Active Learning for Undirected Graphical Model Selection

Divyanshu Vats
Rice University

Robert D. Nowak
University of Wisconsin - Madison

Richard G. Baraniuk
Rice University

Abstract

This paper studies graphical model selection, i.e., the problem of estimating a graph of statistical relationships among a collection of random variables. Conventional graphical model selection algorithms are passive, i.e., they require all the measurements to have been collected before processing begins. We propose an active learning algorithm that uses junction tree representations to adapt future measurements based on the information gathered from prior measurements. We prove that, under certain conditions, our active learning algorithm requires fewer scalar measurements than any passive algorithm to reliably estimate a graph. A range of numerical results validate our theory and demonstrates the benefits of active learning.

1 Introduction

An important problem that arises in many applications is that of inferring the statistical relationships between a large collection of random variables. For example, the random variable could represent expression values of a gene, opinions of a person, or stock returns of a company. Graphical models compactly represent statistical relationships using a graph. The vertices in the graph represent random variables, and the edges in the graph represent statistical relationships between random variables [1]. Although the graph may be of three types, namely directed, undirected, or mixed, we only study undirected graphs here. Given measurements drawn from a graphical model, there are now several algorithms for estimating the graph of statistical relationships. See [2–4] for Gaussian graphical models, [5–7] for discrete graphical models, and [8] for nonparametric graphical models.

All conventional algorithms for learning graphical models are *passive*, i.e., they rely on all the measurements being collected before any processing begins. We envision several applications of active learning for graphical models, where future measurements are collected based on the information gathered from prior measurements and/or prior knowledge. For example, in gene expression analysis, once enough measurements have been obtained from a large collection of genes, subsequent measurements can be focused on a subset of genes with more complex interactions. In social network analysis, measurements can be focused on a small subset of people rather than all people in the social network.

Although there exists active learning algorithms for various statistical inference problems, including classification [9], sparse signal recovery [10], clustering [11], multiple testing [12], matrix completion [13], and causal structure discovery [14], the methods in these works do not necessarily apply to learning graphical models. Furthermore, although there exists methods for designing optimal experiments for learning statistical models [15], we are not aware of any work that studies active learning for graphical models.

In this paper, we propose an active learning algorithm for learning the structure of the graph in a graphical model. On a high level, our algorithm is summarized as follows. Suppose we have a large graph that is composed of two or more subgraphs that may have complicated structures themselves, but have relatively few edges between them. In principle it should be easier to identify the gross structure of the graph (i.e., the subsets of vertices corresponding to each subgraph and the few edges between these sets of vertices), then to identify the full graph structure. So we pursue a sequential and active approach to learn the graph.

First, we obtain full joint measurements of all the vertices and identify the gross structure. The gross structure allows us to partition the large graph into multiple subgraphs. We then identify the edges and the non-edges in each subgraph that have been estimated reliably. Next, we collect additional, focused measurements, over a subset of the vertices to identify the

Appearing in Proceedings of the 17^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2014, Reykjavik, Iceland. JMLR: W&CP volume 33. Copyright 2014 by the authors.

edges that could not be reliably estimated using the past measurements. The advantage of this sort of approach is that many of the measurements only involve a smaller subset of the vertices. For this reason, the total number of *scalar measurements* required for reliable graph estimation using this sort of active procedure can be significantly lower than the total number of scalar measurements required by conventional passive methods.

Theoretically, we establish sufficient conditions on the number of scalar measurements needed for reliable graph estimation using an active learning algorithm. Next, we analyze our algorithm when given additional knowledge about the absence of certain edges in the graph. We prove that, under certain favorable conditions, an active learning algorithm can estimate walk-summable Gaussian graphical models over p vertices using only $O(p_{\min}\theta_{\min}^{-2}\log p_{\min})$ scalar measurements, while any passive algorithm necessarily requires $O(p\theta_{\min}^{-2}\log p_{\min})$ scalar meausurements. Here, $p_{\min}$ is the size of the smallest cluster in a junction tree representation after incorporating the prior knowledge and $\theta_{\min}$ quantifies the intrinsic difficulty of the graphical model selection problem. The particular conditions in our analysis depend on the positioning certain *"weak edges"* in the graph and the scaling of the parameter $\theta_{\min}$. Finally, we empirically demonstrate the benefits of our algorithm using numerical simulations.

2 Undirected Graphical Models

An undirected graphical model is a joint probability distribution, say P_X, defined on a graph $G^* = (V, E(G^*))$, where $V = \{1, ..., p\}$ indexes the random vector $X = (X_1, ..., X_p)$. For any graph G, we use the notation $E(G)$ to denote its edges. The vertices V index the random variables and the edges $E(G^*)$ encode statistical relationships between the random variables. In particular, when $P_X > 0$, undirected graphical models can be characterized using Markov properties. One such Markov property is the *global Markov property* which says that whenever a set of vertices A and B are separated by S, then X_A is independent of X_B given X_S. Note that a set S separates A and B if all paths from A to B pass through S. In this paper, we consider the *graphical model selection problem* of estimating the unknown graph G^* given measurements drawn from the probability distribution P_X.

3 Active Learning Algorithm

In this section, we present our active learning algorithm for graphical model selection. Recall that our goal is to actively draw measurements from P_X. Section 3.1 discusses our algorithm. Section 3.2 discusses a key step in our algorithm that determines the future measurements given prior measurements.

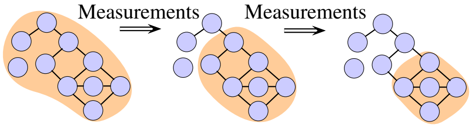


Figure 1: Shaded regions represent the active vertices. As measurements are acquired, the number of active vertices decrease.

