorithm also appears to be an offline algorithm.

1.0.0.2 Our contributions.

The example $X = \{1, 4, 5, 6\}$ underscores that the restriction of a synopsis coefficient to be a coefficient of the data results in a suboptimal strategy. The problem disappears in the unrestricted case and matches the intuitive notion of a synopsis. We will also show that the removal of the restriction gives us a streaming computation. For the purpose of the paper a data stream computation is a space bounded algorithm, where the space is sub-linear in the input. Any input items are accessed sequentially and any item not explicitly stored cannot be accessed again in the same pass. We discuss streaming models and its relevance to our problem at the appropriate place.

1. We propose the first one-pass streaming Wavelet synopsis construction algorithms for (several) non-Euclidean error measures and we provide a solution with an additive error. For most error measures considered the additive error is ϵM where the the data is integers in a range $[-M, M]$. For relative error, the additive error is ϵ, but the running time depends on the ratio of the largest to smallest number in the input.

2. We show that the general version of the problem produces up to 30% better quality synopsis on a few real and synthetic data sets compared to the restricted version considered in [5, 20, 16, 9] where the coefficients stored in the synopsis are restricted to be coefficients of the data.

3. We also propose the first streaming approximation algorithm for the version of the problem considered in [5, 20, 16, 9]. As expected, the streaming algorithms are significantly faster than the offline algorithms suggested in [5, 20, 16, 9] and is guaranteed to be no worse than the offline algorithms plus the additive error. Interestingly since we choose between the better of the two solutions (rounding up or down) we got better solution than the offline algorithms!

4. We also propose a new algorithm *Hybrid* which stores rounded coefficients of the data except at the root. Since at the root we have already seen all the data and can choose the best number (or none at all) easily. Surprisingly we show that this extremely small modification already shows significant improvements over the restricted version and is almost of the same quality as the general solution.

1.0.0.3 Simultaneous and Independent work.

While this paper was submitted for review, Karras and Mamoulis [14] have proposed a greedy one pass algorithm for the ℓ_∞ and related maximum measures for the restricted version (storing coefficients of the data) which runs in $O(n \log n)$ time and $O(n)$ space. The algorithm is extended to a streaming setting by repeatedly adding two new coefficients and discarding two old coefficients. Note that the authors of [14] do not provide any guarantees for the synopsis quality for any of the algorithms proposed, but observe on the basis of experiments that their synopses are good. Since all of their algorithms store the coefficients of the input, the example $X = \{1, 4, 5, 6\}$ applies to them as well.

1.0.0.4 Overview.

In Section 2 we discuss the preliminaries of Wavelet transforms and various terminology. In Section 3 we provide the basic algorithm and running time analysis without getting into the streaming or space complexity aspect. In Section 4, we indicate how the algorithm is adapted to a one pass data stream and analyze the space complexity. We also include a discussion on streaming models and the issue of precision. In Section 6 we provide some experimental results showing the proof of concept of these algorithms.

2. PRELIMINARIES

We will work with *non-normalized* wavelet transforms where the inverse computation is simply adding the coefficients that affect a coordinate. For normalized wavelets the normalization constant appears both in forward and inverse transform, all the results in the paper will carry over in that setting as well,with the introduction of the normalization constants at several places. The wavelet basis vectors are defined as (assume n is a power of 2):

$$\begin{array}{rcl} \psi_1(j) = & 1 & \text{for all } j \\ \psi_{2^s+t}(j) = & \left\{ \begin{array}{rl} 1 & \text{if } (t-1)\frac{n}{2^s} + 1 \le j \le \frac{tn}{2^s} - \frac{n}{2^{s+1}} \\ -1 & \text{if } \frac{nt}{2^s} - \frac{n}{2^{s+1}} + 1 \le j \le \frac{tn}{2^s} \end{array} \right. & \\ & & \text{where } (1 \le t \le 2^s, 0 \le s \le \log n) \end{array}$$

The above definitions ensure $\mathcal{W}^{-1}(Z) = \sum_i Z_i \psi_i$. To compute $\mathcal{W}(X)$, we can compute the average $\frac{x_{2i+1}+x_{2i+2}}{2}$ and the difference $\frac{x_{2i+1}-x_{2i+2}}{2}$ for each pair of consecutive elements as i ranges over $0, 1, 2, 3, \ldots$. The difference coefficients form the last $n/2$ entries of $\mathcal{W}(X)$. The process is repeated on the $n/2$ average coefficients - *their difference coefficients yield the $n/4+1, \ldots, n/2$'th coefficients of $\mathcal{W}(Z)$.* The process stops when we compute the overall average, which is the first element of $\mathcal{W}(Z)$.

The wavelet basis functions 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 non-leaf nodes of the tree. Assigning a value c_i to the coefficient corresponds to assigning $+c_i$ to all leaves j that are *left descendants* (descendants of the left child) and $-c_j$ to all right descendants. The leaves that are descendants of a node in the coefficient tree are termed as the *support* of the coefficient.

2.1 Previous Algorithm(s)

In this section we briefly describe the algorithm framework proposed in [5]. Recall that the algorithm only retains coefficients of the input signal. The algorithm uses the coefficient tree, and each node decides the best solution for the subtree *given the choices made at all ancestor nodes in the coefficient tree.* To find the best solution given such a configuration of the ancestors the algorithm needs to allocate the coefficients to the two subtrees. The number of choices of configurations at a node is 2^{depth} (root is at $depth = 0$), and the number of ways of dividing the coefficients (at most B) is $O(B)$. To find the best division we need $O(\log B)$ time (using binary search) and thus the time spent at each node in the coefficient tree is $O(2^{depth} B \log B)$. Since the depth of any node is at most $\log n + 1$ and there are n nodes, the total time taken is $O(n 2^{\log n+1} B \log B)$ which is $O(n^2 B \log B)$. As noted earlier, [20, 16, 9] present better analyses of the algorithm but the computation is $\Omega(n^2)$.

