Distributed Sparse Random Projections
for Refinable Approximation

Wei Wang[*]
Department of
Electrical Engineering
and Computer Science,
University of California,
Berkeley
wangwei@eecs.
berkeley.edu

Minos Garofalakis[†]
Yahoo! Research
and Department of
Computer Science,
University of California,
Berkeley
minos@
yahoo-inc.com

Kannan Ramchandran[*]
Department of
Electrical Engineering
and Computer Science,
University of California,
Berkeley
kannanr@eecs.
berkeley.edu

ABSTRACT

Consider a large-scale wireless sensor network measuring compressible data, where n distributed data values can be well-approximated using only $k \ll n$ coefficients of some known transform. We address the problem of recovering an approximation of the n data values by querying any L sensors, so that the reconstruction error is comparable to the optimal k-term approximation. To solve this problem, we present a novel distributed algorithm based on sparse random projections, which requires no global coordination or knowledge. The key idea is that the sparsity of the random projections greatly reduces the communication cost of pre-processing the data. Our algorithm allows the collector to choose the number of sensors to query according to the desired approximation error. The reconstruction quality depends only on the number of sensors queried, enabling robust refinable approximation.

Categories and Subject Descriptors: G.1.2, C.2.4
General Terms: Algorithms
Keywords: sparse random projections, wireless sensor networks, refinable approximation, compressed sensing, AMS sketching

1. INTRODUCTION

Suppose a wireless sensor network measures data which is compressible in an appropriate transform domain, so that n data values can be well-approximated using only $k \ll n$ transform coefficients. In order to reduce power consumption and query latency, we want to pre-process the data

in the network so that only k values need to be collected to recover the data with an acceptable approximation error. Computing the deterministic transform in a distributed manner is difficult in an unreliable wireless network, requiring global coordination and knowledge of the network state. In addition, one must then locate the largest transform coefficients in the network to recover the best approximation.

There is a rich literature on using random projections to approximate functions of data. It is a well-known result from compressed sensing [1, 2] that $O(poly(k, \log n))$ random projections of the data are sufficient to recover a representation very close to the optimal approximation using k transform coefficients. Similarly, in the AMS sketching literature [7, 8, 9], random projections are used to approximate wavelet representations of streaming data. Random projections are also used in the Johnson-Lindenstrauss (JL) embedding theorems [10, 11, 12, 13] to estimate pairwise distances of high-dimensional points in a low-dimensional space. However, previous results in compressed sensing and AMS sketching rely on *dense random projection matrices*. Computing such matrices in a distributed setting would require $\Omega(n^2)$ communications, equivalent to flooding the network with data.

Our technical contributions are twofold. First, we show that $O(poly(k, \log n))$ sparse random projections are sufficient to recover a data approximation which is comparable to the optimal k-term approximation, with high probability. The expected degree of sparsity, that is, the average number of nonzeros in each random projection vector, can be $O(\log n)$. In fact, there is a trade-off between the sparsity of the random projections and the number of random projections needed.

Second, we present a distributed algorithm, based on sparse random projections, which guarantees recovery of a near-optimal approximation by querying any $O(poly(k, \log n))$ sensors. Our algorithm effectively acts as an erasure code over real numbers, generating n sparse random projection coefficients out of which any subset of $O(poly(k, \log n))$ is sufficient to decode. Since sparsity of the random projections determines the amount of communication, the communication cost can be reduced to $O(\log n)$ packets per sensor, routed to randomly selected nodes in the network. There is a corresponding trade-off between the pre-processing communication cost and the number of sensors that need to be queried to recover an approximation with acceptable error.

Our distributed algorithm has the interesting property that the decoder can choose how much or how little to query, depending on the desired approximation error. The reconstruction error of the optimal k-term approximation decreases with increasing values of k. The sensors do not need any knowledge of the data model or the transform necessary for compression, including the value of k. Sensors simply compute and store sparse random projections, which they can do in a completely decentralized manner by acting independently and randomly. Only the decoder chooses k and the number of sensors to query, along with the appropriate transform to recover the approximation. The decoder can then reconstruct the data by collecting a sufficient number of projection coefficients, from anywhere in the network. The approximation error depends only on the number of coefficients collected, and not on which sensors are queried. Therefore, distributed sparse random projections enable efficient and robust approximation with refinable error.

The remainder of the paper will be organized as follows. In Section 2, we precisely define the problem setup, the modeling assumptions, and previously known results. Section 3 presents our main results on near-optimal signal approximation using sparse random projections. In Section 4, we describe our distributed algorithm based on sparse random projections. Section 5 contains comparisons and simulation results. Finally, we give detailed proofs of the main results in Section 6 and conclude.

2. PROBLEM SETUP

We consider a wireless network of n sensors, each of which measures a real data value u_i. Suppose the aggregate data $\mathbf{u} \in \mathbb{R}^n$ is compressible, so that it can be well-approximated using $k \ll n$ coefficients of some orthonormal transform. For simplicity, we assume that each sensor stores one real value. We want to be able to query any L sensors and recover an approximation of the n data values, with reconstruction error comparable to the best k-term approximation.

2.1 Compressible Data

A well-studied phenomenon in signal processing is that many natural classes of signals, such as smooth signals with bounded derivatives and bounded variation signals, are compressible in some transform domain [15, 16]. Sensor networks measuring a smooth temperature field, for example, may efficiently represent the data using only a few large transform coefficients, which record useful structure such as average temperature and sharp temperature changes. The remaining small transform coefficients may be discarded without much loss in the total signal energy.

We consider a real data vector $\mathbf{u} \in \mathbb{R}^n$, and fix an orthonormal transform $\boldsymbol{\Psi} \in \mathbb{R}^{n \times n}$ consisting of a set of orthonormal basis vectors $\{\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_n\}$. $\boldsymbol{\Psi}$ can be, for example, a wavelet or a Fourier transform. The transform coefficients $\boldsymbol{\theta} = [\boldsymbol{\psi}_1^T \mathbf{u}, \ldots, \boldsymbol{\psi}_n^T \mathbf{u}]^T$ of the data can be ordered in magnitude, so that $|\theta|_{(1)} \geq |\theta|_{(2)} \geq \cdots \geq |\theta|_{(n)}$. The best k-term approximation keeps the largest k transform coefficients and discards the remaining as zero. The approximation error is $\|\mathbf{u} - \hat{\mathbf{u}}\|_2^2 = \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|_2^2 = \sum_{i=k+1}^{n} |\theta|_{(i)}^2$.

We now specify the model of compressible data as defined in the compressed sensing literature [1, 2]. We say that the data is compressible if the magnitude of its transform coefficients decay like a power law. That is, the ith largest
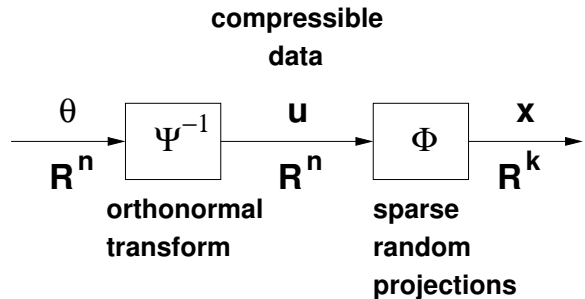

Figure 1: The compressible data model assumes that the largest k transform coefficients of θ in magnitude captures most of the signal energy.

transform coefficient satisfies

$$|\theta|_{(i)} \leq R\,i^{-1/p} \tag{1}$$

for each $1 \leq i \leq n$, where R is a constant, and $0 < p \leq 1$. Note that p controls the compressibility (or rate of decay) of the transform coefficients (i.e., smaller p implies faster decay). The approximation error obtained by taking the k largest transform coefficients and setting the remaining coefficients to zero, is then

$$\|\mathbf{u} - \hat{\mathbf{u}}\|_2 \;=\; \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}\|_2 \;\leq\; \alpha_p\,R\,k^{-1/p+1/2}$$

where α_p is a constant that only depends on p.