3.1 Algorithm Overview

```
Algorithm 1: Active Learning
• Inputs: $A$, $\widehat{E}$, $\widehat{F}$, $q$, $K$, and $\delta$.
• Initialization: $\mathfrak{X} \leftarrow \emptyset$
• For $w = 1, 2, \ldots, K$
    – If $w = K$, then $\delta \leftarrow 1$
    – $m \leftarrow \lfloor \delta q/|A| \rfloor$ ; $q \leftarrow (1-\delta)q$
    – $\mathfrak{X}_A^m \leftarrow$ Draw $m$ i.i.d. samples from $P_{X_A}$.
    – Update measurements: $\mathfrak{X} \leftarrow \mathfrak{X} \cup \mathfrak{X}_A^m$
    – Update $A$, $\widehat{E}$, and $\widehat{F}$ using Algorithm 2.
• Estimate the remaining edges and combine with $\widehat{E}$ and $\widehat{F}$ to output $\widehat{G}$.
```

Algorithm 1, which can be seen as an extension of the active methods for sparse signal recovery [10, 12] applied to graphs, presents our active learning algorithm for graphical model selection with the following inputs:

- Active vertices A: We say that $A \subseteq V$ are *active vertices* if all edges and non-edges over A^c and those connecting A^c to A are *known.*
- Estimated edges $\widehat{E}$: Edges that have been estimated to be in the true graph.
- Estimated non-edges $\widehat{F}$: Edges that have been estimated to *not* be in the true graph.
- Measurement budget q: Total number of scalar measurements Algorithm 1 should draw from P_X.
- Number of measurement rounds K: Number of times Algorithm 1 draws measurements from P_X.
- Fraction of measurements δ: The fraction of scalar measurements drawn in each round.

The main idea in Algorithm 1 is to sequentially draw measurements from P_X and check for edges and non-edges that can be reliably estimated using prior measurements. Algorithm 1 initiates by drawing measurements from the active vertices A, where the number of measurements is determined by q and δ. Next, the sets A, $\widehat{E}$, and $\widehat{F}$ are updated using Algorithm 2, which is discussed in Section 3.2. In general, as illustrated in Figure 1, as measurements are acquired, the size of the set A decreases since parts of the graph are reliably estimated using prior measurements.

3.2 Finding Active Vertices

In this section, we discuss the challenging step in Algorithm 1 of updating the active vertices A, the edges

$\widehat{E}$, and the non-edges $\widehat{F}$. Our main idea is to estimate two graphs, H^+ and H^-, such that H^+ is likely to contain all the true edges and H^- is likely to contain a subset of the true edges. The edges $\widehat{E}$ and $\widehat{F}$ can then be identified from H^- and H^+, respectively. We now want to devise an algorithm to find the active vertices A given H^- and H^+. For a set U and a graph G, let $G[U]$ be the *induced subgraph* over U that contains all edges from G that only involve the vertices U. Note the following:

- If $H^+ = H^-$, we clearly do not need any more measurements.
- Suppose U and U' have the property that $U \backslash U'$ is separated from all other vertices. If $H^+[U] = H^-[U]$, then we must have that $G^*[U] = H^+[U] = H^-[U]$. In this case, there is no need to draw measurements from the vertices $U \backslash U'$ and the sets $\widehat{E}$ and $\widehat{F}$ can be modified accordingly. We may still need to draw measurements from U' since edges in other clusters may depend on U'.
- If $H^+[U] \neq H^-[U]$, all vertices over U may need to be observed further.

To identify appropriate sets U, we use junction tree representations of the graph H^+. Informally, a junction tree clusters vertices in a graph so that the resulting graph over the clusters is a tree; see [16] for more details. In prior work, we have used junction trees to improve the performance of passive graphical model selection algorithms [17]. As it turns out, since we are only interested in the clusters of the junction tree, it is sufficient to identify the cliques in a chordal graph of H^+; see [18] for a definition of chordal graphs.

A graph may have multiple chordal graphs. An optimal chordal graph, which is computationally difficult to find, is defined so that the size of the maximum clique is the smallest. Although finding optimal cliques is "ideal" for our algorithm, it is not necessary for our algorithm to function properly. In our implementation, we use linear time greedy heuristics [19], which are known to output close to optimal chordal graphs [20]. A summary of the above steps is shown in Algorithm 2.

```
Algorithm 2: Find Active Vertices
  • Inputs: $\mathfrak{X}$, $\widehat{E}$ and $\widehat{F}$.
  • Initialize: $A \leftarrow \emptyset$
  • Estimate $H^+$ and $H^-$ (see Remark 3.1).
  • $\mathcal{V} \leftarrow$ Cliques in chordal graph of $H^+$
  • For each clique $V_k \in \mathcal{V}$
      – If $H^+[V_k] \neq H^-[V_k]$, then $A \leftarrow A \cup V_k$
      – If $H^+[V_k] = H^-[V_k]$, then $\widehat{E} \leftarrow$ Edges
        of $H^+[V_k]$, $\widehat{F} \leftarrow$ Nonedges of $H^+[V_k]$
  • Return $A, \widehat{E}$ and $\widehat{F}$
```

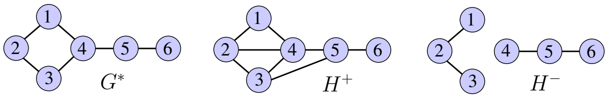


Figure 2: Illustration of Algorithm 2.

To illustrate Algorithm 2, consider the graphs G^*, H^+, and H^- in Figure 2. A simple calculation shows that the cliques of a chordal graph of H^+ are $\{1,2,4\}$, $\{2,3,4\}$, $\{3,4,5\}$, and $\{5,6\}$. Comparing the induced subgraphs of H^+ and H^- on the cliques, we identify that the edge $(5,6)$ is in the true graph. Furthermore, if $(3,5) \notin H^+$, then the edge $(4,5)$ can also be identified to be in the true graph.

Remark 3.1. An important step in Algorithm 2 is computing the graphs H^+ and H^-. Recall that we want $G^* \subseteq H^+$ and $H^- \subseteq G^*$. In our numerical simulations, we use stability selection [21], with appropriate thresholds, to select H^+ and H^-. We refer to Appendix A for more details.

Remark 3.2. Both Algorithms 1 and 2 are independent of the choice of the graphical model selection algorithm. Furthermore, the computational complexity of the active learning algorithm is dominated by the computation of H^+ and H^-. Thus, the overall complexity is roughly $O(K\mathcal{I})$, where $O(\mathcal{I})$ is the complexity of graphical model selection. As will be clear from the theoretical results and the numerical simulations, the additional computations required for active learning is a small price to pay for the potential benefits of using active learning for improved graph estimation.

