RECOVERING K-SPARSE N-LENGTH VECTORS IN $O(K \log N)$ TIME: COMPRESSED SENSING USING SPARSE-GRAPH CODES

Xiao Li and Kannan Ramchandran

Department of EECS, University of California, Berkeley, CA, USA
{xiaoli,kannanr}@eecs.berkeley.edu

ABSTRACT

We study the design of measurement matrices for compressed sensing, where the goal is to stably acquire and reconstruct arbitrary K-sparse N-length signals in the presence of noise. We propose a new design framework that simultaneously leads to low measurement cost and low computational cost. In particular, the proposed framework guarantees successful recovery with high probability using $O(K \log N)$ measurements with a computational complexity of $O(K \log N)$. Both the measurement cost and algorithm run-time are *order-optimal* for support recovery when $K = O(N^\delta)$ for some $0 < \delta < 1$. *To the best of our knowledge, this is the first result that achieves this optimal scaling.* The remarkable gains are brought by the proposed measurement structure based on *sparse-graph codes*, which allows for reconstructions of sparse signals using a simple peeling decoder. More generally, we formally connect general sparse recovery problems with sparse-graph decoding, and demonstrate our design in terms of the measurement cost, computational complexity and performance.

Index Terms— Compressed sensing, sub-linear time, sparse-graph codes, measurement matrix

1. INTRODUCTION

A classic problem of interest in many applications is that of estimating an unknown vector $\mathbf{x}$ of length N

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w}, \tag{1}$$

where $\mathbf{A}$ is an $M \times N$ *measurement matrix* and $\mathbf{w}$ is an additive noise vector. If the signal is K-sparse in some basis with $K \ll N$, it can be recovered from significantly fewer measurements, as studied in *compressed sensing*. In this paper, we focus on recovering the *exact support* of any K-sparse signal. The so-called *support recovery* problem arises in applications including model selection, sparse approximation and subset selection. Therefore, the design of good measurement matrices and efficient reconstruction algorithms are critical. This boils down to two questions of interest:

- **Q1)** Measurement cost: what is the smallest number of measurements M required to guarantee recovery?
- **Q2)** Computational cost: how fast can one reconstruct a K-sparse signal given the M measurements from some $\mathbf{A}$?

The answer to **Q1** is well understood under information-theoretic settings. In the presence of noise, the results in [1–3] indicate a minimum measurement cost of $O(K \log(N/K))$ for exact support recovery, here referred to as the *optimal scaling*. For **Q2**, it is desirable if the complexity scales linearly with the measurement cost $O(K \log(N/K))$. However, so far there are no achievable schemes that achieve $O(K \log(N/K))$ costs in both measurements and run-time in the worst case. Therefore, an intriguing question is:

"Under probabilistic settings, is it possible to achieve the optimal scaling in both measurements and run-time?"

We answer this question in the affirmative under the sparsity regime $K = O(N^\delta)$ for any $0 < \delta < 1$. We propose a novel compressed sensing framework based on sparse-graph codes, and show that *with high probability*, our framework guarantees successful recovery in time $O(K \log N)$ using $M = O(K \log N)$ measurements. Under this sparsity regime, our scaling is order-optimal and the run-time becomes sub-linear in N. *To the best of our knowledge, this is the first constructive design for noisy compressed sensing that achieves the same order-optimal costs in both measurements and complexity under probabilistic guarantees.* This can potentially enable real-time processing for massive datasets featuring sparsity, which is relevant to a multitude of practical applications.

2. MAIN RESULTS AND RELATED WORK

In this work, we break the barrier of super-linear scaling in N. We assume that all the non-zero coefficients belong to

$$\mathcal{X} := \{Ae^{\mathrm{i}\theta} : A \in \mathcal{A}, \theta \in \Theta\}$$

where $\mathcal{A}$ and Θ are arbitrarily large finite sets[1]. Given some estimate $\widehat{\mathbf{x}}$, our error metric is the probability $\mathbb{P}_F$ of recovery failure

$$\mathbb{P}_F := \mathsf{Pr}\left(\mathsf{supp}\left(\widehat{\mathbf{x}}\right) \neq \mathsf{supp}\left(\mathbf{x}\right)\right), \tag{2}$$

where $\mathsf{supp}\left(\mathbf{x}\right) := \{k : \ x[k] \neq 0, \ k \in [N]\}$ and $[N]$ is the set of integers $\{0, 1, \cdots, N-1\}$.

