Sparse Covariance Estimation Based on Sparse-Graph Codes

Ramtin Pedarsani, Kangwook Lee, and Kannan Ramchandran
Dept. of Electrical Engineering and Computer Sciences
University of California, Berkeley
{ramtin, kwlijiang, kannan}@eecs.berkeley.edu

Abstract—We consider the problem of recovering a sparse covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$ from $m$ quadratic measurements $y_i = a_i^T \Sigma a_i + w_i$, $1 \leq i \leq m$, where $a_i \in \mathbb{R}^n$ is a measurement vector and $w_i$ is additive noise. We assume that $\Sigma$ has $K$ non-zero off-diagonal entries. We first consider the simplified noiseless problem where $w_i = 0$ for all $i$. We introduce two low complexity algorithms, the first a “message-passing” algorithm and the second a “forward” algorithm, that are based on a sparse-graph coding framework. We show that under some simplifying assumptions, the message passing algorithm can recover an arbitrarily-large fraction of the $K$ non-zero components with $cK$ measurements, where $c$ is a small constant that can be precisely characterized. As one instance, the message passing algorithm can recover, with high probability, a fraction $1 - 10^{-4}$ of the non-zero components, using only $m = 6K$ quadratic measurements, which is a small constant factor from the fundamental limit, with an optimal $O(K)$ decoding complexity. We further show that the forward algorithm can recover all the $K$ non-zero entries with high probability with $m = \Theta(K)$ measurements and $O(K \log^2(K))$ decoding complexity. However, the forward algorithm suffers from significantly larger constants in terms of the number of required measurements, and is indeed less practical despite providing stronger theoretical guarantees. We then consider the noisy setting, and show that both proposed algorithms can be robustified to noise with $m = \Theta(K \log^2(n))$ measurements. Finally, we provide extensive simulation results that support our theoretical claims.

I. INTRODUCTION

Estimating the covariance matrix of high-dimensional data is an important problem in many applications related to big data information processing. In this paper, we tackle the problem of estimating a sparse covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$ from quadratic measurements of the form $y = a_i^T \Sigma a + w$, where $a \in \mathbb{R}^n$ is a measurement column vector, and $w$ is an independent additive noise. Sparse covariance matrices arise in applications such as finance, biology, and spectrum estimation [1], [2], where few pairs of random variables have significant correlation. In high-dimensional data stream computing, it is preferable to maintain a lower-dimensional sketch of the data due to constraints on memory complexity and computational complexity. The main motivation behind considering quadratic measurements is to maintain a sketch $(a_i^T x)^2$ of the real-time data $x_t \in \mathbb{R}^n$, $t \geq 1$ for various sketch vectors $a_i$, $1 \leq i \leq m$ repeatedly to form an empirical estimate of $E[(a_i^T x)^2]$. Then, the quadratic measurement will be $y_i = E[(a_i^T x)^2] + w_i = a_i^T \Sigma a_i + w_i$, where $w_i$ is the error of the empirical estimate. Other applications for quadratic measurements are also presented in [1].

We now provide a brief literature review. Covariance estimation from full data samples has been extensively studied in the literature, e.g. [2], [3]. Estimating the covariance matrix from quadratic measurements is considered in [1]. The authors use convex optimization methods, and propose a robust recovery algorithm when the matrix is low-rank, sparse, or both. Covariance sketching from measurements of the form $Y = A\Sigma B^T$ is considered by [4], [5]. Recovery of sparse matrices from similar sketching measurements is also studied in [6]. In [7], the authors introduce a rank-one projection model for low-rank matrix recovery, and propose a constrained nuclear norm minimization method for stable recovery of the matrix. Our work is significantly different from all the previous related works in certain aspects. First, we design a measurement system that is based on the use of sparse-graph codes. Second, we propose an iterative decoding algorithm, which has computational complexity that is almost linear in $K$, the sparsity dimension. This is the time to find all the significant off-diagonal components of the matrix, not the time to reconstruct the full estimated matrix $\Sigma$. We remark that our scheme can be applied to the recovery of any symmetric matrix such as the adjacency matrix of a graph via quadratic measurements, and is not specific to the covariance matrix. The key contribution of this work is to exploit powerful coding theory techniques to come up with a provably fast and robust recovery algorithm. We present a high-level idea of our measurement design and decoding procedure. Our measurement design follows similar architecture to those of [8]–[11]. Let $A = [a_1^T, a_2^T, \ldots, a_m^T]^T$ be the measurement matrix constructed from the measurement row vectors $[a_i^T]_{i=1}^m$. We design a sparse measurement matrix based on a sparse-graph code. The design of $A$ is such that the measurements provide opportunities to find each off-diagonal non-zero entry of $\Sigma$ iteratively, and peel the recovered components from the other measurements by subtracting their contributions. We use techniques from coding theory such as density evolution to show that the peeling decoding algorithm terminates successfully by recovering almost all of the significant entries of $\Sigma$. To the best of our knowledge, we are the first to advocate the use of coding theory for the sparse covariance estimation problem.

