Mining Complex Networks Notes

Ian Chen

May 21, 2026

Part I

Main Material

These notes very briefly summarize[1] Chapters $2, 3, 5, 6$ from Kamiński, Prałat, and François Théberge (2021), focusing on data analysis of simple graphs. As much of the material is review, these notes mostly provide an overview and the reader should see the original text for the specific details.

1 Random Graphs

The authors motivate the study of random graphs for the following applications:

- As a set of conceptual and mathematical examples, with well understood properties.
- As models for empirical (real, complex) networks.
- As procedures to create synthetic networks, often for testing and benchmarking algorithms.

1.1 ER graphs

The Erdös-Renyi graph $\mathcal{G}(n,p)$ has vertex set $V = [\![n]\!]$, with edges sampled iid from the support $\binom{V}{2}$ according to $\mathrm{Ber}(p)$. It is closely related to the Erdös process:

1. $\mathcal{G}_0$ has no edges
2. $\mathcal{G}_{t+1}$ adds a single edge to $\mathcal{G}_t$, uniformly among all non-present edges

where $\mathcal{G}(n,p) = \mathcal{G}_t$, where $t \sim \mathrm{Binom}(\binom{n}{2}, p)$, and hence it is often also called the Binomial graph. The asymptotic behavior of ER graphs are incredibly well-understood, and we illustrate it with two examples, developed precisely in Section 5.1.

Components and Connectivity: Let $\langle k \rangle = (n-1)p$ denote the expected average degree of $\mathcal{G}(n,p)$. For $\langle k \rangle < 1$, then asymptotically almost surely (aas), $\mathcal{G}(n,p)$ "looks like" collections of small trees and uni-cycle components. For $\langle k \rangle > 1$, then aas $\mathcal{G}(n,p)$ "looks like" a single giant component of size $\Omega(n)$. Indeed, the intuition for this phase transition is clear: it is the stability threshold of exponential growth!

If we write $p(n) = \frac{\ln n + c}{n}$, then the expected number of isolated nodes is aas e^{-c}. This is the main barrier to connectivity, as if $p = \omega(\frac{\ln n}{n})$, then aas $\mathcal{G}(n,p)$ is connected.

[1] Scattered throughout the notes are additional material that I found helpful.

Motifs: Let $d(G) = \frac{1}{2}\langle k \rangle = \frac{|E(G)|}{|V(G)|}$ be the density of a graph, and consider $m(G) = \max_{H \subset G} d(H)$ the maximum subgraph density. Then, G will appear as a subgraph aas if and only if the densest subgraph will too, and moreover is 0 or 1 based on the thresholded probability $p^* = n^{-1/m(G)}$. Indeed, the intuition is clear for why the densest subgraph H^* is the hardest to reproduce. With $O(n^{V(H^*)})$ possible vertex sets where H^* can appear, with $O(p^{E(H^*)})$ probability of this graph appearing, then $p^* = n^{-V(H^*)/E(H^*)}$ is indeed the transition point.

1.2 Power-Law

Significant empirical evidence supports that many properties of real-world networks exhibit power-law degree distributions, i.e. according to Zipf distribution with support over $\mathbb{Z}^+$,

$$D \sim f(d) \propto d^{-\gamma} \qquad \mathbf{E}(D^k) \approx \frac{\gamma - 1}{\gamma - 1 - k}, \gamma > (k+1) \tag{1}$$

The ER graph is a poor model for this. Indeed, the degree of a vertex follows $\mathrm{Binom}(n-1, p)$, and thus as $n \to \infty$, then this degree follows $\mathrm{Poisson}(\langle k \rangle)$. In particular, Poisson has an exponential tail, in contrast to a power-law tail.

The Barabási–Albert (BA) preferential attachment process produces power-law graphs with $\gamma = 3$ (Section 5.2):

1. $\mathcal{B}_1$ has a single vertex
2. $\mathcal{B}_{t+1}$ adds a single vertex to $\mathcal{B}_t$, adding m edges sampled with probability proportional to $\deg_{\mathcal{B}_t}(v)$.

Indeed, if we use a *continuum* approximation to the degree distribution, so that for vertex v_i,

$$\frac{\partial \deg(v_i)}{\partial t} = m\frac{\deg(v_i)}{2mt} \implies \frac{\mathrm{d}\deg(v_i)}{\deg(v_i)} = \frac{\mathrm{d}t}{2t}$$

and solving it, with the initial condition that $t_0/t \sim U(0,1)$, and $\deg(v_i)\big|_{t_0} = m$, then

$$\deg(v_i) = m\sqrt{\frac{t}{t_0}}, \qquad \frac{\partial}{\partial k}\mathbf{Pr}(\deg(v_i) \le k) = \frac{\partial}{\partial k}\mathbf{Pr}\left(\frac{t_0}{t} \ge \frac{m^2}{k^2}\right) = \frac{\partial}{\partial k}\left(1 - \frac{m^2}{k^2}\right) = 2m^2k^{-3}$$

1.3 Chung-Lu and Configuration

Another approach for better realism in our network models is to specifically enforce the degree distribution. For a given degree sequence $\boldsymbol{d}$[2], we consider the uniform distribution over all graphs with $\boldsymbol{d}$ as its sequence.

