Why Steiner-tree type algorithms work for community detection

Mung Chiang Princeton University
Henry Lam Boston University
Zhenming Liu Princeton University
Vincent Poor Princeton University

Abstract

We consider the problem of reconstructing a specific connected community $S \subset V$ in a graph $G = (V, E)$, where each node v is associated with a signal whose strength grows with the likelihood that v belongs to S. This problem appears in social or protein interaction network, the latter also referred to as the *signaling pathway reconstruction* problem (Bailly-Bechet et al., 2011).

We study this community reconstruction problem under several natural generative models, and make the following two contributions. First, in the context of social networks, where the signals are modeled as bounded-supported random variables, we design an efficient algorithm for recovering most members in S with well-controlled false positive overhead, by utilizing the network structure for a large family of "homogeneous" generative models. This positive result is complemented by an information theoretic lower bound for the case where the network structure is unknown or the network is heterogeneous. Second, we consider the case in which the graph represents the protein interaction network, in which it is customary to consider signals that have unbounded support, we generalize our first contribution to give the first theoretical justification of why existing Steiner-tree type heuristics work well in practice.

1 Introduction

In a community detection problem, we are given a graph and the goal is to identify the nodes in the graph that have strong ties to each others, or belong to so-called a "community". In the context of social network analysis, the graph refers to the social network; a community refers to a group of people who interact closely with each others, such as researchers on the same topic or college students living in the same dorm (Leskovec et al., 2009; Chen et al., 2010; Sozio and Gionis, 2010; Abraham et al., 2012; Arora et al., 2012; Balcan et al., 2012). In systems biology, the network can represent a protein-to-protein interaction process, with each node representing a protein and each edge representing the interaction between two proteins. Here, a community refers to the molecules that belong to the same functional unit of some kind (Newman, 2006; Dittrich et al., 2008; Deo et al., 2010; Fortunato, 2010; Bailly-Bechet et al., 2011).

This line of problems have been extensively studied. In this paper, we shall revisit it with a primary focus on a signal detection component that deviates from the standard literature. The following explains this motivation.

Finding a highly asymmetric group in a social network. We are interested in finding an important group of individuals in a social network. Such a subgroup, for example, could be a terrorist network. In this case, one can use communication data from mobile phone carriers to construct the social network (Shapiro and Weidmann, 2012). Also, security agencies are often able to provide an incomplete list of terrorists. Our goal is to find the rest of the terrorists in the network. Another example is the placement of personalized ads in social network services such as Facebook or Linkedin. For instance, when Facebook wants to help a local language school to find potential customers for its French class, it essentially needs to find a community in the town that is interested in foreign languages or cultures. Beyond the social network structure of the users in the town, Facebook also possesses user profiles, which may be used to infer a subset of members in the community. It remains for Facebook to uncover the rest of the members.

Finding protein association from cell signaling. Here, we are interested in using the trajectories of external information propagation to identify func-

Appearing in Proceedings of the 16^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2013, Scottsdale, AZ, USA. Volume 31 of JMLR: W&CP 31. Copyright 2013 by the authors.

tional modules in a protein-to-protein network (Bailly-Bechet et al., 2011; Dittrich et al., 2008). Specifically, in a graph that represents the protein-to-protein network, we may initiate a signal propagation process as follows: at the beginning a piece of signal starts to propagate from an unknown node. The signal can propagate from one node to another only if they are connected in the graph. Also, one node may propagate the signal to more than one of its neighbors. At the end of the process, the subset of nodes visited by the signal, namely *the signaling pathway*, often belongs to the same functional unit. Finding the signaling pathway is important for medical studies because such pathway could be connected with diseases such as cancer or Alzheimer's (See Bailly-Bechet et al. (2011) and the references therein). The measurement tools for these signals could bear reading errors. Thus we can only observe a likelihood on each node being part of the pathway. This problem is sometimes named as the *signaling pathway reconstruction* problem.

Both of these problems essentially need to find a specific subset of nodes in a network. Despite the dissimilar motivations, they call for a natural and unified model that simultaneously leverages the knowledge of the network structure and the extra information available at the node level. Below are some highlights of the key features in our setting and the differences with the standard community detection problem.

1. *Only one asymmetric group.* In standard community detection problems, the goal is to recover *all* the (possibly overlapping) communities in the network. Here we are interested in recovering *only one* community. Furthermore, it is often expected that the size of this community is small compared to the size of the entire network.
2. *Rich node level information is available.* In the standard setting, the algorithm needs to infer the community structure by using only the (possibly weighted) network structure. In our scenario, we often possess a rich amount of information at the node level.
3. *Tradeoff between false positive and false negative.* In the standard setting, the notion of false positive and false negative is often missing. In our problems, the costs of false positives and false negatives are often asymmetric, with false negatives being substantially higher.

1.1 Existing community detection algorithms

Despite the differences with the standard literature, one might still be hopeful to tweak existing solutions to solve our problems. Here we briefly review the existing community detection algorithms. We shall then discuss the fundamental barriers from using these solutions that are unlikely to bypass, and hence it becomes necessary to design and analyze a new model. A comprehensive survey on community detection algorithms can be found in Fortunato (2010).

Roughly speaking, the existing community detection algorithms can be divided into two categories. We refer to the first category as the *data mining* approach. In this approach, one starts with identifying intuitive combinatorial structures of the communities in a network. One then proceeds to execute combinatorial algorithms to find the clusters with the combinatorial properties they want. These properties often imply that the nodes in the same community have stronger ties than those belonging to different communities. Algorithms that minimize the *network modularity* (Newman, 2006) or the *conductance* (Leskovec et al., 2009) are examples in this category.

The second category is referred to as the *inference based* approach. Here one describes the formation of the social network as a stochastic process governed by a set of hidden parameters that characterizes the community structure. The social network is treated as an observed dataset from the process and inference algorithms are designed to learn the hidden parameters. One well-known example is the planted random graph model (Hastings, 2006). In this model, the node set V is partitioned into two subsets V_1 and V_2 so that $\Pr[\{u, v\} \in E]$ is p if u and v belong to the same partition and is q otherwise, where $p > q$ are two parameters controlling the density of the graph. The corresponding inference algorithm will assume the network is generated from this process to recover the set V_1 and V_2. Other examples in this category include Ball et al. (2011), Balcan et al. (2012), and Arora et al. (2012).

The following three obstacles make these existing solutions unsuitable to our setting.

Deficiency in stand-alone use of a *data mining* solution or an *inference-based* solution. The data-mining approach is not a principled approach in that it does not allow us to *reason* probabilistically why a community is formed in a specific way. This is undesirable in many social network analysis applications. For example, when Facebook wants to use community information to place personalized ads, it needs to explain to the clients (who buy the ads) why they think they correctly find the communities. On the other hand, inference-based algorithms usually are not robust and are designed only for very specific and simple generative models. It is quite unrealistic to believe a social or protein network is generated from a set of simple rules, and it is unclear what an inference-based algorithm can give us when the formation of the network deviates from the assumed generative model.

Profound computational barriers. We also observe that many algorithms (in both categories) are grounded in the assumption that the interactions among nodes in the same community are stronger than interactions among nodes from different communities. Such assumption often inevitably leads to solving some variations of the *densest subgraph*, the *conductance*, or the *modularity* problem. All these problems have been known to be difficult to tackle both in theory and in practice.

No usage of the node level information. All the existing community detection algorithms take the network structure but nothing else as input. We do not have a unified model that allows us to leverage both the network structure and the node level information simultaneously.

1.2 Our contribution

We position our contributions in both modeling and algorithmic design: We propose a natural theoretical model that allows us to use both the network structure and the node level information to model the formation of the communities. Then we design efficient community detection algorithms for our model. In particular, we make the following two distinct contributions for social networks and protein interaction networks.

Asymmetric group detection. Our model is the first unified and tractable model to solve the problem of this kind. Moreover, our model bypasses the aforementioned computational barriers by *not* using the assumption that the interaction in the same community is dense, thus avoiding computationally intractable problems. Furthermore, our algorithm is not designed for a specific generative model. Instead, it works for networks that come from a *wide range of generative models*.

Finding undetected protein association. For the "signaling pathway reconstruction" problem, a few heuristics that are quite effective in practice have *already* been proposed, *e.g.,* Dittrich et al. (2008) and Bailly-Bechet et al. (2011). Our model gives the first mathematically grounded explanation on why these heuristics work. Specifically, if the protein interaction network comes from a random graph family, namely the exponential tail graph (defined in the forthcoming sections), then some of the existing heuristics are *guaranteed* to work well. The exponential tail graph family is a large family of graphs that include the Erdös-Rényi model, Kleinberg's small world model (Kleinberg, 2000), and other latent space models such as the inner product model (Kim and Leskovec, 2012).

1.3 Organization

In Section 2, we describe our model and summarize our theoretical results. Section 3 presents a lower bound for the case where the network structure is unknown. In Section 4 we present our main results. Finally in Section 5, we use our results to explain why existing signaling pathways algorithms work.

2 The model

We now describe our model. A social or protein interaction network is represented by an undirected graph $G = \{V, E\}$, where $V = \{v_1, ..., v_n\}$.

The goal of the problem. In this network, there is a subset $S \subset V$ of special nodes that we need to find out. Let $k = |S|$, where $k = o(n)$.

We make the following assumptions regarding the combinatorial structure of S and the structure of the information associated with each individual node.

The community structure. We make only the weakest combinatorial assumption here that the subgraph induced by S is connected. In the context of detecting a social community, violating this assumption would imply that the community could be unrealistically decomposed into two subgroups so that members in different subgroups *do not know* each other. This connectedness assumption is also very natural in the context of finding pathways in protein interaction networks (Bailly-Bechet et al., 2011).

The signal structure. We shall assume each node is associated with a signal that represents how likely it is that the node belongs to S. The stronger the signal is, the more likely the node belongs to S. Specifically, we shall assume that each signal is a real number. The real numbers are independently sampled from one of two possible distributions. When $v_i \in S$, its associated signal is generated from $\mathcal{D}_1$. When $v_i \notin S$, its signal is generated from $\mathcal{D}_0$. Furthermore, we shall assume both distributions are from the same distributional family but the mean of $\mathcal{D}_1$ is higher. For exposition purpose, we shall assume that both $\mathcal{D}_0$ and $\mathcal{D}_1$ are uniform distributions with the same support size but $\mathcal{D}_1$ has higher mean, *i.e.,* $\mathcal{D}_0$ is a uniform distribution from $[0, 1]$ and $\mathcal{D}_1$ is a uniform distribution from $[1 - \gamma, 2 - \gamma]$ for some constant $0 < \gamma < 1$. Section 5 will explain how our results can be generalized to Gaussian distributions.

