DIMACS Technical Report 2004-42
September 2004

Nonuniform Sparse Approximation with Haar Wavelet Basis [1]

by

S. Muthukrishnan [2]
Dept. of Computer Science
Rutgers University
New Brunswick, New Jersey 08903.

[1]This report is an expanded version of "Optimal weighted wavelet synopsis algorithm" by the author, DIMACS TechReport 2004-25.

[2]Supported by NSF ITR 0220280.

ABSTRACT

Sparse approximation theory concerns representing a given signal on N points as a linear combination of at most B ($B \ll N$) vectors from a dictionary so that the error of the representation is minimized; traditionally, error is taken *uniformly* as the sum of squares of errors at each point.

In this paper, we initiate the study of *nonuniform* sparse approximation theory where each point has an *importance*, and we want to minimize the sum of individual errors weighted by their importance. In particular, we study this problem with the basic Haar wavelet dictionary that has found many applications since being introduced in 1910. Parseval's theorem from 1799 which is central in solving uniform sparse approximation for Haar wavelets does not help under nonuniform importance.

We present first known polynomial time algorithms for the problem of finding B wavelet vectors to represent a signal of length N so that the representation has the smallest error, averaged over the given importance of the points. The algorithms take time $O(N^2B/\log B)$ in general and take near-linear time when the importance function is well-behaved. Our methods also give first known, efficient algorithms for a number of related problems with nonuniform importance: for example, for a dual problem in nonuniform sparse approximation theory, we present a subquadratic time algorithm. We also pose and discuss new open problems.

1 Introduction

The main problem of *sparse approximation theory* is to approximate functions compactly, using a small number of elements from a given dictionary. Formally, we are given a *signal* $\mathbf{A} \in \mathcal{R}^N$ and a *dictionary* $\mathcal{D} = \{\phi_i\}$ of unit vectors that span $\mathcal{R}^N$. A representation $\mathbf{R}$ of B terms for input $\mathbf{A} \in \mathcal{R}^N$ is a linear combination of dictionary elements,

$$\mathbf{R} = \sum_{i \in \Lambda} \alpha_i \phi_i,$$

for $\phi_i \in \mathcal{D}$ and some Λ, $|\Lambda| \leq B$. Typically, $B \ll N$. The error of the representation is measured by how well it approximates $\mathbf{A}$. Following common practice, we define the error of approximation to be

$$\epsilon_{\mathbf{R}} = \|\mathbf{A} - \mathbf{R}\|_2^2 = \sum_i |\mathbf{A}[i] - \mathbf{R}[i]|^2,$$

henceforth written $\|\mathbf{A} - \mathbf{R}\|$. Sparse approximation theory studies (a) which dictionaries are suitable for different applications, (b) which signals are effectively represented using small number B of elements, (c) how to find the representation with smallest error, etc. It is a mature area with many decades of significant mathematical achievements and successful applications both classical and novel; see *Constructive Approximation* [4] and the *Journal of Approximation Theory* for survey and recent mathematical developments. Some of the areas where sparse approximation theory is applied include adaptive PDE solvers, audio signal compression, image compression, statistical estimation and query optimization in databases.

In this paper, we initiate the study of algorithms for *nonuniform* sparse approximation. That is, there is bias in the importance of error due to the different points in the domain; the bias in practice depends on the application. Formally, there is an *importance function* π where $\pi[i]$ is the importance of i and it is assumed to be nonnegative. (It does not matter if π is normalized to 1 but that may be convenient to think of $\pi[i]$'s as probabilities.) The uniform importance case is the one when say $\pi[i] = 1/N$ for all i. In the general importance case then, one wants to minimize, analogous to the uniform importance case, the total squared-error over all i's, but now normalized for the importance of each point i. More precisely, we wish to minimize

$$\epsilon^{\pi}_{\mathbf{R}} = \sum_i \pi[i]\ (\mathbf{R}[i] - \mathbf{A}[i])^2.$$

The problem of *nonuniform sparse approximation* is as follows:

Problem 1 *Given a signal $\mathbf{A} \in \mathcal{R}^N$, importance function π, and a dictionary $\mathcal{D} = \{\phi_i\}_{i \in I}$ of unit vectors that span $\mathcal{R}^N$, find a representation $\mathbf{R}$ of B terms for $\mathbf{A}$ given by $\mathbf{R} = \sum_{i \in \Lambda} \alpha_i \phi_i$, for $\phi_i \in \mathcal{D}$ and some Λ, $|\Lambda| \leq B$, $B \ll N$, such that $\epsilon^{\pi}_{\mathbf{R}}$ is minimized.*

Nonuniform sparse approximation is well motivated in many applications where sparse approximation finds uses. In the next section, we will give a precise application in databases.

In this paper, we study Problem 1 under the dictionary comprising Haar wavelets [11] which has become one of the fundamental dictionaries in sparse approximation theory since its introduction in 1910. If the importance were uniform, the classical Parseval's theorem from 1799 immediately gives us an optimal algorithm, namely, choose the B largest inner products of the signal with the Haar basis vectors in magnitude. However, for nonuniform importance, Parseval's theorem does not apply and no prior algorithmic result is known. In general when Parseval's theorem does not apply (say with more a general $\mathcal{D}$ not necessary orthonormal), there are few algorithmic techniques if any that we know. Hence, for nonuniform sparse approximation, new algorithmic approaches are needed.