A more theoretically pleasing model is to enforce degrees only in expectation, i.e. attaching edges independently:

$$p_{ij} = \begin{cases} \frac{d_i d_j}{w} & i \neq j \\ \frac{d_i^2}{2w} & i = j \end{cases} \qquad w = \mathbf{1}^T \boldsymbol{d} \tag{2}$$

where we allow self-loops (which contribute 2 to the degree).

These models nicely allow us to generate many other graphs by correctly specifying the degree distribution:

- The power law graphs, where the coefficient is typically fit to a training set of data
- Random d-regular graphs, which is aas connected for $d \ge 3$, aas disconnected for $d < 3$

and also serve nicely as a *null model* to normalize several measures discussed later (Sections **??**).

[2]Not all degree sequences are realizable. The Havel-Hakimi theorem is illustrative, asserting handshaking and pigeonhole are sufficient and necessary. In practice, the Erdös-Gallai formula is efficient.

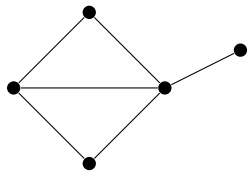
1.4 Quick checks

Checks

1. For what p would $\mathcal{G}(n,p)$ be the uniform distribution over all graphs on $V = [\![n]\!]$?
2. Let X be the number of occurrences of a subgraph H in $\mathcal{G}(n,p)$, easy to calculate using linearity of expectation. If $\mathbf{E}(X) \to \infty$, does this imply H is a motif aas?
3. Approximate the Zipf distribution over $\mathbb{R}^{\geq 1}$ by $f(d) \propto d^{-\gamma}$. Verify the approximate moments in Equation 1.
4. Verify the expected degree in the Chung-Lu model (Equation 2).
5. How might one generate a graph according to the configuration model?

Solutions

1. $p = 1/2$, where indeed each graph has probability 2^{-n} of being generated
2. No. Even if the probability tends to 0, the number of occurrences conditioned on existence may tend to infinity. Consider H to the right, on $\mathcal{G}(n, p = n^{-9/11})$, and X number of occurrences in $\mathcal{G}(n,p)$. The maximum subgraph density is $5/4$, and $p = n^{-9/11}$ is below threshold $n^{-4/5}$, so $\mathbf{Pr}(X = 0) \xrightarrow{P} 1$. However, we may find $\mathbf{E}(X)$ explicitly:



$$\mathbf{E}(X) = \frac{1}{\mathbf{aut}(H)} \sum_{v_1,\ldots,v_5 \in V} 1_{H \subset G[v_1,\ldots,v_5]} = \binom{n}{5} \frac{5!}{2} p^6 = \Theta(n^{1/11}) \to \infty$$

3. First, we find the normalizing factor, $f(d) = cd^{-\gamma}$, with $\gamma > 1$, and moments $\mathbf{E}(D^k)$ for $\gamma > k+1$,

$$\int_1^\infty cy^{-\gamma}\,\mathrm{d}y = \frac{c}{\gamma - 1} = 1 \implies c = \gamma - 1 \qquad \mathbf{E}(D^k) = (\gamma-1)\int_1^\infty y^k y^{-\gamma}\,\mathrm{d}y = (\gamma-1)\frac{1}{\gamma-k-1}$$

4. Apply linearity of expectation:

$$\mathbf{E}(\deg(v_i)) = 2p_{ii} + \sum_{v_i \neq v_j \in V} p_{ij} = d_i \left(2\frac{d_i}{2w} + \sum_{v_i \neq v_j} \frac{d_j}{w} \right) = d_i \frac{\sum_j d_j}{w} = d_i$$

5. A simple algorithm is to create *stubs* for each edge. Each vertex gets d_i source stubs, and d_i target stubs, and then a simple permutation (with no fixed points) suffices to match them together. It is well-known that $1/e$ proportion of such permutations are valid de-rangements, so rejection-sampling is efficient.

2 Centrality Measures

This section addresses the question of detecting influential *nodes* in a network. We assign an influence score to each node, and then a typical pipeline proceeds by ranking them. Indeed, as several methods are expensive on very large graphs, filtering by only collecting nodes in the k-core (i.e. maximal subgraph with min-degree at least k) is often performed ($k = 2$ usually). It is also useful to describe the distribution of centrality. A histogram or a survival curve is good, and the centralization $\max c(v) - \mathbf{E}(c(v))$ is also used.

For simplicity, the measures presented here are not normalized (cf. exercises). We note that it is desirable for measures to have unit L1-norm, as well as between $[0,1]$ to be interpretable across networks.

Our first set of measures are based on the adjacency matrix A, or potential normalized $\hat{A}$ with (column) sum 1, where $A_{ij} = 1\{i \to j \in E\}$ and are enumerated in Table 1. These measure influence primarily via their ego-net.