The community detection algorithm. Given the network $G = \{V, E\}$ and the signals associated with the nodes, our goal is to output a set $\hat{S}$ so that $\hat{S}$ is as close to S as possible. Specifically, we call an algorithm

an (α, β)-*detector* if and only if the algorithm can output a set $\hat{S}$ such that $|\hat{S}| \leq \alpha k$ and $|S - \hat{S}| \leq \beta k$ (with high probability). Notice that the parameters α and β indirectly control the tradeoffs between false positive rate and false negative rate: when α and β are fixed, the total number of false negatives is at most βk and the sum of false positives and false negatives is at most $(\alpha + 2\beta - 1)k$. In our applications, we want $\alpha = 1 + \delta$ for a small constant δ and β be as small as possible because it is more costly to miss a member in S than to make a mistaken claim on a non-member.

Before we continue, we remark on some aspects of our model.

Applying the model. In a social network, the signals can be interpreted as a lousy classifier that makes mistakes with constant probability. Often times, implementing a high quality classifier may not be completely infeasible (*e.g.,* one can hire human beings to monitor the communication among individuals in order to accurately label the set of terrorists). But executing high quality classifiers is usually very costly and thus cannot scale to giant networks. Thus, it is important to use a time-efficient classifier even at the cost of having reduced accuracy. Another way of interpreting our problem is to find an algorithm to boost the performance of a low-quality classifier by leveraging the network structure information. As mentioned before, we shall also use a generalization of this model to explain why some existing algorithms for finding pathways in protein interaction networks work.

Relationship to the sparse signal recovery problem. If the network structure is not given, our problem degenerates to the sparse signal recovery problem (See Haupt et al. (2011) and the references therein). In the latter context, one is given a set of real numbers $x_1, x_2, ..., x_n$ such that most of the numbers are sampled from a zero-mean distribution and only a small portion, say S, are sampled from an alternate positive-mean distribution. The goal is to identify the set of positive-mean variables. One major result in this paper is to show that knowing the structure of the network can substantially improve the algorithmic performance to recover S.

Combinatorial constraints in statistical models. Because our model takes into account both the network structure and the signal structure, the combinatorial constraints naturally melt with the statistical inference problem. We notice that recent works of Arias-Castro et al. (2008), Addario-berry et al. (2010), Abraham et al. (2012), and Soufiani and Airoldi (2012) also studied relevant latent space inference problems in networks.

Highlight of results and techniques. We now informally describe our results. We focus on understanding "the value of the network structure", *i.e.,* how much the connectivity constraints can help in our community detection algorithm.

Roughly speaking, our main result states that in a homogeneous and sparse network, the knowledge of the network structure and the connectivity constraint is very helpful in detecting S. In Section 3, we first show a lower bound for the case where the network structure is not given, *i.e.,* for any constants α and γ, there does not exist an $(\alpha, 0.999\gamma)$-detector (notice that getting an $(\alpha, (1+o(1))\gamma)$-detector is trivial). Then in Section 4, we show that when the network is generated from an "exponential tail random graph family"— a family of homogenous and sparse random graphs — then there exists an $(1.55, \lambda\gamma)$-detector for any arbitrarily small λ.

The power law graph family is a natural set of graphs that is *not* homogenous. For this case, we have a negative result: knowing the structure of the graph is information-theoretically *valueless*. On the other hand, if none of the highly connected nodes are in S, then finding S in a power law network becomes easy again. To summarize, we may interpret the value of the network structure as follows: when the nodes are homogenous and have sparse connections, the network structure has the highest value. When the degrees start to become skewed and some nodes are better connected than others, the value of knowing the network structure starts to decrease. Finally, when the network exactly follows the power law distribution, knowing the network structure will not be helpful at all.

Regarding methodology, central to our analysis is an understanding of how likely a random subset of nodes can be connected in a random graph. Intuitively, the less likely a random subset is connected, the more "powerful" the connectivity constraint is. In our analysis, we derive a set of coupling techniques to reduce the connectivity problem for different generative models into simpler objects, such as the sum of independent variables and branching processes. These techniques for understanding subgraph connectivities could be of independent interest.

3 Lower bound

This section presents a lower bound result when the network structure is unknown (the proof is in Appendix B). This result can also be viewed as a special case of the sparse signal recovery problem.

Theorem 3.1. *Let γ and α be arbitrary constants. Consider the community detection problem where the graph structure is not given. When $k = o(n)$, for any*

algorithm that returns a set $\hat{S}$ of size $\leq \alpha k$, we have $\mathrm{E}\left[|S - \hat{S}|\right] \geq (1 - o(1))\gamma k$.

We shall imagine γ as a sufficiently small constant and α a large constant. Theorem 3.1 implies that there exists no $(O(1), (1 - o(1))\gamma)$-detector for any constant γ when the network structure is unknown.

4 Algorithms for generative models

We next move to analyze the scenarios where the network structure is known. We focus on two generative models: Erdös-Rényi graphs and Kleinberg's small world (Kleinberg, 2000). The result for the small world model can be further generalized for the so-called "family of exponential tail graphs" (defined in Section 4.2). The technique developed for the small world model is strictly stronger but is more complicated. The connectivity analysis for subgraphs in $G_{n,p}$ appears to be a folklore. For completeness, we also present the analysis.

The reader is also referred to Appendix C for the analysis of a toy example, namely the line graph case, to get a quick intuition on how knowledge on the network structure may help. We also remark that our analysis assumes we know the value of k. This assumption can easily be relaxed because k can be estimated accurately from the signals.

4.1 The Erdös-Rényi random graph model.

We now analyze the Erdös-Rényi model. The following is our main theorem in this subsection.

Theorem 4.1. *Let $p = \frac{c}{n}$ for some constant c and λ be an arbitrary small constant. Consider the community detection problem where the network is sampled from $G_{n,p}$ and $k = o(n)$ is a polynomial in n. There exists a constant γ_0 such that for all $\gamma < \gamma_0$:*

- *There is no $(1.55, \gamma(1 - o(1)))$-detector that does not use the network structure information.*
- *There is an efficient $(1.55, \lambda\gamma)$-detector that uses the network structure.*

Before proceeding to our analysis, let us make a few remarks.

Setting $\alpha = 1.55$. First, our detector is only able to return a set of size $1.55k$ instead of k. This is because an intermediate step in our algorithm is to solve a Steiner tree problem and 1.55 is the best approximation ratio among Steiner tree algorithms (Robins and Zelikovsky, 2005).

The interpretation of λ. The parameter λ can be viewed as the "value" of the network structure: when the network structure is unknown, the portion of misses from S is approximately γ; but when we know the network structure, the portion of misses can reduce to $\lambda\gamma$.

We now proceed to our analysis. First, we need to show a combinatorial property about $G_{n,p}$.

Lemma 4.2. *Let G be a sample from $G_{n,p}$, where $p = \frac{c}{n}$ for some c. Let $C_\ell(G)$ be the number of connected subgraphs of size ℓ. There exists a constant τ_0 such that for any ϵ, we have* $\Pr\left[C_\ell(G) \geq \frac{1}{\epsilon p}(\tau_0)^\ell\right] \leq \epsilon$.

Proof. We shall first compute the expected number of connected subgraphs of size ℓ. Let J be a subset of size ℓ. Let $G(J)$ be the subgraph induced by J and let $\chi(J)$ be an indicator random variable that sets to 1 if and only if the subgraph induced by J is connected. We have $\mathrm{E}_{G \leftarrow G_{n,p}}[C_\ell(G)] = \sum_{J:|J|=\ell} \mathrm{E}[\chi(J)]$.

Thus, we only need to find $\mathrm{E}[\chi(J)]$, *i.e.,* the probability that J is a connected subgraph. Wlog, let $J = \{v_1, ..., v_\ell\}$. A necessary condition for J to be connected is that the number of edges in J is at least $|J| - 1$. Thus, we focus on finding $\Pr[E(G(J)) \geq \ell - 1]$.

Let us define an indicator random variable $X_{i,j}$ $(i < j)$ that sets to 1 if and only if $\{v_i, v_j\} \in E(G)$. We can see that $\{X_{i,j}\}_{i<j\leq\ell}$ are independent Bernoulli random variables with parameter p. We have

$$\Pr\left[\sum_{i,j\in J} X_{i,j} \geq \ell - 1\right] = \sum_{t=\ell-1}^{\frac{\ell(\ell-1)}{2}} \binom{\frac{\ell(\ell-1)}{2}}{t} p^t (1-p)^{\frac{\ell(\ell-1)}{2} - t}. \tag{1}$$

Let us consider the terms $\Pr[\sum_{i<j\in J} X_{i,j} = t] = \binom{\frac{\ell(\ell-1)}{2}}{t} p^t (1-p)^{\frac{\ell(\ell-1)}{2} - t}$ for all t. One can see that $\Pr[\sum_{i<j\in J} X_{i,j} = t]$ is maximized when t is near the expectation of $\sum_{i<j\in J} X_{i,j}$, *i.e.,* when t is either $[p\frac{\ell(\ell-1)}{2}]$ or $[p\frac{\ell(\ell-1)}{2}] + 1$. Using the assumption that $p = \Theta(\frac{1}{n})$, we have $\frac{\ell(\ell-1)p}{2} \ll \ell - 1$. Thus, the largest term in the summands at the right hand side of (1) is the term with $t = \ell - 1$, *i.e.,* $\binom{\ell(\ell-1)}{\ell} p^{\ell-1}(1-p)^{\ell(\ell-1)-\ell+1}$. Therefore, $\Pr\left[\sum_{i<j\in J} X_{i,j} \geq \ell - 1\right] \leq \ell^2 p^{\ell-1} \binom{\frac{\ell(\ell-1)}{2}}{\ell-1}$. We thus have $\mathrm{E}\left[\sum_{J:|J|=\ell} \chi(J)\right] \leq \tau_0^\ell / p$ for a suitable constant τ_0. Finally, by using a Markov inequality, we complete the proof of Lemma 4.2. □

We now prove Theorem 4.1. The analysis for the first part is similar to the one for Theorem 3.1 (whose proof is in Appendix B). Thus, we focus on proving the second part of the theorem. Specifically, we shall design an algorithm that works for a sufficiently small

γ. Our algorithm also needs to invoke the following building block:

Definition 4.3. *Let $G = \{V, E\}$ be an arbitrary graph and W be a subset of V. The* **MinConnect** *problem finds the smallest superset U of W so that the subgraph induced by W is connected.*

It is not difficult to see that the **MinConnect** problem is equivalent to the Steiner tree problem (See *e.g.,* Vazirani (2001) or Definition A.1 in Appendix) when the edges have uniform weights, *i.e.,*

Lemma 4.4. *The* **MinConnect** *problem is equivalent to the Steiner tree problem in which all the edges have the same weight.*

The proof of Lemma 4.4 is in Appendix D. Since there exists a 1.55-approximation algorithm for the Steiner tree problem, there also exists a 1.55-approximation algorithm for the **MinConnect** problem. We next describe how we use the **MinConnect** problem to solve the community detection problem.

