Proceedings of Machine Learning Research vol 125:1–22, 2020 33rd Annual Conference on Learning Theory

Information Theoretic Optimal Learning of Gaussian Graphical Models

Sidhant Misra SIDHANT@LANL.GOV
Marc Vuffray VUFFRAY@LANL.GOV
Andrey Y. Lokhov LOKHOV@LANL.GOV

Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545

Editors: Jacob Abernethy and Shivani Agarwal

Abstract

What is the *optimal* number of independent observations from which a sparse Gaussian Graphical Model can be correctly recovered? Information-theoretic arguments provide a lower bound on the minimum number of samples necessary to perfectly identify the support of any multivariate normal distribution as a function of model parameters. For a model defined on a sparse graph with p nodes, a maximum degree d and minimum normalized edge strength κ, this necessary number of samples scales at least as $d \log p/\kappa^2$. The sample complexity requirements of existing methods for perfect graph reconstruction exhibit dependency on additional parameters that do not enter in the lower bound. The question of whether the lower bound is tight and achievable by a polynomial time algorithm remains open. In this paper, we constructively answer this question and propose an algorithm, termed DICE, whose sample complexity matches the information-theoretic lower bound up to a universal constant factor. We also propose a related algorithm SLICE that has a slightly higher sample complexity, but can be implemented as a mixed integer quadratic program which makes it attractive in practice. Importantly, SLICE retains a critical advantage of DICE in that its sample complexity only depends on quantities present in the information theoretic lower bound. We anticipate that this result will stimulate future search of computationally efficient sample-optimal algorithms.

Keywords: Gaussian graphical model, information theoretic bound, sample-optimal learning, slice, dice

1. Introduction

Gaussian Graphical Models (GGMs) are powerful modelling tools for representing statistical dependencies between variables in the form of undirected graphs that are widely used throughout a large number of fields, including neuroscience Huang et al. (2010); Varoquaux et al. (2010), gene regulatory networks Basso et al. (2005); Menéndez et al. (2010) and protein interactions Friedman (2004); Jones et al. (2012). The popularity of GGMs in applications can be explained by the fact that multivariate Normal distribution approximately describes physical variables represented by sums of independent factors and has maximum entropy among all continuous-variable distributions with unbounded support and a given mean and covariance. Moreover, the sparsity pattern of the graph underlying the GGM provides interpretable structural information on the conditional dependencies between variables through the so-called separation property of Markov Random Fields (MRFs).

In this paper, we study the inverse problem of learning a sparse GGM from a small number of observations. Consider a multivariate Normal distribution defined on a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = p$ and bounded maximum degree d:

$$\mathbb{P}(\mathbf{x}) = \frac{\sqrt{\det(\Theta)}}{(2\pi)^{\frac{p}{2}}} \exp\left(-\frac{1}{2} \sum_{i \in \mathcal{V}} \Theta_{ii}(x_i - \mu_i)^2 - \sum_{(i,j) \in \mathcal{E}} \Theta_{ij}(x_i - \mu_i)(x_j - \mu_j) \right), \tag{1}$$

where μ_i denotes the mean of the variable x_i and Θ is the *precision matrix* whose support is determined by the sparsity pattern of the graph $\mathcal{G}$. GGMs have a special property that Θ is equal to the inverse of the covariance matrix Σ, meaning that $[\Sigma^{-1}]_{ij} = 0$ for all $(i,j) \notin \mathcal{E}$. In our reconstruction problem, the data is given as a collection of n independent samples $\{x_i^k\}_{i \in \mathcal{V}}$ indexed by $k = 1, \dots, n$ and drawn from the distribution (1). We are interested in finding tractable algorithms that with high probability output an accurate estimate $\hat{\mathcal{G}}$ of the graph $\mathcal{G}$, i.e. $\mathbb{P}(\hat{\mathcal{G}} = \mathcal{G}) > 1 - \delta$ for a given confidence $\delta > 0$.

The minimum number of samples n^* required for perfect sparse graph reconstruction is bounded by an information-theoretic (IT) lower bound in Wang et al. (2010) that reads

$$n^* > \max\left\{ \frac{\log\binom{p-d}{2} - 1}{4\kappa^2}, \frac{2\left(\log\binom{p}{d} - 1\right)}{\log\left(1 + \frac{d\kappa}{1-\kappa}\right) - \frac{d\kappa}{1+(d-1)\kappa}} \right\}, \tag{2}$$

where the parameter κ denotes the minimum normalized edge strength and is defined as

$$\kappa = \min_{(i,j) \in \mathcal{E}} \frac{|\Theta_{ij}|}{\sqrt{\Theta_{ii}\Theta_{jj}}}. \tag{3}$$

Notice that the IT lower bound (2) depends solely on three parameters of the problem: dimension p, maximum degree d, and minimum edge strength κ. A weak logarithmic dependence on p indicates that it might be possible to reconstruct $\mathcal{G}$ in the high-dimensional regime, and the inverse square dependence on κ is natural because it becomes more difficult to distinguish an edge of low strength from its absence as $\kappa \to 0$. It remained unknown if the bound (2) is tight, i.e., if there exists a *sample-optimal* algorithm that does not depend on additional parameters and achieves this bound.

Numerous algorithms have been suggested to reconstruct sparse GGMs; a non-exhaustive list includes Meinshausen and Bühlmann (2006); Yuan and Lin (2007); Cai et al. (2011); Anandkumar et al. (2012); Cai et al. (2016); Johnson et al. (2012); Wang et al. (2016). However, the sample complexity analysis of previously proposed methods reveals that none of them is converse to the IT bound (2). For most algorithms, the required number of samples depends on additional parameters of the problem that are not present in (2), often due to the assumptions made in the analysis. The regression-type approach of Meinshausen and Bühlmann (2006) for estimating the neighborhood of each vertex based on LASSO Tibshirani (1996) requires certain *incoherence* properties of the precision matrix, reminiscent of the compressed sensing problem. A variant of the incoherence condition is assumed in the analysis Ravikumar et al. (2009) of the ℓ_1 regularized log-likelihood estimator, commonly known as GRAPH LASSO Yuan and Lin (2007); d'Aspremont et al. (2008). The proof for ℓ_1-based estimators CLIME Cai et al. (2011) and ACLIME Cai et al. (2016) require that the eigenvalues of the precision matrix are bounded. Other methods such as the conditional covariance thresholding algorithm Anandkumar et al. (2012) was analyzed only for the class of

so-called *walk-summable* models. The analysis of non-convex optimization based methods Johnson et al. (2012) and Wang et al. (2016) require bounded eigenvalues of Θ matrix.

Although the above methods have been shown to successfully exploit sparsity and reconstruct the underlying graph $\mathcal{G}$ perfectly with $O(\log p)$ samples, all of them exhibit dependence on the condition number of the precision matrix among other quantities. In particular, the bound on the condition number of Θ follows from the the most prevalent assumption in the literature, the so-called Restricted Eigenvalues (RE) condition. Consequently, it is widely believed that the RE condition is in fact *necessary* to enable model reconstruction. Notice, however, that the condition number has no impact on the IT lower bound in (2), as stated above. Consider the following example of a simple precision matrix:

$$\Theta = \begin{bmatrix} 1 & \kappa_0 & \kappa_0 \\ \kappa_0 & 1 & 1-\epsilon \\ \kappa_0 & 1-\epsilon & 1 \end{bmatrix}, \tag{4}$$

where $1 - \epsilon > \kappa_0 > 0$. The minimum normalized edge strength for (4) is given by $\kappa = \kappa_0$. On the other hand, the condition number is given by $\frac{\lambda_{max}(\Theta)}{\lambda_{min}(\Theta)} \geq \epsilon^{-1}$. This means that if we keep κ_0 fixed and let $\epsilon \to 0$, the minimum number of samples n^* according to the IT bound in (2) remains fixed whereas the condition number, and hence the sample complexity of existing algorithms diverges.

Several approaches that might appear quite natural in this context surprisingly do not successfully eliminate additional parametric dependencies in their sample complexity. For example, a natural path is to consider conditional independence testing, since it directly exploits the so-called separation property of graphical models. Along these lines one might attempt methods similar to the SGS and PC algorithms found in Spirtes et al., 2001 Spirtes et al. (2000) and Kalisch et al., 2007 Kalisch and Bühlmann (2007). However, in Van de Geer et al. (2013), which is the follow up of Kalisch and Bühlmann (2007), the authors explicitly pointed out that conditional independence testing requires strong faithfulness assumptions in the analysis of the PC algorithm in Kalisch and Bühlmann (2007). The analysis of Bresler et al. (2008) that is based on neighborhood testing do not directly apply to the current setting: first, their analysis uses properties of discrete distributions that do not hold for Gaussians, and second, their analysis requires additional assumptions. Even in Van de Geer et al. (2013), Maximum-Likelihood with exhaustive search requires bounds on eigenvalues of the covariance matrix; therefore, the sample complexity depends on these eigenvalue bounds. A different approach to utilizing conditional independence testing is based on convergence of the empirical correlation coefficients, such as the one studied in Anandkumar et al. (2012). However methods based on this idea do not seem to close the IT lower bound. The example below gives an intuitive explanation why. Consider the precision matrix

$$\Theta = \begin{bmatrix} 1 & \kappa & \kappa & 0 \\ \kappa & 1 & 1-\epsilon & 0 \\ \kappa & 1-\epsilon & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

The correlation coefficient of variable corresponding to rows 1 and 2 conditioned on variable 4 can be computed as $\rho_{12|4} = \frac{Cov(X_1,X_2|X_4)}{\sqrt{Var(X_1|X_4)Var(X_2|X_4)}} = \frac{\kappa\sqrt{\epsilon}}{\sqrt{(1-\kappa^2)(2-\epsilon)}}$. This implies that one needs $n = O(1/\epsilon)$ samples to assert that there exists an edge between 1 and 2 in this approach, whereas for $\epsilon < \kappa$ the IT lower bound does not depend on ϵ.