The main contribution of this paper is to initiate the study of algorithms for nonuniform sparse approximation of signals. Our contributions are as follows:

1. We present the first known polynomial time algorithm for the problem with Haar wavelet dictionary. Our algorithm takes time $O(N^2B/\log B)$ to obtain the optimal solution for any general importance function π.

2. For well-behaved importance functions (defined later in the paper with motivations), we present an algorithm that takes only near-linear time to get the optimal solution.

3. We also present efficient algorithms for a number of extensions of the basic problem. For example, for a dual problem where a budget of some error is provided and we need to minimize B to meet the budget, we present algorithms that take time subquadratic in N.

4. Finally, we present several new open problems and discuss their significance. Some of these open problems represent nontrivial intuition about the role of the importance fuction π.

Taken together we hope the results inspire more developments in nonuniform sparse approximation.

All our results are obtained by a common framework that involves the combination of two different kinds of searches: external and internal. External searches are based on dynamic programming. Depending on the problem, internal searches are either exhaustive or based on structural properties of the problem. In particular, we prove a version (Lemma 2) of local application of Parseval's theorem which is very effective for internal searches with well-behaved π's and this may be of independent interest. External search alone often suffices to get polynomial time algorithms; internal search typically helps improve the running time, in some cases, interestingly, to be subquadratic in N. The entire framework succeeds because of the orthogonality and other structural properties such as the dyadic support structure (defined later) of the Haar wavelet vectors.

In Section 2, we will present some background on Haar wavelets, show how to represent signals using Haar wavelets, define the nonuniform sparse approximation problem for Haar wavelets and present one concrete application to database query processing. Since study of algorithms for Haar wavelets and their various applications is not common in the TCS community, this section will be introductory. In Section 3, we will present the external and internal search framework for the problem with both general π's and well-behaved π's. We will present extensions in Section 4 where we discuss the dual problem. Finally, in Section 5, we present new open problems and describe our intuitions. Many of the proofs of technical lemmas have been omitted and the algorithms have been provided only in sketch.

2 Background on Wavelets

2.1 Definition of Haar Wavelets

We first define wavelets. There are many kinds of wavelets. We work with one of the simplest, most popular and most fundamental, ie., Haar wavelets [11].

Definition 1 *Let N be a power of 2. We define $N-1$ proper wavelet functions on $[0, N-1)$ as follows. For integer j, $0 \le j < \log(N)$ and integer k, $0 \le k < 2^j$, we define a* proper wavelet *by*

$$\phi(x)[j,k] = \begin{cases} -\sqrt{2^j/N}, & x \in [kN/2^j, kN/2^j + N/2^{j+1}) \\ +\sqrt{2^j/N}, & x \in [kN/2^j + N/2^{j+1}, (k+1)N/2^j) \\ 0, & \textit{otherwise}. \end{cases}$$

Additionally, we define a wavelet function ϕ, also known as a scaling function*, that takes the value $+1/\sqrt{N}$ over the entire domain $[0, N)$. We number the $\phi(x)[j,k]$'s as $\psi_0, \psi_1, \ldots, \psi_{N-2}$ in some arbitrary order and ϕ as ψ_{N-1}.*

The *support* of a vector v, denoted $\mathrm{supp}(v)$, is $\{t : v(t) \neq 0\}$. Thus the support of a wavelet vector is either the entire interval $[0, N)$ or, recursively, the left half or right half of the support of some other wavelet. The wavelet ψ_{N-1} is constant on its entire support; each other wavelet is constant on the left half and right half of its support and takes values on its left and right halves that are negatives of each other. The set of possible supports of a wavelet is the set of dyadic intervals of length at least 2. (A *dyadic* interval is the whole space a power of two in length, or recursively, a dyadic interval comprising the left or right half. The intervals that x belong to in the definition above are dyadic.)

2.2 Haar Wavelet Synopsis for Signals

Wavelets can be used to represent signals. Consider a signal $\mathbf{A}[0...N-1]$, where N is a power of 2 for convenience and $\mathbf{A}[i] \in \mathcal{R}$. Any signal $\mathbf{A}$ is exactly recoverable using the wavelet basis, ie.,

$$\mathbf{A} = \sum_i \langle \mathbf{A}, \psi_i \rangle \, \psi_i.$$

Typically, we are not interested in recovering the signal exactly using all the N *wavelet coefficients* $c_i = \langle \mathbf{A}, \psi_i \rangle$; instead, we want to represent the signal using no more than B wavelet coefficients for some $B \ll N$. Say Λ is a set of wavelets of size at most B. Signal $\mathbf{A}$ can be represented as $\mathbf{R}$ using these coefficients as follows:

$$\mathbf{R} = \sum_{i \in \Lambda} \langle \mathbf{A}, \psi_i \rangle \, \psi_i.$$

Necessarily $\mathbf{R}$ is an approximation of $\mathbf{A}$. The *best B-term representation* (aka *wavelet synopsis*) of $\mathbf{A}$ is the choice of Λ that minimizes the error $\epsilon_{\mathbf{R}}$ of representing $\mathbf{A}$ which is typically the sum-squared-error, i.e., $\epsilon_{\mathbf{R}} = ||\mathbf{R} - \mathbf{A}||_2^2 = \sum_i (\mathbf{R}[i] - \mathbf{A}[i])^2$.