The algorithm: We first partition the nodes into three sets: V_H contains the set of nodes whose associated signals are in $H \triangleq [1, 2-\gamma]$; V_M contains the set of nodes whose associated signals are in $M \triangleq [1-\gamma, 1]$, and V_L contains the set of nodes whose associated signals are in $L \triangleq [0, 1-\gamma]$. Notice that when $v \in V_H$, we are sure $v \in S$. When $v \in V_L$, we are sure $v \notin S$. Our algorithm consists of the following two steps:

- *Step 1. Truncate:* Let G' be the subgraph induced by V_H and V_M.
- *Step 2. Solve* **MinConnect**: Find the minimum connected subgraph in G' that contains all nodes in V_H. When the returned subset contains less than k nodes, we add arbitrary nodes in G' to the solution, as long as the solution remains connected, until the size reaches k.

We remark that this algorithm appears to be one of the most natural heuristics. We next analyze the algorithm's performance. Let us define a collection of subgraphs $\mathcal{C}(\epsilon)$ parametrized by ϵ as $\mathcal{C}(\epsilon) = \left\{G : C_\ell(G) \leq \frac{1.55k(\tau_0)^\ell}{\epsilon p} \text{ for any } 1 \leq \ell \leq 1.55k\right\}$, where τ_0 is the constant defined in Lemma 4.2. By using straightforward analysis, one can see that $\Pr\{G \in \mathcal{C}(\epsilon)\} > 1 - \epsilon$

Next we shall show that when G is in $\mathcal{C}(\epsilon)$, our algorithm succeeds with high probability. First observe that the subgraph induced by the set S contains k nodes, is connected (by definition), and contains all nodes in V_H (by definition). Thus the optimal solution for our **MinConnect** problem contains at most k nodes. Therefore, a Steiner-tree based approximation algorithm will give us a set of size $\leq 1.55k$.

We then argue that with low probability our algorithm returns a subset S' such that $|S - S'| > \lambda\gamma k$. We need the following definition.

Definition 4.5. *A subset of nodes T is a* good subset *if and only if 1. its size is between k and $1.55k$, 2. the subgraph induced by T is connected, and 3. $T \subseteq V_H \cup V_M$,* i.e., *all signals associated with nodes in T are either in H or in M.*

It suffices to show that with probability at least $(1 - \epsilon/(1.55k))$, any good subset S' of size ℓ ($k \leq \ell \leq 1.55k$) will be that $|S - S'| \leq \lambda\gamma k$. To prove this, consider, on the contrary, any S' such that $|S - S'| > \lambda\gamma k$. Since $|S' \cap S| \leq (1-\lambda\gamma)k$, we have $|S' - S| \geq \ell - (1-\lambda\gamma)k = (\ell - k) + \lambda\gamma k$. Let $\Delta k = \ell - k$. We then have $|S' - S| - \Delta k \geq \lambda\gamma k =: k_0$. Observe that a necessary condition for S' being a good subset is that all the signals associated with nodes in $S' - S$ are in M. This happens with probability $\leq \gamma^{k_0 + \Delta k}$. On the other hand, the total number of connected subgraph of size ℓ is bounded above by $\frac{1.55k(\tau_0)^\ell}{\epsilon p}$ with high probability. By using a union bound, the probability there exists at least one good S' with $|S - S'| > \gamma\lambda k$ is at most

$$\gamma^{k_0+\Delta k} \frac{1.55k(\tau_0)^\ell}{\epsilon p} = \frac{1.55k}{\epsilon p} \gamma^{k_0+\Delta k} \tau_0^{k+\Delta k} \leq c_0^{-k} \leq \frac{\epsilon}{1.55k} \tag{2}$$

for a suitable constant c_0. Appendix J.1 explains the deviation of (2) in detail.

To sum up we have shown that 1. $\Pr[G \in \mathcal{C}(\epsilon)] > 1 - \epsilon$; 2. When $G \in \mathcal{C}(\epsilon)$, the probability that our algorithm will output a good S' but $|S - S'| > \lambda\gamma k$ is $\leq \epsilon$. Therefore, with probability at most 2ϵ our algorithm will output a set S' such that $|S - S'| > \lambda\gamma k$, which proves Theorem 4.1.

4.2 The small world model and its generalization

We next move to the small world model. Appendix A.2 reviews the definition of the model. We have the following main proposition of this subsection.

Proposition 4.6. *Let G be a sample from the small world model with normalization constant $C = \Theta(\log n)$. Let $C_\ell(G)$ be the number of connected subgraphs of size ℓ, where $\ell \leq n^{1/3}$. There exists a constant τ_0 such that for any ϵ, we have $\Pr\left[C_\ell(G) \geq \frac{n}{\epsilon}(\tau_0)^\ell\right] \leq \epsilon$.*

The requirement that $\ell \leq n^{1/3}$ is chosen rather arbitrarily and is not optimized. Proposition 4.6 is the small world model's counterpart of Lemma 4.2. Thus, from Proposition 4.6 we use the same algorithm that appeared in Section 4.1 to achieve the same performance as described in Theorem 4.1, as long as $k = o(n^{1/3})$ and is a polynomial in n.

Our analysis, which presents a major technical contribution, couples the random subgraph induced by S with a branching process (See Appendix E for the proof).

We can continue to generalize Proposition 4.6 to cover a wider family of random graph models. Let us define the *exponential tail family* of random graphs as follows: the node set is $V = \{v_1, ..., v_n\}$. Each node v_i is associated with a hidden state s_i. A generative model in the exponential tail family defines a function h such that:

- The edge between v_i and v_j is included in the graph with probability $h(s_i, s_j)$, which is independent of the rest of the edges.
- Let D_i be the degree of the node v_i. Then: 1. $\mathrm{E}[D_i] = O(1)$ and 2. There exists a constant g_0 such that for any $g \geq g_0$ and D_i, $\Pr[D_i \geq g] < 2^{-g}$.

We have the following Corollary.

Corollary 4.7. *Let G be a random sample from an arbitrary exponential tail family of grpahs. There exists a constant γ_0 such that for all $\gamma < \gamma_0$: 1. There is no $(1.55, \gamma(1-o(1)))$-detector when the network structure is unknown, and 2. There is an efficient $(1.55, \lambda\gamma)$-detector when the network structure is given.*

We remark that a large number of generative models can be characterized by a set of latent states $\{s_1, ..., s_n\}$ and the probability function h, such as the inner product model, the exchangeable graph model, the planted random graph model, and the Chung and Lu's random graph model with expected degree (Chung and Lu, 2002; Hastings, 2006; Goldenberg et al., 2009; Kim and Leskovec, 2012). So long as the degree variables have small expectations and exponentially small tails, Corollary 4.7 is applicable.

4.3 The power law graph

It is also natural to ask whether there exist algorithms for the family of power law graphs, which clearly does not belong to the exponential tail family. In this section, we focus on understanding a specific power law graph model, namely Chung and Lu's model (Chung and Lu, 2002) when the expected degrees follow a power law distribution. We shall present a negative result and a positive result for this model. In our negative result, we show that no algorithm will perform better than the optimal algorithm for the case where the network *is not given*. In other words, the network structure *does not* add any value to solving the community detection problem. In our positive result, we show that, under the additional information that the community does not contain, say, the top 1% most densely connected nodes, there exists a sufficiently small constant γ so that our algorithm presented in Section 4.1 works well.

Recall that in Chung and Lu's random graph model, each node v_i is associated with a value w_i that represents its expected degree. The probability that $\{v_i, v_j\} \in E$ is $w_i w_j \rho$, where ρ is a normalization term that is linear in n. Here, we shall make standard assumptions that the sequence w_i follows a power law distribution and the average degree is a constant.

Let us start with our negative result.

Proposition 4.8. *Consider Chung and Lu's random graph model, in which the largest expected degree is $\Theta(\sqrt{n})$ and $k = o(\sqrt{n})$. Then with high probability there exists a connected group S such that any algorithm that outputs $\hat{S}$ with $|\hat{S}| = O(k)$ satisfies $\mathrm{E}|S - \hat{S}| \geq (1 - o(1))\gamma k$.*

This proposition shall be contrasted with Theorem 3.1: in the present setting, the structure of the graph is essentially useless. We remark on the choice of the parameters in Proposition 4.8. Here, we implicitly assume that the largest expected degree is larger than the size of the community. This assumption is supported by existing experiments (Mislove et al., 2007; Leskovec et al., 2009). The proof of Proposition 4.8 is deferred to Appendix F.

With Proposition 4.8, a natural question is whether we can do better if the highly connected nodes are known to be not in the set S. Our observation here is that for any constant ϵ, if we remove the ϵ-portion of highly connected nodes, the subgraph induced by the remaining nodes will have constant expected degree everywhere. In this case, the problem will be no harder than the problem for the $G_{n,p}$ case. Thus, we have the following corollary:

Corollary 4.9. *Consider Chung and Lu's model with the same set of parameters described in Proposition 4.8. Let ϵ and λ be arbitrary positive constants. There exists a γ such that if the top ϵ most connected nodes (in expectation) are not in the community, we can use the algorithm described in Section 4.1 to find a subset $\hat{S}$ of size $1.55k$ and $|S - \hat{S}| \leq (1 + o(1))\lambda\gamma k$.*

5 The Gaussian signal case: why existing pathways algorithms work.

We now generalize our result to explain why existing algorithms for finding signaling pathways in protein-to-protein networks work. In this problem, given a network G, we are required to recover the pathways of a signal cascading process, which means that we need to find a special subset of nodes S whose induced subgraph is connected. Furthermore, we also know

the p-value of each node between the hypotheses of being and not being in S (see our discussion on the Gaussian hypotheses that will come shortly). Existing solutions (Ideker et al., 2002; Dittrich et al., 2008; Deo et al., 2010; Bailly-Bechet et al., 2011; Jahid and Ruan, 2012) use the following algorithmic framework to recover the pathways: first, the algorithm assigns scores to each of the nodes according to their p-values. Nodes with low p-values will get high scores. Then the algorithm proceeds to find a subset of nodes whose score sum is maximized subject to the constraints that the nodes are connected, hoping to find a connected component with a large number of nodes having small p-value. In order to control the size of the output, the algorithm also introduces a regularization term to favor solutions with smaller number of nodes. Different algorithms have different ways of assigning the scores and the regularization terms. For example, in Bailly-Bechet et al. (2011), the score of a node v_i is defined as $-\log(p_i)$, where p_i is the p-value of v_i; next, with each edge assigned a weight, the regularization term of an output set $\hat{S}$ is the cost of the minimum spanning tree of $\hat{S}$. The final score of $\hat{S}$ is then the sum of the scores of all nodes in $\hat{S}$ minus the cost of the minimum spanning tree of $\hat{S}$.