2.2 Random Projections

Recent results in compressed sensing [1, 2] have shown that random projections can guarantee the recovery of a near-optimal approximation of compressible data, with a very small hit in performance. Specifically, $O(k \log n)$ random projections of the data can produce an approximation with error comparable to the best approximation error using the k largest transform coefficients.

More concretely, consider the random projection matrix $\boldsymbol{\Phi} \in \mathbb{R}^{k \times n}$ containing i.i.d. entries

$$\Phi_{ij} = \begin{cases} +1 & \text{with prob. } \frac{1}{2} \\ -1 & \text{with prob. } \frac{1}{2} \end{cases} \tag{2}$$

Then the k random projections $\frac{1}{\sqrt{n}} \boldsymbol{\Phi} \mathbf{u} \in \mathbb{R}^k$ produce an approximation $\hat{\mathbf{u}}$ of the data $\mathbf{u}$ with error

$$\|\mathbf{u} - \hat{\mathbf{u}}\|_2 \;\leq\; \beta_p\,R\,(k/\log n)^{-1/p+1/2}$$

with probability of failure decaying polynomially in n. The β_p is equal to some function of p. Compressed sensing decoding is achieved by solving a linear program, which, in general, has $O(n^3)$ computational complexity [1, 2].

Random projections have also been used to recover approximate wavelet representations of streaming data in the AMS sketching literature [7, 8, 9]. The encoding random projection matrix has entries Φ_{ij} defined in (2), except only four-wise independence is required within each row. This relaxation allows the matrix to be generated pseudo-randomly and stored in small space. The decoding process estimates the largest k wavelet coefficients using random projections of the data and the wavelet bases. The sketching decoder requires $O(k^2 \log n)$ random projections to produce an approximation with error comparable to the best-k wavelet coefficients. However the decoding computational complexity
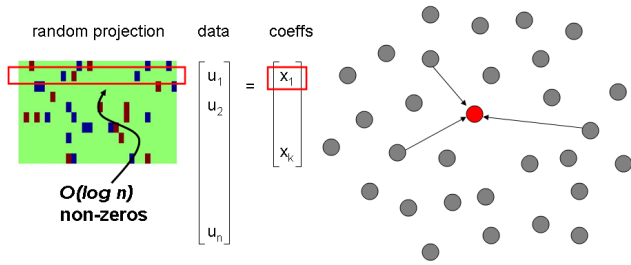
Figure 2: Sparsity of the random projection matrix leads to a more efficient distributed algorithm with fewer communications.

is reduced to $O(Ln \log n)$, where L is the number of random projections used. In some distributed applications, it would be useful for sensors or other low-powered collectors to be able to decode a coarse approximation of the data cheaply and quickly. Meanwhile, collectors with greater resources can query for more random projections and reconstruct a good approximation.

Random projections are also used for dimensionality reduction in the Johnson-Lindenstrauss (JL) embedding theorems [10, 11, 12, 13]. Any set of $n \geq d$ points can be mapped from $\mathbb{R}^d$ to $\mathbb{R}^k$ while preserving all pairwise distances within a factor of $(1 \pm \epsilon)$, where $k = O\left(\frac{\log n}{\epsilon^2}\right)$. [11, 12, 13] explore using sparsity for efficient JL embedding.

2.3 Distributed Data Processing

The random projection matrices used in both compressed sensing and sketching are dense. The key idea of this paper is that *sparse* random projections can reduce the computational complexity, and in our distributed problem setting, reduce the communication cost. Sparsity in the random projection matrix may also be exploited to reduce the decoding complexity.

Distributed compressed sensing schemes have been proposed in [6, 4, 5, 3]. The problem setups in prior works are very different from our setup. [6, 5] pose the scenario where all sensors communicate directly to a central fusion center, without any in-network communication. [6] defines a joint sparsity model on the data, and uses knowledge of this correlation structure to reduce communications from the sensors to the fusion center. [5] uses uncoded coherent transmissions through an AWGN multiple access channel to simultaneously communicate and compute random projections from the sensors to the fusion center. [4] poses the scenario where ultimately every sensor has an approximation of the network data, by using gossip algorithms to compute each random projection.

3. SPARSE RANDOM PROJECTIONS

We first summarize our main result that sparse random projections can be used to reconstruct an approximation of the data, with error very close to the optimal transform-based approximation. We then state this result more precisely in section 3.1, and give proofs in section 6.

For data in $\mathbb{R}^n$, we want to find the minimum number of sparse random projections L to recover an approximation with error comparable to the best k-term approxima-

tion. We consider an L-by-n sparse random matrix with entries that have a probability $1/s$ of being nonzero, so that on average there are n/s nonzeros per row. We show that $L = O(sM^2k^2 \log n)$ sparse random projections are sufficient, if the data satisfies a peak-to-total energy condition $\|data\|_\infty / \|data\|_2 \leq M$. This condition bounds the largest component of the data, and guarantees that the energy of the signal is not concentrated in a few elements. Intuitively, sparse random projections will not work well when the data is also very sparse. Interestingly, we can relate M to the compressibility of the data, as defined in (1). Our sparse random projections algorithm uses the low-complexity sketching decoder.

Sparsity of the random projection matrix produces an extra factor of sM^2 in the number of random projections. Therefore, there is an interesting trade-off between the number of random projections L, the average number of nonzeros n/s in the random projections, and the peak-to-total energy ratio (or compressibility) of the data M. For data compressible in the discrete Fourier transform (as in (1)) with $p = 1$, if the sparsity is $\frac{n}{s} = \log^2 n$, then $sM^2 = O(1)$. In this case, there is no hit in the number of sparse random projections needed for approximation. If the sparsity is $\frac{n}{s} = \log n$, there is a hit of $sM^2 = O(\log n)$ in the number of sparse random projections. If $\frac{n}{s} = 1$, then the hit in the number of projections is $sM^2 = O(\log^2 n)$. For more compressible data with $0 < p < 1$, if $\frac{n}{s} = 1$, then the hit in the number of sparse random projections is $sM^2 = O(1)$.

We shall see in Section 4 that this trade-off, between the sparsity of the random projections and the number of projections, will have a corresponding trade-off in pre-processing communication cost and querying latency.

3.1 Main Results

The intuition for our analysis is that sparse random projections preserve inner products within a small error, and hence we can use random projections of the data and the orthonormal bases to estimate the orthonormal transform coefficients. Thus, we can estimate all the transform coefficients to within a small error given only the sparse random projections of the data. However, we need to bound the sum squared error of our approximation over all the transform coefficients. If the data is compressible, and k of the transform coefficients are large and the others are close to zero, then we only need to accurately estimate k coefficients. The remaining small transform coefficients can be approximated as zero, incurring the same error as the best k-term approximation.