Notation and Paper Organization. We define the following notations. We denote $f = \Theta(g)$ if there are positive constants $c_1$ and $c_2$ such that $c_1 \leq |f/g| \leq c_2$, and $f = O(g)$ if there exists a positive constant $c$ such that $|f/g| \leq c$. We define the set $[n] \triangleq \{1, 2, \ldots, n\}$. We define the entry-wise product of two vectors $u, v \in \mathbb{R}^n$ to be $u \odot v = [u_i v_i]_{i=1}^n$. We define $\log(\cdot)$ to be the natural logarithm and $\log_2(\cdot)$ to be the base-2 logarithm.
The rest of this paper is organized as follows. In Section II, we present the problem formulation. We state the main theoretical results of the paper in Section III. We consider the noisecase in Section IV, and present two recovery schemes: the message passing algorithm and the forward algorithm. We robustify the algorithms to noise in Section V. We provide simulation results in Section VI that support our theoretical results. We conclude the paper in Section VII.

II. PROBLEM FORMULATION

Consider a covariance matrix $\Sigma \in \mathbb{R}^{n \times n}$ of random vector $X = \{X_i\}_{i=1}^n$ that is $K$-sparse on the off-diagonal components. Assume that the variance of the random variables $\{X_i\}_{i=1}^n$ are known to be $\sigma_i^2$. That is, the diagonal components of $\Sigma$ are known. Let $\sigma_{ij} = \text{cov}(X_i, X_j)$. We consider the covariance estimation problem where the measurements are of the form of quadratic measurements $y_i = a_i^T \Sigma a_i + w_i$ for $i = 1, \ldots, m$, where $a_i \in \mathbb{R}^n$ is the measurement vector that can be designed, and $w_i \sim N(0, \sigma^2)$ is the i.i.d. noise of the measurement.

In this paper, we focus on the support recovery of the covariance matrix. Let $\text{supp}(\Sigma)$ be the set of non-zero off-diagonal entries of $\Sigma$:

$$\text{supp}(\Sigma) = \{(i, j) \in [n]^2 : \sigma_{ij} \neq 0, i < j\}.$$

Let $\hat{\Sigma}$ be the estimated covariance of some estimation algorithm. We define an error event of the estimation algorithm to be the event that $\Sigma$ and $\hat{\Sigma}$ differ in more than $\epsilon K$ components for some chosen reliability $\epsilon \geq 0$. Note that setting $\epsilon = 0$ leads to asking for exact recovery of the covariance matrix.

We assume that the components of $\Sigma$ come from a set $\mathcal{X} = \{s_1, s_2, \ldots, s_D\}$, where $s_i \in \mathbb{R} \setminus \{0\}$, for some arbitrarily large but finite $D$. Let $c_0 = \min_{x, y \in \mathcal{X}, x \neq y} |x - y|$ and $c_0 = \min_{x \in \mathcal{X}} |x|$ be constants independent of $K$ and $n$. The main objectives of the covariance estimation problem via quadratic measurements are the following.

1) Reliability: We want the error probability of the algorithm to be vanishing as the problem size $N = \binom{n}{2}$ and the number of measurements $m$ get large.
2) Measurement complexity: We want the number of measurements $m$ to be as small as possible, and close to the fundamental limits of the problem.
3) Decoding complexity: We want to have a low-complexity decoding algorithm that finds the non-zero entries of $\Sigma$ in time that is almost linear in $K$ and sublinear in $N$.

III. MAIN RESULTS

We propose two algorithms to tackle the sparse covariance estimation problem that are based on sparse-graph codes: 1) the message passing algorithm, 2) the forward algorithm. We first discuss these algorithms for the noiseless case, and then propose a robustified version of the algorithms for the noisy case. For our theoretical results, we consider the following simplifying assumption.

Assumption 1: The $K$ non-zero off-diagonal entries of $\Sigma$ are uniformly distributed in the set of $\binom{n}{2}$ entries.

We remark that while Assumption 1 is quite restrictive for covariance matrices, we need it mainly for theoretical purposes. In practice, we observe that if the sparsity is distributed enough, our proposed algorithms perform well. The necessity of having distributed sparsity (though with some difference) is also brought up in [5].

The main results of this paper are as follows.