Theorem 1. *Given any K-sparse signal $\mathbf{x}$ with its non-zero coefficient $x[k] \in \mathcal{X}$ and $\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I})$, then with a vanishing failure probability $\mathbb{P}_F = O(1/K)$, our framework recovers the exact $\mathbf{x}$ with $M = O(K \log N)$ measurements in time $O(K \log N)$.*

Proof. See the Appendix in [24]. □

Here we give a brief account of related work. The most relevant scheme is the class of *greedy pursuit* algorithms, which detects the sparse support iteratively and refines the approximation in each iteration. Examples include Orthogonal Matching Pursuit (OMP) [4], CoSaMP [5], Regularized OMP (ROMP) [6] and so on. Typically, these algorithms run in polynomial time $\mathrm{poly}(N)$ for both noiseless and noisy settings, except for the StOMP algorithm [7] with a near-linear run-time $O(N \log N)$. Successful recovery in the worst case typically requires a measurement scaling $M = O(K^2)$ even in the absence of noise [8]. Although the measurement scaling can be reduced to $M = O(K \log(N/K))$ under probabilistic settings [9], the best achievable complexity still scales super-linearly with N.

This work was supported by NSF CCF EAGER 1439725 and AFOSR MURI FA9550-10-1-0567 grants.

[1]This is imposed to simplify our analysis. Since its cardinality can be arbitrarily large, it subsumes all signals quantized with finite precisions.

Another popular class of algorithms is based on ℓ_1-*norm minimization* using convex optimization [10], which rests on the assumption of the *Restricted Isometry Property* (RIP) of the measurement matrix. It has been shown that the RIP condition can be satisfied with high probability using $M = O(K\log(N/K))$ measurements by randomized constructions. However, this class of algorithms run in polynomial time $\text{poly}(N)$. To reduce the complexity, there are coding-theoretic designs leveraging the properties of expander graphs [11–14] to achieve lower computational costs [15–17], whose complexities scale super-linearly in N. In particular, [18] introduces a RIP-1 condition and shows that the adjacency matrix of good *expander graphs* satisfies the condition, which allows for near-linear time recovery $O(N\log(N/K))$ using $O(K\log(N/K))$ measurements. However, the run-time still scales super-linearly with N. To achieve a sub-linear scaling in N, the results in [19, 20, 22] achieve $O(K\log^{O(1)}N)$ run-time with optimal measurement scaling but at a constant failure probability.

3. MAIN IDEA

Now, we describe our design framework through a simple example. In this simple example, we make a few assumptions and gradually unfold our design by getting rid of the assumption one-by-one. More specifically, we first introduce in Section 3.1 the main idea of our measurement design by connecting compressed sensing with sparse-graph codes. In this running example, we first illustrate our recovery algorithm with the help of an "oracle", and then explain in Section 3.2 how to get rid of the "oracle".

3.1. Oracle-based Sparse-Graph Decoding

Consider a signal $\mathbf{x}$ of length $N = 16$ with $K = 5$ non-zero coefficients $\text{supp}(\mathbf{x}) = \{1, 3, 5, 10, 13\}$. We construct a bipartite graph with 16 left nodes and 9 right nodes with the following properties:

- Each *left node* labeled with k is associated with $x[k]$;
- Each *left node* is connected to each *right node* based on the *sparse* bipartite graph[2] in Fig. 1;
- Each *right node* labeled with r is assigned a value y_r equal to the sum of its neighbors[3]

$$\mathbf{y} = \mathbf{H}\mathbf{x}, \tag{3}$$

where $\mathbf{H} \in \{0,1\}^{R\times N}$ is the **adjacency matrix**.

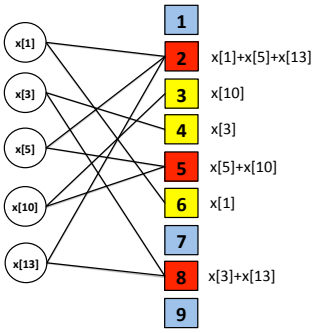

Fig. 1: A bipartite graph consisting of 5 left nodes and 9 right nodes, where each left node represents a non-zero coefficient $x[k]$.