Measure	Formula	Notes
Degree †	$A\mathbf{1}_n$	BA is based on degree centrality.
Eigenvector †	$(I - A)^{-1}\mathbf{0}_n$	Perron eigenvector is unique.
Authority	$(I - \alpha\beta AA^T)^{-1}\mathbf{0}_n$	Center of incoming influence.
Hub	$(I - \alpha\beta A^T A)^{-1}\mathbf{0}_n$	Point to authorities.
Katz	$(I - \alpha A^T)^{-1}\mathbf{1}_n$	Based on damped random walk.
PageRank	$(I - \alpha\hat{A}^T)^{-1}\mathbf{1}_n$	Damped walk in stochastic matrix.

Table 1: **Adjacency matrix based centrality measures**. See text for additional details. † indicates undirected graphs.

Of course, it is straightforward to generalize degree and eigenvector centrality, although their interpretation may be less clear. Authority and hub centrality are solutions to the following mutual recurrence $\boldsymbol{x} = \alpha A^T \boldsymbol{y}$, and $\boldsymbol{y} = \beta A\boldsymbol{x}$, and are not necessarily unique, but must correspond to the (same) principal eigenvalue. PageRank and Katz centrality measure the occupancy of a node according to a random walk, e.g. $\mathbf{1}_n^T \sum_{k=1}^{\infty} (\alpha A)^k = (I - \alpha A^T)^{-1}\mathbf{1}$. Iterative numerical methods (related to power iteration) are useful here, especially for sparse matrices (e.g. Lanczos).

Our next set of centrality measures are based instead on paths and distances (Table 2). Let $\Pi(i, j, k)$ be the set of shortest paths between v_i and v_j containing v_k, $\Pi(i, j) = \bigcup_k \ell(i, j, k)$, and $\ell(\cdot) = |\Pi(\cdot)|$.

Measure	Formula	Notes
Closeness	$\frac{1}{\sum_v d(u,v)}$	$d(u, u) = 0$ and n if unreachable
Harmonic	$\sum_{v \neq u} \frac{1}{d(u,v)}$	More stable than closeness, typically
Eccentricity	$\frac{1}{\max_v d(u,v)}$	Related to diameter and radius
Betweenness	$\sum_{i,j \neq u} \frac{\ell(i,j,u)}{\ell(i,j)}$	Influential nodes are congested.
Efficiency	$-\sum_{i,j \neq u} \frac{1}{d_{G-u}(i,j)}$	Delta influence in the book.

Table 2: **Path based centrality measures**. See text for additional details.

Each of the measures considered here straightforward to extend to groups of nodes, e.g. *communities*. They can help evaluate the overall organization of the network, as well as rank relative importance of various subsets of nodes.

2.1 Quick checks

Checks

1. Normalize each of the centrality measures between $[0, 1]$.
2. How might one describe the correlation between centrality measures? What pairs have high or low correlations?
3. What is the computational cost for each of the centrality measures?

Solutions

1. As eigenvector based measures are only defined up to scale, normalizing just requires picking a unit norm eigenvector. Katz is already normalized and pagerank requires a $(1 - \alpha)/n$ factor. Table 3 enumerates the normalization for the path based measures.

Closeness	Harmonic	Eccentricity	Betweenness	Efficiency
$\frac{n-1}{\sum_v d(u,v)}$	$\sum_{v \neq u} \frac{1}{(n-1)d(u,v)}$	$\frac{1}{\max_v d(u,v)}$	$\sum_{i,j \neq u} \frac{(n-1)(n-2)\ell(i,j,u)}{\ell(i,j)}$	$1 - \frac{\sum_{i,j \neq u} d_{G-u}(i,j)^{-1}}{\sum_{i,j} d_G(i,j)^{-1}}$

Table 3: **Normalized path based centrality measures**.

2. We plot the pairwise Kendall τ correlation between centrality measures on the restricted Airport dataset[3]. This correlation measures the number of concordant pairs minus discordant pairs in the ranking, normalized between 0 and 1 by $\binom{n}{2}$.

   The graph is disconnected, leading to 0 variance of the eccentricity scoring and an undefined correlation between eccentricity and other measures. Overall, we see each of the measures are correlated, with the strongest correlation being between Katz and degree, as well as harmonic and closeness centralities. Moreover, the eigenvector centrality is similar to hub and authority.