4 Conditional Independence Testing

```
Algorithm 3. $\mathsf{CIT}(\mathfrak{X}^n_V, \kappa, \eta)$: Conditional independence testing for graphical model selection
  • Inputs: $\mathfrak{X}^n_V$: $n$ i.i.d. measurements; $\kappa$: An
    integer that controls the computational complexity; $\tau_n$: threshold that controls the sparsity of graph.
  • $\widehat{G} \leftarrow$ Complete graph over $p$ vertices.
  • for each $(i,j) \in E(\widehat{G})$
      – If $\exists$ $S$, $|S| \leq \kappa$, s.t. $|\widehat{\rho}_{ij|S}| \leq \tau_n$, then
        delete edge $(i,j)$ from $\widehat{G}$.
  • Return $\widehat{G}$.
```

In this section, we review a graphical model selection algorithm to study the advantages of our active learning algorithm. In particular, we review Algorithm 3, called CIT, which uses conditional independence tests to estimate a graph. This method is not new, and goes back to the SGS-Algorithm [22] for learning Bayesian networks. The conditional independence test used in CIT is to threshold the empirical conditional correlation coefficient (see (A3) for definition). Recently, [4, 7] studied the regimes under which a conditional independence test based graphical model selection algorithm has attractive sample complexity. Although the computational complexity of Algorithm 3 is $O(p^{\kappa+2})$, where κ is an input to the algorithm, the PC-Algorithm [23] can be used to significantly speed up the computations.

To characterize the performance of Algorithm 3, we consider the following assumptions.

(A1) P_X is a multivariate normal distribution with mean zero and covariance Σ such that $\max_{i,i} \Sigma_{i,i} \leq M < \infty$, where M is a constant.

(A2) $X_i \perp\!\!\!\perp X_j | X_S \iff i$ and j are separated by S.

(A3) $\sup |\rho_{ij|S}| < 1$, where $\rho_{ij|S} = \frac{\Sigma_{ij|S}}{\sqrt{\Sigma_{i,i|S}\Sigma_{j,j|S}}}$ and $\Sigma_{i,i|S} = \Sigma_{i,j} - \Sigma_{i,S}\Sigma_{S,S}^{-1}\Sigma_{S,j}$. Note that $\widehat{\rho}_{ij|S}$ is computed using the empirical covariance matrix.

(A4) If $(i,j) \notin E(G^*)$, there exists a *minimal separator* of size η that separates i and j.

The Gaussian assumption in (A1) is for simplicity. We can use the results in [7] to generalize the analysis to discrete distributions. Assumption (A2) is sometimes called the faithfulness condition. The parameter $\rho_{ij|S}$ in (A3) is the conditional correlation coefficient. Whenever $(i,j) \notin G^*$, then $\rho_{ij|S} = 0$. Moreover, using (A2), we have that $\rho_{ij|S} = 0$ if and only if $(i,j) \notin G^*$. This justifies the use of the empirical conditional correlation coefficient, $\widehat{\rho}_{ij|S}$, to test for conditional independence in Algorithm 1. The *minimal separator* in (A4) is defined as a separator S for $(i,j) \notin E(G^*)$ such that no proper subset of S separates i and j. The parameter η in (A4) implicitly places limits on the sparsity of the graph. For example, we can easily upper bound η by the maximum degree of the graph. However, for many graphs, this upper bound is very loose. For example, $\eta = 1$ for trees, but the maximum degree can be as large as $p-1$. Finally, we define the minimal conditional correlation coefficient as follows:

$$\rho_{min} := \min_{(i,j)\in G^*, |S| \leq \eta} |\rho_{ij|S}| \,. \tag{1}$$

Now, suppose we are given n i.i.d measurements $\mathfrak{X}_V^n = (X_V^{(1)}, \ldots, X_V^{(n)})$ drawn from P_X. We work within a high-dimensional framework so that the various problem parameters can scale arbitrarily as $p \to \infty$. We have the following theorem.

Theorem 4.1. *Suppose Assumptions (A1)-(A4) hold and let $\widehat{G} = \mathsf{CIT}(\mathfrak{X}_V^n, \eta, \tau_p)$, where $\tau_p = 0.9\rho_{\min}$. For constants $c_1, c_2 > 0$, if $\rho_{\min} > \frac{c_1(\eta+2)\log p}{(n-\eta)}$ and*

$$n \geq \eta + c_2 \rho_{\min}^{-2}(\eta+2)\log(p)\,,$$

then $\mathbb{P}(\widehat{G} = G^*) \to 1$ as $p \to \infty$.

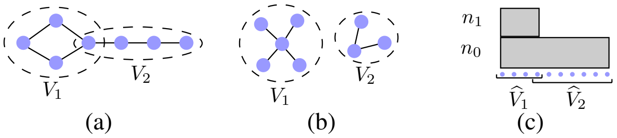


Figure 3: (a)-(b) Examples of the graphs in Section 5. (c) Measurement scheme in Algorithms 4 and 5.

The proof of Theorem 4.1, outlined in Appendix B, is based on methods in [24] and [4]. We note that Theorem 4.1 differs slightly from the results on conditional covariance based testing in [4]. In particular, the result in [4] is based on local separators, while the result in Theorem 4.1 is based on exact separators between non-edges of the graph. In general, the size of a local separator is less than the size of the exact separator. Although we use Theorem 4.1 to analyze our active learning algorithms, our analysis can be easily derived using the results from [4].

5 A Graph Family with Two Clusters

In this section, we define a family graphical model family to highlight the advantages of active learning. In the definitions that follow, operations over graphs correspond to operations over the vertices and edges.