Theorem 1: Under Assumption 1, if $K = \Theta(n^{1-\delta})$ for some $0 < \delta < 1$, the message passing algorithm recovers an arbitrarily-large fraction, $1 - \epsilon$, of the non-zero components of $\Sigma$ using $m = c(\epsilon) K$ quadratic measurements with probability $1 - O(e^{-m^{\Theta(1)}})$, where the constant $c(\epsilon)$ can be precisely characterized. See Table I for some operating points $(c, \epsilon)$.

Theorem 2: Under Assumption 1, the forward algorithm recovers all the non-zero components of $\Sigma$ using $m = \Theta(K)$ with probability $1 - O\left(\frac{\log(m)}{m}\right)$.

Theorem 3: In the presence of noise, both message passing and forward algorithms can be robustified to noise with $m = \Theta(K \log^2(n))$ measurements and with probability $1 - O\left(\frac{1}{n}\right)$.

We provide the details of the algorithms as well as the proofs of Theorems 1 and 2 in Section IV, and the proof of Theorem 3 in Section V.

Remark While the forward algorithm provides stronger theoretical guarantees, it suffers from significantly larger constants in terms of the number of required measurements compared to the message passing algorithm, and is indeed less practical.

IV. NOISELESS CASE

We design the measurement matrix $A$ to be a row tensor product of a modulation matrix $T$ and a code matrix $H$. Let $A = T \otimes H$, where $T \in \mathbb{R}^{P \times n}$, $H \in \mathbb{R}^{M \times n}$ is a code matrix, and $m = PM$. The precise definition of the row tensor product is as follows. Let $h_i \in \mathbb{R}^n$ be the $i$-th row of matrix $H$. Let $A_i = T \text{diag}(h_i) \in \mathbb{R}^{P \times n}$. Then, $A = [A_1^T, A_2^T, \ldots, A_M^T]^T$.

Example Consider matrices

$$T = \begin{bmatrix} 0.1 & 0.2 & 0.3 \\ 0.4 & 0.5 & 0.6 \end{bmatrix} \quad \text{and} \quad H = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

Then, our measurement matrix $A$ is designed from:

$$A = T \otimes H \quad \text{where} \quad A = \begin{bmatrix} 0 & 0.2 & 0 \\ 0 & 0.5 & 0 \\ 0.1 & 0.2 & 0 \\ 0.4 & 0.5 & 0 \\ 0 & 0 & 0.3 \\ 0 & 0 & 0.6 \end{bmatrix}.$$

In the noiseless case, we set $P = 3$, and design $T$ to be

$$T = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ 1/n & 2/n & \ldots & 1/n \\ 0/n & 1/n & \ldots & (n-1)/n \end{bmatrix}.$$
Consider one noiseless quadratic measurement of the covariance matrix \( y = a^T \Sigma a \) for some vector \( a \). Then,
\[
y = \sum_{i,j} \sigma_{ij} a(i)a(j) = \sum_i \sigma_i^2 a(i)^2 + \sum_{i \neq j} \sigma_{ij} a(i)a(j).
\]
Thus, given the knowledge of \( \sigma_i^2 \) for all \( i \), the decoder can get access to the measurements of the form \( \sum_{i < j} \sigma_{ij} a(i)a(j) \).

From now on, with some abuse of notation we assume that the \( k \)-th measurement is
\[
y_k = \sum_{i < j} \sigma_{ij} a_k(i)a_k(j),
\]
where \( a_k \) is the \( k \)-th measurement vector.

**Remark** If the variances of the random variables are not known, one can easily (at least in the noiseless setting) first find the variances by designing measurements vectors \( e_i \in \mathbb{R}^n \), where \( e_i(j) = 1_{(i=j)} \). The measurement complexity is then at least \( n \) that is also a trivial lower bound to the measurement complexity of the problem, if the variances of the random variables are not known. Moreover, in the noisy setting, these measurements can be robustified to noise as we explain in Section V.

The architecture of our measurement system is based on designing a bipartite graph \( G_1 \) with \( M \) right nodes (corresponding to the rows of matrix \( H \)), each representing a set of \( P \) measurements, and \( n \) left nodes (See the left figure of Fig. 1). This design leads to having \( P \) measurements that are linear combinations of the non-zero off-diagonal entries of the covariance matrix. The following example illustrates the design of our measurements.