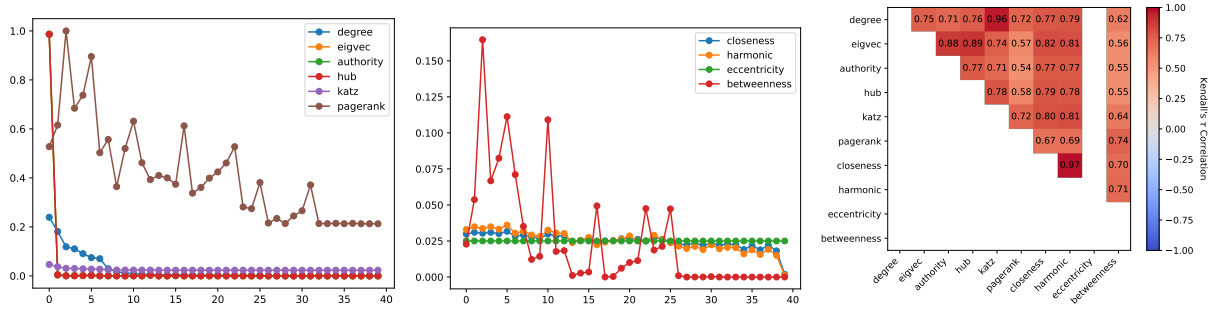


Figure 1: **Centrality on the California Airports dataset**. We show the distance and adjacency based centrality on a subset of the Airports dataset, restricted to within California flights. Adjacency based measures are on the weighted (directed) graph by passengers, with degree and eigvec on $A + A^T$ symmetric graph. Path based are on the unweighted undirected graph. Figures (a) and (b) have nodes ordered by degree centrality, and are normalized to unit L1-norm. Finally, we show the pairwise Kendall τ correlation in (c).

3. The matrix based are cheap (often Lanczos iteration converges in constant number of steps), taking time proportional to $O(m)$. The distance based ones are more expensive, with APSP taking $O(n^2 \log n + nm)$ time with Johnson's algorithm. k-core pruning is useful, as well as probabilistic estimation (Monte Carlo) is good.

3 Community Detection

Community detection, as an unsupervised task, has many formulations. Finding a partition of our vertex set, with certain requirements, we can discuss a node's role, e.g. if we consider degree, then

1. A node's participation is given by $p(v) = 1 - \sum_i \left(\frac{|N(v) | \cap C_i}{|N(v)|} \right)^2$
2. A node's within-module z-score is given by $z(v) = \left(\deg^{\text{int}}(v) - \mu(v) \right) \sigma(v)^{-1}$.
3. The anomaly score is $\text{cd}(v) = \deg(v) / \deg^{\text{int}}(v)$.
4. The community association strength $\deg^{\text{int}}(v) / \deg(v) - \frac{\text{Vol}(C_i) - \deg(v)}{\text{Vol}(V)}$, normalized by Chung-Lu model.

Community detection formulations argue about desirable cluster qualities. Section 5.4 has a modern characterization, but for now, traditionally, for a cluster C, we may consider the following three characterizations:

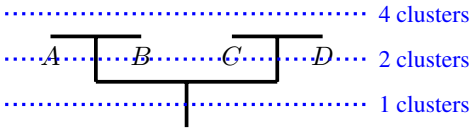
- Weak community, where $\sum_{v \in C} \deg^{\text{int}}(v) \geq \sum_{v \in C} \deg^{\text{ext}}(v)$
- Strong community, where $\deg^{\text{int}}(v) \geq \deg^{\text{ext}}(v) \forall v \in C$
- Community, where $p^{\text{int}}(v) = \frac{\deg^{\text{int}}(v)}{|C|} > \frac{\deg^{\text{ext}}(v)}{|V \setminus C|} = p^{\text{ext}}(v)$

[3] https://www.kaggle.com/datasets/flashgordon/usa-airport-dataset

As an unsupervised approach, there are several useful guiding questions to tailor your method of choice. How many clusters do we want, and what resolution are we studying? What are the desirable qualities in a clustering? Should clusters be overlapping?

3.1 Linkage

First, we sidestep the question of "how many clusters" by providing a view of many clusterings, for different scales. In agglomerative clustering, we start with a singleton partition and merge pairs of clusters. In divisive clustering, we start with a giant partition and split apart clusters. Both are nicely visualized in a dendrogram, see for example the one on the right.



The three typical strategies for agglomerative clustering, given a metric on our vertices, are:

1. Single-linkage, where cluster distance is the minimum distance between points
2. Complete-linkage, where cluster distance is the maximum distance between points
3. Average-linkage, where cluster distance is the mean distance between points

These tend to produce "stringy" and "round" clusters respectively, with average-linkage also usually round (in Euclidean space). For example, Rasvasz algorithm uses the following similarity function with average linkage:

$$s(v_i, v_j) = \frac{|N(v_i) \cap N(v_j)| + 1\{v_i v_j \in E\}}{\min(\deg(v_i), \deg(v_j)) + 1\{v_i v_j \notin E\}}$$

and Girvan-Newman is a divisive clustering algorithm, removing edges by an extension of betweenness for edges.

