Signal Processing/AI/ML versions of (large scale) random graphs

Random models

- We already met split graphs – which *could* model situations where we expect *perfect* connection between members of a “community”/cluster
- However, in real networks/graphs we expect “random” (probabilistic, varying by sample) graphs that allow some of the members of the “community” to have weak or no connection. Likewise variation and randomness in links between other (not in the community) vertices and between those and the community.
- A whole variety of random models have been proposed and studied.

Erdos-Renyi random model

- We start with the perhaps most famous (and first) random graph model (which has some relevance for some physical phenomena – but is often studied for its own sake – what it teaches us about this “landscape” of random models.
- $G_{n,p}$ is the Erdos-Renyi random graph with n vertices and p is the probability that a potential edge (pair of vertices) taken uniformly at random, is an edge.
- A very interesting property is that *samples* taken from this model, will undergo (with very high probability) a phase transition. Under a threshold, the graphs will consist of mostly isolated vertices and a “few” connections between small numbers of vertices – very small connected components. Above the thrsold there will be a Giant Component – a set of vertices that is almost all of the vertices and is connected.
- (Technical terminology – the edges are sample by *independent* Bernoulli (0 or 1 with probability p 1-p) variables)

In a 1960 paper, Erdős and Rényi[6] described the behavior of $G(n, p)$ very precisely for various values of p. Their results included that:

- If $np < 1$, then a graph in $G(n, p)$ will almost surely have no connected components of size larger than $O(\log(n))$.
- If $np = 1$, then a graph in $G(n, p)$ will almost surely have a largest component whose size is of order $n^{2/3}$.
- If $np \to c > 1$, where c is a constant, then a graph in $G(n, p)$ will almost surely have a unique giant component containing a positive fraction of the vertices. No other component will contain more than $O(\log(n))$ vertices.
- If $p < \frac{(1-\varepsilon)\ln n}{n}$, then a graph in $G(n, p)$ will almost surely contain isolated vertices, and thus be disconnected.
- If $p > \frac{(1+\varepsilon)\ln n}{n}$, then a graph in $G(n, p)$ will almost surely be connected.

Thus $\frac{\ln n}{n}$ is a sharp threshold for the connectedness of $G(n, p)$.

(Wikipedia)

Relation to percolation [edit]

In percolation theory one examines a finite or infinite graph and removes edges (or links) randomly. Thus the Erdős–Rényi process is in fact unweighted link percolation on the complete graph. (One refers to percolation in which nodes and/or links are removed with heterogeneous weights as weighted percolation). As percolation theory has much of its roots in physics, much of the research done was on the lattices in Euclidean spaces. The transition at $np = 1$ from giant component to small component has analogs for these graphs, but for lattices the transition point is difficult to determine. Physicists often refer to study of the complete graph as a mean field theory. Thus the Erdős–Rényi process is the mean-field case of percolation.

Some significant work was also done on percolation on random graphs. From a physicist's point of view this would still be a mean-field model, so the justification of the research is often formulated in terms of the robustness of the graph, viewed as a communication network. Given a random graph of $n \gg 1$ nodes with an average degree $\langle k \rangle$. Remove randomly a fraction $1 - p'$ of nodes and leave only a fraction p' from the network. There exists a critical percolation threshold $p'_c = \frac{1}{\langle k \rangle}$ below which the network becomes fragmented while above p'_c a giant connected component of order n exists. The relative size of the giant component, P_∞, is given by[6][1][2][9]

$$P_\infty = p'[1 - \exp(-\langle k \rangle P_\infty)].$$

Caveats [edit]

Both of the two major assumptions of the $G(n, p)$ model (that edges are independent and that each edge is equally likely) may be inappropriate for modeling certain real-life phenomena. Erdős–Rényi graphs have low clustering, unlike many social networks.[10] Some modeling alternatives include Barabási–Albert model and Watts and Strogatz model. These alternative models are not percolation processes, but instead represent a growth and rewiring model, respectively.