$\mathcal{G}_{p,p_1,p_2,\eta,d}$:= Family of graphs over p vertices such that $G = G_1 \cup G_2$, where G_1 and G_2 are characterized as follows. Arbitrarily select two sets of vertices V_1 and V_2, such that $V_1 \cup V_2 = V$ and $T = V_1 \cap V_2$, where $|T| \leq 1$. Let $\mathcal{G}_{p,\eta,d}$ be the set of all graphs over p vertices with maximum degree d and minimal separator of size less than or equal to η. Assume $G_k = (V_k, E(G_k)) \in \mathcal{G}_{p_k,\eta,d}$, for $k = 1, 2$, where $p_k = |V_k|$. Note that, since $|T| \leq 1$, $G[V_1] = G_1$ and $G[V_2] = G_2$.

$\Theta(G)$:= Inverse covariance matrix of a zero mean Gaussian graphical model on a graph G.

$\mathcal{G}_{p,p_1,p_2,\eta,d}(\theta_1, \theta_2)$:= Set of all possible inverse covariance matrices $\Theta(G)$, where $G \in \mathcal{G}_{p,p_1,p_2,\eta,d}$, such that $\min_{(i,j)\in E(G[V_k])} \frac{|\Theta_{ij}(G)|}{\sqrt{\Theta_{ii}(G)\Theta_{jj}(G)}} \geq \theta_k$, for $k = 1, 2$, where θ_1 and θ_2 quantify the minimal conditional covariances over V_1 and V_2 given all other variables.

Throughout this paper, we assume that $G^* \in \mathcal{G}_{p,p_1,p_2,\eta,d}$ and that the Gaussian graphical model has zero mean with inverse covariance $\Theta(G^*) \in \mathcal{G}_{p,p_1,p_2,\eta,d}(\theta_1, \theta_2)$. From the definition, it is clear that G^* admits a two cluster decomposition, as in Figure 3(a)-(b), where there exists a set of vertices T that separates the vertices $V_1 \backslash T$ and $V_2 \backslash T$. In words, this means that all paths from a vertex in $V_1 \backslash T$ to a vertex in $V_2 \backslash T$ pass through T. When $T = \emptyset$, there are no edges between V_1 and V_2. Note that the assumption $|T| \leq 1$ is only enforced to simplify our analysis; see

Remark 6.2 for more details.

Next, we define three parameters on $\Theta(G^*)$ that will be important in the analysis of our algorithm:

$$\rho_0 := \min_{(i,j)\in E(G^*),|S|\le|T|} |\rho_{ij|S}| \,, \tag{2}$$

$$\rho_1 := \min_{(i,j)\in E(G^*[V_1]),|S|\le\eta,S\subset V_1} |\rho_{ij|S}| \,, \tag{3}$$

$$\rho_2 := \min_{(i,j)\in E(G^*[V_2]),|S|\le\eta,S\subset V_2} |\rho_{ij|S}| \,. \tag{4}$$

Informally, ρ_0 quantifies the difficulty in learning the two cluster decomposition, ρ_1 quantifies the difficulty in learning the edges over V_1, and ρ_2 quantifies the difficulty in learning the edges over V_2.

Finally, we use the results in [4] to relate the parameters ρ_1 and ρ_2 to θ_1 and θ_2, respectively. In what follows, the various parameters defined on the graphical model are assumed to scale with p and we use the following notations: $\|M\|$ is the spectral norm of a matrix and $f_p = \Omega(g_p)$ means that for sufficiently large p, there exists a constant c such that $f_p \ge c g_p$.

Theorem 5.1 ([4]). *Let $\Theta(G^*) \in \mathcal{G}_{p,p_1,p_2,\eta,d}(\theta_1,\theta_2)$. Suppose $\Theta_{ii}(G^*) = 1\ \forall\ i$ and $\Theta_{ij}(G^*) \le 0$ for $i \ne j$. If $\|I - |\Theta(G^*)|\| = \alpha < 1$, where α is a constant, then $\rho_1 = \Omega(\theta_1)$ and $\rho_2 = \Omega(\theta_2)$.*

Theorem 5.1 shows that ρ_1 and ρ_2 are asymptotically lower bounded by $c\theta_1$ and $c\theta_2$, respectively, where c is an appropriate constant. The condition on $\Theta(G^*)$, although restrictive, can be generalized so that $\Theta(G^*)$ is a walk-summable graphical model [25]. For simplicity, we avoid stating the conditions and refer to Lemma 14 in [4] for more technical details.

6 Theoretical Analysis of a Two-Stage Active Learning Algorithm

In this section, we derive necessary and sufficient conditions on the number of *scalar measurements* required for reliable estimation of the unknown graph using our active learning algorithm. Recall that if we draw n measurements from p vertices, then the number of scalar measurements is np.

Section 6.1 presents sufficient conditions for a modified version of Algorithm 1 that is designed for graphs in the two-cluster graph family defined in Section 5. Section 6.2 presents sufficient conditions when given prior knowledge about the absence of certain edges in the graph. Section 6.3 compares the sufficient conditions to necessary conditions required by any passive graphical model selection algorithm.

6.1 Sufficient Conditions

Recall that Algorithm 1 uses Algorithm 2 to update the set of active vertices. Unfortunately, an analysis of Algorithm 2 is not within the scope of this paper and is left for future work. Instead, we replace Algorithm 2 with another method, specific to the two-cluster decomposition. The details of the active learning algorithm we analyze is given in Algorithm 4.

Algorithm 4. Two-Stage Active Learning

```
1) Draw $n_0 = \eta + c_2 \log p \max\{3\rho_0^{-2}, \rho_2^{-2}(\eta+2)\}$
   measurements, $\mathfrak{X}_V^{n_0}$, from $V$.
2) $\widehat{G} \leftarrow \mathsf{CIT}(\mathfrak{X}_V^{n_0}, \eta, \tau_0)$, where $\tau_0 = 0.9\min\{\rho_0, \rho_2\}$.
3) Find $\widehat{V}_1$, $\widehat{V}_2$, and $\widehat{T}$ such that $\widehat{T}$ separates $\widehat{V}_1\backslash\widehat{T}$
   and $\widehat{V}_2\backslash\widehat{T}$ in $\widehat{G}$, $|\widehat{T}| = 1$, and $\widehat{G}[V_2] = G^*[V_2]$.
4) Let $\widehat{p}_1 = |\widehat{V}_1|$. Draw $n_1 = \eta + c_2\rho_1^{-2}(\eta +
   2)\log \widehat{p}_1 - n_0$ measurements from $\widehat{V}_1$.
5) $\widehat{G}_1 \leftarrow \mathsf{CIT}(\mathfrak{X}_{\widehat{V}_1}^{n_0+n_1}, \eta, \tau_1)$, where $\tau_1 = 0.9\rho_1$.
6) Return $\widehat{G} = \widehat{G}_1 \cup \widehat{G}[\widehat{V}_2]$.
```