3.2 Modularity

Specializing our discussion of clustering to graphs, modularity is perhaps one of the most commonly used (despite its many flaws). It evaluates the quality of a clustering $\mathcal{C}$ by its improvement against the Chung-Lu null model,

$$q_G(\mathcal{C}) = \frac{1}{|E|} \sum_{C_i \in \mathcal{C}} \left[\underbrace{|E_G(C_i, C_i)|}_{\text{observed edges}} - \underbrace{\mathbf{E}_{G' \sim \mathcal{G}(\boldsymbol{d})} |E_{G'}(C_i, C_i)|}_{\text{expected edges}} \right] = \sum_{C_i \in \mathcal{C}} \left[\underbrace{\frac{|E_G(C_i)|}{|E|}}_{\text{edge contribution}} - \underbrace{\left(\frac{\text{Vol}(C_i)}{\text{Vol}(V)} \right)^2}_{\text{degree tax}} \right]$$

Indeed, $q_G(C) \in [-0.5, 1]$, with 0 being a singleton partition. It is NP-Hard to optimize, but there are many good heuristics. Namely, the Louvain algorithm iterates the following two stages:

1. Stage 1: local node movement; in order of a random permutation, move v to community maximizes $q_G(\mathcal{C}')$.
2. Stage 2: contract all communities into weighted (non-simple) graph, preserving degrees across components.

As it is an unstable algorithm (meaning its performance varies widely with the permutation), it is an ideal candidate as a weak-learner. ECG (Poulin and Théberge 2019) runs stage 1 of Louvain ℓ times to find $\mathcal{C}_\ell$, forming weights

$$w(u, v) = \lambda + (1 - \lambda) 1\{uv \in 2 - \text{core}\} \mathbf{E}_{\mathcal{C}_\ell}(uv \in \mathcal{C}_\ell)$$

with default values $\ell = 16$ and $\lambda = 0.05$. Finally, they run Louvain (repeating both stages) on the weighted graph to obtain their output clustering. Leiden is an improvement over Louvain, with a similar heuristic but enforcing well-connectedness of communities (Louvain may output disconnected communities).

The biggest issue against modularity is its inductive bias against small clusters. Namely, the Chung-Lu model is non-local, so that for two communities C_i and C_j, we expect $\text{Vol}(C_i)\text{Vol}(C_j)/\text{Vol}(G)$ edges. If both communities are

smaller than $\sqrt{2|E|}$, then with even one edge between them, it is beneficial to merge[4]. A simple patch is to add a resolution parameter γ, with the updated modularity function

$$q_G(\mathcal{C}, \gamma) = \sum_{C_i \in \mathcal{C}} \Big[\frac{|E_G(C_i)|}{|E|} - \gamma \Big(\frac{\text{Vol}(C_i)}{\text{Vol}(V)} \Big)^2 \Big]$$

where communities maintain separated as long as the null proportion of edges $\leq \gamma$ the empirical. Of course, there is now the issue of picking a γ, difficult to know *a priori*. This is equivalent to adding a resistance to every node.

3.3 Other Methods

Now, we briefly examine a few other clustering algorithms. The most scalable approach, label propagation, repeatedly moves a node to the partition matching the most of its neighbors. Repeating for randomly selected nodes, it may not necessarily converge.

Spectral clustering is k-means clustering via the RBF kernel. We seek k components minimizing the RatioCut:

$$\mathcal{L}(S_1, \ldots, S_k) = \sum_{k=1}^{K} \frac{1}{|S_k|} A(S_k, \bar{S}_k) = \sum_{k=1}^{K} \frac{1}{|S_k|} \sum_{x \in S_k, y \notin S_k} A_{xy}$$

We can formulate an approximation to the NP-Hard optimization via the graph Laplacian $L = KK^T$, for $K \in \mathbb{R}^{n \times m}$ the (weighted) incidence matrix. Let $H \in \mathbb{R}^{n \times k}$, $H_{ij} = \frac{1}{\sqrt{|S_j|}} 1\{i \in S_j\}$ be a (weighted) assignment, so $\mathcal{L}(S_1, \ldots, S_k) = \mathbf{Tr}(H^T L H)$. Indeed, H is orthogonal (normalized by construction). For any $f \in \mathbb{R}^n$,

$$\|K^T f\|_2^2 = \sum_e (\sum_v k_{v,e} f_v)^2 = \sum_{uv \in E} \Big(\sqrt{A_{uv}} (f_u - f_v) \Big)^2 = \frac{1}{2} \sum_{u,v \in V} A_{uv} (f_u - f_v)^2$$

With $|h_{ix} - h_{iy}| = \frac{1}{\sqrt{|S_i|}} |1\{x \in S_i\} - 1\{y \in S_i\}| = \frac{1}{\sqrt{|S_i|}} 1\{\text{exactly one in } S_i\}$, then letting $f = h$:

$$\mathbf{Tr}(H^T L H) = \|K^T H\|_F^2 = \sum_{i=1}^{k} \|K^T \boldsymbol{h}_i\|_2^2 = \sum_{i=1}^{k} \frac{1}{|S_i|} \sum_{(x,y) \in (S_i, \bar{S}_i)} A_{xy} = \mathcal{L}(S_1, \ldots, S_k)$$

Our (exact) problem seeks an orthonormal H with discrete entries minimizing this trace. Relaxing H with entries in $\mathbb{R}$, our problem is now $\min_{H^T H = I_k} \mathbf{Tr}(H^T L H)$, which by Courant-Fisher are the k smallest eigenvectors of L