Consider the sparse random projection matrix $\boldsymbol{\Phi} \in \mathbb{R}^{L \times n}$ (where $L < n$), containing entries [11]

$$\Phi_{ij} = \sqrt{s} \begin{cases} +1 & \text{with prob. } \frac{1}{2s} \\ 0 & \text{with prob. } 1 - \frac{1}{s} \\ -1 & \text{with prob. } \frac{1}{2s} \end{cases} \tag{3}$$

We assume the entries within each row are four-wise independent, while the entries across different rows are fully independent. This limited independence assumption allows each random projection vector to be pseudo-randomly generated and stored in small space [7].

The parameter s controls the degree of sparsity of the random projections. Thus if $\frac{1}{s} = 1$, the random matrix has no sparsity; and if $\frac{1}{s} = \frac{\log n}{n}$, the expected number of nonzeros in each row of the random matrix is $\log n$.

We want to first show that, with high probability, sparse random projections preserve inner products within a small error. To do this, we demonstrate that inner products are preserved in expectation, and we show concentration about the mean using a standard Chernoff-type argument. Lemma 1 states that an estimate of the inner product between two vectors, using only the random projections of those vectors, are correct in expectation and have bounded variance.

LEMMA 1. *[12] Consider a random matrix* $\mathbf{\Phi} \in \mathbb{R}^{L \times n}$ *with entries* Φ_{ij} *satisfying the following conditions:*

$$\Phi_{ij} \text{ are 4-wise indep. in rows, indep. across rows}$$
$$\mathrm{E}[\Phi_{ij}] = 0, \quad \mathrm{E}[\Phi_{ij}^2] = 1, \quad \mathrm{E}[\Phi_{ij}^4] = s \quad (4)$$

For any two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, *denote the random projections of these vectors as* $\mathbf{x} = \frac{1}{\sqrt{L}}\mathbf{\Phi}\mathbf{u}$, $\mathbf{y} = \frac{1}{\sqrt{L}}\mathbf{\Phi}\mathbf{v} \in \mathbb{R}^L$.

$$\mathrm{E}\left[\mathbf{x}^T\mathbf{y}\right] = \mathbf{u}^T\mathbf{v}$$

$$\mathrm{Var}\left(\mathbf{x}^T\mathbf{y}\right) = \frac{1}{L}\left((\mathbf{u}^T\mathbf{v})^2 + \|\mathbf{u}\|_2^2\|\mathbf{v}\|_2^2 + (s-3)\sum_{j=1}^n u_j^2 v_j^2\right)$$

Note that Lemma 1 and all subsequent results require only the sufficient condtions (4) on the random projection matrix. The sparse random projection matrix $\mathbf{\Phi}$ defined in equation (3) satisfies the conditions (4), with the fourth moment $\mathrm{E}[\Phi_{ij}^4]$ corresponding to the sparsity parameter of the matrix s. It is interesting to note that these conditions also hold for other random projection matrices. For example, the non-sparse matrix containing Gaussian i.i.d. entries $\Phi_{ij} \sim \mathcal{N}(0,1)$ satisfies (4) with $\mathrm{E}[\Phi_{ij}^4] = 3$. Similarly, $\mathrm{E}[\Phi_{ij}^4] = 1$ for the non-sparse random projection matrix containing i.i.d. entries $\Phi_{ij} = \pm 1$ as defined in equation (2).

Theorem 1 now states that sparse random projections of the data vector and any set of n vectors can produce estimates of their inner products to within a small error. Thus, sparse random projections can produce accurate estimates for the transform coefficients of the data, which are inner products between the data and the set of orthonormal bases.

THEOREM 1. *Consider a data vector* $\mathbf{u} \in \mathbb{R}^n$ *which satisfies the condition*

$$\frac{\|\mathbf{u}\|_\infty}{\|\mathbf{u}\|_2} \leq M. \quad (5)$$

In addition, let V *be any set of* n *vectors* $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\} \subset \mathbb{R}^n$. *Suppose a sparse random matrix* $\mathbf{\Phi} \in \mathbb{R}^{L \times n}$ *satisfies the conditions (4), with sparsity parameter* s. *Let*

$$L = \begin{cases} O\left(\frac{(1+\gamma)}{\epsilon^2} sM^2 \log n\right) & \text{if } sM^2 \geq \Omega(1) \\ O\left(\frac{(1+\gamma)}{\epsilon^2} \log n\right) & \text{if } sM^2 \leq O(1) \end{cases}$$

Then, with probability at least $1 - n^{-\gamma}$, *the random projections* $\frac{1}{\sqrt{L}}\mathbf{\Phi}\mathbf{u}$ *and* $\frac{1}{\sqrt{L}}\mathbf{\Phi}\mathbf{v}_i$ *can produce an estimate* $\hat{a}_i$ *for* $\mathbf{u}^T\mathbf{v}_i$ *satisfying*

$$|\hat{a}_i - \mathbf{u}^T\mathbf{v}_i| \leq \epsilon\|\mathbf{u}\|_2\|\mathbf{v}_i\|_2$$

for all $i = 1, \ldots, n$.

Theorem 2 states our main result, namely, that sparse random projections can produce a data approximation with error comparable to the best k-term approximation with high probabiliy.

THEOREM 2. *Suppose data* $\mathbf{u} \in \mathbb{R}^n$ *satisfies condition (5), and a sparse random matrix* $\mathbf{\Phi} \in \mathbb{R}^{L \times n}$ *satisfies conditions (4), with*

$$L = \begin{cases} O\left(\frac{(1+\gamma)}{\epsilon^2\eta^2} sM^2 k^2 \log n\right) & \text{if } sM^2 \geq \Omega(1) \\ O\left(\frac{(1+\gamma)}{\epsilon^2\eta^2} k^2 \log n\right) & \text{if } sM^2 \leq O(1) \end{cases} \quad (6)$$

Let $\mathbf{x} = \frac{1}{\sqrt{L}}\mathbf{\Phi}\mathbf{u}$. *Consider an orthonormal transform* $\mathbf{\Psi} \in \mathbb{R}^{n \times n}$ *and the corresponding transform coefficients* $\boldsymbol{\theta} = \mathbf{\Psi}\mathbf{u}$. *If the* k *largest transform coefficients in magnitude gives an approximation with error* $\|\mathbf{u} - \hat{\mathbf{u}}_{opt}\|_2^2 \leq \eta\|\mathbf{u}\|_2^2$, *then given only* $\mathbf{x}$, $\mathbf{\Phi}$, *and* $\mathbf{\Psi}$, *one can produce an approximation with error*

$$\|\mathbf{u} - \hat{\mathbf{u}}\|_2^2 \leq (1+\epsilon)\eta\|\mathbf{u}\|_2^2$$

with probability at least $1 - n^{-\gamma}$.

The sufficient condition we place on the data (5) is to bound the peak-to-total energy of the data. This guarantees that the signal energy is not concentrated in a small number of components. Intuitively, if the data is smooth in the spatial domain, then it will be compressible in the transform domain. As the following lemma shows, we can precisely relate condition (5) on the data to the compressibility of the data as defined in (1).