(Interlude – software to experiment with models)

- There are MANY “environments” (R/Rstudio, Python packages…) that are free to install/use – and of course some that are commercial (Matlab). What you prefer to use will be a personal choice…..
- I very much recommend “playing” with random graph models (indeed any new concept you meet and can find software for) to develop intuition and test your understanding…
- Here I will illustrate some things using R/Rstudio

Examples – R/Rstudio – 2% below critical

>install.packages("igraph")
>library(igraph)

> n=1000
> p_crit=log(n)/n
> g <- sample_gnp(n,0.98*p_crit)
>components(g)
(some output omitted)
$csize
[1] 999   1

$no
[1] 2

Examples – 2% above critical

```
> g <- sample_gnp(n,1.02*p_crit)
> components(g)
(some output omitted)
$csize
[1] 1000

$no
[1] 1
```

Examples – 2% of critical

```
> g <- sample_gnp(n,0.02*p_crit)
>components(g)
(output omitted)
$no
[1] 938
```

For more detail

MIT notes https://economics.mit.edu/sites/default/files/inline-files/Lecture%204%20-%20Erdos-Renyi%20Graphs%20and%20Phase%20Transitions.pdf

https://ocw.mit.edu/courses/14-15j-networks-spring-2018/pages/lecture-and-recitation-notes/

….or web search ☺

Stochastic Block Model

The **stochastic block model** is a generative model for random graphs. This model tends to produce graphs containing *communities*, subsets of nodes characterized by being connected with one another with particular edge densities. For example, edges may be more common within communities than between communities. Its mathematical formulation has been firstly introduced in 1983 in the field of social network by Paul W. Holland et al.[1] The stochastic block model is important in statistics, machine learning, and network science, where it serves as a useful benchmark for the task of recovering community structure in graph data.

Definition [edit]

The stochastic block model takes the following parameters:

- The number n of vertices;
- a partition of the vertex set $\{1, \dots, n\}$ into disjoint subsets $C_1, \dots, C_r$, called *communities*;
- a symmetric $r \times r$ matrix P of edge probabilities.

The edge set is then sampled at random as follows: any two vertices $u \in C_i$ and $v \in C_j$ are connected by an edge with probability P_{ij}. An example problem is: given a graph with n vertices, where the edges are sampled as described, recover the groups $C_1, \dots, C_r$.

(Wikipedia)

Relationship to some other models…

If the probability matrix is a constant, in the sense that $P_{ij} = p$ for all i, j, then the result is the Erdős–Rényi model $G(n, p)$. This case is degenerate—the partition into communities becomes irrelevant—but it illustrates a close relationship to the Erdős–Rényi model.

The *planted partition model* is the special case that the values of the probability matrix P are a constant p on the diagonal and another constant q off the diagonal. Thus two vertices within the same community share an edge with probability p, while two vertices in different communities share an edge with probability q. Sometimes it is this restricted model that is called the stochastic block model. The case where $p > q$ is called an *assortative* model, while the case $p < q$ is called *disassortative*.

Returning to the general stochastic block model, a model is called *strongly assortative* if $P_{ii} > P_{jk}$ whenever $j \neq k$: all diagonal entries dominate all off-diagonal entries. A model is called *weakly assortative* if $P_{ii} > P_{ij}$ whenever $i \neq j$: each diagonal entry is only required to dominate the rest of its own row and column.[2] *Disassortative* forms of this terminology exist, by reversing all inequalities. For some algorithms, recovery might be easier for block models with assortative or disassortative conditions of this form.[2]

(Wikipedia)

Applications of SBM – what type of tasks?

Typical statistical tasks [edit]

Much of the literature on algorithmic community detection addresses three statistical tasks: detection, partial recovery, and exact recovery.