3. BASIC ALGORITHMS AND ANALYSIS

We now show how to obtain an additive approximation algorithm for the general/unrestricted wavelet synopsis construction problem. Recall that the wavelet synopsis problem is: Given a set of n numbers $X = x_1, \ldots, x_n$, find a $Z \in \mathcal{R}^n$ with at most B non-zero entries such that $\|X - \mathcal{W}^{-1}(Z)\|_p$ is minimized.

3.1 Overview and Intuition

The algorithm will be bottom up, which is convenient from a streaming point of view. In this section we will ignore the streaming aspect and prove correctness of our algorithms and the approximation guarantees.

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 computation 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 first show that *there exists* a set R such that if the coefficients were drawn from it, then *there exists* one solution which is close to the optimum unrestricted solution (where we search over all reals). In a sense the set R "rescues" us from the search. Alternately we can view R as a "rounding" of the optimal solution. Obviously such an R exists if we did not care about the error, e.g., take the all zero solution. We would expect a dependence between the set R and the error bound we seek.

However there is a subtle twist – the existence of R is straightforward if $\pi_i > 0$ for all i. But it is unclear if the values of the largest numbers in R is bounded if π_i is very small. For the cases that π_i is very small we would have to allow the algorithm to use *different* sets R_j at each node j of the coefficient tree. We can show that $|R_j|$ will be bounded - but may be $O(n)$. This would imply that the algorithm cannot be made small space if some of the π_i's are small.

In what follows we first show the additive approximation algorithm for minimizing the ℓ_p norm, $\|X - \mathcal{W}^{-1}(Z^*)\|_p$. Subsequently we show how to get an additive approximation for the weighted ℓ_p norm and the relative error ℓ_p norms.

3.2 The Algorithm for ℓ_k Error

Definition 1. 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 obviously be set later for a subtree as we see more and more data. 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 as we will see in the next section.

The overall answer is clearly $\min_b E[root, 0, b]$ – 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. 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:

1. A non-root node computes $E[i, v, b]$ as follows:

$$\min \left\{ \begin{array}{l} \min_{r,b'} E[i_L, v + r, b'] + E[i_R, v - r, b - b' - 1] \\ \min_{b'} E[i_L, v, b'] + E[i_R, v, b - b'] \end{array} \right.$$

   where the upper term computes the error if the i^{th} coefficient is chosen and it's value is $r \in R$; and the lower term computes the error if the i^{th} coefficient is not chosen.

2. Then root computes:

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

   where i_C is the root's only child.

Time Analysis. The size of the error table at node i, $E[i, \cdot, \cdot]$, is $|R| \min\{B, 2^{t_i}\}$ where t_i is the height of node i in the error tree (The leaves have height 0). Further, computing each entry of $E[i, \cdot, \cdot]$ takes $O(|R| \min\{B, 2^{t_i}\})$ time. Hence, the total running time is $O(|R|^2 B^2)$ for computing the root table plus $O(\sum_{i=1}^{n} \left(|R| \min\{2^{t_i}, B\}\right)^2)$ for computing all the other error tables. Now,

$$\sum_{i=1}^{n} \left(|R| \min\{2^{t_i}, B\}\right)^2 = |R|^2 \sum_{t=1}^{\log n} \frac{n}{2^t} \min\{2^{2t}, B^2\}$$

$$= n|R|^2 \left(\sum_{t=1}^{\log B} 2^t + \sum_{t=\log B+1}^{\log n} \frac{B^2}{2^t} \right) = O(|R|^2 nB) \ ,$$

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 $r \in R$, 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,

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

The algorithm needs to maintain the "state" which is the errors for the set R, and all b s.t. $0 \le b \le \min\{B, 2^t\}$ for a node at level t. The bottom up dynamic programming will require us to store the states along at most two leaf to root paths. Thus the required space is

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

3.3 The Set R

In this section, we prove the existence of the set R, as well as show how to find the set. The first task of the proof of existence would be to show that the values in the set R are bounded by some function of the input. The proof is based on the fact that the all zero solution is a feasible solution.

LEMMA 1. *For any vector Y, if $\max_i |Y_i| = M$, then $\max_i |\mathcal{W}(Y)_i| \le M$.*

PROOF. The 1^{st} coefficient is the average of all values and therefore cannot exceed M. Every other coefficient is half the average value of left half (of the support) minus half the average value of right half. Each cannot be more than M in absolute value. □

LEMMA 2. *Let the optimum solution Z^* be better than the all zero vector $\vec{0}$, then $\max_i |Z_i^*| \le 2n^{\frac{1}{p}} M$.*

PROOF. Observe that,

$$\|X - \mathcal{W}^{-1}(Z^*)\|_p \geq \|\mathcal{W}^{-1}(Z^*)\|_p - \|X\|_p$$

Now $\|X\|_p \leq n^{\frac{1}{p}} M$. Also $\|\mathcal{W}^{-1}(Z^*)\|_p \geq \|\mathcal{W}^{-1}(Z^*)\|_\infty = \max_i \left|\mathcal{W}^{-1}(Z^*)_i\right|$. If $\max_i |\mathcal{W}^{-1}(Z^*)_i| > 2n^{\frac{1}{p}} M$ then we get

$$\|X - \mathcal{W}^{-1}(Z^*)\|_p > n^{\frac{1}{p}} M \geq \|X\|_p = \|X - \mathcal{W}^{-1}(\vec{0})\|_p$$

i.e., setting Z^* as the all zero vector $\vec{0}$ improves the solution. This contradicts that Z^* is the optimum solution.