LEMMA 2. *If data* $\mathbf{u}$ *is compressible in the discrete Fourier transform as in (1) with compressibility parameter* p, *then*

$$\frac{\|\mathbf{u}\|_\infty}{\|\mathbf{u}\|_2} \leq M = \begin{cases} O(\frac{\log n}{\sqrt{n}}) & \text{if } p = 1 \\ O(\frac{1}{\sqrt{n}}) & \text{if } 0 < p < 1 \end{cases} \quad (7)$$

4. DISTRIBUTED ALGORITHM

We now describe an algorithm by which the n sensors of a wireless network each measures a data value u_i, and each computes and stores one sparse random projection of the aggregate data $\mathbf{u}$. Consider an $n \times n$ sparse random matrix $\mathbf{\Phi}$ with entries as defined in (3). For concreteness, let the probability of a nonzero entry be $\frac{1}{s} = \frac{\log n}{n}$. Each sensor will compute and store the inner product $\sum_{j=1}^n \Phi_{ij}u_j$ between the aggregate data $\mathbf{u}$ and one row of $\mathbf{\Phi}$. We think of this as generating a bipartite graph between the n data nodes and the n encoding nodes (see Figure 3).

When the entries of $\mathbf{\Phi}$ are independent and identically distributed, they can be generated at different sensor locations without any coordination between the sensors. To compute one random projection coefficient, every sensor j locally generates a random variable Φ_{ij}. If that random variable is zero the sensor does nothing, and if it's nonzero the sensor sends the product of Φ_{ij} with its own data u_j to one receiver sensor i. The receiver simply stores the sum of everything it receives, which is equal to the random projection coefficient $\sum_{j=1}^n \Phi_{ij}u_j$. This process is repeated until every sensor has stored a random projection coefficient. Thus, computation
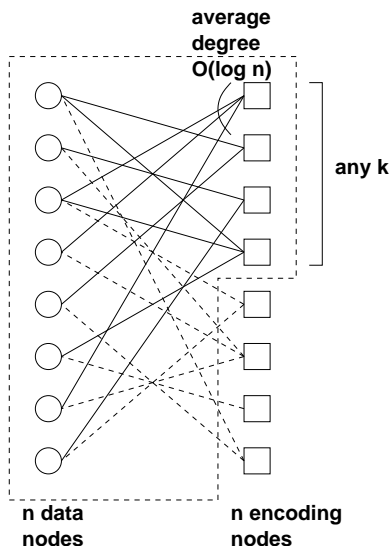
average degree O(log n)

any k

n data nodes

n encoding nodes

Figure 3: Every sensor stores a sparse random projection, so that a data approximation can be reconstructed by collecting coefficients from any k out of n sensors.

of the sparse random projections can be achieved in a decentralized manner with the following *push-based* algorithm.

Distributed Algorithm I:

- Each data node j generates a set of independent random variables $\{\Phi_{1j}, \ldots, \Phi_{nj}\}$. For each i, if $\Phi_{ij} \neq 0$, then data node j sends to encoding node i the value $\Phi_{ij}u_j$. Repeat for all $1 \leq j \leq n$.

- Each encoding node i computes and stores the sum of the values it receives, which is equal to $\sum_{j=1}^{n} \Phi_{ij}u_j$. Repeat for all $1 \leq i \leq n$.

Since the probability that $\Phi_{ij} \neq 0$ is $\frac{1}{s} = \frac{\log n}{n}$, each sensor independently and randomly sends its data to on average $O(\log n)$ sensors. Now, the decoder can query *any* $L = O(poly(k, \log n))$ sensors in the network and obtain $\mathbf{\Phi_{L \times n} u}$, where $\mathbf{\Phi_{L \times n}}$ is the matrix containing L rows of $\mathbf{\Phi} \in \mathbb{R}^{n \times n}$. By Theorem 2, the decoder can then use $\mathbf{x} = \frac{1}{\sqrt{L}}\mathbf{\Phi_{L \times n} u}$, $\mathbf{\Phi_{L \times n}}$, and $\mathbf{\Psi}$ to recover a near-optimal approximation of the data $\mathbf{u}$. The decoding algorithm proceeds as described in the proofs of Theorems 1 and 2.

4.1 Alternate Algorithm for Limited Independence

We present an alternate, *pull-based*, distributed algorithm, which takes greater advantage of the limited independence of the sparse random projections. Each sensor i locally generates a set of four-wise independent random variables, corresponding to one row of the sparse random projection matrix. If a random variable Φ_{ij} is nonzero, sensor i sends a request for data to the associated data node j. Sensor j then sends its data u_j back to sensor i, who uses all the data thus collected to compute its random projection coefficient. Therefore, different sensors still act with complete independence.

Distributed Algorithm II:

- Each encoding node i generates a set of four-wise independent random variables $\{\Phi_{i1}, \ldots, \Phi_{in}\}$. For each j, if $\Phi_{ij} \neq 0$, then encoding node i sends a request for data to node j.

- If data node j receives a request for data from encoding node i, node j sends the value u_j to node i.

- Encoding node i computes and stores $\sum_{j=1}^{n} \Phi_{ij}u_j$ using the values it receives. Repeat for all $1 \leq i \leq n$.

Since the average number of nonzeros per row of the sparse random projection matrix $\mathbf{\Phi}$ is $\frac{n}{s} = \log n$, the expected communication cost is still $O(\log n)$ packets per sensor, routed to random nodes. Algorithm II has twice the communication cost of Algorithm I, but the four-wise independence in Algorithm II allows each sensor to store a sparse random projection vector in constant rather than $poly(\log n)$ space. This further decreases the querying overhead cost for the collector seeking to reconstruct an approximation.

Both algorithms we described above perform a completely decentralized computation of n sparse random projections of the n distributed data values. In the end, collecting any subset of $O(poly(k, \log n))$ sparse random projections will guarantee near-optimal signal recovery. Thus, our algorithms enables ubiquitous access to a compressed approximation of the data in a sensor network.

4.2 Trading-off Pre-processing Communication and Query Latency

In Section 3, we described the trade-off between the sparsity of the random projection matrix and the number of random projections needed for the desired approximation error. By Theorem 2, when the probability of a nonzero entry in the projection matrix is $\frac{1}{s}$, the number of projections is $O(sM^2k^2 \log n)$. In our distributed algorithms, the average number of packets transmitted per sensor is $O(\frac{n}{s})$, while the number of sensors that need to be queried to recover an approximation is $O(sM^2k^2 \log n)$. The average computation cost per sensor is also $O(\frac{n}{s})$. Therefore, there is a trade-off between the amount of work performed by the sensors to pre-process the data in the network, and the number of sensors the decoder needs to query. Increasing the sparsity of the random projections, decreases the pre-processing communication, but potentially increases the latency to recover a data approximation.