Finally, InfoMap is another algorithm which captures information flow using random walks as a surrogate. With a two-level description scheme (encoding enters/exits, with a code for a community), they optimize

$$\underbrace{L(\mathcal{C})}_{\text{description length}} = \underbrace{q_{\curvearrowright}}_{\text{inter-community steps}} \underbrace{H(\mathcal{C})}_{\text{encoding cost}} + \sum_{i=1}^{\ell} \underbrace{p_{\circlearrowright}^{(i)}}_{\text{intra-community steps}} \underbrace{H(C^{(i)})}_{\text{encoding cost}}$$

3.4 Benchmarks

Evaluating a clustering algorithm for *accuracy* requires a set of benchmarks with a ground-truth clustering. Hence, there is a need to generate synthetic networks with a ground truth clustering[5]. Similarity functions on the contingency tables are useful, e.g. normalized/adjusted mutual information or adjusted rand index (concordant pairs).

[4]The ring of cliques is a canonical example, where small cliques with just one edge between them get merged.

[5]Ground-truth on empirical networks are sketchy. See, e.g. Peel, Larremore, and Clauset (2017).

The Stochastic Block Model extends ER with communities. With communities C_i of size n_i, we specify intra/inter-community edge probabilities p_{ii} and p_{ij}. In the planted partition model, $p_{ij} = p\delta_{ij} + q(1 - \delta_{ij})$.

The LFR model focuses specifically on recreating power-law graphs, with γ and τ the community and degree Zipf parameters, as well as μ the fraction of inter-community *neighbors*. The original proposed model is clunky to specify exactly, and not-scalable. Thus, the ABCD model improves on it, with γ, τ, ξ parameters, with ξ the fraction of inter-community edges. It is a union of ℓ communities (of sizes according to Zipf(γ)), with degrees by $(1 - \xi)$Zipf(τ), and a background graph with the inter-community edges. ABCD+o extension allows for non-clustered outliers.

3.5 Quick Checks

Checks

1. Are all weak communities strong communities, or vice versa? What about the third characterization?
2. How long would an algorithm that checks every partition take?

Solutions

1. Strong communities are also weak. Weak communities are not necessarily strong. There are communities (defined by the proportion of edges) that are weak communities, as well as ones that aren't.
2. The number of partitions of $[\![n]\!]$ is known as the Bell number, with

$$B_n = \sum_{k=0}^{n} \binom{n-1}{k} B_k = \left(\frac{n}{(e + o(1)) \ln n} \right)^n$$

and grows super-exponentially. Indeed, $B_{15} \approx 1.4 \times 10^9$ (OEIS sequence #A000110).

4 Node Embeddings

Another way of trying to obtain useful structural information is by embedding it into Euclidean space, either for reconstruction purposes, visualization, or to apply Machine learning algorithms (e.g. k-means, or deep learning). For example, if our embedding puts two nodes close together, we may expect a missing link to be a false negative. Formally, we seek $\mathcal{E} : V \to \mathbb{R}^k$, preserving some measure of proximity, e.g.

- Adjacency (1st order proximity), or k-th order proximity the cosine distance of k-neighbors
- Katz index $\sum_{i=1}^{\infty} (\alpha A)^i = (I/\alpha - A)^{-1} A$, and personalized PageRank $(1 - \alpha)(I - \alpha \hat{A}^T)^{-1}$
- Adamic-Adar, a measure of volume of common neighbors $s(u, v) = \sum_{w \in N(u) \cap N(v)} (\ln \deg(w))^{-1}$

4.1 Linear Embeddings

Let $E \in \mathbb{R}^{k \times n}$ be an embedding, mapping each node to a (column) vector of dimension $k \ll n$. We constrain E to be zero-mean and orthogonal. Local linear embedding argues E should be similar for neighbors, thus optimizes

$$\Phi_{\text{LLE}}(E) = \|(I - \hat{A})E^T\|_F = \mathbf{Tr}(E(I - \hat{A})^T(I - \hat{A})E^T)$$

Here, E^* are the 2nd to $(k + 1)$-th minimal eigenvectors (ignoring $\mathbf{1}_n$ of eigenvalue 0) of $(I - \hat{A})^T(I - \hat{A})$. Note that isometries preserve Φ. Very related is $\Phi_{\text{LEM}} = \mathbf{Tr}(ELE^T)$, for L Laplacian, or $\Phi(E)_{\text{HOPE}} = \|S - E_s^T E_t\|_F$, for source and target embeddings respectively, and S similarity.

4.2 Nonlinear Embeddings

Decoder-encoder networks D, E increase the expressivity of linear embeddings, e.g. optimizing

$$\Phi_{\text{SDNE}}(E) = \sum_i \|(a_i - D(E(a_i))B_i\|^2 + \alpha\, \mathbf{Tr}(ELE^T) + L_{\text{reg}}$$

for B_i (diagonal) vertex weights. Another set of embeddings are based on the *distribution hypothesis*. Recall that Word2Vec finds an embedding by treating each row as logits for appearing as a neighbor and optimizing MLE. Node2Vec creates "sentences" by running a random walk (DFS or BFS) and applying the Word2Vec procedure.