Therefore $\max_i |\mathcal{W}^{-1}(Z^*)_i| \leq 2n^{\frac{1}{p}} M$. Now we apply Lemma 1, on $Y = \mathcal{W}^{-1}(Z^*)$ and get $\max_i |\mathcal{W}^{-1}(Z^*)_i| \geq \max_i |Z_i^*|$ which completes the proof. □

The above lemma is one of the main reasons for choosing to work with a non-normalized basis. An analogous theorem could be proven for normalized coefficients, but the statements of the lemma would be significantly less clean.

¿From the above lemma $\max_{r \in R} |r| = 2n^{\frac{1}{p}} M$. The next lemma bounds the size of R. The basic intuition is that if we approximate the coefficients the effect seen at a point can be bounded.

LEMMA 3. *If we round each non-zero value of the optimum Z^* to the nearest multiple of δ thereby obtaining $\hat{Z}$, then $\|X - \mathcal{W}^{-1}(\hat{Z})\|_p \leq \|X - \mathcal{W}^{-1}(Z^*)\|_p + \delta n^{\frac{1}{p}} \min\{B, \log n\}$ and $|R| \leq 4n^{\frac{1}{p}} M/\delta$.*

PROOF. The bound on $|R|$ clearly follows from Lemma 2 and the size δ since we are interested in searching in the range $\pm \max_i |Z_i^*|$. Now from the triangle inequality we have,

$$\|X - \mathcal{W}^{-1}(\hat{Z})\|_p \leq \|X - \mathcal{W}^{-1}(Z^*)\|_p + \|\mathcal{W}^{-1}(Z^*) - \mathcal{W}^{-1}(\hat{Z})\|_p$$

In what follows, we will argue that $\|\mathcal{W}^{-1}(Z^*) - \mathcal{W}^{-1}(\hat{Z})\|_p$ is at most $\delta n^{1/p} \min\{B, \log n\}$ which will prove the lemma. Note that if $Z_i^* = 0$ then $\hat{Z}_i = 0$ and thus we do not increase the number of coefficients.

For all i we have $|\hat{Z}_i - Z_i^*| \leq \delta$, and each point in $\mathcal{W}^{-1}(Z^*)$ (or $\mathcal{W}^{-1}(\hat{Z})$) is a sum of at most $\min\{B, \log n\}$ wavelet coefficients. Therefore since the rounding errors at each point can at most add up, we get

$$\|\mathcal{W}^{-1}(Z^*) - \mathcal{W}^{-1}(\hat{Z})\|_\infty \leq \delta \min\{B, \log n\}$$

Now we observe that

$$\|\mathcal{W}^{-1}(Z^*) - \mathcal{W}^{-1}(\hat{Z})\|_p \leq n^{\frac{1}{p}} \|\mathcal{W}^{-1}(Z^*) - \mathcal{W}^{-1}(\hat{Z})\|_\infty$$

and the lemma follows. □

Therefore if we set $\delta = \epsilon M/(n^{1/p} \min\{B, \log n\})$ we can say that we have an additive approximation of ϵM as well as $|R| = O(\epsilon^{-1} n^{2/p} \min\{B, \log n\})$. Therefore we conclude with the following:

THEOREM 4. *We can solve the Wavelet Synopsis Construction problem with ℓ_p error with an additive approximation of ϵM in time $O(B\epsilon^{-2} n^{1+4/p} (\min\{B, \log n\})^2)$ where $M = \max_i |x_i|$ using space $O(B\epsilon^{-1} n^{2/p} \min\{B, \log n\} \log \frac{n}{B})$.*

It is immediate that we can achieve a tradeoff of the error and running time. Further,

COROLLARY 5. *For ℓ_∞ error measure the above algorithm runs in time $O(\epsilon^{-2} n \min\{B, \log n\})^2 \log^2 B)$ and uses at most $O(B\epsilon^{-1} \min\{B, \log n\} \log \frac{n}{B})$ space.*

3.4 Weighted and Relative Error

The key to the analysis in Section 3.3 was bounding $\max_i |Z_i^*|$ in the optimal solution Z^*. We will prove a lemma analogous to Lemma 2 above. We will prove the result for the weighted ℓ_p norm; and then show that the result is slightly better for the relative ℓ_p error. Recall that $\pi_i = 1/\max\{|x_i|, c\} > 0$ for the relative ℓ_p error. We begin with the following definition:

Definition 2. Define $\pi_{\min} = \min_i \pi_i$. Recall $\pi_{\max} = \max_i \pi_i$. For the weighted-$\ell_p$ norm $\pi_{\max} = 1$ without loss of generality. For the relative ℓ_p error $\pi_{\max} = 1/\max\{\min_i |x|_i, c\}$ (which is at most $1/c$) and $\pi_{\min} = 1/M$. We can assume $M > c$ since otherwise relative ℓ_p is almost the same as the ℓ_p norm (with a $\pi_i = 1/c$ scaling).

LEMMA 6. *Let the optimum solution be Z^* for the weighted-ℓ_p error measure. If $\max_i |x_i| \leq M$, then $\max_i |Z_i^*| \leq 2n^{1/p} M \frac{1}{\pi_{min}}$. For the relative ℓ_p (if $c < 1$) the bound reduces to $2n^{1/p} \frac{1}{\pi_{min}} = 2Mn^{1/p}$*

PROOF. The proof of this lemma will be similar to Lemma 2 with a small twist. Observe that if $U_i = \pi_i x_i$ and $V_i = \pi_i \mathcal{W}^{-1}(Z^*)_i$ then

$$\|X - \mathcal{W}^{-1}(Z^*)\|_{\pi,p} = \|U - V\|_p \geq \|V\|_p - \|U\|_p$$

We transform the problem to ordinary ℓ_p norm over the weighted vectors. Observe that for relative error $|U_i| \leq 1$ and therefore $\|U\|_p \leq n^{1/p}$. In case of weighted ℓ_p norm $\|U\|_p \leq Mn^{1/p}$ since $\pi_{\max} = 1$.