Does that mean that the IT lower bound (2) is loose, and the bounded condition number of the precision matrix is indeed a necessary condition for the recovery of sparse GGMs? In this paper, we answer this question constructively, and propose a multi-stage algorithm, named *Degree-constrained Inverse Covariance Estimator* (DICE). Without any assumptions, we show that DICE reconstructs the graph G perfectly with high probability $1-\delta$ using $2d + \frac{192}{\kappa^2} d \log p + \frac{64}{\kappa^2} \log\left(\frac{4d}{\delta}\right)$ samples, i.e. achieves the IT lower bound (2). A discrepancy with compared to the IT bound exists in a subtle case such where d and κ are interdependent quantities that scale together such that $\lim \kappa = 0$ and $\lim d(\kappa) = \infty$; however, this discrepancy dissapears in the regime $d = O(1)$, which is the setting considered here. Therefore, in this regime, DICE closes the gap to the IT bound, and shows that (2) is tight. The worst-case computational complexity of DICE is primarily driven by the iterative support testing step, based on comparison of two neighborhoods, and reads $O(p^{2d+1})$, i.e. exponential for dense graphs, but polynomial with respect to p in the setting of sparse graphs where $d = O(1)$; this result shows that at least for bounded degree graphs, the IT bound on sample complexity can be achieved with a polynomial-time algorithm.

We also propose a related algorithm termed *Sparse Least-squares Inverse Covariance Estimator* (SLICE) which uses a subset of the phases used in DICE. Unlike DICE this simpler algorithm allows implementation as a mixed integer quadratic program (MIQP), enabling the use of modern mixed integer solvers that can be very efficient in practice. As a price for the enhanced computational efficiency, the sample complexity of SLICE is $d + \frac{32}{\kappa^4} \log\left(\frac{4p^{d+1}}{\delta}\right)$, i.e., roughly a factor $1/\kappa^2$ higher than the IT lower bound (and the sample complexity of DICE). However the sample complexity is still only dependent on the parameters present in the IT lower bound, thus avoiding dependence on additional assumptions such as restricted eigenvalues.

The paper is organized as follows: In Section 2 we introduce our algorithm DICE and its sub-routines, and provide main results of our study. Section 3 introduces SLICE and states the theorem regarding its sample complexity. Rigorous mathematical guarantees on the sample performance of our algorithms are given in Section A, while Section C contains proofs of technical lemmas. We conclude with Section 4 where we discuss some perspectives and open problems.

2. DICE: Reconstructing Gaussian Graphical Models with Information Theoretically optimal number of samples

In this section we provide details of DICE. The three constituent steps are (i) cardinality constrained regression to obtain an estimate of the conditional variance for each variable, (ii) an iterative support testing method to find a size d neighborhood that contains the right support, and (iii) a clean up phase to eliminate the non-edges in the set found in (ii). For simplicity of notation, we assume that the distribution in consideration has zero mean. All results easily generalize to the non-zero mean case, as stated below.

2.1. Phase 1: Estimating conditional variances

The first step of the algorithm obtains an estimate of the conditional variance of each variable $i \in \mathcal{V}$ where the conditioning is for all neighbors of i. For each $i \in \mathcal{V}$ our estimate $\hat{\Theta}_{ii}$ of Θ_{ii} is given by

$$\frac{1}{\hat{\Theta}_{ii}} = \min_{\hat{\beta}_i \in \mathbb{R}^{p-1}} \quad L_i(\hat{\beta}_i, \hat{\Sigma}), \qquad \text{s.t.} \quad \|\hat{\beta}_i\|_0 \leq d, \tag{6}$$

where

$$L_i(\beta_i, \hat{\Sigma}) = \frac{1}{n}\sum_{k=1}^{n}\left(x_i^k + \sum_{j\neq i}\beta_{ij}x_j^k\right)^2, \tag{7}$$

the ℓ_0-norm counts the number of non-zero components, and $\hat{\Sigma}$ denotes the empirical covariance matrix whose components are given by $\hat{\Sigma}_{ij} = \frac{1}{n}\sum_{k=1}^{n} x_i^k x_j^k$. [1]

Since the ℓ_0 constraint in (6) is equivalent to searching over all possible $\hat{\beta}_i$ with support given by some $A \subset [p]$ with $|A| = d$, the optimization in (6) can be re-written as

$$\frac{1}{\hat{\Theta}_{ii}} = \min_{A\subseteq[p]\setminus i\,:\,|A|=d} \quad \min_{\hat{\beta}\in\mathbb{R}^{p-1}:\text{Supp}(\hat{\beta})\subset A} L_i(\hat{\beta}, \hat{\Sigma}). \tag{8}$$

Since we will be restricting ourselves to the case when $2d+1 < n$, each $d \times d$ submatrix of $\hat{\Sigma}$ has full rank, and the inner minimization in (8) can be explicitly resolved to get

$$\hat{\beta}_{iA} = -\hat{\Sigma}_{AA}^{-1}\hat{\Sigma}_{Ai}. \tag{9}$$

The corresponding optimal value is given by

$$L_i^*(A, \hat{\Sigma}) = L_i(\hat{\beta}_{iA}, \hat{\Sigma}) = \hat{\Sigma}_{ii} - \hat{\Sigma}_{iA}\hat{\Sigma}_{AA}^{-1}\hat{\Sigma}_{Ai} \overset{(a)}{=} \left[\hat{\Sigma}^{-1}_{(iA)(iA)}\right]_{11} \overset{(b)}{=} \widehat{\mathbf{Var}}(X_i|X_A), \tag{10}$$

where (a) is obtained by using the matrix inversion lemma. We obtain (b) from the standard expression for conditional variance of X_i conditioned on X_A in multivariate gaussians, explainig the name of this subsection. Notice that in the limit of large number of samples when $\hat{\Sigma} = \Sigma$, the empirical value $L_i^*(B_i, \hat{\Sigma})$ is equal to the conditional variance of the model:

$$L_i^*(B_i, \Sigma) = \mathbf{Var}(X_i|X_{B_i}) \quad \forall B_i \subset [p] \setminus \{i\}. \tag{11}$$

2.2. Phase 2: Iterative Support Testing

In this phase, all candidate neighborhoods are passed through a testing criterion. This phase constitutes the main part of the algorithm DICE. We describe the testing criterion in detail and give intuitive rationale behind it.

Fix $i \in \mathcal{V}$ and consider a candidate neighborhood $B_1 \subset \mathcal{V}\setminus\{i\}$ with $|B_1| = d$. The goal is to obtain a B_1 such that $B_i \subseteq B_1$, where B_i denotes the true neighborhood of i. The candidate B_1 is tested by using a set of *adversarial neighborhoods* $B_2 \subset \mathcal{V}\setminus\{\{i\}\cup B_1\}$ with $|B_2| = d$. The testing criterion is based on the regression coefficients $\hat{\beta}_{iB_1B_2} = -\hat{\Sigma}^{-1}_{B_1B_2,B_1B_2}\hat{\Sigma}_{B_1B_2,i}$ as in (9), where we use the notation $B_1B_2 = B_1 \cup B_2$ for simplicity. The candidate B_1 is deemed to have PASSED the testing criterion if for all adversarial neighborhoods B_2 we have

$$\max_{j\in B_2}\hat{\kappa}_{ij} := |\hat{\beta}_{ij}|\sqrt{\frac{\hat{\Theta}_{ii}}{\hat{\Theta}_{jj}}} < \frac{\kappa}{2}. \tag{12}$$

1. All results in this paper directly generalize to the case of non-zero mean by replacing the $\hat{\Sigma}$ above by the unbiased covariance estimator given by $\hat{\Sigma}_{ij} = \frac{1}{n-1}\sum_{k=1}^{n}(x_i^k - \bar{x}_i)(x_j^k - \bar{x}_j)$, where $\bar{x}_i = \frac{1}{n}\sum_{k=1}^{n} x_i^k$.