From a fundamental result of Parseval from 1799 [16, 2], it follows that the best B-term representation for signal $\mathbf{A}$ is $\mathbf{R} = \sum_{i\in\Lambda} c_i\psi_i$, where $c_i = \langle \mathbf{A}, \psi_i\rangle$ and Λ of size k maximizes $\sum_{i\in\Lambda} c_i^2$. This is because the error of $\mathbf{R}$ is

$$\epsilon_{\mathbf{R}} = \sum_{i\in\Lambda}(c_i - \langle \mathbf{A}, \psi_i\rangle)^2 + \sum_{i\notin\Lambda} \langle \mathbf{A}, \psi_i\rangle^2 .$$

Hence, choosing the B largest $|c_i|$'s suffices to get the best B term representation for any signal.

2.3 Nonuniform sparse approximation with Haar wavelets

We are interested in the nonuniform sparse approximation with Haar wavelets which is formally defined below.

Problem 2 *Given signal* $\mathbf{A}[0,\ldots,N-1]$*,* N *a power of* 2*, an integer* $B \ll N$*, and importance* $\pi[0,\ldots,N-1]$*, determine set* Λ *of wavelets such that* $|\Lambda| \leq B$*, and* $\epsilon^{\pi}_{\mathbf{R}} = \sum_i \pi[i](\mathbf{R}[i] - \mathbf{A}[i])^2$ *is minimized where* $\mathbf{R} = \sum_{i\in\Lambda} \langle \mathbf{A}, \psi_i\rangle \psi_i$.

It is trivial to observe that the B i's with the largest $|c_i|$'s no longer form the best B-term representation and indeed can be arbitrarily far away from the optimal in specific instances. So, algorithmically the problem is nontrivial.

2.4 Motivation

Nonuniform version of the sparse approximation problem can be motivated for many different dictionaries and applications. Here, we will describe one concrete application for the problem with the Haar wavelets.

Consider how Haar wavelet synopses get used in databases and data analysis. A signal $\mathbf{A}$ is represented using a B-term wavelet synopsis. Then its representation $\mathbf{R}$ is used as the (succinct) approximation or the surrogate of $\mathbf{A}$. When the signal is queried using *point queries*, i.e., queries are of the form "$\mathbf{A}[i]$ =?", the approximation $\mathbf{R}[i]$ is returned instead. *If all point queries are equally likely*, then average of the squared error of using $\mathbf{R}[i]$ rather than $\mathbf{A}[i]$ is precisely $\epsilon_{\mathbf{R}}$, thus, the rationale to minimize $\epsilon_{\mathbf{R}}$.

Our problem here arises in the same setting. However, in practice, all point queries are *not* equally likely. In general, there is bias in how points are queried that depends on the application and the nature of the signal. So, there is a *workload* π where $\pi[i]$ is the *probability* of i being queried.[1] The uniform workload case is the one when $\pi[i] = 1/N$ for all i. In the general workload case then, one wants to minimize, analogous to the uniform workload case, the total squared-error over all i's, but now normalized for the probability of a point i being queried (equivalently the probability of incurring the error on i). More precisely, we wish to minimize $\epsilon^{\pi}_{\mathbf{R}} = \sum_i \pi[i]\ (\mathbf{R}[i] - \mathbf{A}[i])^2$. This is our nonuniform sparse approximation problem with Haar wavelet basis.

This problem is fairly central to query optimization in databases. Databases maintain workload information by profiling queries in a number of ways and would prefer to use wavelet synopsis that accurately reflects their cost for the given workload rather than ignore this information. See details

[1] Henceforth we use "workload" and "importance" interchangeably.

of the LEO optimizer of IBM's DB2 that was a major release recently [12] in order to understand the effort database engines put in to obtain and use workload information. In database literature, there has been several heuristic results related to our problem. In [1, 6], the general concept of synopses that adapt to the nonuniform workload is studied; the synopses the authors study there are histograms and samples respectively, but the issue is equally important with wavelet synopses since they have been shown to be powerful for database applications in [14]. In [17], authors adapted wavelet based estimation to different workloads, however, they studied the order of retrieval of the full set of wavelet coefficients in order to be efficient over a whole set of queries and provide the full reconstruction of a signal; they did not study the problem of finding the best B-term representation. In [3], authors studied how to obtain B-term representation for a *set* of signals with cumulative error bound, but each signal was approximated uniformly.

No algorithmic results are previously known for our problem.

3 Polynomial Time Algorithms

3.1 General π's

Consider a dyadic interval I. We define $E(I, k, S)$ which is the least importance-based error for representing $\mathbf{A}[I]$ with any subset Λ_I of wavelets that have supports contained in I, $|\Lambda_I| \leq k$, together with the contribution for the set S of wavelets whose individual support strictly contain I. Clearly $E([1, N], B, \phi)$ solves the entire problem (where ϕ is the empty set). We will compute $E(I, k, S)$ in two different ways.

External Search. Our emphasis is on searching over the different sets S that impinge on I. We will use dynamic programming.

Each dyadic interval I has a left half denoted I_L and a right half denoted I_R which are themselves dyadic intervals. With each interval I, we associate the wavelet vector ψ_I that is positive on I_L and is negative on I_R. There are two cases to consider while computing $E(I, k, S)$: ψ_I is included in the set of k wavelet terms or not. Either way, we will write $E(I, k, S)$ in terms of similar functions for I_R and I_L recursively.

Say ψ_I is included in the set of k wavelet terms. Then,

$$E(I, k, S) = \min_{0 \leq k' \leq k-1} E(I_L, k', S \cup \psi_I) + E(I_R, k - 1 - k', S \cup \psi_I).$$

In the other case,

$$E(I, k, S) = \min_{0 \leq k' \leq k} E(I_L, k', S) + E(I_R, k - k', S).$$

Hence, $E(I, k, S)$ is the minimum of the two possibilities above. The boundary cases are easy to define, and $E(I, k, S)$ can be calculated using dynamic programming. Each such term takes time $O(\log k)$ to compute using the monotonicity of the terms on the right as k' increases.