5. COMPARISONS AND SIMULATIONS

In this section, we give a numeric example comparing the approximation of piecewise polynomial data using wavelet transforms, sparse random projections, and the non-sparse schemes of AMS sketching and compressed sensing. We know analytically that compressed sensing requires only $O(k \log n)$ random projections to obtain an approximation error comparable to the best k-term approximation, while sketching requires $O(k^2 \log n)$. However, the compressed sensing decoding has $O(n^3)$ computational complexity while the sketching decoding complexity is $O(Ln \log n)$, where L is the number of random projections used. The low decoding
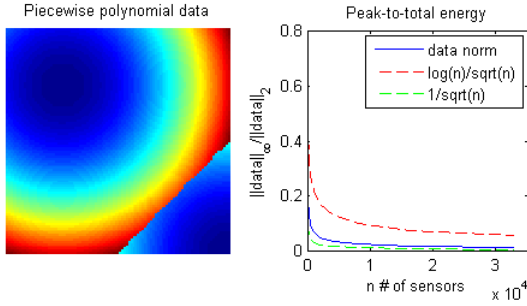
Figure 4: (a) Piecewise polynomial data. (b) Peak-to-total energy condition on data.
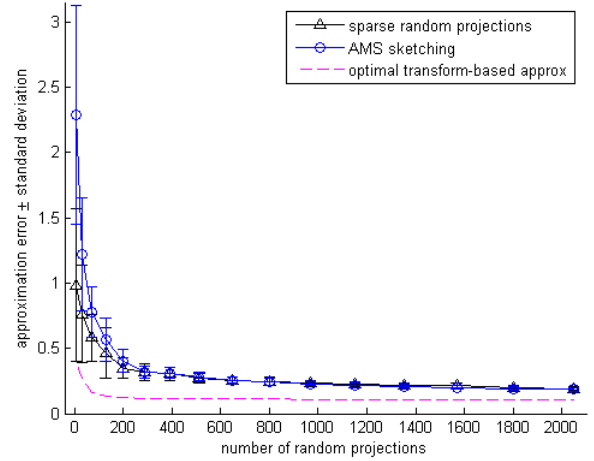

Figure 5: A comparison of the approximation error of piecewise polynomial data using sparse random projections, non-sparse AMS sketching, and optimal Haar wavelet based approximation. The relative approximation error of the data $\frac{\|\mathbf{u}-\hat{\mathbf{u}}\|_2^2}{\|\mathbf{u}\|_2^2}$ is plotted versus the number of random projections $L = k^2 \log n$, for $n = 2048$ sensors. The error bars show the standard deviation of the approximation error.

complexity would make it possible for sensors and other low-powered collectors to query and decode a coarse approximation of the data cheaply and quickly. Collectors with greater resources can still query more sensors and recover a better approximation. Our sparse random projections method uses the low-complexity sketching decoder.

We have seen theoretically that there is a trade-off between the sparsity of the random projections and the number of random projections needed for a good approximation. The degree of sparsity corresponds to the number of packets per sensor that must be transmitted in the pre-processing stage. Sparse random projections can thus reduce the communication cost per sensor from $O(n)$ to $O(\log n)$ when compared to the non-sparse schemes.

We now examine experimentally the effect of the sparsity of the random projections on data approximation. In our experimental setup, n sensors are placed randomly on a unit square, and measure piecewise polynomial data with two second-order polynomials separated by a line discontinuity, as shown in Figure 4 (a). In Figure 4 (b), we verified that the peak-to-total energy condition (5) on the data is satisfied.

Figure 5 compares the approximation error of sparse random projections to non-sparse AMS sketching and the optimal transform-based approximation. The mean approximation error using sparse random projections is as good as the non-sparse random projections, and very close to the optimal k-term approximation. However, the standard deviation of the approximation error increases with greater sparsity.

Figure 6 compares the approximation using sparse random projections for varying degrees of sparsity, along with the non-sparse schemes of sketching and compressed sensing. Sparse random projections with $O(\log n)$ nonzeros perform as well as non-sparse sketching, while sparse random projections with $O(1)$ nonzeros perform slightly worse. As we would expect from the analysis, the compressed sensing decoder obtains better approximation error than the sketching decoder for the same number of random projections. But, the compressed sensing decoder has a higher computational complexity, which was appreciable in our simulations.
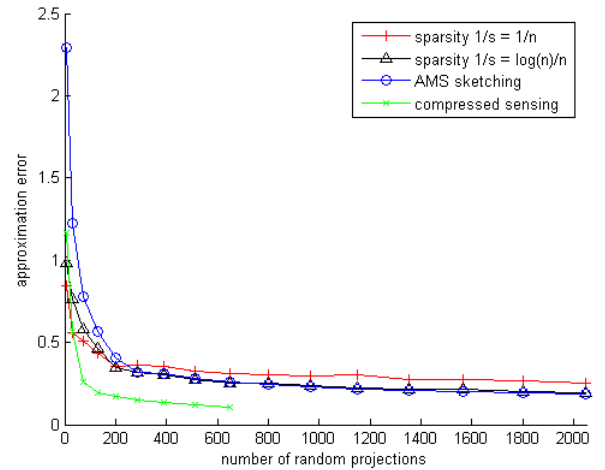

Figure 6: Effect of the sparsity of the random projections on approximation error. Varying degrees of sparsity in the sparse random projections are compared against the non-sparse projections in AMS sketching and compressed sensing. The relative approximation error of the data $\frac{\|\mathbf{u}-\hat{\mathbf{u}}\|_2^2}{\|\mathbf{u}\|_2^2}$ is plotted versus the number of random projections L, for $n = 2048$ sensors. The average number of nonzeros in the sparse random projections is n/s.
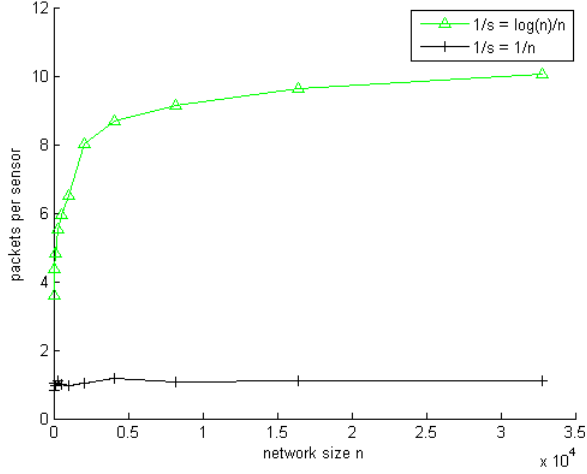
Figure 7: **Communication cost for sparse random projections with varying degrees of sparsity. In comparison, compressed sensing and sketching both require** $O(n)$ **packets per sensor.**

Finally, Figure 7 shows the communication cost of distributed sparse random projections for varying degrees of sparsity. Both compressed sensing and sketching require $O(n)$ packets per sensor to compute the dense random projections in a network of size n. Sparse random projections greatly reduce the overall communication cost.

6. PROOFS

PROOF **Lemma 1**. Let Φ_{ij} satisfy the conditions in (4), and define the random variables

$$w_i = \left(\sum_{j=1}^{n} u_j \Phi_{ij} \right) \left(\sum_{j=1}^{n} v_j \Phi_{ij} \right)$$

so that $w_1, \ldots, w_L$ are independent. Further, define the random variable $z = \mathbf{x}^T \mathbf{y} = \frac{1}{L} \sum_{i=1}^{L} w_i$.

$$
\begin{aligned}
E[w_i] &= E \left[\sum_{j=1}^{n} u_j v_j \Phi_{ij}^2 + \sum_{l \neq m} u_l v_m \Phi_{il} \Phi_{im} \right] \\
&= \sum_{j=1}^{n} u_j v_j E[\Phi_{ij}^2] + \sum_{l \neq m} u_l v_m E[\Phi_{il}] E[\Phi_{im}] \\
&= \mathbf{u}^T \mathbf{v}
\end{aligned}
$$