[2]We show only the edges from variable nodes with $x[k] \neq 0$.

[3]This is similar to encoding a message $\mathbf{x}$ using the parity check matrix of some linear block code (e.g. sparse-graph codes), where the observation $\mathbf{y}$ resembles the parity-check constraints.

Now we briefly introduce how this bipartite graph leads to fast recovery of the 16-length 5-sparse signal $\mathbf{x}$ from the 9 measurements shown on the right nodes in Fig. 1. Depending on the degrees of the right nodes, we categorize the measurements as:

1. **Zero-ton**: a right node that contains no non-zero coefficients (e.g., the color *blue* in Fig. 1).
2. **Single-ton**: a right node that contains only one non-zero coefficient (e.g., the color in *yellow* in Fig. 1). We refer to the index k and value $\alpha = x[k]$ as the **index-value pair** (k, α).
3. **Multi-ton**: a right node that contains more than one non-zero coefficient (e.g., the color *red* in Fig. 1).

We assume that there is an "oracle" that informs the decoder which right nodes are *single-tons* and their index-value pairs. With the oracle information, the peeling decoder repeats the following:

Step (1) select all the edges in the bipartite graph with right degree 1 (identify single-ton bins);

Step (2) peel off these edges and the corresponding pair of variable and right nodes connected to these edges.

Step (3) peel off all other edges connected to the variable nodes that have been removed in **Step (2)**.

Step (4) subtract the variable node contributions from right nodes whose edges were removed in **Step (3)**.

Decoding is successful if all the edges are removed from the graph. However, this example does not work for arbitrary signals. In general, there are specific guidelines for constructing such bipartite graphs for successful peeling, which have been studied extensively in the context of sparse-graph codes [23].

A popular ensemble in sparse-graph codes is the d-regular graph ensemble $\mathcal{G}^N_{\text{reg}}(R, d)$ consisting of N left nodes and $R = \eta K$ right nodes for some $\eta > 0$. In this ensemble, each left node is connected to d right nodes uniformly at random[4]. The number of right nodes R determines the measurement cost, while the number of steps taken to peel off all the edges from the graph corresponds to the computational cost with an oracle-based decoder. Clearly, the more right nodes, the easier to have single-tons and thus more friendly for peeling. The critical question now becomes *what is the minimum number of right nodes* that are necessary for peeling?

Lemma 1. *If $\eta > 0$ is chosen according to the left degree d as*[5]*:*

d	2	3	4	5	6
min. η	2.0000	1.2219	1.2948	1.4250	1.5696

the oracle-based decoder succeeds in peeling off all edges in $O(K)$ steps with probability at least $1 - O(1/K)$.

Proof. Here we provide an outline of the proof highlights, and we provide the details in the Appendix of [24].

- **Density evolution**: We analyze our peeling decoder over a *typical graph* (i.e., cycle-free) of the ensemble $\mathcal{G}^N_{\text{reg}}(R, d)$ for a fixed number of peeling iterations i. We assume that the local neighborhood of every edge in the graph is cycle-free (tree-like) and derive a recursion that tracks the average density p_i of remaining edges in the graph at iteration i:

$$p_i = f(p_{i-1}) = \left(1 - e^{-\frac{d}{\eta}p_{i-1}}\right)^{d-1}, \tag{4}$$

which can be made to strictly satisfy $p_i < p_{i-1}$ as long as η is chosen according to d as Lemma 1.

[4]The graph in Fig. 1 is an instance from the ensemble $\mathcal{G}^{16}_{\text{reg}}(9, 2)$.

[5]Limited by space, we only show the choice of minimum η for $d \leq 6$.

- **Convergence to density evolution**: Using a Doob martingale argument [25], we show that the local neighborhood of most edges of a randomly chosen graph from $\mathcal{G}_{\text{reg}}^{N}(R,d)$ is cycle-free with probability at least $1-O(1/K)$. This proves that with high probability, our peeling decoder removes all but an arbitrarily small fraction of the edges in the graph in finite iterations i.
- **Graph expansion property** for complete decoding: we show that if the sub-graph consisting of the remaining edges is an expander graph, and if our peeling decoder successfully removes all but a small fraction of the edges, then it continues to remove all the remaining edges. This completes the decoding of all non-zero coefficients in $\mathbf{x}$.