Most of the other algorithms also select the scores and regularization terms in a way that the problem reduces to variations of the Steiner tree problem.

In this section we explain why the Steiner tree type algorithms work in practice. In particular, we shall focus on explaining the algorithm proposed in Bailly-Bechet et. al; we believe our arguments remain valid for many other similar algorithms.

In our analysis, we shall model the signals as being drawn from Gaussian distributions instead of from uniform distributions. When we model the signals as uniformly distributed, we implicitly assume that there is a constant portion of nodes have p-values either 0 or 1, which does not appear to be realistic (Dittrich et al., 2008; Bailly-Bechet et al., 2011). Instead, we assume that when $v \in S$, the signal associated with v is sampled from $N(\mu, 1)$ with μ being a positive constant; when $v \notin S$, the signal associated with v is sampled from $N(0, 1)$. We emphasize here that μ does not grow with the size of the network.

Let us recall the solution given in Bailly-Bechet et al. (2011): each edge e is assigned a positive cost $c(e)$ and each node v_i is associated with a positive "price" $b(v_i) = -\log(p_i)$ with p_i being v_i's p-value. The goal is to find a connected subgraph $G' = \{\hat{S}, \hat{E}\}$ that maximizes the following function:

$$\max_{\hat{E}\subset E, \hat{S}\subseteq V} \sum_{i\in\hat{S}} b(v_i) - \sum_{e\in\hat{E}} c(e). \tag{3}$$

Let us further assume that $k = O(n^{1/4})$ and the output $\hat{S}$ is required to be $O(n^{1/3})$ so that the false discovery rate does not approach 1 rapidly (no effort was made to improve the exponents). The following is our main proposition in this section.

Proposition 5.1. *Consider the signaling pathway reconstruction problem where the network is sampled from an exponential tail family of random graphs and $k = O(n^{1/4})$ is a polynomial in n. Let ϵ be an arbitrary small constant. There exists a sufficiently large constant μ_0 and a cost function $c(\cdot)$ such that for any $\mu \geq \mu_0$, the optimal solution S_{opt} for (3), subject to the constraint $|S_{\text{opt}}| \leq n^{1/3}$, satisfies $|S - S_{\text{opt}}| + |S_{\text{opt}} - S| \leq \epsilon k$ with high probability.*

The proof of Proposition 5.1 is in Appendix G. Two natural questions remain to be answered. First, is it plausible to assume the protein-interaction network is an exponential tail graph? Second, the optimization problem in (3) is an NP-hard problem and we cannot exactly solve the problem in polynomial time. What kind of performance guarantee can we get if we use a ρ-approximate algorithm?

Let us start with addressing the first issue. We observe that the proteins reside in a Euclidean space and it is reasonable to assume the likelihood that two proteins interact decreases as their distance grows. These two conditions already give us a model that is very close to Kleinberg's small world model, which belongs to the exponential tail family of graphs.

We now move to the second question. We have the following corollary.

Corollary 5.2. *Let us consider the signaling pathways reconstruction problem such that $k = O(n^{1/4})$ is a polynomial in n. Let $\mathcal{A}$ be an ρ-approximation algorithm for (3) (where $\rho = \tilde{O}(1)$) and outputs a set $\hat{S}$ of size $O(n^{1/3})$. Then for any constant ϵ, with high probability, we have $|\hat{S}| \leq (2+\epsilon-\rho)k$ and $|\hat{S}\cap S| \geq (1-\epsilon)\rho k$.*

In other words, a $(2, (1-\epsilon)\rho)$-detector exists. Furthermore, Corollary 5.2 is complemented by the following lower bound for the case where the network structure is unknown (the proof appears in Appendix H).

Theorem 5.3. *Consider the signaling pathways reconstruction problem with $\mu = \Theta(1)$, where the network structure is unknown and $\rho = \tilde{\Theta}(1)$. For any $(\alpha, 1-\rho)$-detector, its α has to be $\Omega(\rho n)$.*

Thus, if we want $|S \cap \hat{S}| = (1 \pm o(1))\rho k$, knowing the structure of the network will bring down α from ρn to $O(1)$. Notice that Theorem 5.3 gives a much stronger lower bound than Theorem 3.1.

Acknowledgements

This work was supported in part by an ARO MURI Grant W911NF-11-1-0036 and an NSF Grant CNS-0905086.

References

Ittai Abraham, Shiri Chechik, David Kempe, and Aleksandrs Slivkins. Low-distortion inference of latent similarities from a multiplex social network. *CoRR*, abs/1202.0922, 2012.

Louigi Addario-berry, Nicolas Broutin, Gbor Lugosi, and Luc Devroye. Combinatorial testing problems. *Annals of Statistics*, 38(5), 2010.

Ery Arias-Castro, Emmanuel J. Cands, Hannes Helgason, and Ofer Zeitouni. Searching for a trail of evidence in a maze. *Annals of Statistics*, 36(4), 2008.

Sanjeev Arora, Rong Ge, Sushant Sachdeva, and Grant Schoenebeck. Finding overlapping communities in social networks: toward a rigorous approach. In *ACM Conference on Electronic Commerce*, pages 37–54, 2012.

Marc Bailly-Bechet, Christian Borgs, Alfredo Braunstein, Jennifer T. Chayes, A. Dagkessamanskaia, J.-M. Franois, and Riccardo Zecchina. Finding undetected protein associations in cell signaling by belief propagation. *CoRR*, abs/1101.4573, 2011.

Maria-Florina Balcan, Christian Borgs, Mark Braverman, Jennifer T. Chayes, and Shang-Hua Teng. Finding endogenously formed communities. *CoRR*, abs/1201.4899, 2012.

Brian Ball, Brian Karrer, and M. E. J. Newman. An ecient and principled method for detecting communities in networks. *Phys. Rev. E*, 84, 036103, 2011.

Wei Chen, Zhenming Liu, Xiaorui Sun, and Yajun Wang. A game-theoretic framework to identify overlapping communities in social networks. *Data Min. Knowl. Discov.*, 21(2):224–240, September 2010. ISSN 1384-5810.

Fan Chung and Linyuan Lu. The average distances in random graphs with given expected degrees. *Internet Mathematics*, 1:15879–15882, 2002.

Rahul C. Deo, Luke Hunter, Gregory D. Lewis, Guillaume Pare, Ramachandran S. Vasan, Daniel Chasman, Thomas J. Wang, Robert E. Gerszten, and Frederick P. Roth. Interpreting metabolomic profiles using unbiased pathway models. *PLoS Computational Biology*, 6(2), 2010.

Marcus T. Dittrich, Gunnar W. Klau, Andreas Rosenwald, Thomas Dandekar, and Tobias Mller. Identifying functional modules in protein-protein interaction networks: an integrated exact approach. In *ISMB*, pages 223–231, 2008.

Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75 – 174, 2010.

Anna Goldenberg, Alice X. Zheng, Stephen E. Fienberg, and Edoardo M. Airoldi. A survey of statistical network models. *Foundations and Trends in Machine Learning*, 2(2):129–233, 2009.

M. B. Hastings. Community detection as an inference problem. *Phys. Rev. E*, 74(3), 2006.

Jarvis Haupt, Rui M. Castro, and Robert Nowak. Distilled sensing: Adaptive sampling for sparse detection and estimation. *IEEE Transactions on Information Theory*, 57(9):6222–6235, 2011.

Trey Ideker, Owen Ozier, Benno Schwikowski, and Andrew F. Siegel. Discovering regulatory and signalling circuits in molecular interaction networks. In *ISMB*, pages 233–240, 2002.

Md Jamiul Jahid and Jianhua Ruan. A steiner tree-based method for biomarker discovery and classification in breast cancer metastasis. *BMC Genomics*, 13(Suppl 6):S8, 2012.

Myunghwan Kim and Jure Leskovec. Multiplicative attribute graph model of real-world networks. *Internet Mathematics*, 8(1-2):113–160, 2012.

Jon Kleinberg. The small-world phenomenon: An algorithmic perspective. In *in Proceedings of the 32nd ACM Symposium on Theory of Computing*, pages 163–170, 2000.

Jure Leskovec, Kevin J. Lang, Anirban Dasgupta, and Michael W. Mahoney. Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters. *Internet Mathematics*, 6(1):29–123, 2009.

Alan Mislove, Massimiliano Marcon, Krishna P. Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, IMC '07, pages 29–42, New York, NY, USA, 2007. ACM.

M E Newman. Modularity and community structure in networks. 103(23):8577–8582, June 2006.

Gabriel Robins and Alexander Zelikovsky. Tighter bounds for graph steiner tree approximation. *SIAM Journal on Discrete Mathematics*, 19:122–134, 2005.

Jacob N. Shapiro and Nils B. Weidmann. Is the phone mightier than the sword? cell phones and insurgent violence in Iraq. *Working paper*, 2012.

Hossein Azari Soufiani and Edoardo M. Airoldi. Graphlet decomposition of a weighted network. *AISTATS*, abs/1203.2821, 2012.

Mauro Sozio and Aristides Gionis. The community-search problem and how to plan a successful cocktail party. In *ACM SIGKDD*, 2010.

Vijay V. Vazirani. *Approximation Algorithms*. Springer-Verlag, New York, NY, USA, 2001.

David Williams. *Probability with Martingales*. Cambridge University Press, 1991.

A Definitions of models and algorithms we used

A.1 Steiner tree problem

Definition A.1 (Steiner tree)**.** *Let $G = \{V, E\}$ be an edge-weighted graph. Let W be a subset of nodes in V. The* Steiner tree problem *is to find a tree that spans all the nodes in W and its edge weight sum is minimized.*

A.2 Small world

We now review Kleinberg's small world model (Kleinberg, 2000). We remark that Kleinberg's model allows the network to be directed. Here, we shall use a straightforward way to convert a directed graph into an undirected one: there is an undirected edge between u and v if and only if (u, v) or (v, u) is in the directed grpah. Thus, Kleinberg's small world model can be described as follows.

The set of nodes reside in a two-dimensional lattice points $\{(i, j) : i, j \in \{1, ..., \sqrt{n}\}\}$. The *lattice distance* is defined as $d((i, j), (k, \ell)) = |k - i| + |\ell - j|$. Let $r = 2$ and $q = \Theta(\log n)$ be a normalization term. There are two types of edges in the small world graph:

- *local edges:* if $d(u, v) \leq 1$, then there is an edge between $d(u, v)$.
- *long range edges:* if $d(u, v) > 1$, then with probability $q^{-1} \cdot d^{-r}(u, v)$ there is an edge between u and v, which is independent to the rest of the edges.