**Example** Suppose that the first row of \( H \) is \( h_1 = [1, 1, 1, 0, \ldots, 0] \). Then, the first right node of the bipartite graph, \( G_1 \), represents a set of 3 measurement vectors:
\[
a_1 = [1, 1, 1, 0, \ldots, 0], \quad a_2 = [1/n, 2/n, 3/n, 0, \ldots, 0], \quad a_3 = [0, 1/n, 2/n, 0, \ldots, 0].
\]
The 3 measurements corresponding to the right node are then
\[
y_1 = \sigma_{12} + \sigma_{13} + \sigma_{23}, \quad y_2 = \frac{1}{n}(2\sigma_{12} + 3\sigma_{13} + 6\sigma_{23}), \quad y_3 = \frac{1}{n}(0 \times \sigma_{12} + 0 \times \sigma_{13} + 2\sigma_{23}).
\]

We now define another bipartite graph \( G_2 \) of size \( N \) by \( M \). The \( N = n(n-1)/2 \) left nodes of \( G_2 \) are indexed by pairs \((i,j), i < j \) that denote off-diagonal entries of \( \Sigma \). Our designed graph \( G_1 \) will induce \( G_2 \) as follows. If a right node in \( G_1 \) is connected to a subset \( S \) of left nodes, it will be connected to the \( |S| \) \( \sigma_{ij} \) pairs \( \{ \sigma_{ij} \} \) in \( G_2 \). See the left two figures of Fig. I as an illustration. We call the left nodes of \( G_2 \) that correspond to a non-zero entry of \( \Sigma \) as active left nodes. Define a singleton right node to be a right node that is connected to only one non-recovered active left node. We are now ready to explain two recovery algorithms for this problem that are based on singleton detection and recovery: the message passing algorithm and the forward algorithm. The forward algorithm has some similarities to the SUPER algorithm proposed in [12] for compressive phase retrieval in terms of how the right nodes are designed and processed.

**Message Passing Algorithm** The algorithm processes the right nodes (set of \( P \) measurements), and detects if a right node is a singleton, i.e. it is connected to only one non-recovered active left node. Given a singleton is found, the algorithm recovers the active left node and peels it off (subtracts it) from other measurements. The algorithm continues processing the right nodes iteratively until no new singletons can be found. At each iteration of the message passing algorithm, all the right nodes are re-examined to check whether they are (new) singletons (after potential peelings in the previous iteration) or not.

**Forward Algorithm** There are \( L \) stages (groups) of right nodes. The algorithm sequentially processes the right nodes in every stage, and detects if a right node is a singleton, i.e. it is connected to only one active left node. Given a singleton is found, the algorithm recovers the active left node. The found singletons of every stage are peeled from the right nodes of the following stages only after the processing of all the right nodes in the current stage is finished. The algorithm processes each right node only once.

We now explain how our measurement system enables the decoder to detect and recover a singleton. Suppose that right node \( k \) is a singleton of covariance entry \((i,j)\). Then, due to the design of modulation matrix \( T \) in (1), the set of measurements corresponding to right node \( k \) are
\[
y_{k,1} = \sigma_{ij}, \quad y_{k,2} = \sigma_{ij} \times i/n \times j/n, \quad y_{k,3} = \sigma_{ij} \times (i-1)/n \times (j-1)/n.
\]
Thus, the decoder first guesses that the right node is a singleton. By evaluating ratios \( y_{k,3}/y_{k,1} \) and \( y_{k,2}/y_{k,1} \), the decoder finds the values of \( ij \) and \((i-1)(j-1) = ij - (i+j) + 1 \), given that the guess is correct. Solving 2 equations with 2 variables, the decoder finds \( i \) and \( j \). If the results are integer numbers, the decoder declares the right node to be a singleton, and recovers \( \sigma_{ij} = y_{k,1} \). If the right node is not actually a singleton, solving the 2 equations will not lead to integer solutions for \( i \) and \( j \) for generic values of \( \sigma_{ij} \). Thus, the decoder declares that the right node is not a singleton.

\(^2\)One can remove the genericity assumption by adding a 4th random tensor measurement in \( T \) that is uniformly distributed between 0 and 1, and independent of everything else.
Remark In practice, the measurements will always be noisy, and the measurements are not available with infinite precision. Our noiseless study here is of mostly theoretical interest to investigate whether there exists an algorithm with low measurement complexity and decoding complexity that operates close to the fundamental limits of the problem. Further, it provides guidelines on how to design the measurement system for the noisy case using the sparse-graph coding framework.

A. Analysis of the Message Passing Decoding Algorithm

In this section, we analyze the iterative decoding algorithm using density evolution technique from modern coding theory [13]. We first give a high-level description of the proof of Theorem 1. Let \( p_j \) be the fraction of non-zero off-diagonal entries of \( \Sigma \) (active left nodes) that are not recovered at iteration \( j \). Density evolution analysis is a recursive equation that relates \( p_j \) to \( p_{j+1} \). The goal is to design the bipartite graph of matrix \( H \) with as small number of right nodes as possible, while ensuring that \( p_j \) approaches 0 as \( j \) gets large. Concentration techniques can then be used as in [14] to guarantee that the actual random process does not deviate much from the average behavior predicted by the density evolution.