□

There are other irregular graph constructions that achieve $\eta = 1$ asymptotically and hence the measurement cost approaches $R = K$, but it does not affect the scaling of our results so we consider the regular ensemble for simplicity.

3.2. The Noiseless "Oracle": Binary Ratio Test

Now we explain how to obtain the oracle information. For simplicity, we discuss the noiseless scenario first and defer the noisy oracle to Section 4. The trick is to accrue more measurements at each right node by lifting the simple sum into a vector sum, where the n-th left node is weighted by the n-th column of the **bin detection matrix**

$$\mathbf{S} = (-1)^{\mathbf{B}}, \tag{5}$$

where $\mathbf{B} = \begin{bmatrix}\mathbf{b}_0 & \mathbf{b}_1 & \cdots & \mathbf{b}_{N-1}\end{bmatrix}$ is the binary expansion matrix with $n = \lceil \log_2 N \rceil$ such that each column $\mathbf{b}_k$ is an n-bit binary representation for all $k \in [N]$. In our running example with $N = 16$, the 4×16 binary expansion matrix is

$$\mathbf{B} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & \cdots & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & \cdots & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 & \cdots & 1 \end{bmatrix} \tag{6}$$

and the bin detection matrix is:

$$\mathbf{S} = \begin{bmatrix} (-1)^0 & (-1)^0 & (-1)^0 & (-1)^0 & \cdots & (-1)^1 \\ (-1)^0 & (-1)^0 & (-1)^0 & (-1)^0 & \cdots & (-1)^1 \\ (-1)^0 & (-1)^0 & (-1)^1 & (-1)^1 & \cdots & (-1)^1 \\ (-1)^0 & (-1)^1 & (-1)^0 & (-1)^1 & \cdots & (-1)^1 \end{bmatrix}. \tag{7}$$

3.2.1. A Simple Noiseless Case with Known $x[k]$

For simplicity, we assume that the values are all known $x[k] = 1$ for $k \in \mathsf{supp}(\mathbf{x})$ but the locations k are unknown. In Section 3.2.2, we explain how to get rid of this assumption. Given this bin detection matrix and that all $x[k] = 1$ by assumptions, right nodes 1, 2 and 3 are associated with measurements $\mathbf{y}_1 = \mathbf{0}$,

$$\mathbf{y}_2 = \begin{bmatrix}(-1)^0\\(-1)^0\\(-1)^0\\(-1)^1\end{bmatrix} + \begin{bmatrix}(-1)^0\\(-1)^1\\(-1)^0\\(-1)^1\end{bmatrix} + \begin{bmatrix}(-1)^1\\(-1)^1\\(-1)^0\\(-1)^1\end{bmatrix},\ \mathbf{y}_3 = \begin{bmatrix}(-1)^1\\(-1)^0\\(-1)^1\\(-1)^0\end{bmatrix}.$$

Now, one can easily determine if a right node is a zero-ton, a single-ton or a multi-ton as follows:

- **zero-ton bin**: consider the right node 1. A zero-ton can be identified easily since the measurements are all zero $\mathbf{y}_1 = \mathbf{0}$;
- **single-ton bin**: consider the right node 3. A single-ton can be verified by checking if $|y_3[1]| = \cdots = |y_3[4]|$ and the unknown index can be obtained by taking the sign[6] of each measurement $\mathsf{sgn}[y_3[p]]$ such that

$$\widehat{k} = \sum_{p=1}^{n} 2^{p-1} \times \mathsf{sgn}[y_3[p]]. \tag{9}$$

Finally, since the measurement $\mathbf{y}_2$ from right node 2 does not pass the zero-ton and single-ton tests, it can be concluded as a multi-ton.