Internal Search. Our emphasis is now over wavelet vectors whose supports are contained in I. Specifically, the contribution of S, $|S| \leq k$, to a $j \in I$ is denoted

$$v[j] = (\sum_{\psi_i \in S} \langle \mathbf{A}, \psi_i \rangle \, \psi_i)[j].$$

Since all ψ_i's in S have support that contains I, $v[j]$'s will be identical for all $j \in I$. So, we have a generic v to denote the value $v[j]$ for any $j \in I$. Now,

$$E(I,k,S) = \min_{|\Lambda_I| \le k; \mathrm{supp}(\psi_i \in \Lambda_I) \subset I} \sum_{j \in I} \pi[j] \left((\sum_{\psi_i \in \Lambda_I} \langle \mathbf{A}, \psi_i \rangle \psi_i)[j] + v - \mathbf{A}[j] \right)^2 .$$

This can be calculated in $O(1)$ time for each new $\psi_i \in \Lambda_I$ by considering them in the bottom-up order of the wavelet supports. So, the total time taken for populating this table is $O(2^{|I|})$ for each S in the worst case by exhaustive search.

Putting Things Together. The final algorithm involves doing the external search approach for large intervals and finishing it off with internal search for short intervals.

Say we do external search for all dyadic intervals of size strictly greater than 2^j, and internal search for the intervals of size precisely 2^j (called *short*).

The number of short intervals is at most $N/2^j$. Each such interval I has $\log N - j$ wavelet vectors whose support strictly contains I. Hence the set of possible sets S that are involved in computing $E(I,k,S)$ is at most $2^{\log N - j}$. Each term takes $O(2^{2^j})$ time to compute. Hence, the total cost is

$$O(\frac{N}{2^j} 2^{\log N - j} 2^{2^j}).$$

For longer intervals, the total cost is dominated by the intervals of smallest size, i.e., 2^{j+1}, but we will perform the computations as if the smallest size interval is size 2^j for convenience. The total cost here is

$$O(\frac{N}{2^j} 2^{\log N - j} B \log B).$$

Now we will balance the two costs to choose j. When the two costs are equal, we have $2^{2^j} = B \log B$, so we choose $j = \log\log B$. Then, the total cost is $O((\frac{N}{2^j})^2 B \log B) = O(N^2 B / \log B)$. From the optimal cost, one can easily extract the actual choice of wavelets that will give the optimal cost using standard techniques in dynamic programming applications without any asymptotic cost.

Theorem 1 *Problem 2 can be solved in time $O(N^2 B / \log B)$ optimally for any arbitrary π.*[2]

The proof of correctness follows from the following observation which looks natural, but needs a detailed technical proof. The crux is that for any partition of a dyadic interval I into dyadic subintervals, the choice of wavelet vectors to represent each subinterval in the partition is independent of those used to represent other subintervals in the partition. This lemma will not be true if the partition consists of non-dyadic subintervals.

Lemma 1 *Let S be any set of wavelet vectors and $E[I,S]$ the error of representing the dyadic interval I of signal $\mathbf{A}$ using S, that is,*

$$E(I,S) = \sum_{j \in I} \pi[j] \left((\sum_{\psi_i \in S} \langle \mathbf{A}, \psi_i \rangle \psi_i))[j] - \mathbf{A}[j] \right)^2 .$$

[2] Theorem 1 appeared in May 2004 [15]. Recently the running time has been improved to $O(N^2 \log B)$ [10]. No improvements are known to any of our other theorems in this paper.

Further let $I = I_1 \cup I_2 \cup \cdots I_\ell$ for some ℓ, where I_i's are dyadic intervals with any two intervals I_i and I_j being disjoint. Also, let S_i denote those wavelets in i whose support intersects I_i, i.e., $S_i = \{\psi_j \mid \mathrm{supp}(\psi_j) \in I_i \text{ or } I_i \in \mathrm{supp}(\psi_j)\}$. Then,

$$E[I, S] = E[I_1, S_1] + E[I_2, S_2] + \cdots + E[I_\ell, S_\ell].$$

3.2 Improved algorithm for well-behaved π's.

In practice, π's are seldom arbitrary in real life. In this section, we consider k-flat π's as well-behaved ones.

Definition 2 *A k-flat π for $k \ll N$ comprises k piecewise constant partition, ie., for $p_0 = 0 < p_1 \cdots < p_k = N$, we have for all i, $\pi[j] = \pi[j+1]$ where $j \in [p_i, p_{i+1} - 1)$.*

Obviously any arbitrary π is k-flat for $k \leq N$, but intuitively, we will focus on much better behaved ones where $k \ll N$. We will study Problem 2 for k-flat π's. The intuition behind why k-flat π's are well-behaved is layered and can be found in the discussions in Section 5. For example, it is interesting based on compressibility of π (Problem 4) and maintenance of π under updates in sublinear space (Problem 7). Furthermore, it is natural in some situations: if the domain were time, one can imagine different importance attached to observations from different days (months) but equal importance for all observations of the same day (or month). Another natural setting is one in which N is large and we do not associate $\pi[i]$ with all i's, but only with a handful of i's that are distinguished, and let the remainder be uniformly important. If there are k distinguished importance values, π is $2k-1$-flat. Finally, this is a technically interesting special case of arbitrary π's and understanding this special case provides some insight into the structure of π's that affect the complexity of the general problem.