The density evolution analysis for the covariance estimation problem is quite different from the problems for which the bipartite graph can be arbitrarily designed. Note that we have \( N = n(n-1)/2 \) matrix components to be recovered. However, we design a random bipartite graph, \( G_1 \), of size \( n \) by \( M \) (the coding matrix \( H \)). Recall that our designed graph \( G_1 \) will induce another bipartite graph, \( G_2 \), of size \( N \) by \( M \) as follows. If a right node in \( G_1 \) is connected to a subset \( S \) of left nodes, it will be connected to the \( \binom{|S|}{2} \) pairs \( \{\sigma_{ij}\} \) in \( G_2 \). Thus, we do not have full control in designing the bipartite graph \( G_2 \). Then, given a sparse pattern, one can prune the \( N \) by \( M \) bipartite graph to induce a \( K \) by \( M \) graph, \( G_3 \), for which we derive the density evolution. Figure 1 illustrates this procedure. We consider the following simple design for \( G_1 \) (and thus \( G_2 \)). Each right node is connected to \( |S| \) left nodes chosen uniformly at random in \( G_1 \). Thus, each right node is connected to \( \binom{|S|}{2} \) left nodes in \( G_2 \).

Consider the pruned bipartite graph, \( G_3 \), constructed by the \( K \) active left nodes and the \( M \) right nodes. Define the left (right) degree of an edge to be the degree of the left (right) node corresponding to the edge. Let \( \lambda_i \) be the fraction of edges with left degree \( i \) for \( i \geq 1 \), and let \( \rho_i \) be the fraction of edges with right degree \( i \) for \( i \geq 1 \). Define the left and right degree polynomials of the graph as

\[
\lambda(x) = \sum_{i \geq 1} \lambda_i x^{i-1} \quad \text{and} \quad \rho(x) = \sum_{i \geq 1} \rho_i x^{-i-1},
\]

respectively. We first find the degree distribution of a right node in \( G_3 \). Each right node is connected to \( d_r = \binom{|S|}{2} \) left nodes. Under Assumption 1, the probability that the degree of the right node is \( x \) is

\[
\lambda(x) = \binom{d_r}{x} \left(\frac{1}{|S|}\right)^x \left(1 - \frac{1}{|S|}\right)^{d_r-x}.
\]

Thus, each right node is connected to \( d_r = \binom{|S|}{2} \) left nodes. Under Assumption 1, the probability that the degree of the right node is \( x \) is

\[
\lambda(x) = \binom{d_r}{x} \left(\frac{1}{|S|}\right)^x \left(1 - \frac{1}{|S|}\right)^{d_r-x}.
\]

We now find the density evolution equation. The recursive equation for \( p_j \) is

\[
p_{j+1} = \lambda(1 - \rho(1 - p_j)).
\]

The proof is similar to the one in [15] for peeling decoding over erasure channels. We give a short proof in the following. Consider the unfolded neighborhood of an active left node \( v \) in the graph as shown in Figure 2. Let \( d \) be the degree of \( v \). Node \( v \) passes a “not-recovered” message to neighbor right node \( c \) at step \( j+1 \), if all of the other \( d-1 \) neighbor right nodes of \( v \) pass the “not-recovered” message to \( v \) at step \( j \). That is \( p_{j+1} = q_{j}^{d-1} \) under the tree-like assumption for the neighborhood, where \( q_j \) denotes the probability that the message passed from a right node to a left node at iteration \( j \) of the algorithm is “not-recovered”. We calculate \( q_j \) as follows. At iteration \( j+1 \), a right node \( c \) sends a message to an active left node \( v \) that it cannot get recovered if \( c \) is not a singleton at iteration \( j \), or in other words, if at least one other neighbor of \( c \) is not recovered at iteration \( j \). Thus, \( 1 - q_j = \sum_{i=1}^{\infty} \rho_i (1-p_j)^{i-1} = \rho(1-p_j) \). Given that the degree of \( v \) is \( d \), the recursive equation will be

\[
p_{j+1} = (1 - \rho(1 - p_j))^{d-1}.
\]

Thus, considering the edge degree distribution of the left nodes \( \lambda(x) \), one uses the law of total probability to complete the proof.