The quantities $\hat{\kappa}_{ij}$ can be considered as *estimated* normalized edge strengths. The testing criterion relies on the fact that when the number of samples is sufficient (which we formalize later in Section 2.4), the quantities $\hat{\kappa}_{ij}$ are accurate empirical estimates of the true normalized edge strengths given by $\kappa_{ij} = |\beta_{ij}\beta_{ji}| = \frac{|\Theta_{ij}|}{\sqrt{\Theta_{ii}\Theta_{jj}}}$. The intuitive logic behind the testing criterion in (12) can be explained by considering the following two cases:

- *Case 1: The candidate B_1 in consideration is such that $B_i \subseteq B_1$, where B_i denotes the true neighborhood of i.* In this case, for every adversary B_2, and assuming that the estimates $\hat{\beta}_{iB_1B_2}$ are accurate enough, for each $j \in B_2$ the estimates $\hat{\kappa}_{ij}$ should be close to the true value $\kappa_{ij} = 0$, i.e., $\hat{\kappa}_{ij} < \kappa/2$ for all $j \in B_2$ and B_1 would pass the test in (12).
- *Case 2: There exists $j \in B_i \setminus B_1$:* In this case the set B_1 has missed a neigbor $j \in B_i$. Here, any adversary B_2 such that $B_i \subset B_1 \cup B_2$ will make B_1 fail the testing criterion. This is again because, for $j \in B_i \setminus B_1$, the quantity $\hat{\kappa}_{ij}$ is expected to be close to its true value $\kappa_{ij} > \kappa$, i.e., $\hat{\kappa}_{ij} > \kappa/2$ and hence B_1 would FAIL the test in (12).

2.3. Phase 3: Eliminate non-edges

Once a set B_1 is obtained in Phase 2 such that $|B_1| = d$ and $B_i \subseteq B_1$, this clean-up phase consists of appending any B_2 to B_1, computing the estimated normalized couplings $\hat{\kappa}_{ij}$ for all $j \in B_1$ and declaring any $j \in B_1$ such that $\hat{\kappa}_{ij} < \kappa/2$ as a non-edge. The success of this step also relies on the accuracy of the estimates $\hat{\kappa}_{ij}$. The intuitive description of the algorithm and its performance is formalized in the next subsection.

2.4. Formal description of the algorithm and main result

In this subsection, we state our main result regarding the sample complexity of DICE, which is formally presented in Algorithm 1.

The following is the main result of the paper which proves that the algorithm DICE achieves the information theoretic lower bound in (2) up to a universal constant.

Theorem 1 (Converse to IT bound) *Given $\delta > 0$, the probability of perfect graph reconstruction using* DICE *is lower bounded as*

$$\mathbb{P}(\hat{\mathcal{G}} = \mathcal{G}) > 1 - \delta, \tag{13}$$

provided that the number of samples satisfies [2]

$$n > 2d + \frac{192}{\kappa^2} d \log p + \frac{64}{\kappa^2} \log\left(\frac{4d}{\delta}\right). \tag{14}$$

2.5. Proof of Theorem 1

As described intuitively above, the success of DICE relies on the fact that the estimates $\hat{\kappa}_{ij}$ in (12) are accurate. This fact is established in the following two propositions.

2. In particular, Theorem 1 is valid for a number of samples $n > \frac{320}{\kappa^2}(d \log p + \log(1/\delta))$.

Phase 1: Estimatimating conditional variances
for $i \in \mathcal{V}$ **do**
  Estimate $\hat{\Theta}_{ii}$ by solving (6)
end
Phase 2: Iterative support testing
for $i \in \mathcal{V}$ **do**
  for $B_1 \subset \mathcal{V} \setminus \{i\}$ *s.t.* $|B_1| = d$ **do**
    PASSED $\leftarrow$ YES
    for $B_2 \subset \mathcal{V} \setminus \{B_1 \cup \{i\}\}$ *s.t.* $|B_2| = d$ **do**
      Compute $\hat{\beta}_{i,B_1B_2}$ following (9)
      Estimate $\hat{\kappa}_{ij}$ following (12)
      if $\max_{j \in B_2} \hat{\kappa}_{ij} > \kappa/2$ **then**
        PASSED $\leftarrow$ NO
        break
      end
    end
    if *PASSED = YES* **then**
      $\tilde{B}_i \leftarrow B_1$
      break
    end
  end
end
Phase 3: Eliminate non-edges
for $i \in \mathcal{V}$ **do**
  Choose any $B_2 \subset \mathcal{V} \setminus \{B_1 \cup \{i\}\}$ s.t. $|B_2| = d$
  for $j \in \tilde{B}_i$ **do**
    Compute $\hat{\kappa}_{ij}$ following (12)
  end
  $\hat{B}_i \leftarrow \{j \in \tilde{B}_i \mid \hat{\kappa}_{ij} > \frac{\kappa}{2}\}$
end
return $\hat{B}_i$ for $i \in \mathcal{V}$

Algorithm 1: DICE(p, d, κ)

Proposition 2 (Accuracy of $\hat{\Theta}_{ii}$) *Given $\epsilon > 0$, the diagonal entries reconstructed by (6) satisfies*

$$\frac{\Theta_{ii}}{1+\epsilon} \leq \hat{\Theta}_{ii} \leq \frac{\Theta_{ii}}{1-\epsilon}, \quad \forall i \in \mathcal{V}, \tag{15}$$

with probability at least $1 - \delta_1$ provided that the number of samples satisfies

$$n > d + \frac{8}{\epsilon^2} d \log p + \frac{8}{\epsilon^2} \log\left(\frac{2d}{\delta_1}\right). \tag{16}$$

Proposition 3 (Accuracy of $\hat{\beta}_{ij}$) *Given $\epsilon > 0$, the regression coefficients $\hat{\beta}_{ij}$ satisfy*

$$\left|\hat{\beta}_{ij} - \frac{\Theta_{ij}}{\Theta_{ii}}\right| \leq \epsilon \sqrt{\frac{\Theta_{jj}}{\Theta_{ii}}}, \quad \forall j \in A, \quad \forall A \subset \mathcal{V} \setminus i \quad \textit{s.t.} \quad B_i \subset A, \quad |A| = 2d, \tag{17}$$

with probability at least $1-\delta_2$*, provided that the number of samples satisfies*

$$n > 2d + \frac{8}{\epsilon^2} d \log p + \frac{4}{\epsilon^2} \log\left(\frac{2d}{\delta_2}\right). \tag{18}$$

The quantities $\hat{\beta}_{ij}$ *are computed as in* (9)*.*

We show that Propositions 2 and 3 (proved in the Appendix) are sufficient to prove Theorem 1.

Proof [Proof of Theorem 1] By using $\epsilon = \kappa/4$ and $\delta_1 = \delta_2 = \delta/2$ in Proposition 2 and Proposition 3, we get using the union bound and the lower bound on the number of samples n in (14), that the statements in (15) and (17) hold with probability at least $1-\delta$. The proof proceeds by examining all three phases of DICE.

Phase 1: Using $\epsilon = \kappa/4$ and (15) in Proposition 2, we have that

$$\frac{2}{3} \overset{(a)}{<} \sqrt{\frac{4-\kappa}{4+\kappa}} \leq \sqrt{\frac{\hat{\Theta}_{ii}}{\hat{\Theta}_{jj}} \frac{\Theta_{jj}}{\Theta_{ii}}} \leq \sqrt{\frac{4+\kappa}{4-\kappa}} \overset{(b)}{<} 2, \quad \forall i, j \in \mathcal{V}, \tag{19}$$

where (a) and (b) follow by using $\kappa \leq 1$.

Phase 2: To analyze the performance of this phase, we consider the two cases alluded to in Section 2 for each candidate neighborhood B_1 in the outer loop of phase 2 in DICE.

Case 1: $B_1 \subseteq B_i$. Since $\Theta_{ij} = 0$, for any B_2 in the inner for loop, we get using Proposition 3 that for all $j \in B_2$

$$|\hat{\beta}_{ij}| \leq \frac{\kappa}{4} \sqrt{\frac{\Theta_{jj}}{\Theta_{ii}}} \quad \text{since } \Theta_{ij} = 0. \tag{20}$$

Combining with (19), we get

$$\hat{\kappa}_{ij} = |\hat{\beta}_{ij}| \sqrt{\frac{\hat{\Theta}_{ii}}{\hat{\Theta}_{jj}}} < \frac{\kappa}{4} \times 2 = \frac{\kappa}{2}. \tag{21}$$

Therefore for the candidate B_1, the inner loop in DICE will terminate with PASSED = YES.

Case 2: $B_1 \nsubseteq B_i$. In this case, there exists $j \in B_i \setminus B_1$. Consider the case when in the inner loop we have B_2 such that $B_i \subset B_1 \cup B_2$. Then $j \in B_2$ and $j \notin B_1$. Repeating the previous calculation we have

$$\hat{\kappa}_{ij} = |\hat{\beta}_{ij}| \sqrt{\frac{\hat{\Theta}_{ii}}{\hat{\Theta}_{jj}}} > \left(\kappa - \frac{\kappa}{4}\right) \times \frac{2}{3} = \frac{\kappa}{2}. \tag{22}$$

Therefore for the candidate B_1, the inner loop in DICE will terminate with PASSED = NO.