We will solve the case of k-flat π's in two steps. First, we will consider *k-dyadic-flats*, that is, the case where each interval $I_i = [p_i, p_{i+1})$ is dyadic. Later, we will consider the general case.

3.2.1 k-dyadic-flat π's

Consider the full binary tree T with $[0, N-1]$ as the leaves. Associate with each node u in T, the interval J_u spanned by the leaves in the subtree rooted at it. Since each I_i is dyadic, it is an interval J_u associated with some internal node u of T. Let the set of all nodes u in T that have an associated $J_u = I_i$ be denoted L; clearly $|L| = k$ and no two such nodes are the parents of each other since the k-flat intervals form a partition.

Definition 3 *Let t be the portion of tree T that includes the nodes in L and all their ancestors in T.*

Observe that since the k-dyadic-flat intervals form a partition, t is size at most $2k-1$ (k of leaves and $k-1$ internal nodes); this is crucial.

We will apply external search to the internal nodes of t and internal search to the leaves of t.

External Search. At any internal node u of t, we will consider J_u and determine $E(J_u, \ell, S)$ which is the least importance-based error for representing $\mathbf{A}[J_u]$ with any subset Λ_u of wavelets that have supports contained in J_u, $|\Lambda_u| \leq \ell$, together with the contribution for the set S of wavelets whose individual supports strictly contain J_u. As in proof of Theorem 1, we can set up the recurrence for external search. The number of such problems is $O(kB2^{\min\{k,\log N\}})$ since the number of different S can be bounded by either $\log N$, the depth of T or k, the number of internal nodes in t; each such problem can be computed in $O(\log B)$ time as before.

Internal Search. Now we focus on the internal search. Unlike in the proof of Theorem 1, we will not use an exhaustive search here. Instead, we will exploit the property of π, and be much more efficient. Formally, consider computing $E(J_u, \ell, S)$. Let $\mathbf{A}_u$ denote the signal in interval J_u. We make the following crucial observation:

Lemma 2 (Local Parseval's) *Let $\alpha_0, \alpha_1, \ldots, \alpha_{|J_u|-2}$ be the proper wavelet vectors of an interval $[0, |J_u| - 1]$ and $\alpha_{|J_u|-1}$ be the all "average" scaling vector. $E(J_u, \ell, S)$ is minimized by choosing the largest ℓ wavelet coefficients in magnitude from $\langle \mathbf{A}, \alpha_i \rangle$ for $i = 0, \ldots, |J_u| - 2$.*

Proof Sketch. The contribution of S to a $j \in J_u$ is denoted $v[j] = (\sum_{\psi_i \in S} \langle \mathbf{A}, \psi_i \rangle \psi_i)[j]$. Since all ψ_i's in S have support that contains J_u, $v[j]$'s will be identical for all $j \in J_u$. So, we have a generic v to denote the value $v[j]$ for any $j \in J_u$ and $\mathbf{v}$ to denote the vector of all v's of size J_u. Observe that $E(J_u, \ell, S)$ is minimized by choosing the ℓ wavelet coefficients ψ_i with $\mathrm{supp}(\psi_i) \in J_u$ for the signal $\mathbf{A}_u - \mathbf{v}$. Let $\mathbf{R}$ be such a representation. Observe that the ψ_i's with $\mathrm{supp}(\psi_i) \in J_u$ have a one-to-one correspondence with $\alpha_0, \alpha_1, \ldots, \alpha_{|J_u|-2}$. The difference between picking amongst the best wavelet representation of $\mathbf{A}_u - \mathbf{v}$ from α_i's that form an orthonormal basis for it, and from amongst ψ_i with $\mathrm{supp}(\psi_i) \in J_u$ is the presence of $\alpha_{|J_u|-1}$ in the orthonormal basis that is not available in determining $E(J_u, \ell, S)$ since it is not a wavelet vector for the signal $\mathbf{A}$ on $[0, N)$.

We now proceed with the proof of the lemma. We have $\mathbf{A}_u = \sum_i \langle \mathbf{A}, \alpha_i \rangle \alpha_i$ and $\mathbf{v} = \sum_i \langle \mathbf{v}, \alpha_i \rangle \alpha_i$ since α_i's are an orthonormal basis. Note further that $\langle \mathbf{v}, \alpha_i \rangle = 0$ for $i = 0, \ldots, |J_u| - 2$. We have $\mathbf{R} = \sum_{i \in \Lambda,\ |\Lambda| \leq \ell,\ \Lambda \in \{0, \ldots, |J_u|-2\}} c_i \alpha_i$ which we seek. Expanding and doing algebaric manipulations, one can conclude that (fix $\Lambda \in \{0, \ldots, |J_u| - 2\}$)

$$\|\mathbf{A}_u - \mathbf{v} - \mathbf{R}\|^2 = (\langle \mathbf{A}_u, \alpha_{|J_u|-1} \rangle - v)^2 + \sum_{i \in \Lambda} (c_i - \langle \mathbf{A}_u, \alpha_i \rangle)^2 + \sum_{i \notin \Lambda} \langle \mathbf{A}_u, \alpha_i \rangle^2$$

because all the other cross terms cancel out since $\alpha_i \perp \alpha_j$ for $i \neq j$. Now the lemma follows since last two summands do not depend on v. ∎

Local Parseval's has certain nontrivial aspects. For example, v in the proof may be positive or negative, and we do not pick the largest ℓ of the wavelet coefficients for the signal in the interval of interest, but rather the largest ℓ from those *except* the scaling vector. This form of Local Parseval's may have other applications.