Algorithm 4 corresponds to Algorithm 1 with two rounds of measurements ($K = 2$), $A = V$, and $\widehat{E} = \widehat{F} = \emptyset$. The crux of Algorithm 4 is illustrated in Figure 3(c), where we first draw measurements from all the vertices and then focus the next round of measurements over $\widehat{V}_1$. We *do not* draw measurements over $\widehat{V}_2$ since the edges and the non-edges over $\widehat{V}_2$ are estimated using the first round of measurements.

Before presenting our result regarding Algorithm 4, we state three additional assumptions that we impose on the graphical model.

(A5) $\rho_1^{-2}(\eta+2)\log p_1 > \max\{3\rho_0^{-2}, \rho_2^{-2}(\eta+2)\}\log p$

(A6) $0.9\rho_1 > c_1(\eta+2)\log p_1/(n_0+n_1-\eta)$

(A7) $0.9\min\{\rho_0,\rho_2\} > c_1(\eta+2)\log p/(n_0-\eta)$

Informally, (A5) ensures that the subgraph over V_1 has a more complex structure and requires more measurements to reliably estimate all the edges over V_1. Both (A6) and (A7) ensure that the parameters ρ_0, ρ_1, and ρ_2 are not too small so that the true edges can be distinguished from the non-edges.

Theorem 6.1. *Under Assumptions (A1)-(A7), Algorithm 4 outputs the true graph with probability converging to one as $p \to \infty$. Furthermore, for constants $c_1, c_2 > 0$, the number of scalar measurements drawn by Algorithm 4 is equal to*

$$(p-p_1)c_2\max\left\{3\rho_0^{-2}, \rho_2^{-2}(\eta+2)\right\}\log p$$
$$+p\eta + p_1 c_2 \rho_1^{-2}(\eta+2)\log p_1 \,.$$

The proof of Theorem 6.1, outlined in Appendix C, first uses Theorem 4.1 to show that n_0 measurements are sufficient to estimate the two cluster decomposition and the edges over V_2, and then again uses Theorem 4.1 to show that $n_0 + n_1$ measurements are sufficient to estimate the edges over V_1. Note that Algorithm 4 does not necessarily identify the clusters V_1 and V_2 in step 3. However, as shown in the proof of Theorem 6.1, given n_0 measurements, $\widehat{V}_1 \subseteq V_1$ and $V_2 \subseteq \widehat{V}_2$ with high probability. We now make some additional remarks.

Remark 6.1. We emphasize that Algorithm 4 does *not* assume that V_1 and V_2 are known. Instead, Algorithm 4 only assumes that the parameters ρ_0, ρ_1, and ρ_2 are known. Given these parameters, step 3 of Algorithm 4, where we check if $\widehat{G}[V_2] = G^*[V_2]$, can be implemented using the bounds for the CIT algorithm in Theorem 4.1. Furthermore, if G^* does not admit a two-cluster decomposition, then $\widehat{V}_1 = V$ and $\widehat{V}_2 = \emptyset$, in which case Algorithm 4 will mirror the passive CIT algorithm.

Remark 6.2. Recall from the definition of G^* in Section 5 that we imposed the simplistic assumption that $|T| \leq 1$. At the cost of some additional technicalities, Theorem 6.1 can be extended to the case when $|T| > 1$. The main change in the analysis will be to consider a slightly larger set V_1 to ensure that the edges over T can be accurately estimated.

Remark 6.3. The choice of n_0 and n_1, and the subsequent analysis, is assuming the two cluster decomposition. In practice, the graph G^* can admit multiple two cluster decompositions. Subsequently, Algorithm 4 can be tailored for such decompositions. Thus, we can derive multiple bounds for the scalar measurements required for Algorithm 4 and the optimal one will correspond to the minimum over all two cluster decompositions of the graph G^*.

Remark 6.4. It is easy to see that if (A5) holds, then the difference between the the scalar measurements required for Algorithm 4 and the scalar measurements required for the passive CIT algorithm is $O((p-p_1)\rho_1^{-2}\log p_1)$. This suggests that when $p_1 \ll p$, the advantages of using Algorithm 2 may be much more pronounced. Unfortunately, it is not clear if this analysis is tight since we are comparing the differences between two sufficient conditions. Regardless, our numerical simulations in Section 6 clearly show the benefits of active learning.

6.2 Using Prior Knowledge

In this section, we analyze a variant of Algorithm 4 when given a priori knowledge that there exists no edges between $V_1 \backslash T$ and $V_2 \backslash T$ in G^*. This information could be extracted from prior knowledge about the graphical model of interest. For example, when studying financial data from companies, there may be prior knowledge available about the sectors of different companies. When studying gene expression data, there may be prior knowledge available about the different pathways genes belong to. We show that using such prior knowledge to adapt measurements can lead to significant reductions in the sample complexity of learning the true graph.

In Algorithm 5, we modify Algorithm 4 to take into account the prior knowledge about the graph.

```
Algorithm 5. Given that $T$ separates $V_1\backslash T$ and $V_2\backslash T$, implement Algorithm 4 with Steps 1 and 2 replaced by
  1) Draw $n_0$ measurements, $\mathfrak{X}_V^{n_0}$, from $V$ such that $n_0 = \eta + c_2\rho_2^{-2}(\eta+2)\log(p_2)$.
  2) $\widehat{G} \leftarrow \mathsf{CIT}(\mathfrak{X}_V^{n_0}, \eta, \tau_0)$, where $\tau_0 = 0.9\rho_2$.
```

Algorithm 5 simply changes the initial measurements in Algorithm 4 to account for the fact that some non-edges in G^* are already known. In Algorithm 1, this corresponds to appropriately specifying the set $\widehat{F}$. Before stating the main result regarding Algorithm 5, which follows easily from Theorem 6.1, we consider the following assumptions that are analogous to (A5)-(A7).