We shall define $d \triangleq \max_v \mathrm{E}[\text{degree}(v)]$. Since $q = \Theta(\log n)$, d is also a constant.

A.3 Chung and Lu's random graph

Next, we describe Chung and Lu's random graph model (Chung and Lu, 2002). In this model, we are given an expected degree sequence $\mathbf{w} = (w_1, ..., w_n)$, in which w_i represents the expected degree for v_i. The probability that there is an edge between v_i and v_j is $w_i w_j \rho$ for $\rho = \frac{1}{\sum_i w_i}$, which is independent to other edges. Furthermore, we shall assume the distribution of $\{w_1, ..., w_n\}$ follows a power law distribution with exponent parameter between 2 and 3.

B Proof of Theorem 3.1

We shall partition the support of the signals into three regions: the *high interval* $H = [1, 2 - \gamma]$, the *median interval* $M = (1 - \gamma, 1)$, and the *low interval* $L = [0, 1 - \gamma]$. When a node's signal is in H, the node is in S; when a node's signal is in L, the node is not in S. Thus, a detector's only goal is to identify the nodes from S among those nodes with their signals in M.

On the other hand, observe that conditioned on a node's signal is in M, the distribution of the signal is uniform in $(1 - \gamma, 1)$. This always holds regardless whether the node is in S or not. Therefore, no algorithm can do better than random guess in the region M. It remains to analyze the performance of an algorithm that only makes random guesses. One can see that with high probability: 1. the number of signals in M is $\Theta(\gamma n)$. 2. the number of nodes in S with signals in M is $\Theta(\gamma k)$. Thus, on average, a node in S will be picked up with probability $\Theta\left(\frac{\gamma k}{\gamma n}\right) = \Theta(k/n)$. Since we can select up to $O((\alpha - 1)k)$ nodes in the interval M (with high probability), the total number of nodes from S that will finally be in $\hat{S}$ is $O(k^2/n)$. Thus,

$$\mathrm{E}\left[|S - S'|\right] = (1 - o(1))\gamma k - O\left(\frac{k^2}{n}\right) = (1 - o(1))\gamma k$$

for sufficiently large n and k.

C Warmup: the algorithm for a line case.

We now give a solution for the case where the network forms a line, *i.e.,* $E = \{\{v_i, v_{i+1}\} : i < n\}$. While a line graph is not a realistic model for social or biological networks, analyzing this simple example helps us to understand how we may improve the performance of a detector algorithm by utilizing the information of the network structure.

Lemma C.1. *Let γ be an arbitrary constant. Consider the community detection problem, in which the underlying graph is a line and $k = o(n)$ is polynomial in n. There exists an efficient algorithm that returns a set $\hat{S}$ of size k and $|S - \hat{S}| = o(k)$ whp. In other words, for any γ, there exists a $(1, \gamma)$-detector.*

Proof. Observe that the subgraph induced by S has to be connected. Hence, there exists a j such that $S = \{v_j, v_{j+1}, ..., v_{j+k-1}\}$. Our algorithm works as follows: enumerate through all possible connected subgraphs of size k. Output an arbitrary one such that it contains at least $(1-\gamma)k-(\log n)\sqrt{k}$ signals in the high interval H. Also in what follows, we say a connected subgraph of size k *passes the threshold test* if and only if it contains at least $(1-\gamma)k-(\log n)\sqrt{k}$ signals in the high interval.

We need to show the following two events hold with high probability

- *Event 1:* The subgraph induced by S passes the threshold test.
- *Event 2:* There exists no connected subgraph S' such that $|S'| = k$, $|S - S'| = \Omega(k)$, and S' passes the threshold tests.

Event 1 implies that with high probability our algorithm will return at least one subgraph. Event 2 implies that with high probability the node set returned by our algorithm will be almost the same as S.

We may apply a standard Chernoff bound to prove that event 1 happens with high probability. Specifically,

$$\Pr[S \text{ fails the threshold test}] \leq \exp\left(-\frac{\log^2 n}{3}\right) \leq \frac{1}{n^2} \tag{4}$$

for sufficiently large n.

We next move to analyze the second event. Let ϵ be an arbitrarily small constant. We shall first show the probability that there exists a specific connected subset S' such that $|S - S'| \geq \epsilon k$ is small. Then we shall use a union bound to argue whp no bad S' will pass the threshold test.

Let S' be an arbitrary connected subset of size k such that $|S - S'| \geq \epsilon k$. In expectation, the total number of nodes in S that have signals in H is $(1-\epsilon)(1-\gamma)k$. By using a Chernoff bound again, the probability that S' will have more than $(1-\gamma)k - (\log n)\sqrt{k}$ nodes in H is $\leq \frac{1}{n^3}$ (no effort was made to optimize this bound). Finally, since there are at most $(n-k+1)$ connected subgraphs of size k, we may apply a union bound and get that the probability event 2 happens is $\leq \frac{n-k+1}{n^3} \leq 1/n^2$.

To sum up, with probability $1 - \frac{2}{n^2}$, we will find a set $\hat{S}$ of size k such that $|S - \hat{S}| = o(k)$. □

D Proof of Lemma 4.4

We need to prove two directions: 1. a **MinConnect** problem can be reduced to a Steiner tree problem with uniform edge weights. 2. a Steiner tree problem with uniform edge weights can be reduced to a **MinConnect** problem.

We use the following observation for the analysis for both directions: in an arbitrary graph with uniform edge weights, the cost of any spanning tree for a connected subset of size k is $k-1$. This observation is true because of the simple fact that a tree of size ℓ contains $\ell - 1$ edges for arbitrary ℓ.

Now both directions are straightforward. Let W be the set of nodes that need to be included in the **MinConnect** problem. To reduce the **MinConnect** problem to a Steiner tree problem, we require the Steiner tree to cover all the nodes in W in the same graph. Because of the above observation, a Steiner tree that covers W with minimum number of edges is also a minimum connected subgraph that covers W.

We can prove the other direction in a similar fashion.

E Proof of Proposition 4.6

We shall show that with probability $\left(\frac{\tau\ell}{n}\right)^{\ell-1}$ a random subset R of size ℓ is connected, where τ is a suitable constant (the probability is over both the generative model and the choice of the random subset). Then we can show that the expected number of subgraphs of size ℓ is $\binom{n}{\ell}\left(\frac{\tau\ell}{n}\right)^{\ell-1} = n\tau_0^\ell$ for another constant τ_0. We may next apply a Markov inequality to get the Proposition.

First, we shall imagine the set R is sampled in a sequential manner, *i.e.,* the first node v_1 from R is sampled uniformly from V. Then the second node v_2 is sampled uniformly from $V/\{v_1\}$. And in general the i-th node v_i in R is sampled uniformly from $V/\{v_1, ..., v_{i-1}\}$. After ℓ samplings, we would get a set R of size ℓ that is sampled uniformly among all the nodes. The reason we introduce this scheme is to couple the sampling of R with the generative model itself, by using the principle of deferred decision over long range edges: the existence of a long range edge is not revealed until we need it to make conclusions on connectivity. In other words, we want to test the connectivity of R sequentially when each member node of R and its neighbors are revealed to us step-by-step. The bottom line is that we need not know all the member nodes of R to conclude on connectivity.

To carry out our scheme, we introduce a coupled branching process that is represented by a stochastic tree $\{T(t)\}_{t\geq 1}$. This tree traces the current revealed nodes in $V(R)$. It is rooted at v_1 and grows over time according to the following procedure. First, let us introduce a labeling notion to help define our growing process: at each time t, we call any node in V *dead* if its long-range neighbors are revealed to us, and *active* otherwise. The notion of dead and active can apply to any nodes in V.

We start from a uniformly sampled node v_1 in V, decide $v_1 \in R$, and reveal all its long-range neighbors (in addition to the four local neighbors) via the generative model. Then $T(1)$ contains only v_1 and v_1 is labelled dead. To grow from $T(t)$ to $T(t+1)$, we implement the following steps:

- *Step 1.* Pick an arbitrary active node v in $V(T(t))$ and reveal all its neighbors. The active node is then labeled as dead.
- *Step 2.* Decide the set of nodes from $R/V(T(t))$ that are adjacent to v by using this observation: the distribution of $R/(V(T(t)))$ is uniform over $V/\left(V(T(t)) \cup \text{Dead}(t)\right)$, where $\text{Dead}(t)$ is the

union of all neighbors of all the dead nodes at time t.

- *Step 3.* Expand $T(t)$ to $T(t+1)$ by adding all the new neighbors of v that are in R.

Note that $V(T(t)) \in R$ by construction. Moreover, whenever the tree $T(t)$ stops growing before $|T(t)|$ reaches ℓ, we conclude that the current sample of R is not connected.

We next show by induction over ℓ that $\Pr[|T(t)| = k] \leq \left(\frac{\tau\ell}{n}\right)^{\ell-1}$. One potential obstacle in our analysis is that in the second step of the above procedure, the set $\mathrm{Dead}(t) \cup V(T(t))$ evolves in a complex manner, and it is not straightforward to find the neighbors of v that are in R over time. On the other hand, it is not difficult to show that $\Pr[|\mathrm{Dead}(t)| \geq \frac{n^{2/3}}{2}] \leq 2^{-\frac{n^{2/3}}{2}}$ at any moment (by using a Chernoff bound). In the rest of our analysis, we shall silently assume that $\mathrm{Dead}(t) \geq \frac{n^{2/3}}{2}$ does not happen. This assumption will result in an additive error of order $2^{-\frac{n^{2/3}}{2}}$ for the probability quantities we are interested in, which is an asymptotically small term. Since our goal is to give an upper bound on the probability that $|T(t)|$ eventually reaches ℓ, we can imagine the branching process works as follows: there is an (imaginary) adversary that decides how $T(t)$ grows and tries to maximize the probability that $|T(t)|$ eventually reaches ℓ. The adversary basically has to follow the above three-step procedure to grow $T(t)$, but in the second step the adversary needs to decide which set of nodes that is not allowed to choose (instead of using $\mathrm{Dead}(t) \cup V(T(t))$) as long as $|\mathrm{Dead}(t)|$ is sufficiently small, *i.e.,* $|\mathrm{Dead}(t)| \leq \frac{n^{2/3}}{2}$. When the adversary maximizes the probability that $|T(t)|$ eventually reaches ℓ, such quantity is also an upper bound on the same probability for the original process.

Thus, our goal is to understand the probability that $|T(t)|$ hits ℓ under the adversarial setting. Specifically, we want to prove the following statement.

