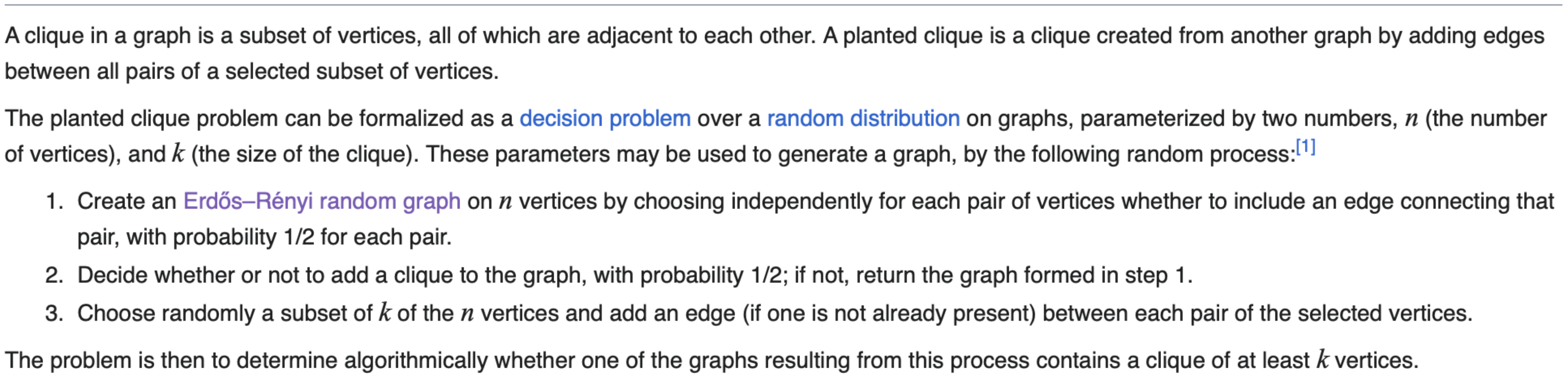
Signal in Noise – Graph versions



A clique in a graph is a subset of vertices, all of which are adjacent to each other. A planted clique is a clique created from another graph by adding edges between all pairs of a selected subset of vertices.

The planted clique problem can be formalized as a decision problem over a random distribution on graphs, parameterized by two numbers, n (the number of vertices), and k (the size of the clique). These parameters may be used to generate a graph, by the following random process:[1]

1. Create an Erdős–Rényi random graph on n vertices by choosing independently for each pair of vertices whether to include an edge connecting that pair, with probability 1/2 for each pair.
2. Decide whether or not to add a clique to the graph, with probability 1/2; if not, return the graph formed in step 1.
3. Choose randomly a subset of k of the n vertices and add an edge (if one is not already present) between each pair of the selected vertices.

The problem is then to determine algorithmically whether one of the graphs resulting from this process contains a clique of at least k vertices.

Above is from wikipedia….put simply (but less elegantly) – given a graph that you suspect might contain a clique of size k, in a background of noise. or might just be all "noise" (an Erdos-Renyi graph) – can you perform a test to give a high probability correct answer to "is this all noise or is there a k-clique in it"?

Of course variations

- Of the detection problem
  - Is there a clique of size at least k etc.
- Of the search problem
  - Find the clique, or
  - Approximately find the clique (for whatever sensible notion of approximately you think relevant).

The detect signal in noise is an important problem!