(A5′) $\rho_1^{-2}\log p_1 > \rho_2^{-2}\log p$

(A6′) $0.9\rho_2 > c_1(\eta+2)\log p_2/(n_0-\eta)$

(A7′) $0.9\rho_1 > c_1(\eta+2)\log p_1/(n_0+n_1-\eta)$

Note that n_0 and n_1 in (A6′)-(A7′) are defined in Algorithm 5.

Theorem 6.2. *Under Assumptions (A1)-(A4), (A5′)-(A7′) and given that T separates $V_1\backslash T$ and $V_2\backslash T$, Algorithm 5 outputs the true graph with probability converging to one as $p \to \infty$. Furthermore, for constants $c_1, c_2 > 0$, the number of scalar measurements is equal to*

$$(p-p_1)c_2\rho_2^{-2}(\eta+2)\log p_2 + p\eta + p_1c_2\rho_1^{-2}(\eta+2)\log p_1.$$

The only difference between Theorem 6.1 and Theorem 6.2 is that the scalar measurements in the later theorem no longer depends on ρ_0, the parameter that quantifies the difficulty in learning the two cluster decomposition. An alternative active learning method is to draw measurements from V_1 and V_2 separately. As long as $|T|$ is small, this strategy will roughly need the same number of scalar measurements as Algorithm 3. However, if there are constraints on the number of joint measurements a system can make, then this later strategy could be more useful. For example, if the measurements are acquired from a sensor network, then there may be limits on the number of joint measurements sensors can transmit so as to conserve the battery life of sensors.

6.3 Comparison to Necessary Conditions

We now compare the sufficient conditions in Theorem 6.2 to the necessary conditions for *any* passive algorithm. Let $\mathfrak{X}_V^n$ be n i.i.d. samples drawn from $\mathcal{N}(0, \Theta^{-1}(G^*))$, where $\Theta(G^*) \in \mathcal{G}_{p,p_1,p_2,\eta,d}(\theta_1,\theta_2)$. Let ψ be a graph decoder that takes as input $\mathfrak{X}_V^n$ and outputs a graph in $\mathcal{G}_{p,p_1,p_2,\eta,d}(\theta_1,\theta_2)$. For any decoder ψ, define the maximal probability of error as

$$p_e(\psi) = \max_{\Theta(G) \in \mathcal{G}_{p_1,p_2,\eta,d}(\theta_1,\theta_2)} \mathbb{P}(\psi(\mathfrak{X}_V^n) \neq G)\,,$$

where the probability is with respect to the product distribution of $(\mathcal{N}(0,\Theta^{-1}(G)))^n$ over n i.i.d. observations. We say a graph decoder is high-dimensional consistent if $p_e(\psi) \to 0$ as $p \to \infty$.

Theorem 6.3. *Suppose $\theta_1, \theta_2 \in [0, 0.5]$ and the decoder ψ is given prior knowledge that there are no edges between $V_1 \backslash T$ and $V_2 \backslash T$. A necessary condition for high-dimensional consistent graphical model selection over a Gaussian graphical model with the inverse covariance matrix in the set $\mathcal{G}_{p,p_1,p_2,\eta,d}(\theta_1,\theta_2)$ is*

$$n > \frac{1}{2}\max\left\{\theta_1^{-2}\log\frac{p_1-d-1}{2e}, \theta_2^{-2}\log\frac{p_2-d-1}{2e}\right\}.$$

The proof of Theorem 6.3, given in the supplement, uses information-theoretic methods from [26].

We now compare the passive and active algorithms. Suppose θ_1 is small enough so that $\theta_1^{-2}\log(p_1-d-1) > \theta_2^{-2}\log(p_2-d-1)$, and $\theta_1^{-2}\log(p_1) > \theta_2^{-2}\log(p_2)$. Then, the necessary conditions on the number of scalar measurements for consistent selection by *any* passive algorithm scales as

$$q_{passive} = \Omega(p\theta_1^{-2}\log(p_1-d-1)). \qquad (5)$$

On the other hand, using Theorem 5.1 in Theorem 6.2, and assuming that η is a constant, the sufficient conditions for the active method in Algorithm 5 scales as

$$q_{active} = \Omega((p-p_1)\theta_2^{-2}\log p_2 + p_1\theta_1^{-2}\log p_1)\,. \qquad (6)$$

Now, consider the condition

$$\theta_1^2 < \theta_2^2\frac{p_1\log p_1}{(p-p_1)\log p_2}\,. \qquad (7)$$

A simple calculation shows that if (7) holds, then $q_{active} = \Omega(p_1\theta_1^{-2}\log p_1)$. Thus, if $d \ll p_1 \ll p$, and (7) holds, then Algorithm 5 requires far fewer number of scalar measurements than any other passive algorithm. To get an understanding of the condition in (7), suppose $p_1 = \sqrt{p}$ and $|T| = 0$. Then, $\theta_1^2 = O(\theta_2^2/\sqrt{p})$. In other words, the advantages of active learning are substantial when θ_1 is much smaller than θ_2.

Finally, we note that our analysis is only for Algorithm 5, where information about the graph decomposition is known to the algorithm. An open problem is to study how the performance of Algorithm 4 compares to the necessary conditions when no information about the graph decomposition is given.

7 Numerical Results

In this section, we present numerical results that highlight the advantages of our active learning algorithm. For all synthetic results, we assume that P_X is multivariate Gaussian with mean zero and covariance Σ. Define the inverse covariance matrix by $\Theta = \Sigma^{-1}$. If P_X is Markov on G^*, it is well known that $(i,j) \notin G^*$ implies that $\Theta_{ij} = 0$.

We consider three different kinds of synthetic graphical models and assume that $\Theta_{ii} = 1$ for $i = 1, 2, \ldots, p$. For all graphs considered below, we assume that the first p_1 vertices are *weak edges* so that the absolute value of the non-zero entries over these vertices is smaller than the other non-zero entries. We refer to Appendix E for results on scale-free graphs.