3.2.2. General Noiseless Case with Unknown $x[k]$

In the general noiseless case where $x[k]$ is unknown, we can easily modify the simple case by concatenating an extra "all-one" row vector with the bin detection matrix $\mathbf{S}$ as

$$\mathbf{S} = \begin{bmatrix} 1 & 1 & 1 & 1 & \cdots & 1 \\ (-1)^0 & (-1)^0 & (-1)^0 & (-1)^0 & \cdots & (-1)^1 \\ (-1)^0 & (-1)^0 & (-1)^0 & (-1)^0 & \cdots & (-1)^1 \\ (-1)^0 & (-1)^0 & (-1)^1 & (-1)^1 & \cdots & (-1)^1 \\ (-1)^0 & (-1)^1 & (-1)^0 & (-1)^1 & \cdots & (-1)^1 \end{bmatrix}. \tag{10}$$

Using this bin detection matrix, for the single-ton right node 3, we would have $\mathbf{y}_3 = x[10] \times [1, (-1)^1, (-1)^0, (-1)^1, (-1)^0]$, which gives us $y_3[0] = x[5]$ and the unknown index k can be obtained as:

$$\widehat{k} = \sum_{p=1}^{n} 2^{p-1} \times \mathsf{sgn}[y_3[p]] \oplus \mathsf{sgn}[y_3[0]]. \tag{11}$$

However, in the presence of noise, these tests no longer work as an oracle. Next we explain how to get rid of the oracle in this setting.

4. ROBUST BIN DETECTION

For convenience, we denote an arbitrary measurement bin as $\mathbf{y}$ by dropping the right node index r

$$\mathbf{y} = \mathbf{S}\mathbf{z} + \mathbf{w} \tag{12}$$

with some sparse vector $\mathbf{z}$. Before the peeling starts, the sparsity of $\mathbf{z}$ depends on the connectivity of the right node, but eventually, the vector $\mathbf{z}$ reduces to a 1-sparse vector when becoming a single-ton.

The bin detection matrix $\mathbf{S}$ can be regarded as a *codebook* for encoding the unknown value and location of the 1-sparse coefficient in $\mathbf{z}$, where each column of $\mathbf{S}$ is a *codeword*. In the noiseless case, each codeword in $\mathbf{S}$ is the bipolar $\{\pm 1\}$ image of the corresponding binary code $\mathbf{b}_k$ of the column index k, and hence it is not difficult to decode the transmitted message $\mathbf{b}_k$ and recover k. However, in the presence of noise, the codebook needs to be re-designed such that it can be robustly decoded. Specifically, the bin detection matrix is constructed as $\mathbf{S} = [\mathbf{S}_0^T, \mathbf{S}_1^T, \mathbf{S}_2^T]^T$, where $\mathbf{S}_i$ is chosen as follows.

Definition 1. *Let $n = \lceil \log_2 N \rceil$ and $\mathbf{B}$ be the $n \times N$ binary expansion matrix in* (6)*. We choose the codebooks $\mathbf{S}_i$ as:*

- *$\mathbf{S}_0 = \mathbf{1}_{P\times N}$ is an all-one codebook (i.e. repetition codes);*
- *$\mathbf{S}_1 = (-1)^{\mathbf{C}}$ and $\mathbf{C} = \mathbf{G}\mathbf{B}$, where $\mathbf{G}$ is the $P \times n$ generator matrix for an expander code with block length P, and $R(\beta) = n/P$ is the rate of the code determined by the normalized minimum distance β with respect to P;*
- *$\mathbf{S}_2$ is a $P \times N$ random codebook with Rademacher entries.*

[6] the sign function is defined slightly different than the usual case:

$$\mathsf{sgn}[x] = \begin{cases} 1, & x < 0 \\ 0, & x \geq 0 \end{cases} \tag{8}$$

The reason for choosing expander codes for $\mathbf{S}_1$ is due to its minimum distance properties and linear decoding time (with respect to its block length) using the bit flipping algorithm [26]. With this design, we obtain three measurement sets in each bin:

$$\mathbf{u}_i = \mathbf{S}_i \mathbf{z} + \mathbf{w}_i, \quad i = 0, 1, 2. \tag{13}$$