For our design of the bipartite graph, the density evolution
First, observe that the equation \( t = f(t) \) has a fixed point that is approximately \( t^* = e^{-\eta_2} \). Note that the term \( e^{-\eta_1 t} \approx 1 \) for \( t \ll 1 \). We call this fixed point as the error floor of the algorithm, or the fraction of non-zero off-diagonal entries of \( \Sigma \) that will not be recovered. This error floor can be made arbitrarily small by increasing the average degree of left nodes \( \eta_2 \). To ensure that the algorithm recovers all but a fraction \( e^{-\eta_2} \) of the significant components, we need to design \( \eta_1 \) large enough such that \( f(t) < t \) for \( t \in (t^*,1] \). To minimize the number of measurements of the algorithm, given a target reliability (error floor) \( \epsilon \), we solve the following optimization problem.

Minimize \( \frac{\eta_2}{\eta_1} \) \hspace{1cm} (6)
Subject to \( e^{-\eta_2} \leq \epsilon \) \hspace{1cm} (7)
\[ t > e^{-\eta_2} e^{-\eta_1 t}, \forall t \in (t^*,1] \] \hspace{1cm} (8)

The optimization problem can be numerically solved. Some of the operating points for \( \eta_1, \eta_2 \) and \( \epsilon \) are presented in Table II. The number of measurements of the algorithm is then \( cK = 3\eta_2 K \) that is consistent with Table I. Note that the factor 3 comes from the set of 3 measurements per right node that are presented in (2)–(4).

Figure 3 illustrates the density evolution for design parameters \( \eta_1 = 5 \) and \( \eta_2 = 14 \). We observe that after only 15 iterations, \( p_j \) gets as small as \( 9 \times 10^{-7} \).

Density evolution finds the average fraction of non-zero components that are not recovered by the algorithm assuming that the depth-\( \ell \) neighborhood of an edge is a tree for any finite \( \ell \). Consider a directed edge \( \vec{e} = (v,c) \) from an active left node \( v \) to a right node \( c \). The neighborhood of depth \( \ell \) of \( \vec{e} \) is defined as \( N_\ell^{\vec{e}} \), that is the subgraph of all the edges and nodes on paths having length less than or equal to \( \ell \), that start from \( v \) and the first edge of the path is not \( \vec{e} \).

**Lemma 1:** If \( K = \Theta(n^{1-\delta}) \) for some \( \delta = \Theta(1) > 0 \), under Assumption 1, \( N_\ell^{\vec{e}} \) is a tree-like neighborhood with probability \( \mathcal{O}(1 - \frac{1}{\mathrm{e}^{\mathrm{e}^{\mathrm{e}^{\cdots}}}}) \) for any finite \( \ell \).

**Proof:** See Appendix A in [16].

Given the result of Lemma 1, one can use the standard Doob’s martingale argument first used in [14] for LDPC codes to show the concentration of the fraction of non-recovered significant components around its mean.

**Lemma 2:** Let \( Z \) be the number of non-zero off-diagonal components of \( \Sigma \) after \( \ell \) iterations of the message passing algorithm that are not recovered. Then, for any \( \epsilon_1 > 0 \), there exists large enough \( K \) and positive constants \( \beta \) and \( \gamma \) such that

\[
|\mathbb{E}[Z] - K p_t| < K \epsilon_1 / 2 \tag{9}
\]
\[
P(|Z - K p_t| > K \epsilon_1) < 2 e^{-\beta \epsilon_1^2 K^{\gamma}}, \tag{10}
\]

where \( p_t \) is derived from the density evolution equation (5).

The proof of Lemma 2 is similar to the proof of Lemma 2.7. in [10], which we skip in the interest of space and readability. Note that the error probability of the algorithm is characterized by (10). Since \( 2 e^{-\beta \epsilon_1^2 K^{\gamma}} = \mathcal{O}(e^{-K^{\Theta(1)}}) \), the proof of Theorem 1 is complete.

**Remark** Since the total number of edges in the pruned graph \( G_3 \) is \( \mathcal{O}(K) \) with high probability, the decoding complexity of the message passing algorithm is \( \mathcal{O}(K) \). Note that this is not the time to reconstruct matrix \( \Sigma \), but only the time to decode which off-diagonal entries of \( \Sigma \) are non-zero, and to find their values.

**B. Analysis of the Forward Algorithm**

In the previous section, we observed that the message passing algorithm provides small constants \( c \) for the number of required measurements \( m = cK \), and works well in practice (as we will see in the simulation results). However, it suffers from error floor (an arbitrarily small fraction of significant components will be missed) and the theoretical guarantees are only for the very sparse case where \( K = o(n) \).

In this section, we present a forward algorithm that has no error floor, and works for any sparsity regime. However, we emphasize that in practice the forward algorithm leads to much larger constants and sub-optimal performance compared to the message passing algorithm.