If $\max_i |V_i| > 2\|U\|_p$, then

$$\|V\|_p - \|U\|_p \geq \|V\|_\infty - \|U\|_p > \|U\|_p$$

Again, the all zero solution provides an error of $\|U\|_p$. Thus we arrive at a contradiction of the optimality of Z^*. Therefore, $\max_i |V_i| \leq 2\|U\|_p$. Now from Lemma 2, we have that $\max_i |\mathcal{W}^{-1}(Z^*)_i| \geq \max_i |Z_i^*|$.

For relative ℓ_p error we get $2n^{1/p} \geq \max_i |V_i| \geq \pi_{\min} \max_i |Z_i^*|$ and the lemma follows. For the weighed ℓ_p norm, we get $(\pi_{\min} \max_i |Z_i^*|) \leq 2Mn^{1/p}$ and the lemma is true. □

The next lemma is immediate from the proof of Lemma 3.

LEMMA 7. *If we round each non-zero value of the optimum Z^* to the nearest multiple of δ thereby obtaining $\hat{Z}$, then $\|X - \mathcal{W}^{-1}(\hat{Z})\|_{\pi,p}$ is at most $\|X - \mathcal{W}^{-1}(Z^*)\|_{\pi,p} + \delta n^{1/p} \min\{B, \log n\}$ since $\pi_{max} = 1$. For relative ℓ_p the error is at most $\|X - \mathcal{W}^{-1}(Z^*)\|_p^{rel} + \delta \frac{n^{1/p}}{\max\{\min_i |x_i|, c\}} \min\{B, \log n\}$.*

Based on the above we get the following

THEOREM 8. *We can solve the Wavelet Synopsis Construction problem for minimizing the relative ℓ_k error in time $O(B\epsilon^{-2} n^{1+4/p} \frac{M^2}{(\max\{c, \min_i |x|_i\})^2} (\min\{B, \log n\})^2)$ and space $O(B\epsilon^{-1} n^{2/p} \frac{M}{\max\{c, \min_i |x|_i\}} \log \frac{n}{B} (\min\{B, \log n\}))$ with an additive error of ϵ. The running time for ℓ_∞ reduces by $B/log^2 B$.*

For the weighted-ℓ_p error the above gives an additive ϵM approximation in $O(B\epsilon^{-2} n^{1+4/p} \frac{1}{\pi_{\min}^2} (\min\{B, \log n\})^2)$ time using space $O(B\epsilon^{-1} n^{2/p} \frac{1}{\pi_{\min}} \log \frac{n}{B} (\min\{B, \log n\})^2)$ where $M = \max_i |x_i|$. Clearly the above result is useful when $\pi_{\min} > 0$. In what follows we will show how to handle $\pi_i = 0$ for the weighted-ℓ_p.

3.5 Weighted ℓ_p and $\pi_i = 0$

Recall the algorithm outline, $E[i, v, b]$ was defined to be the minimum possible contribution to the overall error from all descendants of 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:

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

Denote each entry $E[i, v, *]$ as a "line" – based on the notion that the entries correspond to a table.

LEMMA 9. *At a leaf node i, for weighted-ℓ_p error, if $\pi_i = 0$ then the range does not matter and we can describe the dynamic programming table in one line.*

LEMMA 10. *For any node i there exist two unique lines s.t. the entries $E[i, v, *]$ for $v \notin [-M_i, M_i]$ where $M_i \leq M + Mn^{1/p}/\min_j\{\pi_j | \pi_j > 0$ and j is a descendant of $i\}$ can be represented by those two lines (corresponding to $v > M_i$ and $v < -M_i$).*

PROOF. The proof is by induction on the level of i. For a leaf node with $\pi_i = 0$ clearly we can set $M_i = 0$. For any v, b the error is 0 and therefore one "line" suffices.

Assuming $\pi_i > 0$ for a leaf node i. Then $M_i = M + Mn^{1/p}/\pi_i$ suffices because any value of v outside this range will ensure that the error is at least $Mn^{1/p}$, which we have seen, is more than the error of the all zero solution $\vec{0}$. Thus for any v, b in this range we can set $E[i, v, b] = \infty$ since these entries will never be useful for the optimum solution.

For an internal node the two children (by induction) will return tables which are in the range $[-M_L, M_L]$ and $[-M_R, M_R]$. Let $M_i = \max\{M_L, M_R\}$. For a $v \in [-M_i, M_i]$ the computation of $E[i, v, b]$ is the same as before, except that if we consider storing a coefficient at i whose value v_i is such that $v + v_i$ (or $v - v_i$) exceeds the range $[-M_L, M_L]$ (or $[-M_R, M_R]$) then we use the unique line for the left (or right) hand side. The important point is that v_i cannot be larger than $\pm(|v| + M_i)$ since in that case we would be focusing on the unique lines on both sides and the optimum allocation of the buckets is fixed (does not depend on v).

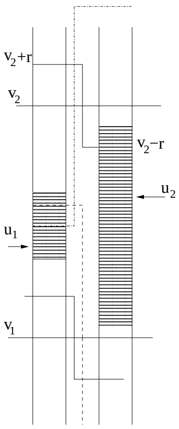


Figure 1: An example of merging tables at a node i

Now consider a $v \notin [-M_i, M_i]$. Suppose we are looking at v_2 as shown in Figure 1. If we do not store a coefficient at node i then the minimization is between the unique lines on the left and right hand sides and is fixed. If we do want to store a coefficient at i, for v_2 consider a wavelet which adds r to the left hand side (wavelet represented by solid line around v_2). The optimization varies only in the right hand side of the table since $v_2 + r$ uses the unique line on the left hand side. For a fixed b, there is exactly one line in the entire range on the right hand side (as $v_2 - r$ varies) which gives the optimum answer to

$$\min_{b_1, r > 0} E[i_L, v_2 + r, b_1] + E[i_R, v_2 - r, b - b_1 - 1]$$

Likewise for $r < 0$ (shown by the dashed line) the right hand side uses the unique line and for every b there is a fixed $u_1(b)$ which minimizes the above equation. Therefore, for every $v \notin [-M_i, M_i]$ and every $0 \leq b \leq B$ the error $E[i, v, b]$ is the minimum of three quantities that are independent of v. Therefore for all such $v > M_i$ (and likewise $v < -M_i$) we can use the same line. □