Chain Graph: $\Theta_{i,i+1} = \rho_1$ for $i = 1, \ldots, p_1$ (weak edges) and $\Theta_{i,i+1} = \rho_2$ for $i = p_1+1, \ldots, p$ (strong edges). Let $\rho_1 = 0.1$, $\rho_2 = 0.3$, and $\Theta_{ij} = \Theta_{ji}$.

Hub Graph: The first p_1 vertices are partitioned into vertices of size 10 and the remaining vertices are partitioned into vertices of size 5. For each partition, all vertices are connected to one vertex. Θ_{ij} is constructed so that $\Theta_{ij} = 1/d_{ij} - \epsilon$, where d_{ij} is either the degree of vertex i or the degree of vertex j, depending on which one is larger. The scalar $\epsilon = 10^{-4}$. The above construction ensures that the matrix Θ is positive and symmetric.

Cluster Graph: All vertices are partitioned into clusters of size 20. For the first $p_1/20$ clusters, the edges over each cluster are generated using an Erdos-Renyi (ER) random graph model so that the probability that each edge appears in a cluster is 0.2. For the remaining clusters, the graph over each cluster is an ER graph with edges appearing with probability 0.1. The inverse covariance matrix is constructed as in the Hub graph case. This construction ensures that the edges corresponding to the first $p_1 = 0.2p$ vertices have lower partial correlation values than all other edges.

7.1 Methodology

We use the CIT with $\kappa = 1$ to perform the active learning component (computing H^+ and H^-) of our algorithm and CIT with $\kappa = 2$ to estimate the final graph. In all experiments, $K = 5$ and $\delta = 0.5$.

We specify $q = n \times p$, the desired number of scalar measurements, to our algorithm and obtain a matrix $\bar{X}$ of size $\bar{n} \times p$, where $\bar{n} \geq n$. Each column in this matrix corresponds to the samples obtained from a random variable. Since we perform active learning, some entries in $\bar{X}$ will be missing. Moreover, in general, we may not be able to get exactly q scalar measurements, so we stop obtaining measurements until

Table 1: Cluster graph with $p = 400$ vertices

		Oracle Results			Model Selection Results		
n	Alg	TPR	FDR	ED	TPR	FDR	ED
200	Nonactive	0.409 (0.003)	0.149 (0.003)	283 (0.969)	0.29 (0.000)	0.029 (0.000)	307 (0.145)
	Active	0.405 (0.002)	0.120 (0.003)	278 (0.697)	0.296 (0.000)	0.022 (0.000)	303 (0.157)
400	Nonactive	0.675 (0.002)	0.0726 (0.003)	162 (0.838)	0.568 (0.001)	0.0148 (0.000)	188 (0.199)
	Active	0.695 (0.003)	0.0666 (0.002)	152 (0.765)	0.575 (0.001)	0.0111 (0.000)	184 (0.202)
600	Nonactive	0.787 (0.002)	0.0381 (0.001)	104 (0.689)	0.739 (0.000)	0.015 (0.000)	116 (0.144)
	Active	0.819 (0.002)	0.0488 (0.001)	95.5 (0.702)	0.747 (0.000)	0.001 (0.000)	111 (0.161)

the maximum possible number of measurements have been made. Note that, when learning the final graph, we only need to consider the random variables over which we have $\bar{n}$ observations. This is because the edges, and the non-edges, for all other random variables are estimated in the active learning component.

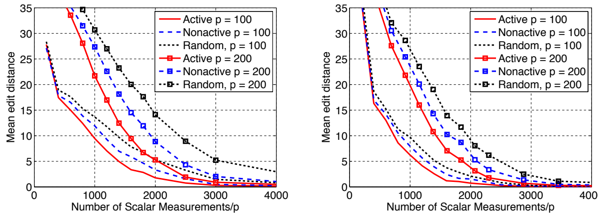
We compare the active learning graphs to two other estimated graphs. The first is the graph estimated using $n \times p$ *nonactive* or *passive* measurements. The second is the graph estimated using an $\bar{n} \times p$ measurement matrix $\widetilde{X}$ that contains randomly chosen missing entries that sum to the number of missing entries in $\bar{X}$. CIT can be easily applied to $\widetilde{X}$. We emphasize that all three graphs, active, nonactive, and random, are estimated using roughly *the same number of scalar measurements.*

We use the extended Bayesian information criterion (EBIC) for model selection [27]. EBIC requires an appropriate input parameter γ that controls the sparsity of the graph. We use $\gamma = 0.5$ as suggested by the authors in [27]. When using $\bar{X}$ and $\widetilde{X}$, we make appropriate modifications to compute the likelihood function, see [28] for more details.

We use three measures to compare an estimated graph to the true graph. The first is the true positive rate (TPR), which is the number of true edges estimated divided by the total number of true edges. The second is the false discovery rate (FDR), which is the number of falsely detected edges divided by the number of edges estimated. The third is the edit distance (ED), which is the number of true edges missed plus the number of falsely detected edges.

7.2 Results

Figure 4(a)-(b) show results for the chain and hub graphs with $p = 100$ and $p = 200$ vertices. The graphs estimated in each case are oracle estimates, i.e., the true graph was used to select an optimal threshold for CIT. This allows us to quantify the benefits of active learning and also validate our theoretical results. The plots show the variation of the edit distance for active, nonactive, and random graph estimates as the number of scalar measurements increase. For small q, no active learning is done since there are not enough measurements to separate the weak parts of the graph from the strong parts. As q increases, we clearly see the benefits of using active learning.



(a) Chain Graph (b) Hub Graph

Figure 4: Mean edit distance vs. number of scalar measurements over 50 trials for chain and hub graphs.

Table 1 shows results for the cluster graph when $p = 400$ and $n = 200, 400, 600$. Each entry in the table is the mean value of the metric over 50 trials with the standard error given in brackets. We present both oracle and model selection results. In both cases, the benefits of active learning is clear.

8 Conclusions

We have proposed an active learning algorithm for graphical model selection by adapting measurements drawn from a graphical model to certain subsets of vertices in a graph. We have identified a broad class of graphical models for which active learning can lead to significant savings in the total number of measurements needed for consistent graph recovery.