Phase 3: By appending to B_1, any $B_2 \in \mathcal{V} \setminus \{\{i\} \cup B_1\}$ with $|B_2| = d$, and using the computations in (21) and (22), we get that for all $j \in B_1$,

$$\hat{\kappa}_{ij} > \kappa/2 \quad \text{if} \quad j \in B_i, \tag{23}$$

$$\hat{\kappa}_{ij} < \kappa/2 \quad \text{if} \quad j \notin B_i, \tag{24}$$

and the proof is complete. ■

3. SLICE: Reconstructing Gaussian Graphical Models with near optimal number of samples using Mixed Integer Quadratic Programming

In this section we state the details of the SLICE algorithm. SLICE trades-off some optimality with respect to sample complexity for better computational complexity and enable implementation using a mixed integer quadratic programming formulation. With the rapid progress in mixed integer programming technology, this offers a significant advantage over the exhaustive search required for DICE in terms of practical efficiency. The algorithm SLICE simply utilizes the Phase 1 (Section 2.1) of DICE followed by a variation of the product and threshold procedure in Phase 3 (Section 2.3) of DICE in order to eliminate non-edges and estimate the exact support. By skipping the iterative neighborhood testing in Phase 2 (Section 2.2), SLICE improves upon the computational efficiency of DICE, theoretically by a factor of p^d and more significantly in practice since it can be implemented as a mixed integer linear program, but with a penalty of an additional $\frac{1}{\kappa^2}$ factor in the required number of samples. The various phases of SLICE are described in the following sections.

3.1. Phase 1: Least Squares with ℓ_0-constraint

The first step of the algorithm is identical to that of DICE, but the purpose is different. While for DICE, the only purpose was to estimate the conditional variances, SLICE requires the estimates of the regression coefficients:

$$\hat{\beta}_i = \underset{\beta_i \in \mathbb{R}^{p-1}}{\operatorname{argmin}} \quad L_i(\beta_i, \hat{\Sigma}), \qquad \text{s.t.} \quad \|\beta_i\|_0 \leq d, \tag{25}$$

where $L_i(\beta_i, \hat{\Sigma})$ is defined in (7).

3.2. Phase 2: Estimate the support

Once the estimates $\hat{\beta}_i$ have been obtained for all $i \in \mathcal{V}$, we estimate the edge-set $\hat{\mathcal{E}}$ through the following thresholding procedure

$$\hat{\mathcal{E}} = \left\{ (i,j) \in \mathcal{V} \times \mathcal{V} : \sqrt{|\hat{\beta}_{ij} \times \hat{\beta}_{ji}|} > \kappa/2 \right\}. \tag{26}$$

The estimated graph is then declared as $\hat{\mathcal{G}} = (\mathcal{V}, \hat{\mathcal{E}})$.

3.3. Implementation as a mixed integer quadratic program

Phase 1 of the SLICE algorithm has a computational complexity of $O(p^{d+1})$ since it is equivalent to an exhaustive search over all possible size d neighborhood of each vertex $i \in \mathcal{V}$. The second step can be implemented with a much lower computational complexity of $O(pd)$ leading to an overall complexity of $O(p^{d+1})$.

However when d is not small enough, performing an exhaustive search can be prohibitively expensive. Instead, the problem can be reformulated as a Mixed Integer Quadratic Program (MIQP), which in practice is significantly faster, especially when using modern mixed integer solvers such as CPLEX or GUROBI. In the context of compressive sensing and sparse regression, the use of MIQP

has been explored in Bertsimas et al. (2016) to solve a ℓ_0 constrained quadratic objective. We present one such formulation:

$$\min_{\beta_i \in \mathbb{R}^{p-1}} \quad \beta_i^T \hat{\Sigma}_{\bar{i}\bar{i}} \beta_i + 2\hat{\Sigma}_{\bar{i}i}\beta_i + \hat{\Sigma}_{ii} \tag{27a}$$

$$\text{s.t.} \quad s_{ij}L \leq \beta_{ij} \leq s_{ij}U, \quad \forall j \neq i \tag{27b}$$

$$\sum_{j \neq i} s_{ij} = d, \tag{27c}$$

$$s_{ij} \in \{0,1\}, \quad \forall j \neq i. \tag{27d}$$

In the above L and U denote known or estimated upper and lower bounds on the regression variables. For a more detailed discussion on obtaining these bounds, and formulations that avoid them, we refer the reader to Bertsimas et al. (2016).

3.4. Sample complexity of SLICE

In this subsection, we state the theoretical result regarding the sample complexity of SLICE.

Theorem 4 (Sample complexity of SLICE) *Given $\delta > 0$, the probability of perfect graph reconstruction using* SLICE *is lower bounded as $\mathbf{P}(\hat{\mathcal{G}} = \mathcal{G}) > 1 - \delta$, provided that the number of samples satisfies*

$$n > d + \frac{32}{\kappa^4} \log\left(\frac{4p^{d+1}}{\delta}\right). \tag{28}$$

SLICE retains a critical advantage of DICE, which is its insensitivity to parameters absent in the IT lower bound 2. In the next subsection, we demonstrate this advantage of SLICE through some illustrative numerical examples.

3.5. Numerical illustration of condition number independence of SLICE

In this section, we construct a very simple counterexample, consisting of a sequence of matrices with growing condition number $\frac{\lambda_{max}}{\lambda_{min}}$ but fixed minimum normalized edge strength κ. The primary purpose of this experiment is to demonstrate that the sample complexity of existing reconstruction algorithms are indeed sensitive to the condition number of the precision matrix Θ, whereas the sample complexity dictated by the IT lower bound in (2) as well as our proposed algorithm SLICE shows no such dependence.

The counter example sequence inspired by (4) consists of a triangle with two weak links and one stronger link and a collection of independent nodes. This family of GGMs is parametrized by the following inverse covariance matrix,

$$\Theta_{\kappa,\epsilon,\sigma} = \begin{bmatrix} 1 & \kappa & \kappa & 0 \\ \kappa & 1 & 1-\epsilon & 0 \\ \kappa & 1-\epsilon & 1 & 0 \\ 0 & 0 & 0 & \frac{1}{\sigma^2} I_{(p-3)\times(p-3)} \end{bmatrix}, \tag{29}$$

where $1-\epsilon$ is the strength of the strong link, $\kappa < 1-\epsilon$ is the strength of the weak links and σ^2 is the variance of the independent nodes. This family of graphs are chosen such that κ in (29) corresponds

to the minimum normalized edge strength in (3). Note that the maximum degree is $d = 2$. This problem can be interpreted as detecting a triangle within a cloud of independent nodes, a situation that is very plausible in practice.

The simulations are performed for matrix dimension $p = 200$ and $n = 175$ samples which satisfies $n < p$. We repeat the reconstruction procedure 50 times with independent samples for different values of $\sigma^2 \in \{1, \ldots, 10^4\}$ while $\kappa = 0.4$ and $\epsilon = 0.01$ are fixed. The regularizer parameters in ACLIME, LASSO and GRAPH LASSO have been optimized to yield the best possible results for each value of σ^2, an advantage that cannot be availed in practice. SLICE inherently does not have this issue. For each algorithm we compute its estimate $\hat{\kappa}_{12}$ and $\hat{\kappa}_{14}$ of the normalized link values (1,2) and (1,4),

$$\kappa_{12} = \sqrt{\frac{\Theta_{12}\Theta_{21}}{\Theta_{11}\Theta_{22}}}, \quad \kappa_{14} = \sqrt{\frac{\Theta_{14}\Theta_{41}}{\Theta_{11}\Theta_{44}}}. \tag{30}$$

We declare that an algorithm fails to reconstruct the graph whenever $\hat{\kappa}_{12} \leq \hat{\kappa}_{14}$: if this condition is satisfied, then links (1,2) or/and (1,4) are incorrectly reconstructed regardless of the thresholding procedure. Note that this choice of reconstruction failure criterion is quite generous. It is very unlikely that one can devise a successful thresholding procedure solely based on the criterion $\hat{\kappa}_{12} \leq \hat{\kappa}_{14}$ when $\hat{\kappa}_{12}$ and $\hat{\kappa}_{14}$ are close to each other. This is particularly true for several reconstructions provided by GRAPH LASSO and ACLIME as illustrated in Figure 1, whereas the procedure appears to provide no advantage to SLICE. Note that we compare normalized link strengths κ_{ij}, which are invariant to rescaling of the Θ matrix, instead of matrix element ratios β_{ij} or matrix elements Θ_{ij}. Thus reconstruction based on the latter quantities would fail for some rescaling of Θ.