We perform the bin detection using each set differently in the **zero-ton test** and the **single-ton test**. For the zero-ton test, we use the measurement set $\mathbf{u}_2$ and perform an energy test with some parameter $\gamma \in (0, 1)$. The node is accepted as a zero-ton if

$$\frac{1}{P} \|\mathbf{u}_2\|^2 \leq (1+\gamma)\sigma^2. \tag{14}$$

After ruling out zero-tons, now it remains to identify the index-value pairs (k, α) in two steps:

- the **single-ton search** estimates the index-value pair $(\widehat{k}, \widehat{\alpha})$ assuming that the bin is a single-ton. Specifically, the measurement set $\mathbf{u}_1$ is used for obtaining the estimate $\widehat{k}$ of the index k and the measurement set $\mathbf{u}_0$ is used for obtaining the estimate $\widehat{\alpha}$ of $x[k]$, the details of which are given next.
- the **single-ton verification** uses the measurement set $\mathbf{u}_2$ to confirm whether the bin is in fact a single-ton

$$\frac{1}{P} \left\|\mathbf{u}_2 - \mathbf{S}_2 \cdot \widehat{\alpha} 1_{\widehat{k}}\right\|^2 \leq (1+\gamma)\sigma^2.$$

It can be shown (see the proof of Lemma 2) that zero-tons and multi-tons will be ruled out with high probability, and hence we focus on the case of single-tons and discuss the single-ton search.

If the underlying bin has an index-value pair (k, α), the measurement $\mathbf{u}_1$ is the noisy version of some coded message $\mathbf{G}\mathbf{b}_k$

$$\mathbf{u}_1 = \alpha \times (-1)^{\mathbf{G}\mathbf{b}_k} + \mathbf{w}_1, \tag{15}$$

where $\mathbf{b}_k$ is the k-th column of the binary expansion matrix $\mathbf{B}$.

Proposition 1. *Given a single-ton bin with an index-value pair (k, α), the sign of the measurement set $\mathbf{u}_1$ satisfies*

$$\mathsf{sgn}\left[\mathbf{u}_1\right] = \mathbf{G}\mathbf{b}_k \oplus \mathsf{sgn}\left[\alpha\right] \oplus \mathbf{e}, \tag{16}$$

where $\mathbf{e}$ is a binary error vector containing P Bernoulli variables with a cross probability upper bounded as $\mathbb{P}_{\mathsf{e}} = e^{-\frac{|x[k]|^2}{2\sigma^2}}$.

Proof. The proof can be obtained by standard Gaussian tail bounds, and hence we omit it here due to lack of space. □

Although α is unknown, it can be estimated using $\mathbf{u}_0$:

$$\widehat{\alpha} = \min_{x \in \mathcal{X}} \left| x - \mathbf{1}^T \mathbf{u}_0 / P \right|^2. \tag{17}$$

It can be shown that the $\mathsf{Pr}\left(\widehat{\alpha} \neq x[k]\right)$ also decays exponentially (see the proof of Lemma 2 below) and therefore, we have

$$\mathsf{sgn}\left[\mathbf{u}_1\right] \oplus \mathsf{sgn}\left[\widehat{\alpha}\right] = \mathbf{G}\mathbf{b}_k \oplus \mathbf{e}. \tag{18}$$

Because the index k can be obtained from $\mathbf{b}_k$ directly, we only need to decode $\mathbf{b}_k$ reliably over a binary symmetric channel (BSC) with a cross probability $\mathbb{P}_{\mathsf{e}}$. Since the codebook $\mathbf{C} = \mathbf{G}\mathbf{B}$ from Definition 1 has a minimum distance βP, it is obvious that the message $\mathbf{b}_k$ can be decoded with exponentially decaying error probability by the tail bound on the errors (see Lemma 2) as long as $\beta > \mathbb{P}_{\mathsf{e}}$.

Lemma 2. *The error probability of the robust bin detection scheme by Definition 1 decays exponentially in P if the generator matrix $\mathbf{G}$ has a normalized minimum distance $\beta > \mathbb{P}_{\mathsf{e}}$.*

Proof. We refer readers to the Appendix of [24]. □

It has been well established [26] that for any given minimum distance βP, one can construct an expander code that satisfies this minimum distance property with high probability. Thus we can randomly generate the matrix $\mathbf{G}$ offline and verify its minimum distance, and then keep using it for all instances.