Detection [edit]

The goal of detection algorithms is simply to determine, given a sampled graph, whether the graph has latent community structure. More precisely, a graph might be generated, with some known prior probability, from a known stochastic block model, and otherwise from a similar Erdos-Renyi model. The algorithmic task is to correctly identify which of these two underlying models generated the graph.[3]

(Wikipedia)

Partial recovery [edit]

In partial recovery, the goal is to approximately determine the latent partition into communities, in the sense of finding a partition that is correlated with the true partition significantly better than a random guess.[4]

Exact recovery [edit]

In exact recovery, the goal is to recover the latent partition into communities exactly. The community sizes and probability matrix may be known[5] or unknown.[6]

Statistical threshold behaviour

Stochastic block models exhibit a sharp threshold effect reminiscent of percolation thresholds.[7][3][8] Suppose that we allow the size n of the graph to grow, keeping the community sizes in fixed proportions. If the probability matrix remains fixed, tasks such as partial and exact recovery become feasible for all non-degenerate parameter settings. However, if we scale down the probability matrix at a suitable rate as n increases, we observe a sharp phase transition: for certain settings of the parameters, it will become possible to achieve recovery with probability tending to 1, whereas on the opposite side of the parameter threshold, the probability of recovery tends to 0 no matter what algorithm is used.

An example of the assortative case for the stochastic block model.

For partial recovery, the appropriate scaling is to take $P_{ij} = \tilde{P}_{ij}/n$ for fixed $\tilde{P}$, resulting in graphs of constant average degree. In the case of two equal-sized communities, in the assortative planted partition model with probability matrix

$$P = \begin{pmatrix} \tilde{p}/n & \tilde{q}/n \\ \tilde{q}/n & \tilde{p}/n \end{pmatrix},$$

partial recovery is feasible[4] with probability $1 - o(1)$ whenever $(\tilde{p} - \tilde{q})^2 > 2(\tilde{p} + \tilde{q})$, whereas any estimator fails[3] partial recovery with probability $1 - o(1)$ whenever $(\tilde{p} - \tilde{q})^2 < 2(\tilde{p} + \tilde{q})$.

For exact recovery, the appropriate scaling is to take $P_{ij} = \tilde{P}_{ij} \log n/n$, resulting in graphs of logarithmic average degree. Here a similar threshold exists: for the assortative planted partition model with r equal-sized communities, the threshold lies at $\sqrt{\tilde{p}} - \sqrt{\tilde{q}} = \sqrt{r}$. In fact, the exact recovery threshold is known for the fully general stochastic block model.[5]

Algorithms for SBM

- Depending on whether aiming for detection, partial recovery or full recovery….various algorithms can be shown to perform well in the average case (in worst case, cost can be exponential or the algoirithm may actually fail)
  - Spectral clustering (see later)
  - Semidefinite programming (tends not to scale that well)
  - Belief propagation (simple – essentially the algorithm that is known perform well over a wider range of parameter choices of the (assumed) underlying model

Generalizations to Hypergraphs

- The Erdos- Renyi and SBM models can be generalized to hypergraphs (done relatively recently)….some details still subject of research