Thus $E[z] = \mathbf{u}^T \mathbf{v}$. Similarly, we can compute the second moments,

$$
\begin{aligned}
E[w_i^2] &= E \left[\left(\sum_{j=1}^{n} u_j v_j \Phi_{ij}^2 \right)^2 + \left(\sum_{l \neq m} u_l v_m \Phi_{il} \Phi_{im} \right)^2 \right. \\
&\quad \left. + 2 \left(\sum_{j=1}^{n} u_j v_j \Phi_{ij}^2 \right) \left(\sum_{l \neq m} u_l v_m \Phi_{il} \Phi_{im} \right) \right] \\
&= \sum_{j=1}^{n} u_j^2 v_j^2 E[\Phi_{ij}^4] + 2 \sum_{l<m} u_l v_l u_m v_m E[\Phi_{il}^2] E[\Phi_{im}^2] \\
&\quad + \sum_{l \neq m} u_l^2 v_m^2 E[\Phi_{il}^2] E[\Phi_{im}^2] \\
&\quad + 2 \sum_{l<m} u_l v_m u_m v_l E[\Phi_{il}^2] E[\Phi_{im}^2] \\
&= s \sum_{j=1}^{n} u_j^2 v_j^2 + 2 \sum_{l \neq m} u_l v_l u_m v_m + \sum_{l \neq m} u_l^2 v_m^2 \\
&= 2 \left(\sum_{j=1}^{n} u_j^2 v_j^2 + \sum_{l \neq m} u_l v_l u_m v_m \right) \\
&\quad + \left(\sum_{j=1}^{n} u_j^2 v_j^2 + \sum_{l \neq m} u_l^2 v_m^2 \right) + (s-3) \sum_{j=1}^{n} u_j^2 v_j^2 \\
&= 2(\mathbf{u}^T \mathbf{v})^2 + \|\mathbf{u}\|_2^2 \|\mathbf{v}\|_2^2 + (s-3) \sum_{j=1}^{n} u_j^2 v_j^2
\end{aligned}
$$

$$Var(w_i) = (\mathbf{u}^T \mathbf{v})^2 + \|\mathbf{u}\|_2^2 \|\mathbf{v}\|_2^2 + (s-3) \sum_{j=1}^{n} u_j^2 v_j^2$$

$$
\begin{aligned}
Var(z) &= \frac{1}{L^2} \sum_{i=1}^{L} Var(w_i) \\
&= \frac{1}{L} \left((\mathbf{u}^T \mathbf{v})^2 + \|\mathbf{u}\|_2^2 \|\mathbf{v}\|_2^2 + (s-3) \sum_{j=1}^{n} u_j^2 v_j^2 \right)
\end{aligned}
$$

$\square$

PROOF **Theorem 1**. Fix any two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^n$, with $\|\mathbf{u}\|_\infty / \|\mathbf{u}\|_2 \leq M$. Define positive integers L_1 and L_2, which we will determine, and set $L = L_1 L_2$. Partition the $L \times n$ matrix $\mathbf{\Phi}$ into L_2 matrices $\{\mathbf{\Phi}_1, \ldots, \mathbf{\Phi}_{L_2}\}$, each of size $L_1 \times n$. The corresponding random projections are $\{\mathbf{x}_1 = \frac{1}{\sqrt{L_1}} \mathbf{\Phi}_1 \mathbf{u}, \ldots, \mathbf{x}_{L_2} = \frac{1}{\sqrt{L_1}} \mathbf{\Phi}_{L_2} \mathbf{u}\}$, and $\{\mathbf{y}_1 = \frac{1}{\sqrt{L_1}} \mathbf{\Phi}_1 \mathbf{v}, \ldots, \mathbf{y}_{L_2} = \frac{1}{\sqrt{L_1}} \mathbf{\Phi}_{L_2} \mathbf{v}\}$.

Define the independent random variables $z_1, \ldots, z_{L_2}$, where $z_l = \mathbf{x}_l^T \mathbf{y}_l$. Applying Lemma 1 to each z_l, we find that $E[z_l] = \mathbf{u}^T \mathbf{v}$ and

$$Var(z_l) = \frac{1}{L_1} \left((\mathbf{u}^T \mathbf{v})^2 + \|\mathbf{u}\|_2^2 \|\mathbf{v}\|_2^2 + (s-3) \sum_{j=1}^{n} u_j^2 v_j^2 \right).$$

Thus, by the Chebyshev inequality

$$P(|z_l - \mathbf{u}^T\mathbf{v}| \geq \epsilon\|\mathbf{u}\|_2\|\mathbf{v}\|_2)$$

$$\leq \frac{Var(z_l)}{\epsilon^2\|\mathbf{u}\|_2^2\|\mathbf{v}\|_2^2}$$

$$= \frac{1}{\epsilon^2 L_1}\left(\frac{(\mathbf{u}^T\mathbf{v})^2}{\|\mathbf{u}\|_2^2\|\mathbf{v}\|_2^2} + \frac{\|\mathbf{u}\|_2^2\|\mathbf{v}\|_2^2}{\|\mathbf{u}\|_2^2\|\mathbf{v}\|_2^2} + (s-3)\frac{\sum_{j=1}^n u_j^2 v_j^2}{\|\mathbf{u}\|_2^2\|\mathbf{v}\|_2^2}\right)$$

$$\leq \frac{1}{\epsilon^2 L_1}\left(1 + 1 + s\frac{M^2\|\mathbf{u}\|_2^2\sum_{j=1}^n v_j^2}{\|\mathbf{u}\|_2^2\|\mathbf{v}\|_2^2}\right)$$

$$= \frac{1}{\epsilon^2 L_1}\left(2 + sM^2\right) \quad \triangleq \quad p$$

where in line 3 we used the fact that the data is component-wise upper bounded $\|\mathbf{u}\|_\infty \leq M\|\mathbf{u}\|_2$. Thus we can obtain a constant probability p by setting $L_1 = O(\frac{2+sM^2}{\epsilon^2})$.

Now we define the estimate $\hat{a}$ as the median of the independent random variables $z_1, \ldots, z_{L_2}$, each of which lies outside of the tolerable approximation interval with probability p. If the event that at least half of the z_l's are outside the tolerable interval occurs with arbitrarily small probability, then the median $\hat{a}$ is within the tolerable interval. Formally, let ζ_l be the indicator random variable of the event that $\{|z_l - \mathbf{u}^T\mathbf{v}| \geq \epsilon\|\mathbf{u}\|_2\|\mathbf{v}\|_2\}$, which occurs with probability p. Furthermore let $\zeta = \sum_{l=1}^{L_2}\zeta_l$ be the number of z_l's that lie outside the tolerable interval, where $E[\zeta] = L_2 p$. So, we can set p to be a constant less than $1/2$, say $p = 1/4$, and apply the Chernoff bound

$$P\left(\zeta > (1+c)\frac{L_2}{4}\right) < e^{-c^2 L_2/12}$$

where $0 < c < 1$ is some constant.