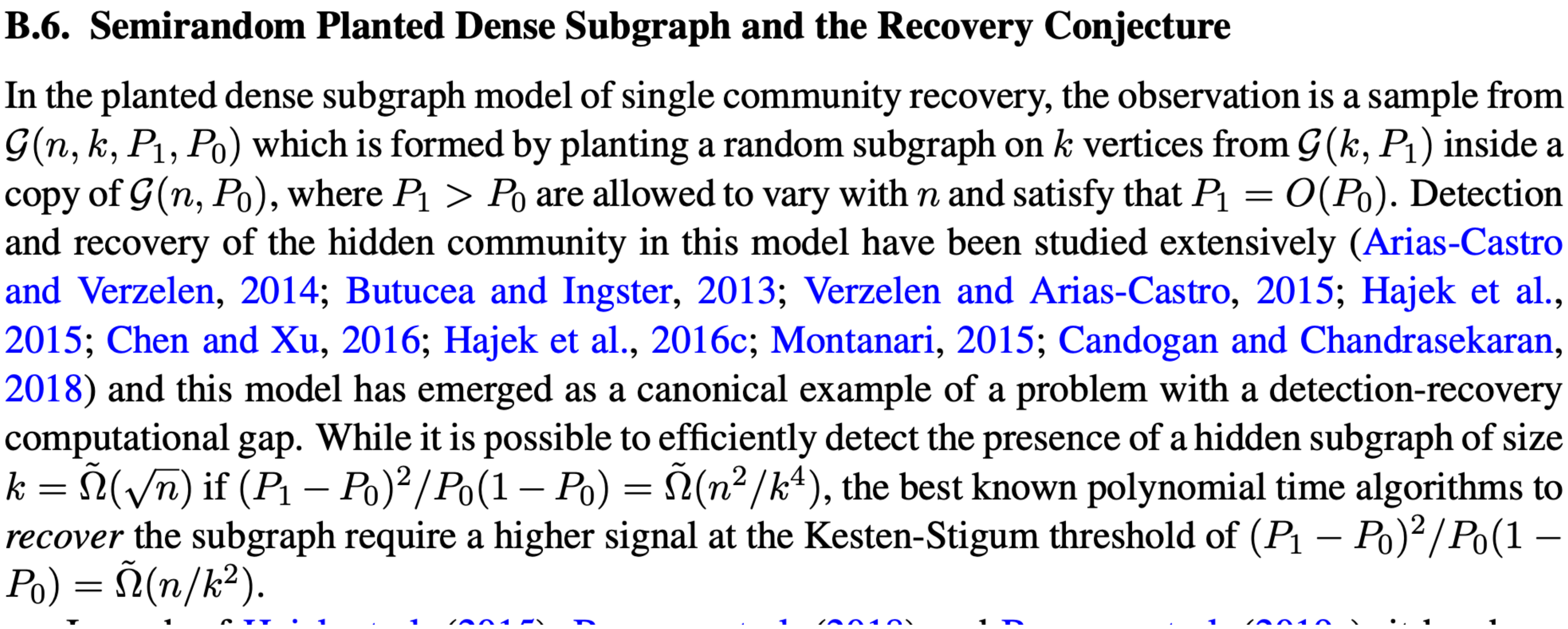
- Practically – most algorithms assume there is a signal. When given “white noise” – they still return an answer.
- Theoretically – since detection ought to be typically easier than finding…then if you can’t detect with reliability/speed then you almost certainly can’t find (in a reasonable way)
- There is a subtlety here – the Erdos-Renyi graph process will almost certainly have a k-clique above a certain edge density/probability of adding an edge. So finding a k-clique doesn’t mean it is a “meaningful k-clique”. Notion of statistically impossible to decide (if the clique ius planted….meaningful).
- This is not just a theoretical issue…arises in my own area of research - robust fitting…is the “structure” you found just noise (outliers) or a real structure.

Planted dense subgraph

- A k-clique is the densest (most fully connected) graph on k-vertices.
- Maybe our problem only requires finding (or checking whether exists) a dense subgraph (for suitable definition of just how dense)?
- Reasonably obvious generalization/weakening of planted clique.
  - Model
    - 1. generate $G_{n,p}$
    - Select randomly k-vertices
    - For pairs of vertices in this k-subset, if not already connected (by the noise - $G_{n,p}$) then add an edge with suitable probability (to get the required density of edges)
- Unsurprisingly similar algorithms can be use to find/detect, and similar things can be said about the thresholds at which is it impossible (statistically), or hard, or easy…..
- See for example https://arxiv.org/abs/2004.13978 or do your own web search! ☺