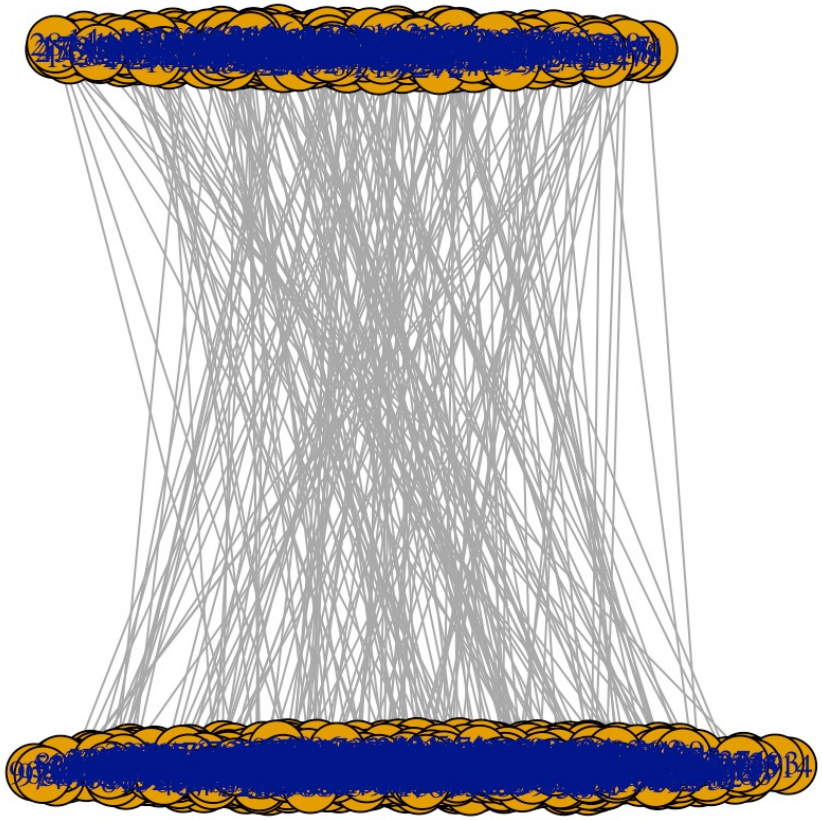
Erchuan Zhang, David Suter, Giang Truong, and Syed Zulqarnain Gilani. “Sparse Hypergraph Community Detection Thresholds in Stochastic Block Model”. In: Advances in Neural Information Processing Systems. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 34012–34023. url: https://proceedings.neurips.cc/paper_files/paper/2022/file/dbdea7859f1d2fc10f2c9e79b8f5ae54- Paper-Conference.pdf.

Example of SBM samples

```
install.packages("igraph")
library(igraph)
pm <- cbind( c(.1, .001), c(.001, .05) )
g <- sample_sbm(1000, pref.matrix=pm, block.sizes=c(300,700))
plot(g)
```

The above installs igraph (if you don't already have it…) and then calls a function to sample a SBM with two communities (one of size 300, other 700) and the first has internal connections with probability 0.1 and to external with probability 0.001……

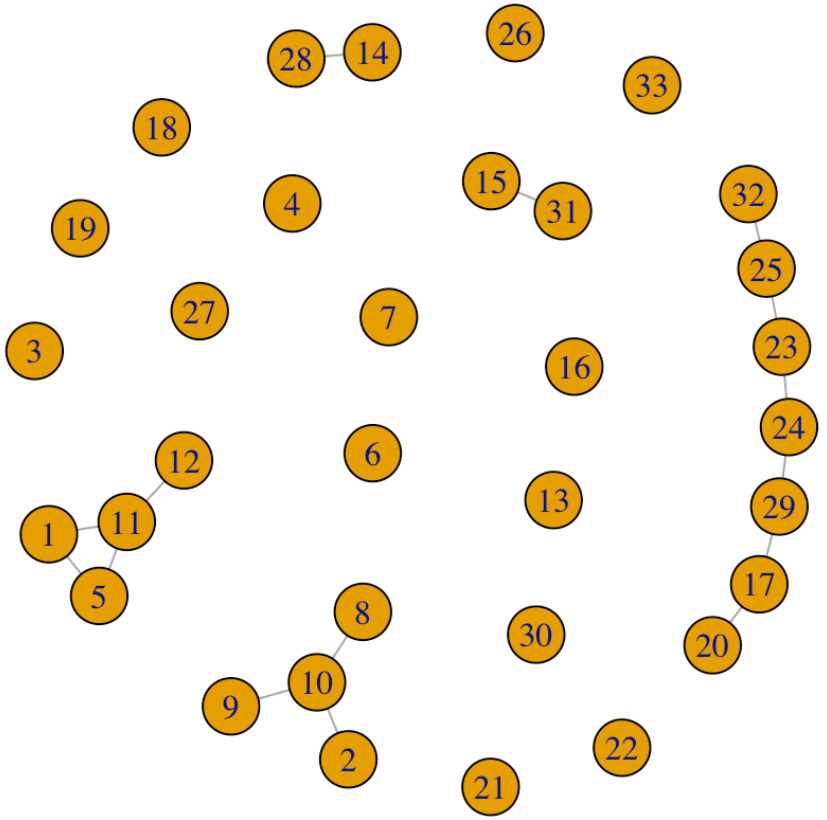
The display software has been reasonably clever in deciding how to layout this graph – allowing us to immediately see the communities! (albeit crowded).
But maybe we should choose a smaller example…



```
g <- sample_sbm(33, pref.matrix=pm, block.sizes=c(13,20))
plot(g)
```