Let $\ell \leq n^{1/3}$ and let G be a small world graph of size n. Let F be an arbitrary set of size $\frac{n}{2} - \frac{\ell}{2} \cdot n^{2/3}$. We shall refer F as the forbidden set*. Let R be a random subgraph from V/F of size ℓ. Let $P(\ell, n)$ be the probability that the coupled branching process $T(t)$ reaches ℓ nodes eventually. Then*

$$P(\ell, n) \leq \left(\frac{\tau\ell}{n}\right)^{\ell-1}. \tag{5}$$

Roughly speaking, the forbidden set F allows the adversary to choose the set $\mathrm{Dead}(t) \cup V(T(t))$ over time. Also, notice that the set F has to shrink as ℓ grows. Imposing such a technical assumption will makes the recursive analysis easier at the cost of weakening bound on k. Also, we shall refer to the branching process in which the adversary controls the set F as $T'(t)$.

The base case where $\ell = 1, 2$ is straightforward. Let us now move to the induction step for computing $P(\ell+1, n)$ when $P(1, n), ..., P(\ell, n)$ satisfy (5). Let v_1 be the first node in R. Let us define the random variables X and Y as follows: X is the total number of neighbors of v_1 and Y is the total number of nodes from R that are adjacent to v_1. Notice that $\mathrm{E}[X_1] \leq 4 + d$ (4 is the number of direct neighbors and d is the maximum expected number of long range edges among all the nodes). By using a Chernoff bound, we also have $\Pr[X_1 \geq g] \leq 2^{-g}$, when g is a sufficiently large constant (Chernoff bound is applicable because the number of long range edges can be expressed as the sum of independent variables).

Furthermore, we may also compute $\Pr[Y = j]$ asymptotically. Specifically, we have the following lemma.

Lemma E.1. *Let Y be the variable defined above. For any $i \leq \ell$, there exists a constant c_2 such that*

$$\Pr[Y = i] \leq \left(\frac{c_2\ell}{n}\right)^{i}. \tag{6}$$

Proof of Lemma E.1. Recall that we let X be the number of nodes that are adjacent to v_1. Also, recall that $\Pr[X > g] < 2^{-g}$, when g is a sufficiently large constant. We can also write $\Pr[X > g] \leq c_4 \cdot 2^{-g}$ for a sufficiently large c_4.

We have

$$\begin{aligned} \Pr[Y = i] &= \textstyle\sum_{1 \leq j \leq n} \Pr[Y = i | X = j] \Pr[X = j] \\ &\leq c_4 \textstyle\sum_{i \leq j \leq n} \frac{\binom{j}{i}\binom{n_0 - j}{\ell - i}}{\binom{n_0}{\ell}} \cdot 2^{-j}, \end{aligned} \tag{7}$$

where n_0 is $n - |F \cup \{v_1\}|$ is the total number of nodes that R may choose from. Notice that $n_0 = \Theta(n)$.

Let us focus on the terms $\frac{\binom{j}{i}\binom{n_0-j}{\ell-i}}{\binom{n_0}{\ell}} \cdot 2^{-j}$. One can see that when $j \geq 3i$, the terms $\frac{\binom{j}{i}\binom{n_0-j}{\ell-i}}{\binom{n_0}{\ell}} \cdot 2^{-j}$ decrease more sharply than a geometric progression with ratio $3/4$. Thus, the sum of the first $3i$ terms is the dominating term. We now give an upper bound on $\frac{\binom{j}{i}\binom{n_0-j}{\ell-i}}{\binom{n_0}{\ell}} \cdot 2^{-j}$ for the case $j \leq 3i$. Let $j = \alpha i$, where

$\alpha \le 3$, we have

$$\begin{aligned}
&\frac{\binom{\alpha i}{i}\binom{n_0-\alpha i}{\ell-i}}{\binom{n_0}{\ell}} \\
\le\ & \frac{c_5}{\sqrt{\ell}} \frac{\alpha^i \frac{(n_0-\alpha i)^{n_0-\alpha i}}{(\ell-i)^{\ell-i}(n_0-\ell-(\alpha-1)i)^{n_0-\ell-(\alpha-1)i}}}{\frac{n_0^{n_0}}{\ell^\ell (n_0-\ell)^{n_0-\ell}}} \\
&\qquad (c_5 \text{ is a sufficiently large constant.}) \\
\le\ & c_5 \cdot \alpha^i \frac{\{(n_0-\alpha i)(n_0-\ell)\}^{n_0-\ell-(1-\alpha)i}}{\{(n_0-\ell-(\alpha-1)i)n_0\}^{n_0-\ell-(\alpha-1)i}} \\
&\cdot \frac{\{\ell(n_0-\alpha i)\}^{\ell-i}}{\{(\ell-i)n_0\}^{\ell-i}} \cdot \frac{(n_0-\ell)^{(\alpha-1)i}}{n_0^{(\alpha-1)i}} \frac{\ell^i}{n_0^i}.
\end{aligned}$$

We can see that $(n_0-\alpha i)(n_0-\ell) \le (n_0-\ell-(\alpha-1)i)n$, $\ell(n_0-\alpha i) \le (\ell-i)n_0$, and $n_0-\ell \le n_0$. Thus,

$$\frac{\binom{\alpha i}{i}\binom{n_0-\alpha i}{\ell-i}}{\binom{n_0}{\ell}} \le \left(\frac{c_6 \ell}{n}\right)^2.$$

Together with (7), we finish the proof of Lemma E.1 □

Next, by the law of total probability, we also have $\Pr[|T'(t)| = \ell+1]$ equals to $\sum_{1\le i\le n} \Pr[|T'(t)| = \ell+1 | Y=i] \Pr[Y=i]$. We first walk through the analysis for $\Pr[|T'(t)| = \ell+1 | Y=1]$ and $\Pr[|T'(t)| = \ell+1 | Y=2]$. Then we give an asymptotic bound on $\Pr[|T'(t)| = \ell+1 | Y=i]$ for general i.

Let us start with the case $Y=1$. Let v_2 be the node in S that is connected with v_1. Here, $\Pr[|T'(t)| = \ell+1 \mid Y=1]$ reduces to a case that is covered by the induction hypothesis, *i.e.,* the tree $T'(t)$ contains $\ell+1$ nodes if and only if the subtree rooted at v_2 contains ℓ nodes. This v_2-rooted subtree is another branching process coupled with ℓ nodes, which are uniformly chosen from $V/(\{v_1\} \cup \Gamma(v_1) \cup F)$, where $\Gamma(v_1)$ is the set of v_1's neigbhors. Thus, by induction hyptothesis, we have $\Pr[|T'(t)| = \ell+1 \mid Y=1] \le P(\ell,n) \le \left(\frac{\tau\ell}{n}\right)^{\ell-1}$.

Next, let us move to the case for $Y=2$. Let v_2 and v_3 be the nodes in S that are connected with v_1. We may first grow the tree $T'(t)$ from v_2, then grow the tree from v_3. At the end, let i be the number of children of v_2 (and v_2 itself). The number of children of v_3 (and v_3 itself) is thus $\ell-i$. One may check that both subprocesses can be understood by using induction hypothesis (we also need to check that the size of the forbidden set does not violate the requirements in the induction hypothesis; but this is straightforward because we assumed that the total number of neighbors of R is less than $n^{2/3}/2$).

Thus, we have

$$\begin{aligned}
&\Pr[|T(t)| = \ell+1 | Y=2] \\
=\ & \sum_{i=1-1}^{\ell} \Big(\Pr[v_2\text{-rooted tree connected}, \\
&\qquad v_3\text{-rooted tree connected} \mid L=i, Y=2] \\
&\qquad \Pr[L=i \mid Y=2] \Big) \\
\le\ & \sum_{i=1}^{\ell-1} P(i,n) P(\ell-i,n) \binom{\ell-2}{i-1} \\
\le\ & \sum_{i=1}^{\ell-1} \left(\frac{\tau i}{n}\right)^{i-1} \left(\frac{\tau(\ell-i)}{n}\right)^{\ell-i-1} \binom{\ell-2}{i-1} \quad \text{(Induction)} \\
=\ & \frac{\tau^{\ell-2}}{n^{\ell-2}} \sum_{i=1}^{\ell-1} i^{i-1} (\ell-i)^{\ell-i-1} \binom{\ell-2}{i-1}
\end{aligned}$$

Let us define $a_j = j^{j-1}(\ell-j)^{\ell-j-1}$. Notice that $a_j = a_{\ell-j}$. So we have

$$\sum_{1\le j\le \ell} a_j \le 2 \sum_{1\le j\le \ell/2} a_j.$$

We may continue to compute the sum of a_j's:

$$\begin{aligned}
&\sum_{1\le j\le \ell/2} a_j \\
=\ & \sum_{1\le j\le \ell/2} j^{j-1}(\ell-j)^{\ell-j-1} \binom{\ell-2}{j-1} \\
\le\ & \sum_{1\le j\le \ell/2} j^{j-1} \ell^{\ell-j-1} \left(\frac{\ell-j-1}{\ell}\right)^{\ell-j-1} \binom{\ell-2}{j-1} \\
\le\ & e \sum_{1\le j\le \ell/2} \ell^{j-1} \ell^{\ell-j-1} \left(\frac{\ell-j-1}{\ell}\right)^{\ell-j-1} \\
\le\ & c_3 \ell^{\ell-2}
\end{aligned}$$

for some constant c_3. The second inequality uses the fact that

$$\begin{aligned}
j^{j-1}\binom{\ell-2}{j-1} &\le j^{j-1} \left(\frac{\ell-2}{j-1}\right)^{j-1} \\
&= j^{j-1} \left(\frac{\ell-2}{j}\right)^{j-1} \left(\frac{j}{j-1}\right)^{j} \\
&\le e\ell^{j-1}
\end{aligned}$$

Thus, we have

$$\Pr[|T(t)| = \ell+1 \mid Y=2] = c_3 \left(\frac{\tau\ell}{n}\right)^{\ell-2}. \quad (8)$$

In general, we may define the ℓ-hitting probability $P_j(\ell,n)$ for a "branching forest" that grows from j

different roots. When $j = 1$, it corresponds with the case for $Y = 1$, *i.e.,* $P_1(\ell, n) = P(\ell, n)$. When $j = 2$, (8) gives us $P_2(\ell, n) \leq c_3 \left(\frac{\tau\ell}{n}\right)^{\ell-2}$.