Finally, we mention an emerging field of structural embeddings, which encode data beyond adjacency. Indeed, in the ReFeX procedure, given a set of initial structural features $\boldsymbol{f}_i^{(0)}$ for v_i,

$$\underbrace{\boldsymbol{h}_i^{(k+1)} \leftarrow g(\left\{\boldsymbol{f}_j^{(k)} : v_j \in N(v_i)\right\},}_{\text{Aggregate}} \qquad \underbrace{\boldsymbol{f}_i^{(k+1)} \leftarrow \Pi_k \boldsymbol{h}_i^{(k+1)}}_{\text{Prune}}$$

In particular, Graph Convolution Network, or the more scalable GraphSAGE (extension via sampling), runs

$$e^{(s+1)} = \sigma(\tilde{D}^{-1/2}\tilde{A}\tilde{D}^{-1/2}e^{(s)}W^{(s)})$$

where σ activation and $W^{(s)}$ (trained) set of weights. This can be semi-supervised, classifying unlabeled nodes.

4.3 Quick Checks

Checks

1. Typically, our formulation of embedding supervised or unsupervised? Compare to spectral k-means.
2. What is the vocabulary space of Node2Vec? What is its size relative to English?
3. How might one evaluate an embedding algorithm?

Solutions

1. Our description of recovering a proximity score is unsupervised. Spectral k-means clusters on the node-embedding of LEM (when normalized).
2. The vocabulary space is each node, which can be millions or billions. Compare to English with thousands of words. DeepWalk addresses scalability with hierarchical softmax and Node2Vec implements negative sampling.
3. Kamiński, Prałat, and Théberge (2019) propose using the Geometric Chung-Lu model, parameterized by degree distribution and node embeddings. Good embeddings reconstruct edges of G, (compare Hosmer-Lemeshow).

Part II

Appendix

This part has content and derivations that are supplementary or illustrative to the content from book. Much of it is from external readings, but many take inspiration from the material.

5 Techniques for Random Graphs

5.1 ER-Graphs: connectivity and motifs

We consider the ER process $\mathcal{G}_t$, seeking to understanding the behavior of the components as $t \to \infty$. The main tool here will be Doob's Optional Stopping Theorem. Fix t, and $v \in \mathcal{G}_t$, and consider running BFS starting from v, letting A_k be the size of our queue, i.e. $A_0 = 1$. Then, the size of the components is $\tau_A \coloneqq \min_k(A_k = 0)$. We shall show[6]that 1. $\mathbf{E}(\tau_A) \leq \frac{1}{1-c}$ when $c < 1$, and 2. $\mathbf{Pr}(\tau_A \geq \epsilon n) > \delta$ for ϵ satisfying $1 - \epsilon > e^{-c\epsilon}$, $\delta > 0$.

Suppose $\langle k \rangle = c < 1$, i.e. $p = \frac{c}{n-1}$. The number of adjacent neighbors is binomial distributed, in particular,

$$A_{k+1} = A_k + \text{Binom}(n - k - A_k, p) - 1$$

As we shall see, t and A_t are both small, so it suffices to bound it by $B_{k+1} = A_{k+1} + \sum_{j=0}^{k} Y_k$ where $Y_k \sim \text{Binom}(k + A_k - 1, p)$. That is, we have the recurrence $B_{k+1} = B_k + \text{Binom}(n-1, p) - 1$. As $B_k \geq A_k$, then we bound $\tau_A \leq \tau_B \coloneqq \min_k(B_k = 0)$. Since $\mathbf{E}(\text{Binom}(n-1, p)) = (n-1)p = c$, then by optionally stopping,

$$1 = \mathbf{E}(B_0) = \mathbf{E}(B_{\tau_B} + \tau_B(1-c)) = 0 + (1-c)\,\mathbf{E}(\tau_B) \implies \mathbf{E}(\tau_B) = \frac{1}{1-c}$$

Thus, for $\langle k \rangle < 1$, then each component is of constant size in expectation. Otherwise, supposing $\langle k \rangle > 1$, then we can apply similar argument with $Z_k \sim \text{Binom}((n\epsilon - A_k - k), p)$, consider $C_{k+1} = A_{k+1} - \sum_{j=0}^{k} Z_k$ satisfying $C_{k+1} = C_k + \text{Binom}(n(1-\epsilon), p)$. By standard results in random walks, as long as $\mathbf{E}(Z_k) - 1 > 0$, then with positive probability $\tau_C \geq k$. Indeed, this holds for $\epsilon < 1 - c^{-1}$. Further, with complementary counting, $\text{Binom}(n - k - A_k, p) \sim \text{Binom}(n - k, (1-p)^k)$, so we can moreover bound ϵ satisfying $1 - \epsilon - e^{-c\epsilon} > 0$.