Hmmmm…now the connection probabilities are so weak for the number of vertices that most vertices are isolated and the is no longer clearly 2 communities.

Statistical models often need large numbers to see what you want!

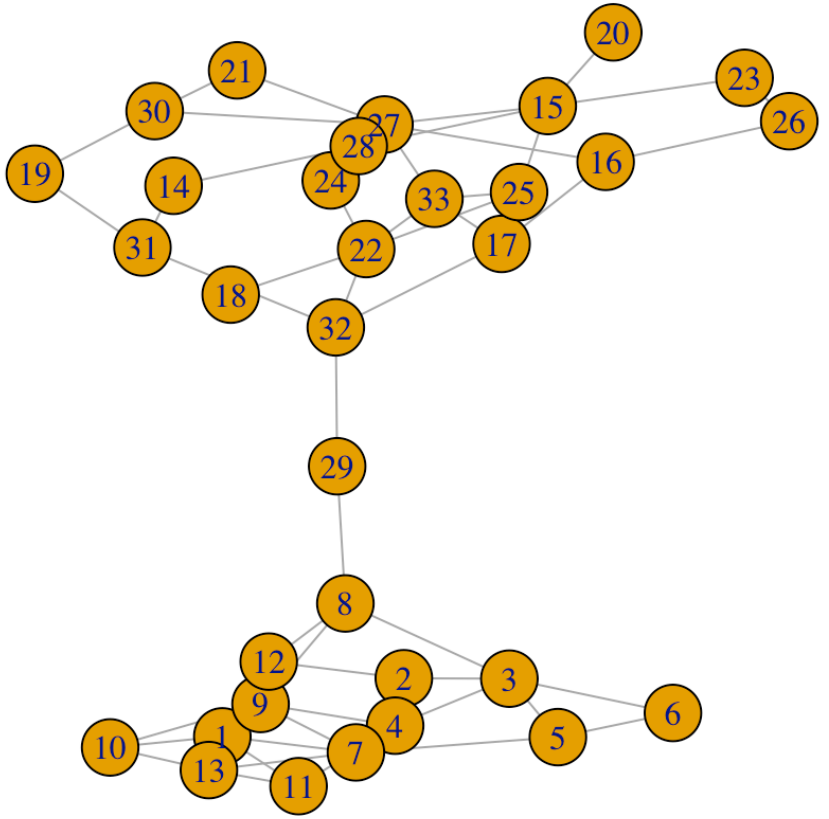


```
pm <- cbind( c(.3, .003), c(.003, .15) )
g <- sample_sbm(33, pref.matrix=pm, block.sizes=c(13,20))
plot(g)
```

OK now it looks like two communities….but are they the right communities – right vertices in the right cluster?

What about vertex 29? It looks to be between the two communities but is that simply an artefact of the drawing software or somehow statistically justified?

What about vertex 20?



Statistical/Machine Learning analysis of SBM

- Computer scientists, mathematicians, physicists,statisticians,….have devoted a lot of time to studying when (and how good) one can, given a sample from an SBM, recover the communities – AND how costly need the computations be…
- It is an interesting and still emerging picture!