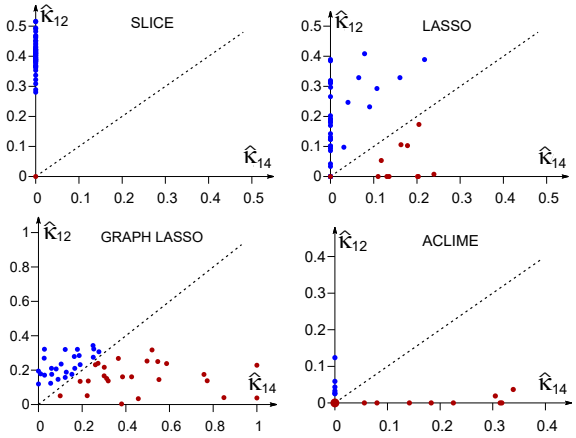

Figure 1: **Illustration of reconstruction failure for $\sigma^2 = \sqrt{1000}$.** We show the scatter plot of reconstructed values $\hat{\kappa}_{12}$ and $\hat{\kappa}_{14}$ obtained through 50 trial reconstructions. We declare that reconstruction fails when $\hat{\kappa}_{12} < \hat{\kappa}_{14}$ which corresponds to points in the lower right part of the graphs (highlighted in red). SLICE demonstrates an almost ideal behavior with $\hat{\kappa}_{14} = 0$ and $\hat{\kappa}_{12} > \kappa/2 = 0.2$. Note that the other algorithms often yield $\hat{\kappa}_{12} < \hat{\kappa}_{14}$ although (1,4) is not an existing edge.

Simulation results are summarized in Figure 2 where the probability of failure is plotted against the variance σ^2 of the independent nodes. For σ^2 close to one, all four algorithms succeed with high-probability and are able to correctly identify that there is a link $(1, 2)$ and no link between $(1, 4)$. However for larger value of σ^2, the probability of failure of ACLIME, LASSO and GRAPH LASSO is close to one while SLICE remains insensitive to changes in σ^2. This simple example

highlights that when the sample complexity of algorithms depends on parameters not present in the information theoretic bound, the graph reconstruction can be adversarially affected even by the presence of additional independent nodes.

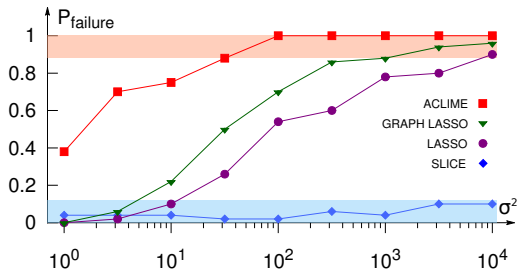

Figure 2: **Detecting a triangle in a cloud of independent nodes.** Empirical probability of failure averaged over 50 trial reconstructions with a fixed number of samples $n = 175$. Optimal regularization hyperparameters are used for all algorithms except SLICE. All algorithms except SLICE fail for large σ^2 values.

In Appendix B, we conduct additional numerical studies on synthetic and real data that show that the use of modern MIQP solvers allows one to scale up SLICE even to relatively large problems.

4. Conclusions

In this paper, we propose the polynomial-time algorithm DICE that provably recovers the support of sparse Gaussian graphical models with an information-theoretic optimal number of samples. On the theoretical side, this result confirms that the incoherence properties and condition number of the precision matrix are not necessary for the reconstruction task, and that the previously derived information-theoretic bound Wang et al. (2010) is tight. From the algorithmic perspective, reconstruction with the least number of samples is critical when the available data is scarce. Hence, even though the computational time of DICE can be large, it might still represent a valuable tool in the applications where the cost of additional data collection is larger than the cost of computations and where we expect the condition number of the precision matrix to be large. We also propose a simplified algorithm called SLICE with slightly higher sample complexity than DICE but with better computational complexity and possibility of implementation as a mixed integer quadratic program, making it attractive in practice. Importantly, like for DICE, the sample complexity of SLICE is also independent of any spurious quantities such as the condition number of the precision matrix.

Since we have now established that learning GGMs with an information-theoretic optimal number of samples given in (2) is achievable, the challenge for future work is to design new algorithms that improve the computational complexity of DICE and SLICE while still keeping the sample complexity optimal. One step in this direction has been made in a recent work Kelner et al. (2019) that shows that it is possible to efficiently learn attractive and walk-summable subclasses of GGMs by keeping sample complexity near-optimal. In future work, it would be interesting to see if the ideas behind assumption-free algorithms for reconstructing discrete graphical models such as Vuffray et al. (2016); Lokhov et al. (2018); Vuffray et al. (2019) could be extended to the case of GGMs. From a theoretical point of view, a fundamental question remains open – what is the minimal computational complexity of any algorithm that can achieve the information-theoretic optimal sample complexity for general Gaussian graphical models?

Acknowledgments

Research presented in this article was supported by the Laboratory Directed Research and Development program of Los Alamos National Laboratory under project numbers 20190059DR, 20180468ER, 20190195ER, 20190351ER, 20200121ER.

References

Animashree Anandkumar, Vincent YF Tan, Furong Huang, and Alan S Willsky. High-dimensional gaussian graphical model selection: Walk summability and local separation criterion. *Journal of Machine Learning Research*, 13(Aug):2293–2337, 2012.

Katia Basso, Adam A Margolin, Gustavo Stolovitzky, Ulf Klein, Riccardo Dalla-Favera, and Andrea Califano. Reverse engineering of regulatory networks in human b cells. *Nature genetics*, 37(4): 382–390, 2005.

Dimitris Bertsimas, Angela King, Rahul Mazumder, et al. Best subset selection via a modern optimization lens. *The Annals of Statistics*, 44(2):813–852, 2016.

Guy Bresler, Elchanan Mossel, and Allan Sly. Reconstruction of markov random fields from samples: Some observations and algorithms. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 343–356. Springer, 2008.

Peter Bühlmann, Markus Kalisch, and Lukas Meier. High-dimensional statistics with a view toward applications in biology. *Annual Review of Statistics and Its Application*, 1:255–278, 2014.

T Tony Cai, Weidong Liu, and Harrison H Zhou. Estimating sparse precision matrix: Optimal rates of convergence and adaptive estimation. *The Annals of Statistics*, 44(2):455–488, 2016.

Tony Cai, Weidong Liu, and Xi Luo. A constrained ℓ_1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494):594–607, 2011.

Alexandre d'Aspremont, Onureena Banerjee, and Laurent El Ghaoui. First-order methods for sparse covariance selection. *SIAM Journal on Matrix Analysis and Applications*, 30(1):56–66, 2008.

Iain Dunning, Joey Huchette, and Miles Lubin. Jump: A modeling language for mathematical optimization. *SIAM Review*, 59(2):295–320, 2017.

Nir Friedman. Inferring cellular networks using probabilistic graphical models. *Science*, 303(5659): 799–805, 2004.

Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2016. URL http://www.gurobi.com.

Shuai Huang et al. Learning brain connectivity of Alzheimer's disease by sparse inverse covariance estimation. *NeuroImage*, 50(3):935–949, 2010.

Christopher Johnson, Ali Jalali, and Pradeep Ravikumar. High-dimensional sparse inverse covariance estimation using greedy methods. In *Artificial Intelligence and Statistics*, pages 574–582, 2012.

David T Jones, Daniel WA Buchan, Domenico Cozzetto, and Massimiliano Pontil. Psicov: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics*, 28(2):184–190, 2012.

Markus Kalisch and Peter Bühlmann. Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *Journal of Machine Learning Research*, 8(Mar):613–636, 2007.

Jonathan Kelner, Frederic Koehler, Raghu Meka, and Ankur Moitra. Learning some popular gaussian graphical models without condition number bounds. *arXiv preprint arXiv:1905.01282*, 2019.

Andrey Y Lokhov, Marc Vuffray, Sidhant Misra, and Michael Chertkov. Optimal structure and parameter learning of ising models. *Science Advances*, 4(3):e1700791, 2018.

Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. *The annals of statistics*, pages 1436–1462, 2006.

Patricia Menéndez, Yiannis AI Kourmpetis, Cajo JF ter Braak, and Fred A van Eeuwijk. Gene regulatory networks from multifactorial perturbations using graphical lasso: application to the dream4 challenge. *PloS one*, 5(12):e14147, 2010.

Diane Valerie Ouellette. Schur complements and statistics. *Linear Algebra and its Applications*, 36: 187–295, 1981.

Pradeep Ravikumar, Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Model selection in gaussian graphical models: High-dimensional consistency of ℓ_1-regularized MLE. In *Advances in Neural Information Processing Systems 21*, pages 1329–1336. 2009.

Peter Spirtes, Clark N Glymour, and Richard Scheines. *Causation, prediction, and search*. MIT press, 2000.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

Sara Van de Geer, Peter Bühlmann, et al. ℓ_0-penalized maximum likelihood for sparse directed acyclic graphs. *The Annals of Statistics*, 41(2):536–567, 2013.

Gael Varoquaux, Alexandre Gramfort, Jean-Baptiste Poline, and Bertrand Thirion. Brain covariance selection: better individual functional connectivity models using population prior. In *Advances in Neural Information Processing Systems 23*, pages 2334–2342. 2010.

Marc Vuffray, Sidhant Misra, Andrey Lokhov, and Michael Chertkov. Interaction screening: Efficient and sample-optimal learning of Ising models. In *Advances in Neural Information Processing Systems 29*, pages 2595–2603. 2016.

Marc Vuffray, Sidhant Misra, and Andrey Y Lokhov. Efficient learning of discrete graphical models. *arXiv preprint arXiv:1902.00600*, 2019.

Lingxiao Wang, Xiang Ren, and Quanquan Gu. Precision matrix estimation in high dimensional gaussian graphical models with faster rates. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 177–185, 2016.