Based on the above and Lemma 7 we get the following

THEOREM 11. *We can solve the Wavelet Synopsis Construction problem for minimizing the weighted-ℓ_k error in time $O(B\epsilon^{-2}n^{1+4/p}(1/\pi^{+}_{min})^2(\min\{B, \log n\})^2)$ and in space $O(B\epsilon^{-1}n^{2/p}(1/\pi^{+}_{min})\log\frac{n}{B}(\min\{B, \log n\}))$ with an additive error of ϵM. The running time for ℓ_∞ reduces by $B/log^2 B$.*

4. DATA STREAMS

For the purpose of the paper a data stream computation is a space bounded algorithm, where the space is sub-linear in the input. Any input items are accessed sequentially and any item not explicitly stored cannot be accessed again in the same pass. In the specific streaming model we will assume, we are given number $X = x_1, \ldots, x_i, \ldots, x_n$ which correspond to the signal to be summarized in the increasing order of i. This model is often referred to as the aggregated model and has been used widely [12, 7, 10]. This model is specially suited to model streams of time series data [15, 2].

As noted before, our algorithms will not depend on M, but the approximation guarantee of the streaming algorithm will depend on this parameter. This is not a very restrictive assumption, if stock prices rose or fell exponentially, or the temperature readings from a sensor network deployed in a nuclear plant rose exponentially, typically there would be more radical issues at stake. For most reasonable analysis tasks, the input has bounded precision and the guarantee is a non-issue.

The streaming algorithm will build upon the previous section and 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 x_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 [12] for stream clustering. The stream computation of wavelets in [7] can be viewed as a similar idea – where the divisions corresponds to the support of the wavelet basis vectors.

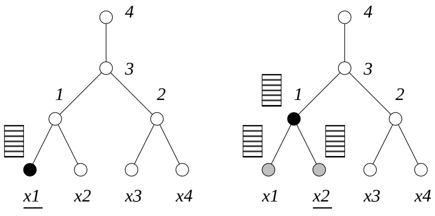


Figure 2: The arrival of the first 3 elements. Upon seeing x_2 node 1 computes $E[1,\cdot,\cdot]$ and the two error arrays associated with x_1 and x_2 are discarded.

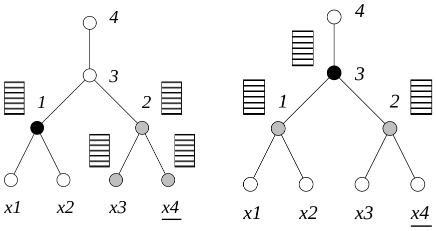


Figure 3: The arrival of x_4 triggers the computation of $E[2,\cdot,\cdot]$ and the two error arrays associated with x_3 and x_4 are discarded. Subsequently, $E[3,\cdot,\cdot]$ is computed from $E[1,\cdot,\cdot]$ and $E[2,\cdot,\cdot]$ and both the latter arrays are discarded. If x_4 is the last element on the stream, the root's error array, $E[3,\cdot,\cdot]$, is computed from $E[2,\cdot,\cdot]$.

4.1 The Streaming Algorithm

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 Sec. 2, form a complete binary tree. For example, the basis vectors for nodes $4, 3, 1$ and 2 in the tree of Fig. 2 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. Hence, as mentioned in Sec. 2, assigning the value c to node 2 (equivalently, setting $z_2 = c$) for example corresponds to adding c to $\mathcal{W}^{-1}(Z)_1$ and $\mathcal{W}^{-1}(Z)_2$, and adding $-c$ to $\mathcal{W}^{-1}(Z)_3$ and $\mathcal{W}^{-1}(Z)_4$.

However, we need not store the error array for every internal node since, in order to compute $E[i, v, b]$ our algorithm from Sec. 3.2 only requires that $E[i_L,\cdot,\cdot]$ and $E[i_R,\cdot,\cdot]$ be known. Hence, it is natural to perform the computation of the error arrays in a post-order fashion. An example best illustrates the procedure. In Fig 2 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 computations of the rest of the arrays as in Fig. 3.

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}, and E_{t_1} is discarded; (3) E_{t_2} and E_{q_1} are used to create E_{q_2} which in turn is added to the queue; and (4) E_{t_2} is discarded. Figure APX shows the implementation of our algorithm for relative ℓ_∞.

```
Algorithm $APX(B, M, \delta)$
Let $|R| = 4M/\delta$.
Initialize a queue $Q$ with one node $q_0$
(* Each $q_i$ contains an array $E_{q_i}$ of size *)
(* $|R| \min\{B, 2^i\}$ and a flag isEmpty *)
repeat Until there are no elements in the stream
Get the next element from the stream, call it $e$
if $q_0$ is empty
   then Initialize $E_{q_0}[r \in R, 0] = |r/e - 1|$
   else Create $t_0$ and Initialize $E_{t_1}[r \in R, 0] = |r/e-1|$
        for $i = 1$ until the 1st empty $q_i$ or end of $Q$
             do Create a temporary node $t_2$.
                Compute $E_{t_2}[r, b \in B]$ from $t_1$ and $q_{i-1}$
                Set $t_1 \leftarrow t_2$ and Discard $t_2$
                Set $q_i$.isEmtpy = true
        if we reached the end of $Q$
           then Create the node $q_i$
        Compute $E_{q_i}[r \in R, b \in B]$ from $t_1$ and $q_{i-1}$
        Set $q_i$.isEmtpy = false and Discard $t_1$
```