Thus, for any pair of vectors $\mathbf{u}$ and $\mathbf{v}_i \in \{\mathbf{v}_1, \ldots, \mathbf{v}_n\} \subset \mathbb{R}^n$, the random projections $\frac{1}{L}\boldsymbol{\Phi}\mathbf{u}$ and $\frac{1}{L}\boldsymbol{\Phi}\mathbf{v}_i$ produce an estimate $\hat{a}_i$ for $\mathbf{u}^T\mathbf{v}_i$ that lies outside the tolerable approximation interval with probability at most $e^{-c^2 L_2/12}$. Taking the union bound over all such of vectors, the probability that at least one estimate $\hat{a}_i$ lies outside the tolerable interval is upper bounded by $p_e \leq ne^{-c^2 L_2/12}$. Setting $L_1 = O(\frac{2+sM^2}{\epsilon^2})$ obtains $p = 1/4$, and setting $L_2 = O((1+\gamma)\log n)$ obtains $p_e \leq n^{-\gamma}$ for some constant $\gamma > 0$. Therefore for $L = L_1 L_2 = O\left(\frac{(1+\gamma)}{\epsilon^2}(2+sM^2)\log n\right)$, the random projection $\boldsymbol{\Phi}: \mathbb{R}^n \to \mathbb{R}^L$ can preserve all pairwise inner products within an approximation error ϵ, with probability at least $1 - n^{-\gamma}$. If $sM^2 \geq \Omega(1)$, then $L = O\left(\frac{(1+\gamma)}{\epsilon^2}sM^2\log n\right)$. If $sM^2 \leq O(1)$, then $L = O\left(\frac{(1+\gamma)}{\epsilon^2}\log n\right)$

$\square$

PROOF **Theorem 2**. Fix an orthonormal transform $\boldsymbol{\Psi}$ consisting of n basis vectors $\{\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_n\} \subset \mathbb{R}^n$. Let $\boldsymbol{\theta} = [\mathbf{u}^T\boldsymbol{\psi}_1, \ldots, \mathbf{u}^T\boldsymbol{\psi}_n]^T$. If we order the transform coefficients $\boldsymbol{\theta}$ in decreasing magnitude, $|\theta|_{(1)} \geq |\theta|_{(2)} \geq \cdots \geq |\theta|_{(n)}$, then the approximation error by taking the largest k coefficients in magnitude, and setting the remaining coefficients to zero, is $\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{opt}\|_2^2 = \sum_{i=k+1}^n |\theta|_{(i)}^2$. Assume that $\|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{opt}\|_2^2 \leq \eta\|\boldsymbol{\theta}\|_2^2$.

Suppose data $\mathbf{u}$ satisfies condition (5), and a random matrix $\boldsymbol{\Phi}$ satisfies conditions (4), with positive integer $L = O\left(\frac{(1+\gamma)}{\alpha^2}sM^2\log n\right)$ if $sM^2 \geq \Omega(1)$, $L = O\left(\frac{(1+\gamma)}{\alpha^2}\log n\right)$

otherwise. Then by Theorem 1, the random projections $\frac{1}{\sqrt{L}}\boldsymbol{\Phi}\mathbf{u}$ and $\{\frac{1}{\sqrt{L}}\boldsymbol{\Phi}\boldsymbol{\psi}_1, \ldots, \frac{1}{\sqrt{L}}\boldsymbol{\Phi}\boldsymbol{\psi}_n\}$ produces (w.h.p.) estimates $\{\hat{\theta}_1, \ldots, \hat{\theta}_n\}$, each satisfying

$$|\hat{\theta}_i - \theta_i| \leq \alpha\|\boldsymbol{\theta}\|_2$$

where we plugged in $\|\boldsymbol{\psi}_i\|_2 = 1$ and $\|\mathbf{u}\|_2 = \|\boldsymbol{\theta}\|_2$ by orthonormality. By triangle inequality, $\left||\hat{\theta}_i| - |\theta_i|\right| \leq |\hat{\theta}_i - \theta_i|$, so the above condition implies that

$$|\theta_i| - \alpha\|\boldsymbol{\theta}\|_2 \leq |\hat{\theta}_i| \leq |\theta_i| + \alpha\|\boldsymbol{\theta}\|_2 \quad \forall i. \quad (8)$$

Order the estimates $\hat{\boldsymbol{\theta}}$ in decreasing magnitude $|\hat{\theta}|_{(1)} \geq |\hat{\theta}|_{(2)} \geq \cdots \geq |\hat{\theta}|_{(n)}$. We define our approximation $\tilde{\boldsymbol{\theta}}$ as keeping the k largest components of $\hat{\boldsymbol{\theta}}$ in magnitude, and setting the remaining components to zero. Let $\tilde{\Omega}$ be the index set of the k largest estimates $\hat{\theta}_i$'s which we keep (and thus $\tilde{\Omega}^C$ is the index set of the estimates we set to zero). Let Ω be the index set of the k largest transform coefficients θ_i's.

$$\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2^2 = \sum_{i\in\tilde{\Omega}}|\theta_i - \hat{\theta}_i|^2 + \sum_{i\in\tilde{\Omega}^C}|\theta_i|^2$$

$$\leq k\alpha^2\|\boldsymbol{\theta}\|_2^2 + \sum_{i\in\tilde{\Omega}^C}|\theta_i|^2$$

In the ideal case, $\tilde{\Omega} = \Omega$ (or equivalently $\tilde{\Omega}^C = \Omega^C$), in which event $\sum_{i\in\tilde{\Omega}^C}|\theta_i|^2 = \sum_{i\in\Omega^C}|\theta_i|^2$. If $\tilde{\Omega} \neq \Omega$, that means that we chose to keep the estimate of a transform coefficient which was not one of the k largest, and consequently we set to zero the estimate of a coefficient which was in the k largest. So there exists some $i \in \tilde{\Omega}, i \notin \Omega, j \notin \tilde{\Omega}, j \in \Omega$. This implies that $|\hat{\theta}_i| > |\hat{\theta}_j|$, but $|\theta_i| < |\theta_j|$. Since the estimates are within a $\pm\alpha\|\boldsymbol{\theta}\|_2$ interval around the transform coefficients (by (8)), this confusion can only happen if $|\theta_j| - |\theta_i| \leq 2\alpha\|\boldsymbol{\theta}\|_2$. Furthermore, $|\theta_j|^2 + |\theta_i|^2 \leq \|\boldsymbol{\theta}\|_2^2$ implies that $|\theta_j| + |\theta_i| \leq \sqrt{3}\|\boldsymbol{\theta}\|_2$. Thus $|\theta_j|^2 - |\theta_i|^2 = (|\theta_j| - |\theta_i|)(|\theta_j| + |\theta_i|) \leq 2\sqrt{3}\alpha\|\boldsymbol{\theta}\|_2^2$. For each time this confusion happens, we get an additional error of $+|\theta_j|^2 - |\theta_i|^2$, and this confusion can happen at most k times. Therefore,

$$\sum_{i\in\tilde{\Omega}^C}|\theta_i|^2 \leq \sum_{i\in\Omega^C}|\theta_i|^2 + k(2\sqrt{3}\alpha\|\boldsymbol{\theta}\|_2^2)$$

$$\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2^2 \leq k\alpha^2\|\boldsymbol{\theta}\|_2^2 + 2\sqrt{3}k\alpha\|\boldsymbol{\theta}\|_2^2 + \sum_{i\in\Omega^C}|\theta_i|^2$$

$$= k\alpha^2\|\boldsymbol{\theta}\|_2^2 + 2\sqrt{3}k\alpha\|\boldsymbol{\theta}\|_2^2 + \|\boldsymbol{\theta} - \hat{\boldsymbol{\theta}}_{opt}\|_2^2$$

$$\leq k\alpha^2\|\boldsymbol{\theta}\|_2^2 + 2\sqrt{3}k\alpha\|\boldsymbol{\theta}\|_2^2 + \eta\|\boldsymbol{\theta}\|_2^2$$

Setting $k\alpha^2\|\boldsymbol{\theta}\|_2^2 + 2\sqrt{3}k\alpha\|\boldsymbol{\theta}\|_2^2 = \delta\|\boldsymbol{\theta}\|_2^2$ and solving for the positive root, we find that $\alpha = -\sqrt{3} + \sqrt{3 + \frac{\delta}{k}} = O(\frac{\delta}{k})$.