Wei Wang, Martin J Wainwright, and Kannan Ramchandran. Information-theoretic bounds on model selection for gaussian markov random fields. In *IEEE International Symposium on Information Theory Proceedings (ISIT)*, pages 1373–1377, 2010.

Ming Yuan and Yi Lin. Model selection and estimation in the gaussian graphical model. *Biometrika*, pages 19–35, 2007.

Appendix A. Proof of main results

In this section, we prove essential propositions and Theorem 4.

A.1. Proof of Proposition 2

To prove Proposition 2, we make use of the following lemma regarding the statistical fluctuations of the various conditional variances involved in (6).

Lemma 5 (Large deviations on $L_i^*(.,\hat{\Sigma})$) *Let $0 < \epsilon < 1$ be given. Then for every $i \in \mathcal{V}$ and every subset $A \subset [p] \setminus \{i\}$ with $|A| = d$, we have*

$$(1-\epsilon)L_i^*(A,\Sigma) \leq L_i^*(A,\hat{\Sigma}) \leq (1+\epsilon)L_i^*(A,\Sigma), \tag{31}$$

with probability at least $1 - 2p\binom{p-1}{d}e^{-(n-d)\epsilon^2/8}$. Here $L_i^*(.,.)$ is defined as in (10).

The proof of the above lemma is deferred to Appendix C. We now show that Proposition 2 follows from Lemma 5.

Proof [Proof of Proposition 2] Fix i and consider a subset $A \subseteq \mathcal{V} \setminus \{i\}$ and $|A| = d$ such that $B_i \subseteq A$. Since $B_i \subseteq A$, we have that $L_i^*(A,\Sigma) = 1/\Theta_{ii}$. Further, using Lemma 5 we get that

$$L_i^*(A,\hat{\Sigma}) \leq (1+\epsilon)L_i^*(A,\Sigma) = \frac{1+\epsilon}{\Theta_{ii}}. \tag{32}$$

We consider the reformulation of (6) as given in (8). Since A is feasible for (8), we must have

$$\frac{1}{\hat{\Theta}_{ii}} \leq \frac{1+\epsilon}{\Theta_{ii}}. \tag{33}$$

For all $A \subseteq \mathcal{V} \setminus \{i\}$ with $|A| = d$, we have

$$L_i^*(A,\Sigma) = \mathbf{Var}\left(X_i \mid X_A\right) \overset{(a)}{\geq} \mathbf{Var}\left(X_i \mid X_{[p]\setminus\{i\}}\right) \overset{(b)}{=} \mathbf{Var}\left(X_i \mid X_{B_i}\right) = \frac{1}{\Theta_{ii}}, \tag{34}$$

where (a) follows from the well-known property of multivariate gaussians that conditioning reduces variance, and (b) follows from the so-called *separation property* of graphical models. Using Lemma 5, this shows that

$$\frac{1}{\hat{\Theta}_{ii}} = \min_{A \subseteq [p]\setminus i \,:\, |A|=d} L_i^*(A,\hat{\Sigma}) \geq \min_{A \subseteq [p]\setminus i \,:\, |A|=d} (1-\epsilon)L_i^*(A,\Sigma) \geq \frac{1-\epsilon}{\Theta_{ii}}. \tag{35}$$

The proof follows by combining (33) and (35). ■

A.2. Proof of Proposition 3

We first state the essential technical lemmas that form the ingredients of the proof.

Lemma 6 *Fix $i \in \mathcal{V}$ and $A \subseteq \mathcal{V}\{i\}$ such that $B_i \subseteq A$ and $|A| = 2d$. The conditional distribution of $\hat{\beta}_{ij}$ for any $j \in A$ is given by*

$$\hat{\beta}_{ij} \mid \hat{\Sigma}_{AA} \sim \mathcal{N}\left(\frac{\Theta_{ij}}{\Theta_{ii}}, \Theta_{ii}^{-1}\left(\hat{\Sigma}_{AA}^{-1}\right)_{jj}\right), \tag{36}$$

where $\mathcal{N}(.,.)$ denotes the normal distribution.

Lemma 7 *Fix $i \in \mathcal{V}$ and $A \subseteq \mathcal{V} \setminus \{i\}$ such that $B_i \subseteq A$ and $|A| = 2d$. Then for any $\epsilon > 0$ the random variable $\left(\hat{\Sigma}^{-1}_{AA}\right)_{jj}$ satisfies the following inequality*

$$\mathbb{P}\left(\left[\hat{\Sigma}^{-1}_{AA}\right]_{jj} > (1-\epsilon)^{-1}\Theta_{jj}\right) \leq e^{-\frac{(n-2d+1)\epsilon^2}{8}}. \tag{37}$$

These lemmas are proved in Appendix C.

Proof [Proof of Proposition 3] Define the event $E = \left[\hat{\Sigma}^{-1}_{\hat{B}_i\hat{B}_i}\right]_{jj} \leq (1-\epsilon_1)^{-1}\Theta_{jj}$. We bound the deviation of $\hat{\beta}_{ij}$ from $\frac{\Theta_{ij}}{\Theta_{ii}}$ as

$$\begin{aligned}
\mathbb{P}\left(|\hat{\beta}_{ij} - \frac{\Theta_{ij}}{\Theta_{ii}}| \geq \epsilon\sqrt{\frac{\Theta_{jj}}{\Theta_{ii}}}\right) &= \mathbb{P}\left(|\hat{\beta}_{ij} - \frac{\Theta_{ij}}{\Theta_{ii}}| \geq \epsilon\sqrt{\frac{\Theta_{jj}}{\Theta_{ii}}} \mid E\right)\mathbb{P}(E) \\
&\quad + \mathbb{P}\left(|\hat{\beta}_{ij} - \frac{\Theta_{ij}}{\Theta_{ii}}| \geq \epsilon\sqrt{\frac{\Theta_{jj}}{\Theta_{ii}}} \mid E^c\right)\mathbb{P}(E^c) \\
&\overset{(a)}{\leq} 2\Phi^c\left(\epsilon\sqrt{1-\epsilon_1}\sqrt{n}\right) + e^{-(n-2d+1)\epsilon_1^2/8},
\end{aligned} \tag{38}$$

where (a) follows by bounding the first term using Lemma 6 and the definition of E, and bounding the second term by the probability of the event E using Lemma 7. Setting $\epsilon_1 = 2\epsilon$, we get

$$\begin{aligned}
\mathbb{P}\left(|\hat{\beta}_{ij} - \frac{\Theta_{ij}}{\Theta_{ii}}| \geq \epsilon\sqrt{\frac{\Theta_{jj}}{\Theta_{ii}}}\right) &\leq 2\Phi^c\left(\epsilon\sqrt{1-2\epsilon}\sqrt{n}\right) + e^{-(n-2d+1)\epsilon^2/2} \\
&\leq \frac{2}{\sqrt{2\pi}}\frac{e^{-\epsilon^2(1-2\epsilon)n/2}}{\epsilon\sqrt{1-2\epsilon}\sqrt{n}} + e^{-(n-2d+1)\epsilon^2/2} \\
&\overset{(a)}{\leq} e^{-(\epsilon^2 n/4)} + e^{-(n-2d+1)\epsilon^2/2} \\
&\leq 2e^{-(n-2d+1)\epsilon^2/4},
\end{aligned}$$

where (a) follows by using $\epsilon \leq 1/4$ and $n > \frac{2}{\epsilon^2}$. Using the union bound, we have that for all $i \in \mathcal{V}$ and all $A \subset \mathcal{V} \setminus i$ such that $B_i \subset A$ and $|A| = 2d$,

$$\mathbb{P}\left(\left|\hat{\beta}_{ij} - \frac{\Theta_{ij}}{\Theta_{ii}}\right| \leq \epsilon\sqrt{\frac{\Theta_{jj}}{\Theta_{ii}}}\right) \leq 2pd\binom{p-1}{2d}e^{-(n-2d+1)\epsilon^2/4} \leq \delta_2, \tag{39}$$

where the last inequality follows from the assumption on n given in (18). ■

A.3. Proof of Theorem 4

We prove Theorem 4 through the results below that provide guarantees for each step of the SLICE estimator.

Proposition 8 (Optimal support contains the true support) *For each $i \in \mathcal{V}$, let $\hat{B}_i \subset [p]$ be the support of the optimal solution in (25) and let $B_i \subset [p]$ be the neighbors of i. Then for any $\delta > 0$,*

the support $\hat{B}_i$ satisfies $B_i \subseteq \hat{B}_i$ with probability greater than $1-\delta/2$, provided that the number of samples satisfies

$$n - d > \frac{32}{\kappa^4} \log\left(\frac{4p^{d+1}}{\delta}\right). \tag{40}$$

Proposition 9 (Post-processing Proposition) *Assume that*

$$n - d > \frac{64}{\kappa^2} \log\left(\frac{8dp}{\delta}\right). \tag{41}$$

Then with probability greater than $1-\delta/2$, the post processing procedure consisting of Product and Threshold terminates with exactly the correct support.