The algorithm that uses the Local Parseval's for internal search is simple and takes $O(|J_u|)$ time and therefore $O(N)$ over all u's that form the leaf of t. That completes the description of the algorithm for the k-dyadic-flat case.

Theorem 2 *Problem 2 can be solved in $O(N + kB \log B 2^{\min\{k,\log N\}})$ time for k-dyadic-flat π's.*

Note that when k is small, say a constant, this algorithm takes time linear in N; for general k, the running time is $O(NkB \log B)$ which is near-linear since $k, B \ll N$. We suspect that the running time analysis can be somewhat tightened.

3.2.2 k-flat π's

When π is k-flat but not k-dyadic-flat, we can not directly follow the previous algorithm. Since any dyadic interval J_u in T may overlap *two* partitions of a k-flat π (one at either end), the external and internal search problems do not divide nicely into independent problems. Our approach here is to replace any k-flat π with $O(k \log N)$-dyadic-flat π' by replacing each constant partition in π using at most $O(\log N)$-sized partitions that are dyadic (in a straightforward way). We can conclude with a near-linear time algorithm again since $k, B \ll N$. :

Theorem 3 *Problem 2 can be solved in time $O(NkB \log N \log B)$ for k-flat π's.*

4 Extensions

We can solve several variations of our problem using a combination of internal and external search methods. For example, we can minimize other error measures such as $\epsilon^{\pi}_{\mathbf{R}} = \sum_i \pi[i]\ |\mathbf{R}[i] - \mathbf{A}[i]|$ or maximum absolute error $\max_i \pi[i]\ |\mathbf{R}[i] - \mathbf{A}[i]|$ or maximum relative error, etc. Minimizing the maximum errors has a nice history in sparse approximation (for example, see work on the pointwise convergence of hard thresholding schemes for wavelets by Terry Tao and others) and has applications in databases [7]. Our results are interesting even in the uniform importance case with these errors: for example, using our approach of coupling the external and internal searches will improve the time bounds in [7] by a $\log^2 B$ term where they find optimal B wavelets that minimizes the maximum relative error under uniform importance. We can also solve the nonuniform sparse approximation problem for other dictionaries. In particular, our work directly extends to tree-structured Haar transforms [5] which are quite general orthogonal basis vectors defined on height-balanced trees atop the signals.

We will show one interesting example of our extensions, in this case, to a dual problem.

Problem 3 *Given a bound δ such that $\max_i \pi[i]|\mathbf{R}[i] - \mathbf{A}[i]| \le \delta$, minimize the number B of wavelet synopsis terms needed to represent the signal.*

We will outline our solution for the uniform importance case; extension to the nonuniform case is straightforward. Given an interval I and set S of wavelets whose individual supports strictly contain I, consider $B(I, \delta, S)$, the least number of wavelets from Λ_I needed for representing $\mathbf{A}[I]$ such that for each $i \in I$, $|\mathbf{R}[i] - a[i]| \le \delta$. Say ψ_I is included in the set of k wavelet terms. Then,

$$B(I, \delta, S) = B(I_L, \delta, S \cup \psi_I) + B(I_R, \delta, S \cup \psi_I) + 1.$$

In the other case,

$$B(I, \delta, S) = B(I_L, \delta, S) + B(I_R, \delta, S).$$

Hence, $B(I, \delta, S)$ is the minimum of the two possibilities above. This external search method gives an algorithm that takes time $O(N^2)$.

Using internal search, we will develop a more efficient algorithm. We need the key observation: we do not have to solve $B(I, \delta, S)$ for each possible S. Instead, *it suffices to keep the maximum and minimum absolute errors for* each *size ℓ of the subset of wavelets in Λ_I*. This is because:

Lemma 3 *We have*

$$\min_{S\ ,\, i \in 2^{\Lambda_I}} \max_{j \in I} |\delta_i^j - v_S|,$$

where δ_i^j is the error in $\mathbf{A}[j]$ due to set i of wavelets chosen from Λ_I and v_S is the contribution of S to each $i \in I$, is equal to

$$\min_{S\ ,\, i \in 2^{\Lambda_I}} \max\{|(\max_{j \in I} \delta_i^j) - v_S|, |(\min_{j \in I} \delta_i^j) - v_S|\}.$$

As before, we run external search until we have intervals of size 2^j and following that, we run the internal search; the combined algorithm will take time $O(\frac{N}{2^j}\frac{N}{2^j}2^j + \frac{N}{2^j}2^{2^j})$, which, when balanced, reduces to $O(N^2/\log N)$.

Another improvement is possible. Say B^* is the optimal answer. It suffices to keep the maximum and minimum absolute errors over all B-sized subsets. Then in the internal search phase, we will spend only time $O(|I|^{B^*})$ (this relies on the monotonicity of the error with increasing wavelet terms in each subproblem at the external search phase). Then we can balance the two running times by choosing $2^j = n^{1/B^*+1}$ so the total running time is $O(N^{2-\frac{2}{B^*+1}}B^*)$. Both these improvements make the running time marginally subquadratic ($o(N^2)$) which is of some interest.

Theorem 4 *There is an $O(N^2/\log N)$ (or $O(N^{2-\frac{2}{B^*+1}}B^*)$, where B^* is the optimal number) time algorithm to solve Problem 3.*

This result for the dual problem is not only of independent interest, but also may be used to approximate the solution to the primal problem studied in [7] by binary searching with guesses of δ, which may be more efficient in certain cases.