5. NUMERICAL EXPERIMENTS

We provide the empirical performance of our design. Each data point is generated by averaging 200 experiments, where the signals $\mathbf{x}$ are generated once and kept fixed for all the subsequent experiments. The signal-to-noise ratio (SNR) is defined as

$$\mathsf{SNR} = \frac{\mathbb{E}\left[\|\mathbf{A}\mathbf{x}\|^2\right]}{\mathbb{E}\left[\|\mathbf{w}\|^2\right]} = \frac{\|\mathbf{x}\|^2}{\sigma^2}\frac{d}{R} \tag{19}$$

where d is the regular left node degree of the bipartite graph, R is the number of right nodes in the graph. The noise is generated as i.i.d. Gaussian with variance σ^2 according to SNR$=$ 20dB. The measurement matrix $\mathbf{A}$ is constructed as follows:

- the coding matrix $\mathbf{H}$ is constructed using the regular graph ensemble $\mathcal{G}_{\mathrm{reg}}^N(R, d)$ with a regular degree $d = 3$ and a redundancy $R = 2K$;
- we choose the $P \times \log_2 N$ generator matrix $\mathbf{G}$ based on a $(3, 6)$-regular LDPC code (i.e. $P = 2\log_2 N$), and the single-ton search utilizes Gallager's bit flipping algorithm.

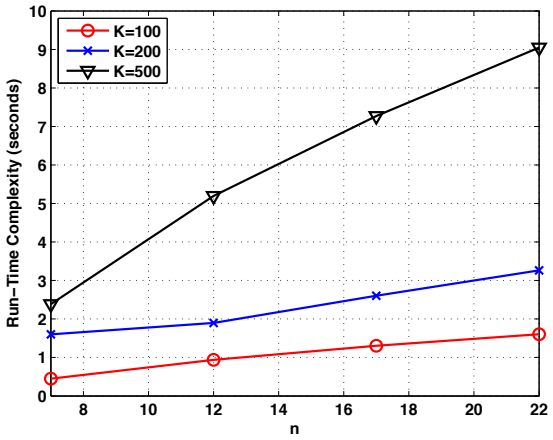

Fig. 2: Run-time against $N = 2^n$ with SNR=20dB.

Under our settings, since $R = 2K$ and $P = 2\log_2 N$, the measurement cost can be obtained accordingly as $M = 12K\log_2 N$. Specifically, it can be seen from Fig. 2 that the run-time scales sub-linearly with respect to N. For instance, when $N = 2^{22} \approx 4$ million and $K = 500$, the measurement cost is approximately $M = 0.13$ million and the run-time is less than 10 seconds.

6. CONCLUSIONS

We propose a new compressed sensing design using sparse-graph codes that simultaneously leads to low measurement cost and low computational cost. Our scheme uses binary measurement matrices that are practical and robust to numerical precisions in implementation. To the best of our knowledge, this is the first known constructive scheme for noisy compressed sensing that achieve $O(K \log N)$ costs in both measurements and complexity. We also provide simulation results to corroborate our theoretical findings.

7. REFERENCES

[1] Michael Gastpar and Yoram Bresler, "On the necessary density for spectrum-blind nonuniform sampling subject to quantization," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2000, vol. 1, pp. 348–351.

[2] Martin J Wainwright, "Information-theoretic limits on sparsity recovery in the high-dimensional and noisy setting," *IEEE Trans. on Information Theory*, vol. 55, no. 12, pp. 5728–5741, 2009.

[3] Mehmet Akçakaya and Vahid Tarokh, "Shannon-theoretic limits on noisy compressive sampling," *IEEE Trans. on Information Theory*, vol. 56, no. 1, pp. 492–504, 2010.

[4] Joel A Tropp and Anna C Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. on Information Theory*, vol. 53, no. 12, pp. 4655–4666, 2007.

[5] Deanna Needell and Joel A Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[6] Deanna Needell and Roman Vershynin, "Uniform uncertainty principle and signal recovery via regularized orthogonal matching pursuit," *Foundations of computational mathematics*, vol. 9, no. 3, pp. 317–334, 2009.