Examples of what is known

http://proceedings.mlr.press/v125/brennan20a/brennan20a.pdf



B.6. Semirandom Planted Dense Subgraph and the Recovery Conjecture

In the planted dense subgraph model of single community recovery, the observation is a sample from $\mathcal{G}(n, k, P_1, P_0)$ which is formed by planting a random subgraph on k vertices from $\mathcal{G}(k, P_1)$ inside a copy of $\mathcal{G}(n, P_0)$, where $P_1 > P_0$ are allowed to vary with n and satisfy that $P_1 = O(P_0)$. Detection and recovery of the hidden community in this model have been studied extensively (Arias-Castro and Verzelen, 2014; Butucea and Ingster, 2013; Verzelen and Arias-Castro, 2015; Hajek et al., 2015; Chen and Xu, 2016; Hajek et al., 2016c; Montanari, 2015; Candogan and Chandrasekaran, 2018) and this model has emerged as a canonical example of a problem with a detection-recovery computational gap. While it is possible to efficiently detect the presence of a hidden subgraph of size $k = \tilde{\Omega}(\sqrt{n})$ if $(P_1 - P_0)^2/P_0(1 - P_0) = \tilde{\Omega}(n^2/k^4)$, the best known polynomial time algorithms to *recover* the subgraph require a higher signal at the Kesten-Stigum threshold of $(P_1 - P_0)^2/P_0(1 - P_0) = \tilde{\Omega}(n/k^2)$.

Examples of what is known

http://proceedings.mlr.press/v125/brennan20a/brennan20a.pdf

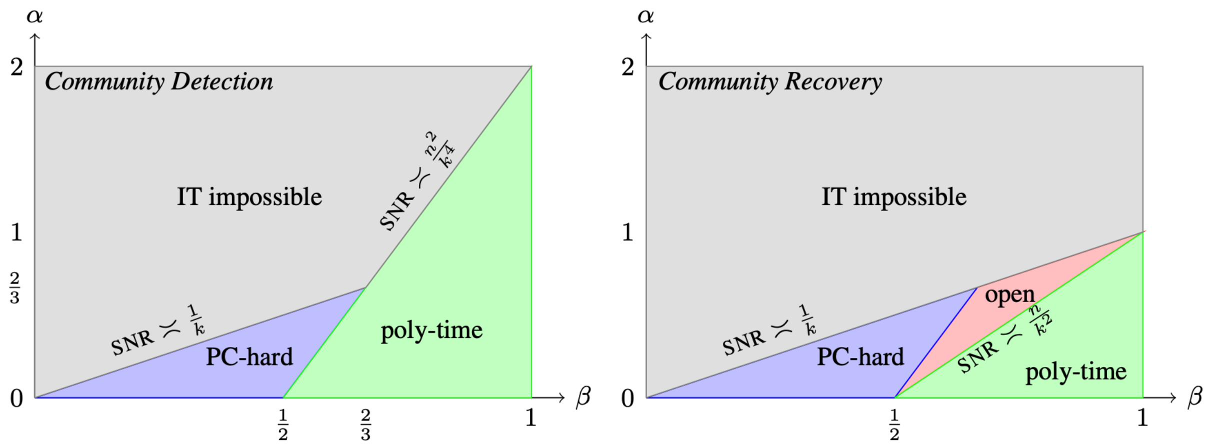


Figure 2: Prior computational and statistical barriers in the detection and recovery of a single hidden community from the PC conjecture (Hajek et al., 2015; Brennan et al., 2018, 2019a). The axes are parameterized by α and β where $\mathrm{SNR} = \frac{(P_1 - P_0)^2}{P_0(1-P_0)} = \tilde{\Theta}(n^{-\alpha})$ and $k = \tilde{\Theta}(n^{\beta})$. The red region is conjectured to be hard but no PC reductions showing this are known.

Whole “Zoo” of planted models…

- Get model of backround noise for that situation (Graph, Hypergraph, Simplicial Complex, etc.)
- Define way of “planting” a signal in that noise
- Look at ways to theoretically determine what is possible (for detection, find solution)
- Look at ways to efficiently get a good solution in practice
- Look at situations that this model (and associated algorithms and theory) might be useful (explain, give good ways to solve etc.).