We now compute the general form of $P_j(\ell, n)$. Specifically, we shall show by induction that

$$P_j(\ell, n) \leq c_3^{j-1} \left(\frac{\tau\ell}{n}\right)^{\ell-j}. \tag{9}$$

The base cases for $j = 1, 2$ are already analyzed above. We now move to the induction case. We have the following recursive relation:

$$P_j(\ell, n) \leq \sum_{1\leq i\leq \ell-j+1} P_1(i, n) P_{j-1}(\ell - i, n) \binom{\ell - j}{i - 1}.$$

Thus, we have

$$\begin{aligned} & P_j(\ell, n) \\ \leq & \sum_{1\leq i\leq \ell-j+1} \left(\tfrac{\tau i}{n}\right)^{i-1} \left(\tfrac{\ell-i}{n}\right)^{\ell-i-j+1} c_3^{j-2} \tau^{\ell-i-j+1} \\ & \qquad \cdot \tbinom{\ell-j}{i-1} \\ = & \left(\tfrac{\tau}{n}\right)^{\ell-j} c_3^{j-2} \left(\sum_{1\leq i\leq \ell-j+1} i^{i-1} (\ell - i)^{\ell-i-j+1} \tbinom{\ell-j}{i-1}\right) \end{aligned} \tag{10}$$

It remains to analyze the term $\sum_{1\leq i\leq \ell-j+1} i^{i-1}(\ell - i)^{\ell-i-j+1}\binom{\ell-j}{i-1}$. Via some straightforward manipulation, we have

$$\sum_{1\leq i\leq \ell-j+1} i^{i-1}(\ell - i)^{\ell-i-j+1} \binom{\ell - j}{i - 1} \leq c_3 \ell^{\ell-j} \tag{11}$$

for some sufficiently large c_3. (11) and (10) together give us (9). Now we are ready to compute $P(\ell + 1, n)$ (and thus $\Pr[|T'(t)| = \ell + 1]$):

$$\begin{aligned} P(\ell + 1, n) &= \sum_{1\leq j\leq n} P(\ell + 1, n \mid Y = j) \Pr[Y = j] \\ &\leq \sum_{1\leq j\leq n} \left(\frac{\tau\ell}{n}\right)^{\ell-j} c_3^{j-1} \left(\frac{c_2\ell}{n}\right)^{j} \end{aligned}$$

When $\tau \geq 4c_2c_3$, we have

$$\begin{aligned} & \sum_{1\leq j\leq n} \left(\frac{\tau\ell}{n}\right)^{\ell-j} c_3^{j-1} \left(\frac{c_2\ell}{n}\right)^{j} \\ \leq & \sum_{1\leq j\leq n} \left(\frac{\tau\ell}{n}\right)^{\ell} 2^{-i} \leq \left(\frac{\tau\ell}{n}\right)^{\ell}. \end{aligned}$$

This completes the proof of Proposition 4.6.

F Proof of Proposition 4.8

We shall first describe a way to construct S. Then we will argue that γ portion of the nodes in S is statistically indistinguishable with a large number of nodes from V/S.

Recall that w_i is the average degree for the node v_i. Wlog, we shall let $w_1 \leq w_2 \leq ... \leq w_n$, where $w_n = c_0\sqrt{n}$ for some constant c_0. Since the degree distribution is a power law distribution, there exists a constant c_1 such that for all $i \leq c_1 n$, $w_i = \Theta(1)$. Let us randomly partition the set of nodes $\{v_1, v_2, ..., v_{c_1 n}\}$ into two subsets of equal size, namely $S_1 = \{v_{i_1}, ..., v_{i_{c_1 n/2}}\}$ and $S_2 = \{v_{i'_1}, ..., v_{i'_{c_1 n/2}}\}$. Notice that for any $v_i \in S_1 \cup S_2$, we have $\Pr[\{v_i, v_n\} \in E] = w_i w_n \rho \geq \frac{c_2}{\sqrt{n}}$ for some constant c_2.

Next, let us construct the set S: we shall first let $v_n \in S$ and the rest of the nodes in S will be picked up from S_1. Since with probability at least $\frac{c_2}{\sqrt{n}}$ there is an edge between a node in S_1 and v_n, in expectation the number of nodes that are connected with v_n is $\Omega(\sqrt{n})$. Thus, with high probability we are able to find a subset of size $k - 1$ that are all connected to v_n.

Now we analyze an arbitrary algorithm's performance. By using a Chernoff bound, we can see that with high probability in S there are $(1 - \gamma \pm o(1))k$ nodes that have signals in H and $(\gamma \pm o(1))$ nodes that have signals in M. Let us refer to the subset of nodes in S whose signals are in M as S_M. Recall that when the algorithm does not know the network structure, it will not be able to discover most of the nodes in S_M. Here, we shall show that the algorithm will behave in a similar way even that it knows the network structure.

Let us focus on the nodes in S_2. It is strightforward to see that with high probability the number of nodes in S_2 that are both connected with v_n and associated with signals in M is at lesat $(1 - \epsilon)\gamma\sqrt{n}$ for an arbitrary constant ϵ. Let us call the set of these nodes S'_2. The nodes in S'_2 are statistically indistinguishable from the nodes in S_M. Furthermore, the connectivity constraint is met for nodes in both sets. Thus, no algorithm can do better than randomly guessing. In other words, if an algorithm outputs a set of size $O(k)$, then the number of nodes in S_M that will be included is $o(\gamma k)$. This completes the proof of Proposition 4.8.

G Proof of Proposition 5.1

Let $\ell = |S_{\text{opt}}|$ be the size of the output. Wlog, we shall focus on the case $\ell \geq (1 + \epsilon)k$. The case for $\ell \leq (1 + \epsilon)k$ can be analyzed in a similar manner. Recall that $\Phi(x)$ is the cdf for the Gaussian variable $N(0, 1)$. Let $\nu_0 = \mathrm{E}[-\log(1 - \Phi(X))]$, where $X \sim N(0, 1)$, *i.e.,* the expected score of a node from V/S. Similarly, let $\nu_1 = \mathrm{E}[-\log(1 - \Phi(X))]$, where $X \sim N(\mu, 1)$. Furthermore, let $\eta \triangleq \nu_1/\nu_0$. Since μ is a constant, we have ν_0, ν_1, and $\nu_1 - \nu_0$ are all constants. Also, η is a function that grows with μ. Our way of setting the function $c(\cdot)$ is straightforward: we set $c(e) = \omega \triangleq \frac{\nu_1 - \nu_0}{2}$ for all e.

We first show that when S is substituted into (3), the objective value is at least $(1-\frac{\epsilon}{10})k\nu_1$. This can be seen by using a Chernoff type bound for the independent variables $-\log(1-\Phi(X))$ with $X \sim N(\mu, 0)$ (See Theorem I.1 in Appendix I for the statements and the proofs):

$$\Pr\left[\sum_{v\in S} b(v_i) \leq (1-\frac{\epsilon}{10})k\nu_1\right] \leq \exp(-\Theta(\epsilon^2\nu_1 k)). \tag{12}$$

Thus, the objective value is at least $E_s \triangleq (1-\frac{\epsilon}{10})k\nu_1 - \omega\cdot(k-1)$ with high probability. We next show that for any specific S' of size ℓ such that $|S'-S|+|S-S'| \geq \epsilon k$, we have

$$\Pr\left[\sum_{v_i\in S'} b(v_i) - \omega(|S'|-1) \geq E_s\right] \leq \exp(-g(\mu)\epsilon^2\ell), \tag{13}$$

where $g(\mu)$ is a monotonic function in μ. Then, using the fact that with probability $\geq 1-\epsilon$ the number of connected subgraphs of size ℓ is $\leq \frac{n}{\epsilon}(\tau_0)^\ell$ for some constant τ_0, we can conclude that whp any S' of size ℓ such that $|S'-S|+|S-S'| \geq \epsilon k$ cannot be an optimal solution.

We now move to prove (13). Our goal thus is to give a bound on the event $\sum_{v_i\in S'} b(v_i) - \omega(|S'|-1) \geq E_s$. The probability is maximized when $S \subset S'$. Let us write $S_2' = S' - S$. Thus, we shall find a bound for $\Pr[\sum_{v_i\in S} b(v_i) + \sum_{v_i\in S_2'} b(v_i) - \omega(\ell-1) \geq E_s]$.

When $\sum_{v_i\in S} b(v_i) + \sum_{v_i\in S_2'} b(v_i) \geq E_s + \omega(\ell-1)$, there exists a $\Delta \in \{-n, ..., n\}$ such that

$$\sum_{v_i\in S} b(v_i) \geq k\nu_1 + \Delta$$

and

$$\sum_{v_i\in S_2'} b(v_i) \geq -\frac{\epsilon\nu_1 k}{10} - \Delta + (\ell-k)\omega - 1.$$

Let us write $\Delta = \delta k\nu_1$ and $\ell = hk$. When $\Delta \geq 0$, we have

$$\Pr\left[\sum_{v_i\in S} b(v_i) = k\tau_1 + \Delta\right] \leq \exp(-c\delta^2\eta k\nu_0)$$

and when $\Delta \leq \Delta_m \triangleq \frac{\eta-1}{2}(\ell-k)\nu_0$:

$$\begin{aligned}
&\Pr\left[\sum_{v_i\in S_2'} b(v_i) \geq \frac{-\epsilon\nu_1 k}{10} - \Delta + (\ell-k)\omega - 1\right] \\
\leq\ & \exp\left(-\frac{c\left(\frac{\eta-1}{2}(\ell-k)\nu_0 - \frac{\epsilon}{10}\nu_1 k - \Delta\right)^2}{\ell\tau_0}\right) \\
\leq\ & \exp\left(-\frac{ck\nu_0}{h}\left(\frac{\eta-1}{2}(h-1) - \frac{\epsilon\eta}{10} - \delta\eta\right)^2\right).
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\Pr[\sum_{v_i\in S} b(v_i) + \sum_{v_i\in S_2'} b(v_i) - \omega(\ell-1) \geq E_s] \\
\leq\ & \sum_{\Delta\leq 0} \exp\left(-\frac{ck\nu_0}{h}\left(\frac{\eta-1}{2}(h-1) - \frac{\epsilon\eta}{10} - \delta\eta\right)^2\right) \\
&+ \sum_{1\leq\Delta\leq\Delta_m} \exp(-c\delta^2\eta k\nu_0) \\
&\quad\cdot \exp\left(-\frac{ck\nu_0}{h}\left(\frac{\eta-1}{2}(h-1) - \frac{\epsilon\eta}{10} - \delta\eta\right)^2\right) \\
&+ \sum_{\Delta>\Delta_m} \exp(-c\delta^2\eta k\nu_0).
\end{aligned}$$

It is not difficult to see that both the first and third summations $\leq \exp(-\Theta(\epsilon^2(\eta-1)^2\nu_0\ell/\eta))$. Therefore, it remains to analyze the second summation in the above inequality. Specifically, we want to understand when

$$\exp\left(-c\delta^2\eta k\nu_0 - \frac{ck\nu_0}{h}\left(\frac{\eta-1}{2}(h-1) - \frac{\epsilon\eta}{10} - \delta\eta\right)^2\right) \tag{14}$$

is maximized. One can see that the exponent is a quadratic function in δ, which is maximized when

$$\delta = \frac{2(h-1)(\frac{2\eta}{5} - \frac{1}{2})}{\frac{\eta}{h}+1}.$$

When we plug in the optimal value of δ to (14), we have

$$\begin{aligned}
&\exp\left(-c\delta^2\eta k\nu_0 - \tfrac{ck\nu_0}{h}\left(\tfrac{\eta-1}{2}(h-1) - \tfrac{\epsilon\eta}{10} - \delta\eta\right)^2\right) \\
&\leq \exp(-\Theta(\epsilon^2(\eta-1)^2\nu_0\ell/\eta)).
\end{aligned}$$

Thus, (13) indeed holds.