5 Concluding Remarks

We have initiated the study of nonuniform sparse approximation with Haar wavelet basis and presented some results that are general or exploit properties of π. From a mathematical and algorithmic viewpoint, lot remains to be done in nonuniform sparse approximation. In what follows, we will list a set of open problems and provide some comments. We have chosen to discuss only those problems that we believe are relevant to understanding the role of π in sparse approximation problems, and those that have some nontrivial intuition or challenge.

5.1 Foundations Open Problems

Problem 4 *Consider some* compression measure *of π, ie., the size of π using a compression algorithm, denoted $C(\pi)$. Find fast algorithms for Problem 2 that adopts to $C(\pi)$.*

The problem above is motivated by an interest to develop faster algorithms that exploit properties of the importance vector π. In particular, we would like to study the information content of π. The uniform $\pi = 1/N$ is most compressible and arbitrary π is incompressible; they lead to the widely-studied uniform sparse approximation problem and the nonuniform sparse approximation problem studied in this paper, respectively. Problem 4 is an attempt to parameterize the complexity of the problem in terms of the information content of π as determined by a compression method. Two simple examples arise:

- If we used *run-length compression* to compress π, then π is piecewise constant with $C(\pi)$ disjoint pieces. Then our Theorem 3 solves it in near-linear time.
- If π can be summarized exactly using say k Haar wavelet coefficients, then π can be rewritten as piecewise constant functions on $O(k)$ *dyadic* pieces. Then our Theorem 2 solves it in nearly linear time.

But much better compressors (such as Lempel-Ziv) are known and Problem 4 is open for those cases.

Problem 5 *Say π is* periodic *with period p, ie., $\pi[j+p] = \pi[j]$ for all $j \in 0, \ldots, N-1$. Solve Problem 2 under this assumption.*

Again, this problem tries to exploit properties of π but from a different perspective from Problem 4. Periodic π's can be nicely compressed, but compressible π's are not strictly periodic. This problem brings out the difficulty in computing partial solutions at two disjoint portions of the signal which have identical importances, ie., for the same importance or information content of π, what is the role of different signals on the complexity of Problem 2?

Problem 6 *Given signal $\mathbf{A}[0, \ldots, N-1]$, N a power of 2, an integer $B \ll N$, and importance $\pi[0, \ldots, N-1]$, determine set Λ of wavelets with $|\Lambda| \leq B$ and coefficients d_i for $i \in \Lambda$ such that $\epsilon^{\pi}_{\mathbf{R}} = \sum_i \pi[i]\ (\mathbf{R}[i] - \mathbf{A}[i])^2$ is minimized where $\mathbf{R} = \sum_{i \in \Lambda} d_i \psi_i$.*

The problem above is more general than Problem 2 because d_i is not necessarily chosen to be $\langle \mathbf{A}, \psi_i \rangle$. In principle, choosing d_i's more effectively does not hurt the applications. If the set Λ of wavelets were *fixed*, then the choice of d_i's is a simple optimization problem taking time $O(N + B^3)$ in the worst case using the least-squares-fit approach. But the problem above is to search simultaneously over the choice of wavelets ψ_i's as well as d_i's. Note that if π was uniform, then from the orthogonality of ψ_i's, it will follow that $d_i = \langle \mathbf{A}, \psi_i \rangle$. But that does not hold for arbitrary π's.

Problem 7 *Solve Problem 2 dynamically as the workload changes under updates; we need to design a streaming algorithm to process such a stream of updates to both the workload as well as the signal, and still maintain an* approximate *B-term wavelet represenation of the current signal under the current workload using sublinear space. A simpler variation is the one in which the workload is fixed and stored explicitly.*

This problem is motivated by work in database research such as [1, 12], and is a natural extension of theory research on data stream maintenance of wavelets and histograms [8]. One way to think about it is that using sublinear space, one can summarize π's to some extent. How does that affect the complexity of approximately maintaining Haar synopses in sublinear space under updates? For example, using methods in [8], one can find near-best approximation of π using k-flats or wavelets. Does this help solve the problem above?

Finally, rather than study Problem 2, we can study the more general Problem 1.

Problem 8 *Solve Problem 1 with the dictionary of Fourier basis.*

We think this is a fairly basic question and surprisingly, perhaps because this problem was not formulated explicitly before, we do not know of an efficient algorithm for this problem from prior work. The same question applies to other dictionaries as well: Haar wavelet packets, curvelets, ridgelets, wavelets on sphere, etc. which have been studied in the past few decades motivated by different applications. Finally, from a theoretical point of view, it will be interesting to study the problem with an arbitrary $\mathcal{D}$ or $\mathcal{D}$ with some "incoherence" [9].

5.2 Technical Open Problems

Many other technical problems arise natually (study piecewise *linear* π's or study two dimensional signals) and we hope they will be the subject of future study. The following is a straightforward open problem, but listed here because it is important.

Problem 9 *Design near-linear time approximation algorithms for Problem 2.*

Haar wavelets can be managed in $O(N)$ time for the uniform π case: this partly explains their popularity in many applications. So, in order to be persuasive in exploiting the full information of nonuniform π, it will be useful to have algorithms that are competitive in their running times, say near-linear, even if they sacrifice some accuracy. Some natural approximation schemes suggest themselves such as rounding off the effect of S in $E(I, k, S)$ so a smaller number of problems are explored. It will be of interest to get new ideas for highly efficient approximation algorithms.