We highlight two interesting directions of future research. First, our algorithm depends on successfully selecting a superset of the true graph. Although we used a heuristic in our implementation, it will be extremely useful to have a consistent estimator that can reliably prune out several edges. Second, it will be interesting to study active learning algorithms for parameter estimation in graphical models.

Acknowledgment

Thanks to Vincent Tan and Gautam Dasarathy for valuable feedback and discussions. The work of D. Vats was supported in part by an Institute for Mathematics and Applications Postdoctoral Fellowship.

References

[1] M. J. Wainwright and M. I. Jordan, *Graphical Models, Exponential Families, and Variational Inference*, Now Publishers Inc., Hanover, MA, USA, 2008.

[2] N. Meinshausen and P. Bühlmann, "High-dimensional graphs and variable selection with the lasso," *Annals of Statistics*, vol. 34, no. 3, pp. 1436–1462, 2006.

[3] O. Banerjee, L. El Ghaoui, and A. d'Aspremont, "Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data," *Journal of Machine Learning Research*, vol. 9, pp. 485–516, 2008.

[4] A. Anandkumar, V. Y. F. Tan, F. Huang, and A. S. Willsky, "High-dimensional Gaussian graphical model selection: Walk summability and local separation criterion," *Journal of Machine Learning Research*, vol. 13, pp. 2293–2337, 2012.

[5] P. Ravikumar, M. J. Wainwright, and J. Lafferty, "High-dimensional Ising model selection using ℓ_1-regularized logistic regression," *Annals of Statistics*, vol. 38, no. 3, pp. 1287–1319, 2010.

[6] P. Netrapalli, S. Banerjee, S. Sanghavi, and S. Shakkottai, "Greedy learning of Markov network structure," in *Allerton Conference on Communication, Control and Computing*, 2010, pp. 1295–1302.

[7] A. Anandkumar, V. Y. F. Tan, F. Huang, and A. S. Willsky, "High-dimensional structure learning of Ising models: Local separation criterion," *Annals of Statistics*, vol. 40, no. 3, pp. 1346–1375, 2012.

[8] J. Lafferty, H. Liu, and L. Wasserman, "Sparse nonparametric graphical models," *Statistical Science*, vol. 27, no. 4, pp. 519–537, 2012.

[9] B. Settles, "Active learning literature survey," Computer Sciences Technical Report 1648, University of Wisconsin–Madison, 2009.

[10] J. Haupt, R. Castro, and R. Nowak, "Distilled sensing: Adaptive sampling for sparse detection and estimation," *IEEE Transactions on Information Theory*, vol. 57, no. 9, pp. 6222, 2011.

[11] B. Eriksson, G. Dasarathy, A. Singh, and R. Nowak, "Active clustering: Robust and efficient hierarchical clustering using adaptively selected similarities," in *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[12] M. Malloy and R. Nowak, "On the limits of sequential testing in high dimensions," in *Forty Fifth Asilomar Conference on Signals, Systems and Computers*, 2011, pp. 1245–1249.

[13] A. Krishnamurthy and A. Singh, "Low-rank matrix and tensor completion via adaptive sampling," *Advances in Neural Information Processing Systems (NIPS)*, 2013.

[14] S. Tong and D. Koller, "Active learning for structure in bayesian networks," in *International Joint Conference on Artificial Intelligence*, 2001, vol. 17, pp. 863–869.

[15] F. Pukelsheim, *Optimal design of experiments*, vol. 50, SIAM, 1993.

[16] S. L. Lauritzen, *Graphical Models*, Oxford University Press, USA, 1996.

[17] D. Vats and R. D. Nowak, "A junction tree framework for undirected graphical model selection," *Journal of Machine Learning Research*, vol. 15, pp. 141–185, 2014.

[18] D. B. West, *Introduction to Graph Theory*, Prentice Hall, 2nd edition, 2000.

[19] A. Berry, P. Heggernes, and G. Simonet, "The minimum degree heuristic and the minimal triangulation process," in *Graph-Theoretic Concepts in Computer Science*, H. Bodlaender, Ed., vol. 2880 of *Lecture Notes in Computer Science*, pp. 58–70. Springer Berlin / Heidelberg, 2003.

[20] F. Jensen and F. Jensen, "Optimal junction trees," in *Proceedings of the 10th Annual Conference on Uncertainty in Artificial Intelligence (UAI-94)*, San Francisco, CA, 1994, pp. 360–36, Morgan Kaufmann.

[21] N. Meinshausen and P. Bühlmann, "Stability selection," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 72, no. 4, pp. 417–473, 2010.

[22] P. Spirtes, C. Glymour, and R. Scheines, "Causality from probability," in *Advanced Computing for the Social Sciences*, 1990.

[23] P. Spirtes and C. Glymour, "An algorithm for fast recovery of sparse causal graphs," *Social Science Computer Review*, vol. 9, pp. 62–72, 1991.

[24] M. Kalisch and P. Bühlmann, "Estimating high-dimensional directed acyclic graphs with the PC algorithm," *Journal of Machine Learning Research*, vol. 8, pp. 613–636, 2007.

[25] D. M. Malioutov, J. K. Johnson, and A. S. Willsky, "Walk-sums and belief propagation in Gaussian graphical models," *Journal of Machine Learning Research*, vol. 7, pp. 2031–2064, 2006.

[26] W. Wang, M. J. Wainwright, and K. Ramchandran, "Information-theoretic bounds on model selection for Gaussian Markov random fields," in *IEEE International Symposium on Information Theory (ISIT)*, 2010.

[27] R. Foygel and M. Drton, "Extended bayesian information criteria for gaussian graphical models," in *Advances in Neural Information Processing Systems 23*, pp. 604–612. 2010.

[28] N. Städler and P. Bühlmann, "Missing values: sparse inverse covariance estimation and an extension to sparse regression," *Statistics and Computing*, vol. 22, no. 1, pp. 219–235, 2012.

[29] M. J. Wainwright, "Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (lasso)," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2183–2202, 2009.