Figure 4: The Streaming Algorithm

Based on the description of above, the algorithm uses the same space as mentioned in the offline algorithm in the previous section. Therefore we conclude with:

THEOREM 12. *We can solve the Wavelet Synopsis Construction problem in a single pass over the data by providing an algorithm (assuming $M = \max_i |x_i|$) that*

- *For ℓ_p error with an additive approximation of ϵM the algorithm runs in time $O(B\epsilon^{-2}n^{1+4/p}(\min\{B, \log n\})^2)$ using space $O(B\epsilon^{-1}n^{2/p}\min\{B, \log n\}\log\frac{n}{B})$.*
- *For minimizing the weighted-ℓ_k error the algorithm runs in time $O(B\epsilon^{-2}n^{1+4/p}(1/\pi^+_{min})^2(\min\{B, \log n\})^2)$ and in space $O(B\epsilon^{-1}n^{2/p}(1/\pi^+_{min})\log\frac{n}{B}(\min\{B, \log n\}))$ with an additive error of ϵM.*
- *For the relative ℓ_k error the algorithm runs time $O(B\epsilon^{-2}n^{1+4/p}\frac{M^2}{(\max\{c, \min_i |x|_i\})^2}(\min\{B, \log n\})^2)$ and space $O(B\epsilon^{-1}n^{2/p}\frac{M}{\max\{c, \min_i |x|_i\}}\log\frac{n}{B}(\min\{B, \log n\}))$ with an additive error of ϵ.*

The running time for ℓ_∞ reduces by B/log^2B in all cases.

5. QUALITY VERSUS TIME

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

The first observation we make is 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)

$$\min \left\{ \begin{array}{l} \min_{r,b'} E[i_L, v+r, b'] + E[i_R, v-r, b-b'-1] \\ \min_{b'} E[i_L, v, b'] + E[i_R, v, b-b'] \end{array} \right.$$

We only need to search for $r = c_i$ where c_i is the wavelet coefficient at i – observe that a streaming algorithm can compute c_i (See [7]). However we have to "round" the c_i since we are storing the table corresponding to the values in R and c_i may have higher precision. We consider *the better of rounding up or rounding down c_i to the nearest multiples of δ*. Notice the set R still performs the role of "anticipatory values" that are being set by the rounded ancestors. The running time improves by a factor of R in this case – since to compute each entry we are now looking at two values of R (round up/down) instead of the entire set. The overall running time is $O(|R|nB)$ in the general case and $O(|R|n\log^2 B)$ 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 $Z_i = \mathcal{W}(X)_i$ or 0 otherwise.* We cannot show any relationship between the quality of this solution and the general unrestricted case. However in the experiments we found that simply deciding between the better of rounding up or down gives a significant improvement in quality in some case. The rounding also introduces more (but bounded) error in other cases, as is expected from an approximation. We conclude with:

Theorem 13. *We can solve the restricted Wavelet Synopsis Construction problem in a single pass over the data by providing an algorithm (assuming $M = \max_i |x_i|$) that*

- *For ℓ_p error with an additive approximation of ϵM the algorithm runs in time $O(B\epsilon^{-1}n^{1+2/p}\min\{B,\log n\})$ where using space $O(B\epsilon^{-1}n^{2/p}\min\{B,\log n\}\log\frac{n}{B})$.*

- *For minimizing the weighted-ℓ_k error the algorithm runs in time $O(B\epsilon^{-1}n^{1+2/p}(1/\pi^+_{min})\min\{B,\log n\})$ and in space $O(B\epsilon^{-1}n^{2/p}(1/\pi^+_{min})\log\frac{n}{B}(\min\{B,\log n\}))$ with an additive error of ϵM.*

- *For the relative ℓ_k error the algorithm runs time $O(B\epsilon^{-1}n^{1+2/p}\frac{M}{\max\{c,\min_i|x|_i\}}\min\{B,\log n\}c)$ and space $O(B\epsilon^{-1}n^{2/p}\frac{M}{\max\{c,\min_i|x|_i\}}\log\frac{n}{B}(\min\{B,\log n\}))$ with an additive error of ϵ.*

The running time for ℓ_∞ reduces by B/log^2B in all cases.

5.1 Hybrid Algorithms

The previous theorem sets the ground for investigating a variety of *Hybrid algorithms* where we choose different search strategies (i.e., what set does r range over) at each of the nodes i. One of the simplest algorithms is to allow $r \in R$ at the root node since we already have full information from the input, and locally at the root, we can choose the best constant value to add.

Observe that this strategy already gives the optimum solution for $B=1$ in the bad example $\{1,4,5,6\}$ – and in fact this observation is our motivation for studying the strategy. We can show that just this small modification improves the synopsis quality significantly.

6. EXPERIMENTAL RESULTS

We consider two issues in this section, namely (i) the quality of the unrestricted version vis-a-vis the restricted optimum solution and (ii) the running time of the algorithms.

6.1 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.

Due to shortage of space we restrict ourselves to the ℓ_∞ and relative ℓ_∞ for the purposes of this section. 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 [9] see also [5, 20, 16].

UNREST This is the streaming algorithm for the *full general version* described in Figure 4 based on the discussion in Section 3.

JITTER This is the streaming algorithm for the *restricted* version of the problem described in Section 5.

HYBRID This is the streaming hybrid algorithm proposed in Section 5.