5.3 Applied Open Problems

From an applied point of view, finally, experimental studies matter. For example, it will be interesting to evaluate different heuristics for optimal wavelets using experiments under realistic nonuniform workloads. An interesting question is what heuristics to study. In sparse approximation theory, when one deals with non-standard problems where Parseval's does not apply, it is common to (a) use greedy methods and sometimes they give provably good bounds, or (b) transform the problem so that Parseval's may be used. In our problem, for example, (b) can be applied as follows.

Heuristic. We have

$$\epsilon^{\pi}_{\mathbf{R}} = (\mathbf{A} - \mathbf{R})^T D(\mathbf{A} - \mathbf{R}),$$

where D is the diagonal matrix with $D[i, i] = \pi[i]$ and $D[i, j] = 0$ for $i \neq j$. We can rewrite this by setting $\mathbf{A}' = D^{1/2}\mathbf{A}$ and say its approximate representation is $\mathbf{R}'$. Then, we have

$$\epsilon^{\pi}_{\mathbf{R}} = (\mathbf{A}' - \mathbf{R}')^T (\mathbf{A}' - \mathbf{R}').$$

This is in the standard form of the error and hence we can directly apply Parseval's theorem and find the B largest Haar coefficients to get optimal $\mathbf{R}'$ for $\mathbf{A}'$. This whole procedure takes only $O(N)$ time. While this is optimal in error, using this transformation to represent $\mathbf{A}$ involves an inverse transformation that is not akin to Haar wavelet decoding (eg., it involves estimating $\mathbf{A}'[j]$ and scaling it by $1/\sqrt{\pi[j]}$ because even though $\mathbf{A}'$ is represented using the Haar wavelet basis (denote it by Ψ), $\mathbf{A}$ is represented using a transformed basis (ie., $D^{-1/2}\Psi$)). ∎

Recently, another nice heuristic has been suggested that too involves transforming the basis from Haar to a suitably different weighted version [13]. Also, there are heuristics to the Haar wavelet basis case based on greedy methods. The open issue from an applied point of view is to compare these heuristics to each other and to our optimal results with Haar wavelets, and to explore the tradeoffs in practice. An interesting systems challenge is to obtain suitable π's from applications (eg database query logs).

This brings us to the concluding thought, namely, why focus on representing a signal using the Haar basis and why not one of the transformed basis if it has suitable properties for fast coding and decoding. Adapting the standard Haar basis for coding signals and making it more suitable for the workload entails no changes in the *decoding* procedure of a signal using the synopses. In fact, the decoding program does not need to know π. Hence, from a systems point of view, there is some preference to our approach. Of course from a technical point of view, our problem is of great interest because what appears to be a natural variation of the classical method from 1910 seems to pose new challenges and calls for new algorithmic strategies beyond the Parseval's theorem that has been the workhorse of sparse approximation theory since 1799. Also, this inspires the area of nonuniform sparse approximation we have articulated in this paper, which we hope will develop into a theory with a clear understanding of the complexity of the problem in terms of the structure of π and A.

6 Acknowledgements

Thanks to David Applegate for pointing me to the least squares fit method in Problem 6. Thanks to Suhas Diggavi for discussions about wavelet basis transformations.

References

[1] A. Aboulnaga and S. Chaudhuri. Self-tuning histograms: building histograms without looking at data. *Proc. SIGMOD*, 181–192, 1999.

[2] K. G. Beauchamp. *Walsh functions and their applications*. 1975.

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

[4] R. Devore and G. Lorentz. *Constructive approximation.* Springer Verlag, 1991.

[5] K. Egiazarian and J. Astola. Tree-structured Haar transforms. *Journal of Mathematical Imaging and Vision*, 16:269–279, 2002.

[6] V. Ganti, M. Lee and R. Ramakrishnan. ICICLES: Self-tuning samples for approximate query answering. *VLDB Journal*, 176–187, 2000.

[7] M. Garofalakis and A. Kumar. Deterministic wavelet thresholding for maximum error metrics. *Proc. PODS*, 2004.

[8] A. Gilbert, S. Guha, P. Indyk, Y. Kotidis, S. Muthukrishnan and M. Strauss. Fast, small space algorithms for approximate histogram maintenance. *Proc. STOC*, 2002, 389–398.

[9] A. Gilbert, S. Muthukrishnan and M. Strauss. Approximation of functions over redundant dictionaries using coherence. *Proc. ACM-SIAM SODA*, 2003.

[10] S. Guha. A Note on Wavelet Optimization. Personal email communication, August 2004.

[11] A. Haar. "Zur theorie der orthogonalen functionsysteme". *Math Annal.*, Vol 69, 331–371, 1910.

[12] V. Markl, G. Lohman and V. Raman. LEO: An automatic query optimizer for DB2. *IBM Systems Journal*, Vol 42, No 1, 2003. See also VLDB 2002 at http://www.dia.uniroma3.it/ vldbproc/.

[13] Y. Matias and D. Urieli. Optimal workload-based wavelet synopses. TR-TAU. February (revised July) 2004.

[14] Y. Matias, J. Vitter and M. Wang. Wavelet-based histograms for selectivity estimation. *Proc. ACM SIGMOD*, 448–459, 1998.

[15] S. Muthukrishnan. Workload-optimal wavelet synopsis. DIMACS Technical Report, May 2004. http://dimacs.rutgers.edu/TechnicalReports/abstracts/2004/2004-25.html

[16] M. Parseval. http://encyclopedia.thefreedictionary.com/Parseval's+theorem 1799.

[17] R. Schmidt and C. Shahabi. How to evaluate multiple range-sum queries progessively. *Proc. PODS*, 2002.