Proof [Proof of Theorem 4] The result follows by combining Proposition 8 and Proposition 9 and applying the union bound. ■

The two propositions above can be proved by reusing Lemmas 5,6,7 in Section A.2 and the following lemma, the proof of which is provided in Appendix C.

Lemma 10 (Multiplicative gap in noiseless optimal solutions) *Fix $i \in \mathcal{V}$ and let $B_i \subset [p]$ be the neighbors of i. Let $\hat{B} \subset [p]$ be any subset such that $|\hat{B}| = d$ and $B_i \not\subseteq \hat{B}$. Then*

$$L_i^*(\hat{B}, \Sigma) \geq L_i^*(B_i, \Sigma)(1-\kappa^2)^{-1}. \tag{42}$$

Proof [Proof of Proposition 8] Combining Lemma 10 and Lemma 5 and using $\epsilon = \kappa^2/2$ we have for any $i \in \mathcal{V}$ that the sequence of inequalities

$$\begin{aligned} L_i^*(B_i, \hat{\Sigma}) &< (1+\epsilon) L_i^*(B_i, \Sigma) < (1+\epsilon)(1-\kappa^2) L_i^*(\hat{B}, \Sigma) \\ &< \frac{(1-\kappa^2)(1+\epsilon)}{1-\epsilon} L_i^*(\hat{B}_i, \hat{\Sigma}) < L_i^*(\hat{B}, \hat{\Sigma}), \end{aligned}$$

is satisfied for all $\{\hat{B} \subset [p] \backslash \{i\} : |\hat{B}| = d,\ B_i \not\subseteq \hat{B}\}$ with probability at least $1 - 2p\binom{p-1}{d} e^{-(n-d)\kappa^4/32}$. Therefore

$$\mathbf{P}\left(\exists i \in [p] : B_i \not\subseteq \hat{B}_i\right) < 2p\binom{p-1}{d} e^{-(n-d)\kappa^4/32} < \delta/2,$$

where the last inequality follows from $n - d > \frac{32}{\kappa^4} \log\left(\frac{4p^{d+1}}{\delta}\right)$. ■

Proof [Proof of Proposition 9] Similar to the proof of Proposition 3, using Lemma 6,7 we get that for all $i \in \mathcal{V}$ and all $A \subset \mathcal{V} \setminus i$ such that $B_i \subset A$ and $|A| = d$,

$$\mathbb{P}\left(\left|\hat{\beta}_{ij} - \frac{\Theta_{ij}}{\Theta_{ii}}\right| \leq \epsilon \sqrt{\frac{\Theta_{jj}}{\Theta_{ii}}}\right) \leq 2pd\binom{p-1}{d} e^{-(n-d+1)\epsilon^2/4}, \tag{43}$$

Using $\epsilon = \kappa/4$ we get

$$\mathbf{P}\left(|\hat{\beta}_{ij} - \frac{\theta_{ij}}{\theta_{ii}}| \leq \frac{\kappa}{4}\sqrt{\frac{\theta_{jj}}{\theta_{ii}}} \quad \forall j \in \hat{B}_i, i \in \mathcal{V}\right) \geq 1 - 2dpe^{-(n-d+1)\kappa^2/64} \overset{(a)}{\geq} 1 - \frac{\delta}{2}, \tag{44}$$

where the implication (a) is obtained by using $n - d > \frac{64}{\kappa^2} \log\left(\frac{8dp}{\delta}\right)$ in the premise of Proposition 9. Using (44) for both i and j we get with probability greater than least $1 - \frac{\delta}{2}$,

$$|\hat{\beta}_{ij}\hat{\beta}_{ji}| \geq \left(\frac{|\theta_{ij}|}{\theta_{ii}} - \frac{\kappa}{4}\sqrt{\frac{\theta_{jj}}{\theta_{ii}}}\right)\left(\frac{|\theta_{ij}|}{\theta_{jj}} - \frac{\kappa}{4}\sqrt{\frac{\theta_{ii}}{\theta_{jj}}}\right) = \left(\frac{|\theta_{ij}|}{\sqrt{\theta_{ii}\theta_{jj}}} - \frac{\kappa}{4}\right)^2. \tag{45}$$

From (45), we get that for $(i,j) \in \mathcal{E}$ the estimates satisfy $\sqrt{|\hat{\beta}_{ij}||\hat{\beta}_{ji}|} \geq 3\kappa/4 > \kappa/2$. An identical argument can be used to show that $\sqrt{|\hat{\beta}_{ij}||\hat{\beta}_{ji}|} \leq \kappa/4 < \kappa/2$. This proves that the post-processing step recovers the exact support.

■

Appendix B. Tests of SLICE scalability on synthetic and real data

In this section, we present several tests on synthetic and real data. We present examples to illustrate that the use of modern Mixed-Integer Quadratic Programming (MIQP) solvers such as Gurobi Gurobi Optimization (2016) allows one to run SLICE in a reasonable time even on relatively large realistic problems. As a first test, we run SLICE on synthetic random graph instances of different degrees ($d = 3$ and $d = 4$) and sizes ($p = 10$, $p = 100$ and $p = 1000$). The link strengths κ_{ij} have been randomly generated in the ranges $[0.2, 0.4]$ for $d = 3$ and $[0.2, 0.3]$ for $d = 4$ instances. The family of regular random graphs has been chosen to eliminate potential dependencies on the heterogeneity in the degree distributions. For implementation, we used one possible MIQP formulation presented in the Supplementary Material, and the JuMP framework Dunning et al. (2017) in julia for running the Gurobi solver. The running times for SLICE with $n = 10^4$ samples for each problem instance are presented in the Table 1. Notice that the practical scaling of running times is significantly better than what one would expect from the worst-case complexity $O(p^{d+1})$ for the full graph reconstruction.

Table 1: **Comparison of running times for SLICE on various regular random graphs with $n = 10^4$ samples**: Longest MIQP Gurobi solver time and longest total running time for reconstruction of the neighborhood of one node, and total time for learning the entire graph.

GRAPH (p, d)	MAX FOR 1 NODE (GUROBI)	MAX FOR 1 NODE (TOTAL)	FULL PROBLEM (TOTAL)
$(10, 3)$	0.01 SEC	2.7 SEC	7.2 SEC
$(10, 4)$	0.03 SEC	2.8 SEC	7.6 SEC
$(100, 3)$	0.03 SEC	2.7 SEC	19.8 SEC
$(100, 4)$	0.04 SEC	2.8 SEC	21.7 SEC
$(1000, 3)$	15.7 SEC	19.3 SEC	18 HOURS
$(1000, 4)$	92.3 SEC	96 SEC	29.3 HOURS

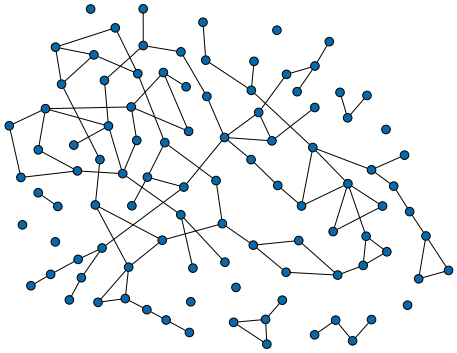

Figure 3: **Graph learned with SLICE from Riboflavin data set.** This real-world data set Bühlmann et al. (2014) contains $p = 101$ variables and $n = 71$ samples. In the reconstruction procedure, the maximum degree has been set to $d = 6$.

For the illustration on real data, we use the biological data set related to the Riboflavin production with *B. subtilis*. This data set contains the logarithm of the Riboflavin production rate alongside the logarithms of normalized expression levels of 100 genes that are most responsive to the Riboflavin production. Hybridization under different fermentation conditions lead to the acquisition of $n = 71$ samples, see Bühlmann et al. (2014) for more details and raw data. The graph reconstructed with SLICE and constraint $d = 6$ is depicted in the Figure 3. It took about 2.5 days for the algorithm to learn this graph (with the proof of optimality of the obtained solution) in this high-dimensional regime. Notice that again the practical running time for SLICE using MIQP technology is much lower than the one required to search over the 10^{14} candidate neighborhoods of size $d = 6$. This example is a perfect illustration of a trade-off between sample and algorithmic complexity in real-world problems where the collection of samples might be very costly.

Appendix C. Proof of technical lemmas

We will need the following result from Ouellette (1981) in the proofs of the technical lemmas.