6.2 The Data Sets

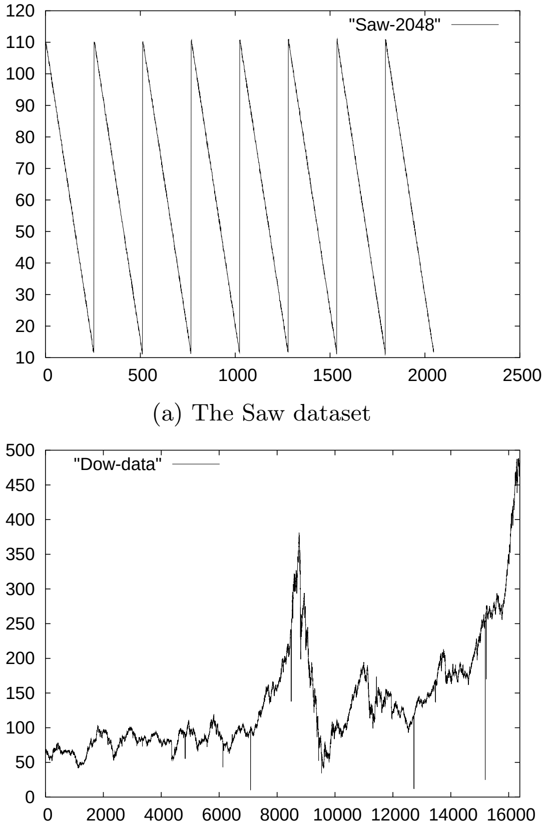
We chose a synthetic dataset to showcase the point made in the introduction about the sub-optimality of the restricted versions. Otherwise we use a publicly available real life data set for our experiment.

- **Saw:** This is a periodic dataset with a line repeated 8 times, with 2048 values total. The dataset is shown in Figure 5(a). This dataset is particularly useful for relative error measures since there is a wide variation in the values.
- **Real life data set:** We used the Dow-Jones Industrial Average (DJIA) data set available at StatLib[*] 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 dataset of size upto 16384. The dataset is shown in Figure 5(b).

6.3 Quality of Synopsis

Maximum Relative Error: The maximum relative errors as a function of B are shown in Figures 6 and 7. The δ in the approximation algorithms UNREST, JITTER and HYBRID, was set to 1, as indicated by the discussion in Section 3.4. We show two figures for Saw data to emphasize that the behavior alluded to in the introduction occurs at a wide range of B values and the differences are highlighted since the overall range changes. The restricted version REST either has 30% more error or requires 20% more coefficients compared to the general unrestricted version. The JITTER and Hybrid algorithms lie in between, HYBRID being better than JITTER as expected. Notice that JITTER follows REST and then switches to the better behavior of UNREST.

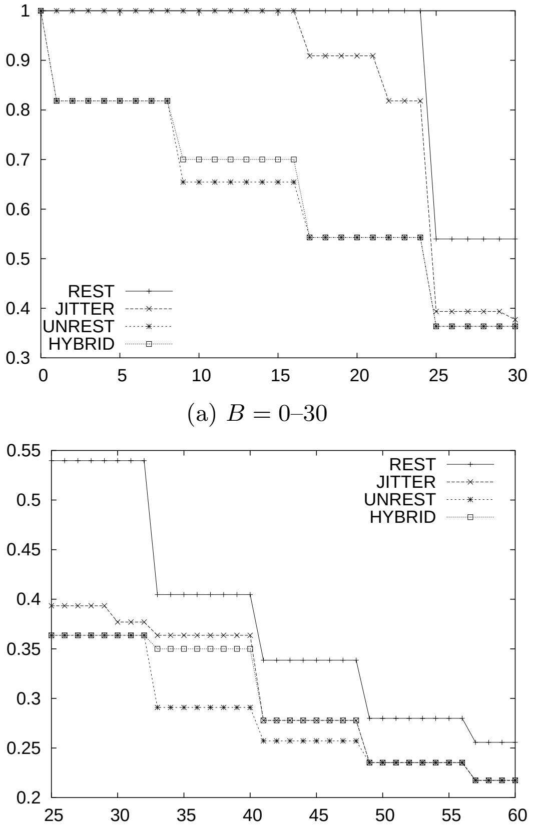
[*] See http://lib.stat.cmu.edu/datasets/djdc0093.



(a) The Saw dataset

(b) The DJIA dataset

Figure 5: The datasets used



(a) $B = 0$–30

(b) $B = 25$–60

Figure 6: Saw data, $\ell_\infty^{\mathbf{rel}}$ Error, $n = 2048$, $\delta = 1$

Observe also that just the simple power of choosing the better of round up or round down achieves a significant improvement. However REST and JITTER are not great for moderate to small B, which is an important range in synopsis construction.

Maximum Error (ℓ_∞): The ℓ_∞ errors as a function of B are shown in Figure 8. The δ in the approximation algorithms UNREST, JITTER and HYBRID, was set to $M/\min\{B, \log n\}$ as described in Section 3. We show only the Dow data since all the algorithms gave very similar synopsis 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) biases the scale making the differences in the more interesting ranges not visible. Once again REST has a 30% worse error compared to UNREST or requires a lot more coefficients (as a ratio of the synopsis size of UNREST). The HYBRID algorithm performs consistently in the middle.

6.4 Running Times

Figure 9 shows the running times of the algorithms as the prefix size n is varied for the Dow data. We report the running time of the ℓ_∞ algorithms only. As mentioned above ϵ was set to 0.1 and δ was set analogously.

The grid in the log-log plot helps us to clearly identify the quadratic nature of REST. The algorithms UNREST, JITTER and HYBRID behave linearly.

6.5 Summary

¿From the preliminary experiments shown in this paper the following properties are immediate:

- The first issue is of quality. The unrestricted synopsis has 30% less error in real life and synthetic data and is significantly better. The Saw data showcases that the problems with the restricted versions demonstrated in the motivating example $\{1, 4, 5, 6\}$ can be realized easily.