The bipartite graph of the forward algorithm is designed based on \( L = \Theta(\log(\log(K))) \) stages. Each stage refers to a set of right nodes in the bipartite graph. Recall that we design a random bipartite graph, \( G_1 \), of size \( n \) by \( M \) (the coding matrix \( H \)) that determines our measurement matrix \( A = T \otimes H \). Then, our designed graph \( G_1 \) will induce another bipartite graph, \( G_2 \), of size \( N \) by \( M \), where \( N \) is the number of off-diagonal entries of \( \Sigma \). The pruned graph \( G_3 \) is constructed by the active left nodes has size \( K \times M \). If a right node in \( G_1 \) is connected to a subset \( S \) of left nodes, it will be connected to the \( \binom{|S|}{2} \) pairs (corresponding to \( \sigma_{ij} \)) in \( G_2 \). At stage 1, we design \( K \) right nodes and \( |S| = n / \sqrt{K} \).

Then, similar to the previous section, we can calculate the degree distribution of the right nodes in \( G_3 \) to be Poisson-distributed with mean 1. The degree distribution of the left nodes is therefore also Poisson-distributed with parameter 1.

Recall that the forward algorithm only recovers singletons in the first stage. With this construction, the probability that an active left node is not connected to any singletons can be calculated as follows. Let \( E_i \) be the event that active left node \( i \) is not connected to any singletons, and \( D_i \) be the degree of
measurements. Thus, the total number of edges of the pruned bipartite graph for the forward algorithm is $O(K \log(K))$ with high probability. Thus, the decoding complexity is $O(K \log(K))$.

V. NOISY CASE

We robustify the message passing algorithm and the forward algorithm to noise by modifying matrix $T$ while maintaining the code matrix $H$ or the sparse-graph code construction. Clearly, in the presence of noise, the measurement system in (1) and the measurements in (2)–(4) cannot reliably detect whether a right node is a singleton or not. To robustify the algorithm, we increase $P$ from 3 in the noiseless case to
$P = \Theta(\log^2(n))$. Then, the measurement complexity of the algorithm is $m = \Theta(K \log^2(n))$.

We design $T = [T_1, T_2]^T \in \mathbb{R}^{P \times n}$, where $T_1 \in \mathbb{R}^{P \times n}$ denotes the first part of $T$, $T_2 \in \mathbb{R}^{P \times n}$ denotes the second part of $T$, and $P = P_1 + P_2$. Later, we see that $P_1 = \Theta(\log(n))$ and $P_2 = \Theta(\log^2(n))$.

We design $T_i$ to have all 1 entries; that is, $T_i(ij) = 1$ for all $i \in [P]$ and $j \in [n]$. We design random signature vectors for each index pair $(i, j)$, $i > j$ corresponding to the off-diagonal entries of $\Sigma$ as follows. Define the random column vectors $u_i \in \{-1, 1\}^{R_i}$, $1 \leq i \leq n$ such that each entry of $u_i$ is chosen from the set $\{-1, 1\}$ uniformly at random and independently. Let $u_{ij} \in \{-1, 1\}^{R_i}$ be the entry-wise product of $u_i$ and $u_j$: $u_{ij} = u_i \odot u_j$. Note that entries of $u_{ij}$ are also $\pm 1$ with probability $\frac{1}{2}$ and mutually independent. Let $U = [u_1, u_2, \ldots, u_n] \in \{-1, 1\}^{R_1 \times n}$ be the concatenation of column vectors $u_i$, $1 \leq i \leq n$. We design $T_1$ by the repetition of matrix $U$, $R_1$ times: $T_1 = [U^T, U^T, \ldots, U^T]^T \in \{-1, 1\}^{R_1 R_2 \times n}$, where $R_2 = R_1 R_2$. Later we see that we design $R_1 = \Theta(\log(n))$ and $R_2 = \Theta(\log(n))$.

On a high level, assuming that a right node is a singleton, we use $T_1$ to do noise averaging so that the value of the corresponding non-zero covariance entry can be reliably determined. We use $T_2$ to check whether a right node is a singleton or not, and if yes, what the index of the corresponding active left node is.

We now explain how the decoder detects if a right node is a singleton. Let $z \in \mathbb{R}^P$ be the measurement vector corresponding to a particular right node at iteration $l$ of the algorithm. Note that the contribution of the already recovered components are subtracted from $z$. Let $z = [z_0^T, z_1^T, \ldots, z_{R_1}^T]$, where $z_0 \in \mathbb{R}^{P_1}$ and $z_i \in \mathbb{R}^{R_1}$, $1 \leq i \leq R_2$. Let $\mathcal{H}$ be the hypothesis that the right node is a singleton.