Lemma 11 (Ouellette (1981) Eq. 6.78) *Let $X \in \mathbb{R}^{k\times k} \sim W(V, l)$ be a random matrix distributed according to the Wishart distribution with parameter $V \succ 0$ and order $l > k - 1$. Let $Y = X^{-1}$ with $Y \sim W^{-1}(U, l)$ where $U = V^{-1}$. Let*

$$X = \begin{bmatrix} X_{11} & X_{12} \\ X_{21} & X_{22} \end{bmatrix}, \; V = \begin{bmatrix} V_{11} & V_{12} \\ V_{21} & V_{22} \end{bmatrix}$$

be any compatible block matrix representation of X and V. Consider block representations for Y, U with the same dimensions k_1, k_2 that satisfy $k_1 + k_2 = k$. Then,

(a) The Schur complements of X_{11} and Y_{11} are distributed as

$$X_{11} - X_{12}X_{22}^{-1}X_{21} \sim W(V_{11} - V_{12}V_{22}^{-1}V_{21}, l - k_2)$$
$$Y_{11} - Y_{12}Y_{22}^{-1}Y_{21} = X_{11}^{-1} \sim W^{-1}(V_{11}^{-1}, l) = W^{-1}(U_{11} - U_{12}U_{22}^{-1}U_{21}, l),$$

(b) The random matrix $Y_{22}^{-1}Y_{21}$ conditioned on X_{11}^{-1} is distributed as a matrix normal distribution

$$Y_{22}^{-1}Y_{21} \mid X_{11}^{-1} \sim \mathcal{N}\left(U_{22}^{-1}U_{21}, X_{11}^{-1} \otimes U_{22}^{-1}\right).$$

Proof [Proof of Lemma 5] Fix $i \in \mathcal{V}$ and $A \subset [p] \setminus \{i\}$ with $|A| = d$. Then using properties of the Wishart distribution in Lemma 11 part (a), we get that,

$$\begin{aligned}\hat{\Sigma}_{ii} - \hat{\Sigma}_{iA}\hat{\Sigma}_{AA}^{-1}\hat{\Sigma}_{Ai} &\sim (\Sigma_{ii} - \Sigma_{i,A}\Sigma_{AA}^{-1}\Sigma_{Ai})\chi^2_{n-d} \\ &= L_i^*(A,\Sigma)\chi^2_{n-d}, \end{aligned} \tag{46}$$

where χ^2_t denotes the standard Chi-squared distribution with t degrees of freedom. Using the Chernoff bound,

$$\mathbb{P}(\chi^2_{n-d} > 1+\epsilon) < e^{-(n-d)\left(\frac{\epsilon}{2} - \frac{1}{2}\log(1+\epsilon)\right)} < e^{-(n-d)\epsilon^2/8}, \tag{47}$$

$$\mathbb{P}(\chi^2_{n-d} < 1-\epsilon) < e^{-(n-d)\left(-\frac{1}{2}\log(1-\epsilon) - \frac{\epsilon}{2}\right)} < e^{-(n-d)\epsilon^2/8}. \tag{48}$$

The proof is completed by using the union bound for all $A \subset [p] \setminus i$ with $|A| = d$ and over all $i \in \mathcal{V}$. ■

Proof [Proof of Lemma 6] For any $i \in \mathcal{V}$ and $A \subseteq \mathcal{V} \setminus i$ with $B_i \subseteq A$ and $|A| = 2d$, let

$$\hat{\psi}_{(iA)(iA)} = \left(\hat{\Sigma}_{(iA)(iA)}\right)^{-1}, \quad \Psi_{(iA)(iA)} = \left(\Sigma_{(iA)(iA)}\right)^{-1}. \tag{49}$$

Using the block matrix decomposition for matrix inverse

$$\Psi_{(iA)(iA)} = \Theta_{(iA)(iA)} - \Theta_{(iA)D}\Theta_{DD}^{-1}\Theta_{D(iA)}, \quad \text{where} \quad D = \mathcal{V} \setminus \{i \cup A\}. \tag{50}$$

Since $B_i \subseteq A$, we must have $\Theta_{iD} = 0$. Hence the matrix $\Psi_{(iA)(iA)}$ satisfies

$$\Psi_{ii} = \Theta_{ii}, \quad \Psi_{ij} = \Theta_{ij}, \quad \forall j \in A. \tag{51}$$

From Lemma 11, part (b) we get that for all $j \in A$,

$$\hat{\beta}_{ij} \mid \hat{\Sigma}_{AA} \sim \mathcal{N}\left(\frac{\Psi_{ij}}{\Psi_{ii}}, \Psi_{ii}^{-1}\left[\hat{\Sigma}_{AA}^{-1}\right]_{jj}\right) \tag{52}$$

$$\overset{(a)}{=} \mathcal{N}\left(\frac{\Theta_{ij}}{\Theta_{ii}}, \Theta_{ii}^{-1}\left[\hat{\Sigma}_{AA}^{-1}\right]_{jj}\right), \tag{53}$$

where (a) follows from (51). ■

Proof [Proof of Lemma 7] From (50) we get that for all $j \in A$,

$$\Psi_{jj} = \Theta_{jj} - \Theta_{jD}\Theta_{DD}^{-1}\Theta_{Dj} \leq \Theta_{jj}. \tag{54}$$

From Lemma 11, the random matrix $\hat{\Sigma}_{AA}$ is distributed according to the Wishart distribution $\hat{\Sigma}_{AA} \sim W(\Sigma_{AA}, n)$. Hence for any $j \in A$,

$$\begin{aligned}\left(\left[\hat{\Sigma}_{AA}^{-1}\right]_{jj}\right)^{-1} &= \hat{\Sigma}_{jj} - \hat{\Sigma}_{j(A\setminus j)}\hat{\Sigma}_{(A\setminus j)(A\setminus j)}^{-1}\hat{\Sigma}_{(A\setminus j)j} \\ &\overset{(a)}{\sim} (\Sigma_{jj} - \Sigma_{j(A\setminus j)}\Sigma_{(A\setminus j)(A\setminus j)}^{-1}\Sigma_{(A\setminus j)j})\chi^2_{n-2d+1} \triangleq \alpha_j\chi^2_{n-d+1}, \end{aligned} \tag{55}$$

where (a) follows from Lemma 11 and we can bound the constant α_j from above for every $j \in A$ as

$$\alpha_j^{-1} = \left(\Sigma_{jj} - \Sigma_{j(A\setminus j)}\Sigma^{-1}_{(A\setminus j)(A\setminus j)}\Sigma_{(A\setminus j)j}\right)^{-1} = \Psi_{jj} - \Psi_{ji}^2\Psi_{ii}^{-1} \overset{(a)}{\leq} \Psi_{jj} \leq \Theta_{jj},$$

where (a) follows from (54). Hence,

$$\mathbb{P}\left(\left[\hat{\Sigma}^{-1}_{AA}\right]_{jj} > (1-\epsilon)^{-1}\Theta_{jj}\right) = \mathbb{P}\left(\chi^2_{n-2d+1} < (1-\epsilon)\alpha_j^{-1}\Theta_{jj}^{-1}\right) \tag{56}$$

$$\overset{(a)}{\leq} \mathbb{P}\left(\chi^2_{n-2d+1} < (1-\epsilon)\right) \tag{57}$$

$$\overset{(b)}{\leq} e^{-(n-2d+1)\epsilon^2/8}, \tag{58}$$

where (a) follows because by (56) we have $\alpha_j^{-1}\Theta_{jj}^{-1} \leq 1$, and (b) follows from (47). ■

Proof [Proof of Lemma 10] From (11) we have that

$$L_i^*(B_i, \Sigma) = \mathbf{Var}(X_i|X_{B_i}) \overset{(a)}{=} \mathbf{Var}(X_i|X_{[p]\setminus i}) \overset{(b)}{=} \theta_{ii}^{-1}, \tag{59}$$

where (a) follows from the separation property of graphical models, and (b) follows from (10). Similarly,

$$L_i^*(\hat{B}, \Sigma) = \mathbf{Var}(X_i|X_{\hat{B}}). \tag{60}$$

Using the law of total variance we get that

$$\begin{aligned}\mathbf{Var}(X_i|X_{\hat{B}}) &= \mathbf{E}\left[\mathbf{Var}(X_i|X_{B_i\cup\hat{B}})|X_{\hat{B}}\right] \\ &\quad + \mathbf{Var}\left(\mathbf{E}\left[X_i|X_{B_i\cup\hat{B}}\right]|X_{\hat{B}}\right) \\ &= \frac{1}{\theta_{ii}} + \frac{1}{\theta_{ii}^2}\mathbf{Var}\left(\sum_{j\in B_i\cup\hat{B}}\theta_{ij}X_j \mid X_{\hat{B}}\right).\end{aligned} \tag{61}$$

Let $u \in B_i \setminus \hat{B}$. From above, we get

$$\mathbf{Var}(X_i|X_{\hat{B}}) - \mathbf{Var}(X_i|X_{B_i}) \tag{62}$$

$$= \theta_{ii}^{-2}\mathbf{Var}\left(\sum_{j\in B_i\cup\hat{B}}\theta_{ij}X_j \mid X_{\hat{B}}\right)$$

$$\overset{(a)}{\geq} \theta_{ii}^{-2}\mathbf{Var}\left(\sum_{j\in B_i\cup\hat{B}}\theta_{ij}X_j \mid X_{[p]\setminus\{i,u\}}\right)$$

$$= \theta_{ii}^{-1}\left(\frac{\theta_{ii}\theta_{uu}}{\theta_{iu}^2} - 1\right)^{-1} \overset{(b)}{\geq} \mathbf{Var}(X_i|X_{B_i})\frac{\kappa^2}{1-\kappa^2}. \tag{63}$$

The inequality (a) follows from the fact that conditioning reduces variance in Gaussian and observing that $\hat{B} \subseteq [p] \setminus \{i, u\}$. The inequality (b) follows from (3). ■