- The growth rate of REST is clearly quadratic. The algorithm is however faster than UNREST due to the latter searching over a significantly richer space. The algorithm UNREST and the approximation algorithms (for REST), JITTER and HYBRID are linear as is expected from streaming algorithms. Based on the running times, the quality, and the one-pass behavior, the algorithm HYBRID is definitely the best choice, specially if we are seeking a restricted synopsis.

We are currently investigating speeding up the algorithm UNREST by analyzing the search space and pruning the computation.

Acknowledgments We thank Vlad Petric for help with

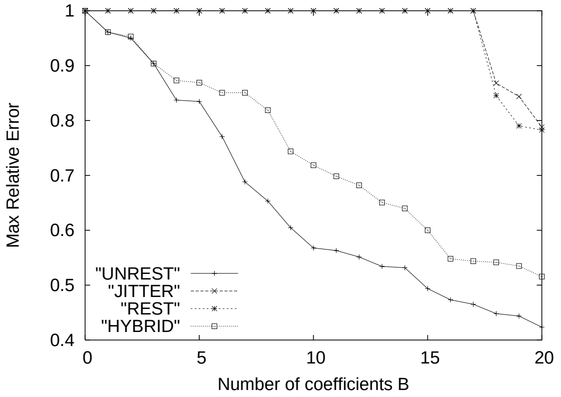


Figure 7: Dow data, $\ell_\infty^{\mathbf{rel}}$ Error, $n = 16384$, $\delta = 1$

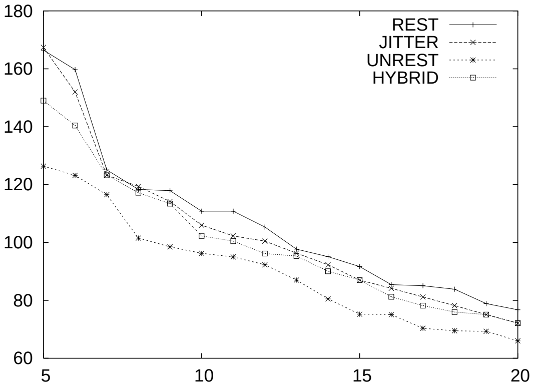


Figure 8: Dow data ℓ_∞ Error, $n = 16384$, $\epsilon = 0.1$

the implementations, and Sampath Kannan for many useful discussions. We thank the referees for useful feedback which improved the paper significantly.

7. REFERENCES

[1] K. Chakrabarti, M. N. Garofalakis, R. Rastogi, and K. Shim. Approximate query processing using wavelets. *VLDB Conference*, 2000.

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

[3] I. Daubechies. *Ten Lectures on Wavelets*. SIAM, 1992.

[4] A. Deligiannakis and N. Roussopoulos. Extended wavelets for multiple measures. *SIGMOD Conference*, 2003.

[5] M. Garofalakis and A. Kumar. Deterministic wavelet thresholding for maximum error metric. *Proc. of PODS*, 2004.

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

[7] A. Gilbert, Y. Kotadis, S. Muthukrishnan, and M. Strauss. Surfing Wavelets on Streams: One Pass Summaries for Approximate Aggregate Queries. *VLDB Conference*, 2001.

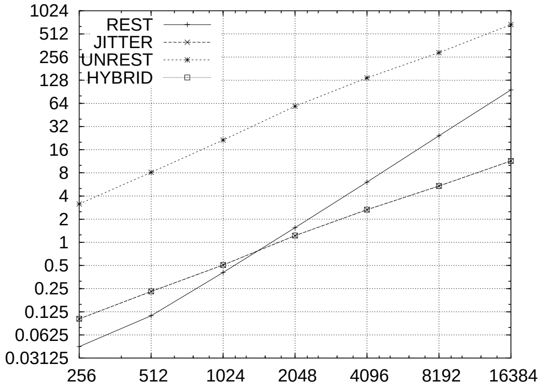


Figure 9: Running times, ℓ_∞, $\epsilon = 0.1$

[8] A. C. Gilbert, S. Guha, P. Indyk, Y. Kotidis, S. Muthukrishnan, and Martin Strauss. Fast, small-space algorithms for approximate histogram maintenance. In *Proc. of ACM STOC*, 2002.

[9] S. Guha. Space efficiency in synopsis construction problems. *VLDB Conference*, 2005.

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

[11] S. Guha, C. Kim, and K. Shim. XWAVE: Optimal and approximate extended wavelets for streaming data. *VLDB Conference*, 2004.

[12] S. Guha, N. Mishra, R. Motwani, and L. O'Callaghan. Clustering data streams. *Proc. of FOCS*, 2000.

[13] C. E. Jacobs, A. Finkelstein, and D. H. Salesin. Fast multiresolution image querying. *Computer Graphics*, 29(Annual Conference Series):277–286, 1995.

[14] P. Karras and N. Mamoulis. One pass wavelet synopis for maximum error metrics. *VLDB Conference*, 2005.

[15] E. Keogh, K. Chakrabati, S. Mehrotra, and M. Pazzani. Locally Adaptive Dimensionality Reduction for Indexing Large Time Series Databases. *Proc. of SIGMOD*, 2001.

[16] Y. Matias and D. Urieli. Manuscript. 2004.

[17] Y. Matias and D. Urieli. Optimal workload-based wavelet synopses. *Proc. of ICDT*, 2005.

[18] Y. Matias, J. S. Vitter, and M. Wang. Dynamic Maintenance of Wavelet-Based Histograms. *VLDB Conference*, 2000.

[19] Y. Matias, J. Scott Vitter, and M. Wang. Wavelet-Based Histograms for Selectivity Estimation. *Proc. of SIGMOD*, 1998.

[20] S. Muthukrishnan. Workload optimal wavelet synopsis. *DIMACS TR*, 2004.

[21] G. Strang and T. Nguyen. *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1996.