**Noisy Right Node Detection Algorithm** Upon observing the measurement vector $z$ for some right node, the decoder finds $\hat{z}_0 = \sum_{k=1}^{R_1} z_0(k) \in \mathbb{R}$. Let $s = \arg\min_{x \in \mathbb{R}^P} |X - \hat{z}_0|$. Fix a small but constant threshold $\delta_0$ such that $0 < \delta_0 \ll \epsilon_0$. Recall that $\epsilon_0 = \min_{x \in \mathbb{R}^P} |X - y|$. If $|s - z_0| > \delta_0$, the decoder rejects $\mathcal{H}$. Otherwise, the decoder forms $\tilde{z} = (\sum_{k=1}^{R_1} z(k)) / R_2 \in \mathbb{R}^{R_2}$. Fix another small but constant threshold $\delta_1$, $0 < \delta_1 \ll 1$. Let $I_1 = [1 - \delta_1, 1 + \delta_1]$ and $I_2 = [-1 - \delta_1, -1 + \delta_1]$. The decoder finds an estimated signature vector $\hat{u}$ as follows. For all $k \in [R_1]$, if $\bar{z}(k) \in I_1$ the decoder sets $\hat{u}(k) = 1$; if $\bar{z}(k) \in I_2$ the decoder sets $\hat{u}(k) = -1$; otherwise, the decoder rejects $\mathcal{H}$. If $\hat{u} = u_{ij}$ for some $i, j$, the decoder declares a singleton right node and recovers $\hat{\sigma}_{ij} = s$. Otherwise, the decoder rejects $\mathcal{H}$.

**A. Analysis of the Noisy Right Node Detection Algorithm**

There are two types of errors for the algorithm: 1) a singleton right node is not detected to be a singleton; 2) a non-singleton right node is detected as a singleton. We find upper bounds on the probability of the two error events.

We first show that with high probability all the signature vectors are distinct.

**Lemma 4:** Suppose that

$$R_1 \geq 5 \log_2(n).$$

Then, the probability that there exist pairs $(i, j) \neq (k, l)$ such that $u_{ij} = u_{kl}$ is $O(\frac{1}{n^3})$.

**Proof:** Note that each entry of the signature vectors is uniformly distributed in $\{-1, 1\}$ and they are independent. It is easy to see that for particular $(i, j)$ and $(k, l)$

$$P_t(u_{ij} = u_{kl}) = \left(\frac{1}{2}\right)^{R_1} \leq \frac{1}{n^3}.$$  

Since there are $\Theta(n^2)$ pairs, using union bound completes the proof of the lemma.

**Remark** The computational complexity of the noisy message passing algorithm is $O(K \log^2(n))$ and the computational complexity of the noisy forward algorithm is $O(K \log(n) \log^2(n))$. Note that each right node is associated with $\Theta(\log^2(n))$ measurements, and there are $\Theta(K)$ right nodes. Further, given that the $\binom{n}{2}$ signature vectors are designed offline and sorted in a look-up table, the computational complexity of finding the index of an active left node corresponding to a singleton right node is $O(\log^2(n))$ by binary search.

**VI. Simulation Results**

We simulate the message-passing algorithm in the noiseless and noisy cases. We first run the message-passing algorithm 1000 times and measure the average fraction of successfully
recovered components: we choose \( n = 100, K = 50, \eta_1 = 4.9, \) and \( 50 \leq M \leq 500 \) (150 \( \leq m \leq 1500\)). Plotted in Figure 4a is the average fraction of correctly recovered components for different values of \( M \). We observe that as the number of right nodes \( M \) increases, the average fraction of recovered components approaches 1. We also simulate the performance of the message-passing algorithm with noisy measurements. We set \( n = 100, K = 50 \) and \( \eta_1 = 4.9 \), and fix the number of right nodes \( M \) as 300. With this fixed number of right nodes, we then vary the number of measurements per right node such that the total number of measurements vary from \( 14K \log_2(n) \) to \( 303K \log_2(n) \). Figure 4b shows the average fraction of recovered components as a function of SNR, defined as 10 \( \log_{10} \left( \frac{\|y-w\|^2}{\|w\|^2} \right) \) in dB.

We also measure the average decoding time of our implementation of the message-passing algorithm and show that the decoding time complexity is \( O(K \log^2(n)) \) in Figures 5a and 5b.

**VII. CONCLUSION**

In this paper, we addressed the sparse covariance estimation problem from quadratic measurements. We proposed a family of measurement matrices for both noiseless and noisy cases that are based on the sparse-graph coding framework. We proved that under some mild assumptions, our iterative decoding algorithm can recover the covariance matrix in time that is almost linear in the sparsity parameter \( K \), and sub-linear to the problem size \( N \). We also provided simulation results to corroborate our theoretical findings.

**REFERENCES**