H Proof of Theorem 5.3

Proof. Let $\phi(x)$ be the pdf of $N(0,1)$. First, observe that for any non-negative functions f and g such that $\phi = f + g$, we may interpret a sample from $N(0,1)$ as a sample from the mixture of two distributions from f and g by using the following procedure:

- First let $F = \int_{-\infty}^{\infty} f(x)dx$ and $G = \int_{-\infty}^{\infty} g(x)dx$.
- Then with probability $\frac{F}{F+G}$, we draw a sample from the ditribution with pdf $\frac{f(x)}{F}$ and with probability $\frac{G}{F+G}$ we draw a sample from the distribution with pdf $\frac{g(x)}{G}$.

Let $\phi_\mu(x)$ be the pdf for $N(\mu, 0)$ and let $\tau = \frac{\phi(\mu)}{\phi_\mu(\mu)}$. We shall decompose $\phi(x)$ into a mixture of $\tau\cdot\phi_\mu(x)$ and $R(x) \triangleq \phi(x) - \tau\cdot\phi_\mu(x)$. We consider the following strictly simpler problem and give a lower bound on α

for this problem: we still have the same setting that nodes from V/S and from S receive samples from different distributions and we are asked to find S. The only difference here is that when $v_i \in V/S$ receives a sample from $N(0,1)$, we assume the sample is generated from the mixture of $\tau \cdot \phi_\mu(x)$ and $R(x)$. Furthermore, when v_i is sampled from $R(x)$, we also *explicitly label* v_i as from $R(x)$. In other words, the algorithm knows the set of nodes that are sampled from $R(x)$. Notice that the new problem gives a strict superset of information and thus is information theoretically easier than the original problem.

Next, let us move to find a lower bound on α for the new problem. When a node is labeled as from $R(x)$, it is clear that the node should not be part of the output. It remains for us to find S from the rest of the non-labeled node. But notice that all the rest of the signals are sampled from $N(\mu, 0)$. Thus, we cannot do better than randomly guessing. Since $\mu = \Theta(1)$, we have $\tau = \Theta(1)$. Thus, the size of the remaining unlabeled nodes is still $\Theta(n)$ (with high probability). One can see that in order to cover ρ portion of nodes from S, the size of the final output has to be $\Theta(\rho n)$. □

I Concentration bounds

In this section, we prove the following large deviations bound.

Theorem I.1. *Let $Y_i \sim N(0,1)$ and $X_i \sim N(\mu, 1)$. Let $\Phi(\cdot)$ be the cdf for $N(0,1)$. Let $\mu_x = \mathrm{E}[-\log(1-\Phi(X_i))]$ and $\mu_y = \mathrm{E}[-\log(1-\Phi(Y_i))]$. We have*

$$\begin{aligned}&\Pr\left[\left|\textstyle\sum_{1\le i\le n} -\log(1-\Phi(X_i)) - n\mu_x\right| \ge \epsilon n\mu_x\right]\\ &\le \exp(-c\epsilon^2 n\mu_x)\end{aligned} \tag{15}$$

and

$$\begin{aligned}&\Pr\left[\left|\textstyle\sum_{1\le i\le n} -\log(1-\Phi(Y_i)) - n\mu_y\right| \ge \epsilon n\mu_y\right]\\ &\le \exp(-c\epsilon^2 n\mu_y)\end{aligned} \tag{16}$$

for a suitable constant $c > 0$.

Proof. We shall prove the lower tail of (17), *i.e.,*

$$\begin{aligned}&\Pr\left[\textstyle\sum_{1\le i\le n} -\log(1-\Phi(X_i)) \le (1-\epsilon)n\mu_x\right]\\ &\le \exp(-c\epsilon^2 n\mu_x)\end{aligned} \tag{17}$$

for some constant $c > 0$. The other cases can be analyzed in a similar manner. Consider the moment generating function (mgf) of $-\log(1-\Phi(X_i))$, where $X_i \sim N(\mu, 1)$ and $\Phi(\cdot)$ is the cdf of $N(0,1)$. For convenience, we also denote $\bar\Phi(\cdot) = 1 - \Phi(\cdot)$ as the tail distribution of $N(0,1)$. We use the bound (Williams, 1991)

$$\bar\Phi(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-y^2/2} dy \ge \frac{1}{\sqrt{2\pi}(x+1/x)} e^{-x^2/2} \tag{18}$$

for $x > 0$. We shall prove that the mgf of $-\log(1-\Phi(X_i))$ exists and is finite in a neighborhood of zero. Namely, consider

$$\begin{aligned}\phi(\theta) &:= E[e^{-\theta \log \bar\Phi(X_i)}]\\ &= E\left[\tfrac{1}{(\bar\Phi(X_i))^\theta}\right]\\ &= \textstyle\int_{-\infty}^{\infty} \frac{1}{(\bar\Phi(x))^\theta} \frac{1}{\sqrt{2\pi}} e^{-(x-\mu)^2/2} dx\end{aligned} \tag{19}$$

Using (18), and by considering the region $\{x \le \eta\}$ and $\{x > \eta\}$ for some $\eta > 0$, the quantity (19) is bounded from above by

$$\begin{aligned}&\max\{\bar\Phi(\eta)^{-\theta}, 1\} \int_{-\infty}^{\eta} e^{-(x-\mu)^2/2} dx\\ &+ \int_\eta^\infty \left(\sqrt{2\pi}\left(x + \frac{1}{x}\right) e^{x^2/2}\right)^\theta \frac{1}{\sqrt{2\pi}} e^{-(x-\mu)^2/2} dx\\ =\ &\max\{\bar\Phi(\eta)^{-\theta}, 1\} \int_{-\infty}^{\eta} e^{-(x-\mu)^2/2} dx + (2\pi)^{(\theta-1)/2}\\ &\int_\eta^\infty \left(x + \frac{1}{x}\right)^\theta e^{\theta x^2/2 - (x-\mu)^2/2} dx\end{aligned} \tag{20}$$

Consider the second term in (20). For $0 < \theta < 1$, it is bounded by

$$(2\pi)^{(\theta-1)/2} \int_{x>\eta} C_1 x^\theta e^{\theta x^2/2 - (x-\mu)^2/2} dx < \infty \tag{21}$$

for some $C_1 > 0$, and for $-1 < \theta < 0$, it is bounded by

$$(2\pi)^{(\theta-1)/2} \int_{x>\eta} C_2 x^{-\theta} e^{\theta x^2/2 - (x-\mu)^2/2} dx < \infty \tag{22}$$

for some $C_2 > 0$. Therefore $(-1, 1)$ is contained in the domain of convergence of $\phi(\theta)$. This implies that $\phi(\theta)$ is infinitely differentiable in $(-1, 1)$. Define $\psi(\theta) = \log \phi(\theta)$ as the logarithmic mgf of $-\log(1-\Phi(X_i))$. The same convergence and differentiability behavior then holds for $\psi(\cdot)$ in the same region $(-1, 1)$.

To proceed, we use the Chernoff inequality

$$\Pr\left[\sum_{1\le i\le n} -\log \bar\Phi(X_i) \le (1-\epsilon)n\mu_x\right] \le e^{\theta(1-\epsilon)n\mu_x + n\psi(-\theta)} \tag{23}$$

for $0 \le \theta < 1$. Using the Taylor expansion $\psi(-\theta) = -\psi'(0)\theta^2 + \psi''(\zeta)\theta^2/2$ for some $\zeta \in (-\theta, 0)$, and the fact that $\psi'(0) = \mu_x$, (23) becomes

$$\begin{aligned}&\exp\left\{\theta(1-\epsilon)n\mu_x - n\psi'(0)\theta^2 + n\psi''(\zeta)\tfrac{\theta^2}{2}\right\}\\ &= \exp\left\{-\epsilon\theta n\mu_x + n\psi''(\zeta)\tfrac{\theta^2}{2}\right\}\\ &\le \exp\left\{-\epsilon\theta n\mu_x + n \sup_{u\in[-\theta,0]} \psi''(u)\tfrac{\theta^2}{2}\right\}\end{aligned}$$

For $\epsilon < N_1$, we choose $\theta = c_1\epsilon$, and the value of $0 < c_1 < N_2$ will be chosen small enough to get our result. Note that for $\epsilon < N_1$ and $c_1 < N_2$, any choice of $\theta = c_1\epsilon$ implies that $\sup_{u\in[-c_1\epsilon,0]} \psi''(u) \leq M$ for some constant $M > 0$. Hence (24) becomes

$$\begin{aligned} & \exp\left\{-c_1\epsilon^2 n\left(\mu_x - \sup_{u\in[-c\epsilon,0]} \psi''(u)\frac{c_1}{2}\right)\right\} \\ \leq \quad & \exp\left\{-c_1\epsilon^2 n\left(\mu_x - M\frac{c_1}{2}\right)\right\} \end{aligned}$$

Choosing c_1 small enough will then give $\mu_x - M\frac{c_1}{2} > 0$, which concludes the theorem. □

J Missing calculations

J.1 Proof of Equation 2

$$\begin{aligned} & \gamma^{k_0+\Delta k}\frac{1.55k(\tau_0)^\ell}{p\epsilon} \\ = \quad & \frac{1.55k}{p\epsilon}\gamma^{k_0+\Delta k}\tau_0^{k+\Delta k} \\ = \quad & \frac{1.55k}{p\epsilon}\gamma^{k_0+\Delta k}(\tau_0)^{k+\Delta k} \\ = \quad & \frac{1.55k}{p\epsilon}(\tau_0)^k\gamma^{k_0}(\gamma\tau_0)^{\Delta k} \\ \leq \quad & \frac{1.55k}{p\epsilon}(\tau_0)^k(\gamma^{\lambda\gamma})^k(\gamma\tau_0)^{\Delta k} \quad \text{(Using } k_0 \geq \lambda\gamma k\text{)} \\ \leq \quad & (2\tau_0\gamma^{\lambda\gamma})^k(\gamma\tau_0)^{\Delta k} \\ & \qquad \text{(using } \tfrac{1.55k}{p\epsilon} \leq 2^k \text{ for sufficiently large } k.\text{)} \\ \leq \quad & c_0^{-k} \end{aligned}$$

The last inequality holds when $\gamma^{\lambda\gamma} \leq \frac{1}{2c_0\tau_0}$ for any constant c_0.