$$\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2^2 \leq \delta\|\boldsymbol{\theta}\|_2^2 + \eta\|\boldsymbol{\theta}\|_2^2$$

$$= \left(1 + \frac{\delta}{\eta}\right)\eta\|\boldsymbol{\theta}\|_2^2$$

Let $\epsilon = \frac{\delta}{\eta}$, so that $\alpha = O(\frac{\epsilon\eta}{k})$. Therefore, the number of random projections we need is $L = O\left(\frac{(1+\gamma)}{\alpha^2}sM^2\log n\right) = O\left(\frac{(1+\gamma)}{\epsilon^2\eta^2}sM^2k^2\log n\right)$ if $sM^2 \geq \Omega(1)$, and $L = O\left(\frac{(1+\gamma)}{\epsilon^2\eta^2}k^2\log n\right)$ if $sM^2 \leq O(1)$. $\square$

PROOF **Lemma 2**. By the definition of the (orthonormal) inverse discrete Fourier transform

$$|u_i| \le \frac{1}{\sqrt{n}} \sum_{m=0}^{n-1} |\theta_m| \left| \mathrm{e}^{\mathrm{j}\frac{2\pi mi}{n}} \right| = \frac{1}{\sqrt{n}} \sum_{m=0}^{n-1} |\theta_m| = \frac{1}{\sqrt{n}} \|\boldsymbol{\theta}\|_1$$

for $i = 0, \ldots, n-1$. Thus $\|\mathbf{u}\|_\infty \le \frac{1}{\sqrt{n}} \|\boldsymbol{\theta}\|_1$. For p-compressible signals, the DFT coefficients obey a power law decay as in (1), then $\|\boldsymbol{\theta}\|_1 \le R \sum_{i=1}^n i^{-1/p}$. For $p = 1$, the summation is a Harmonic series, which diverges slowly like $O(\log n)$. For $0 < p < 1$, the summation is a p-series (or Riemann zeta function) which converges. $\sum_{i=1}^n i^{-1/p} \le 1 + \int_1^n x^{-1/p} \, \mathrm{d}x = 1 + \left(\frac{1}{1/p-1} \right) \left(1 - \frac{1}{n^{1/p-1}} \right)$, which is upper bounded by a constant that depends only on p. Therefore, if the data is compressible with $p = 1$, then $\|\boldsymbol{\theta}\|_1 = O(\log n)$, and $\|\mathbf{u}\|_\infty = O\left(\frac{\log n}{\sqrt{n}} \right)$. If $0 < p < 1$, then $\|\boldsymbol{\theta}\|_1 = O(1)$, and $\|\mathbf{u}\|_\infty = O(\frac{1}{\sqrt{n}})$. Similarly, we can verify that compressible signals have finite energy. By orthonormality, $\|\mathbf{u}\|_2^2 = \|\boldsymbol{\theta}\|_2^2 \le R^2 \sum_{i=1}^n i^{-2/p}$, and $\int_1^{n+1} x^{-2/p} \, \mathrm{d}x \le \sum_{i=1}^n i^{-2/p} \le 1 + \int_1^n x^{-2/p} \, \mathrm{d}x$.

$\square$

7. CONCLUSIONS AND FUTURE WORK

We have proposed distributed sparse random projections and shown how they can enable reliable and refinable access to data approximations. Sensors store sparse random projections of the data, which allows the collector to recover a data approximation by querying a sufficient number of sensors from anywhere in the network. The sensors operate without coordination to compute independent random projections. The decoder has control over the approximation error by choosing the number of sensors it queries. We presented a trade-off between the communication cost to pre-process the data in the network, and the query latency to obtain the desired approximation error. We have shown that this trade-off can be controlled by the sparsity of the random projections.

As future work, our scheme can be applied to a nested tiling of geographic areas in a multiresolution manner, so that an approximation of a local region can be recovered by querying any sensors in that region. We will also study scenarios where information is queried only from a set of boundary sensors or collector nodes. Finally, the ideas presented in this paper can be extended to jointly compress data along both the spatial and temporal dimensions.

8. REFERENCES

[1] E. Candes and T. Tao. Near Optimal Signal Recovery From Random Projections: Universal Encoding Strategies. *IEEE Transactions on Information Theory*, 52(12), pp. 5406-5425, December 2006.

[2] D. Donoho. Compressed Sensing. *IEEE Transactions on Information Theory*, 52(4), pp. 1289-1306, April 2006.

[3] http://www.dsp.ece.rice.edu/CS/

[4] M. Rabbat, J. Haupt, A. Singh, and R. Nowak. Decentralized Compression and Predistribution via Randomized Gossiping. *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*, 2006.

[5] W. Bajwa, J. Haupt, A. Sayeed, and R. Nowak. Compressive Wireless Sensing. *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*, 2006.

[6] D. Baron, M.F. Duarte, S. Sarvotham, M.B. Wakin, and R. Baraniuk. An Information-Theoretic Approach to Distributed Compressed Sensing. *Proceedings of the 43rd Allerton Conference on Communication, Control, and Computing*, 2005.

[7] N. Alon, Y. Matias, and M. Szegedy. The space complexity of approximating the frequency moments. *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, 1996.

[8] A.C. Gilbert, Y. Kotidis, S. Muthukrishnan, and M.J. Strauss. One-Pass Wavelet Decompositions of Data Streams. *IEEE Transactions on Knowledge and Data Engineering*, 15(3), pp. 541-554, May 2003.

[9] G. Cormode, M. Garofalakis, and D. Sacharidis. Fast Approximate Wavelet Tracking on Streams. *Proceedings of the International Conference on Extending Database Technology (EDBT)*, 2006.

[10] W.B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Proceedings of the Conference in Modern Analysis and Probability*, 1984.

[11] D. Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4), pp. 671-687, 2003.

[12] P. Li, T.J. Hastie, and K.W. Church. Very Sparse Random Projections. *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (KDD)*, 2006.

[13] N. Ailon and B. Chazelle. Approximate Nearest Neighbors and the Fast Johnson-Lindenstrauss Transform. *Proceedings of the ACM Symposium on Theory of Computing (STOC)*, 2006.

[14] R.A. Horn and C.R. Johnson. Matrix Analysis. Cambridge University Press, New York, NY, 1985.

[15] M. Vetterli and J. Kovačević. Wavelets and Subband Coding. *Prentice Hall*, Englewood Cliffs, NJ, 1995.

[16] S. Mallat. A Wavelet Tour of Signal Processing. *Academic Press*, San Diego, CA, 1999.

[17] M. Penrose. Random Geometric Graphs. *Oxford University Press*, UK, 2003.

[18] W. Wang and K. Ramchandran. Random Distributed Multiresolution Representations with Significance Querying. *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*, 2006.

[19] A.G. Dimakis, V. Prabhakaran, and K. Ramchandran. Ubiquitous Access to Distributed Data in Large-Scale Sensor Networks through Decentralized Erasure Codes. *Proceedings of the International Conference on Information Processing in Sensor Networks (IPSN)*, 2005.

[20] M. Luby. LT Codes. *Proceedings of the 43rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2002.

[21] R. G. Gallager. Low Density Parity-Check Codes. *MIT Press*, Cambridge, MA, 1963.