[7] David L Donoho, Yaakov Tsaig, Iddo Drori, and J-L Starck, "Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit," *IEEE Trans. on Information Theory*, vol. 58, no. 2, pp. 1094–1121, 2012.

[8] Lloyd Welch, "Lower bounds on the maximum cross correlation of signals (corresp.)," *IEEE Trans. on Information theory*, pp. 397–399, 1974.

[9] J Tropp and Anna C Gilbert, "Signal recovery from partial information via orthogonal matching pursuit," 2005.

[10] Richard G Baraniuk, "Compressive sensing," *IEEE Signal Processing Magazine*, vol. 24, no. 4, 2007.

[11] Farzad Parvaresh and Babak Hassibi, "Explicit measurements with almost optimal thresholds for compressed sensing," in *IEEE International Conference on Acoustics, Speech and Signal Processing,*. IEEE, 2008, pp. 3853–3856.

[12] Shriram Sarvotham, Dror Baron, and Richard G Baraniuk, "Sudocodes - fast measurement and reconstruction of sparse signals," in *IEEE International Symposium on Information Theory*. IEEE, 2006, pp. 2804–2808.

[13] Mayank Bakshi, Sidharth Jaggi, Sheng Cai, and Minghua Chen, "Sho-fa: Robust compressive sensing with order-optimal complexity, measurements, and bits," in *50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2012, pp. 786–793.

[14] Fan Zhang and Henry D Pfister, "Compressed sensing and linear codes over real numbers," in *Information Theory and Applications Workshop, 2008*. IEEE, 2008, pp. 558–561.

[15] Piotr Indyk and Milan Ruzic, "Near-optimal sparse recovery in the l1 norm," in *49th Annual IEEE Symposium on Foundations of Computer Science,*. IEEE, 2008, pp. 199–207.

[16] Radu Berinde, Piotr Indyk, and Milan Ruzic, "Practical near-optimal sparse recovery in the l_1 norm," in *46th Annual Allerton Conference on Communication, Control, and Computing*. IEEE, 2008, pp. 198–205.

[17] Hoa Vinh Pham, Wei Dai, and Olgica Milenkovic, "Sublinear compressive sensing reconstruction via belief propagation decoding," in *IEEE International Symposium on Information Theory*. IEEE, 2009, pp. 674–678.

[18] Weiyu Xu and Babak Hassibi, "Efficient compressive sensing with deterministic guarantees using expander graphs," in *IEEE Information Theory Workshop*. IEEE, 2007, pp. 414–419.

[19] Anna C Gilbert, Martin J Strauss, Joel A Tropp, and Roman Vershynin, "One sketch for all: fast algorithms for compressed sensing," in *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*. ACM, 2007, pp. 237–246.

[20] Anna C Gilbert, Yi Li, Ely Porat, and Martin J Strauss, "Approximate sparse recovery: optimizing time and measurements," *SIAM Journal on Computing*, vol. 41, no. 2, pp. 436–453, 2012.

[21] Anna C Gilbert, Martin J Strauss, Joel A Tropp, and Roman Vershynin, "Algorithmic linear dimension reduction in the l_1 norm for sparse vectors," *arXiv preprint cs/0608079*, 2006.

[22] M Amin Khajehnejad, Juhwan Yoo, Animashree Anandkumar, and Babak Hassibi, "Summary based structures with improved sublinear recovery for compressed sensing," in *IEEE International Symposium on Information Theory*. IEEE, 2011, pp. 1427–1431.

[23] Michael G Luby, Michael Mitzenmacher, Mohammad Amin Shokrollahi, and Daniel A Spielman, "Efficient erasure correcting codes," *IEEE Trans. on Information Theory*, vol. 47, no. 2, pp. 569–584, 2001.

[24] Xiao Li, Sameer Pawar, and Kannan Ramchandran, "Sublinear time support recovery for compressed sensing using sparse-graph codes," `http://www.eecs.berkeley.edu/~xiaoli/TR_CS_sublinear.pdf` 2014.

[25] Thomas J Richardson and Rüdiger L Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *IEEE Trans. on Information Theory*, vol. 47, no. 2, pp. 599–618, 2001.

[26] Tom Richardson and Ruediger Urbanke, *Modern coding theory*, Cambridge University Press, 2008.