Next, we can also discuss the probability of containing a motif, in terms of the maximum subgraph density. Indeed, recall the Second-Moment Method, where for $X \geq 0$, we can bound the probability of X being zero as

$$\mathbf{Pr}(X = 0) \leq \mathbf{Pr}(|X - \mathbf{E}(x)| \geq \mathbf{E}(X)) \leq \frac{\mathbf{Var}(X)}{\mathbf{E}(X)^2}$$

Let X be the number occurrences of H, with h_v and h_e vertices and edges, appearing as a subgraph in $\mathcal{G}(n, p)$:

$$\mathbf{E}(X) = \frac{1}{\mathbf{aut}(H)} \sum_{v_1, \ldots, v_h \in V} 1\{H \subset \mathcal{G}[v_1, \ldots, v_h]\} \propto n^{h_v} p^{h_e}$$

Indeed, if $np^{-1/m(H)} \to 0$, then so does $\mathbf{E}(X)$ and $\mathbf{Pr}(X > 0)$. Conversely, if $np^{-1/m(H)} \to \infty$, then we can bound the variance by writing $X = \sum X_k$ where X_k indicates that a permutation of h_v vertices contains H:

$$\mathbf{Var}(X) = \sum_k \mathbf{Var}(X_k) + \cancel{\sum_{i \not\sim j} \mathbf{Cov}(X_i, X_j)} + \sum_{i \sim j, i \neq j} \mathbf{Cov}(X_i, X_j) = O(n^{2n - h_v} p^{2m - h_e})$$

with $i \sim j \iff X_i \not\perp\!\!\!\perp X_j$, i.e. they share ≥ 2 vertices. Thus, $\frac{\mathbf{Var}(X)}{\mathbf{E}(X)^2} \to 0$, $\mathbf{Pr}(X > 0) \geq 1 - \frac{\mathbf{Var}(X)}{\mathbf{E}(X)^2} \to 1$.

5.2 BA-Graphs: power-law distribution

We briefly mention a more precise derivation of the degree distribution of a BA graph. This is based on the *rate equation*, where if we consider $N(k, t)$ the number of nodes of degree k at a time t, with $p_k(t) = \frac{N(k,t)}{t}$, then

$$\mathbf{E}(\text{edges added to } N(k,t) \text{ per additional vertex}) = m\frac{k}{2mt}N(k,t) = \frac{k}{2}p_k(t)$$

[6]In fact, both of these hold asymptotically almost-surely.

Thus, we get the following difference equation

$$N(k,t+1) = N(k,t) + \underbrace{\frac{k-1}{2}p_{k-1}(t)}_{\text{promoting from degree } k-1} - \underbrace{\frac{k}{2}p_k(t)}_{\text{promoting to degree } k+1}$$

with initial condition $p_k = 0$ for $k < m$. For large t, then $p_k(t)$ is stationary, and so

$$N(k,t+1) - N(k,t) = p_k = \frac{-1}{2}(kp_k - (k-1)p_{k-1})$$

Now, solving this recurrence with one's favorite method, we find that $p_k = \frac{2m(m+1)}{k(k+1)(k+2)}$, indeed very similar to our continuum approximation. In fact, approximating $kp_k - (k-1)p_{k-1} = \frac{\mathrm{d}(kp_k)}{\mathrm{d}k}$ leads back to the standard continuum equation (after product rule).

5.3 Regular-Graphs: number of cycles

For R_n the uniform distribution of 2-regular graphs on n vertices, we show that the number of cycles Y_n is concentrated near $\frac{\ln n}{2}$. Consider the following procedure (resembling DFS) for generating a d-regular graph:

1. Start at a source stub, and pair it with a target.
2. Repeat step 1 from a source stub from the chosen paired vertex, or a uniform one when no remaining sources.

There are $2n$ stubs and thus n steps of the algorithm. Let p_k be the probability of step k closing a loop, so that $\mathbf{E}(Y_n) = \sum_{k=1}^{n} p_k$. Let s_k be the source vertex of the (unique) path at the k-th step. Finally,

$$p_k = \frac{\#\text{target stubs of } s_k}{\#\text{target stubs remaining}} = \frac{1}{2(n-(k-1))-1}$$

Thus we find our desired results:

$$\begin{aligned}\mathbf{E}(Y_n) &= \sum_{k=1}^{n} \frac{1}{2(n-k)+1} = \sum_{j=1}^{2n} \frac{1}{j} - \sum_{j=1}^{n} \frac{1}{2j} \\ &= (\ln 2n + \gamma) - \frac{\ln n + \gamma}{2} + o(1) = \frac{\ln n}{2} + \ln 2 + \frac{\gamma}{2} + o(1) \approx \frac{\ln n}{2} + 1.27\end{aligned}$$

and with Chernoff bounds, for each $X_k \in \{0,1\}$ independent, then for any $0 < \delta < 1$, then[7]

$$\mathbf{Pr}\Big(Y_n \geq (1+\sqrt{3}\delta)\,\mathbf{E}(Y_n)\Big) \leq \mathbf{exp}\left(-\delta^2\,\mathbf{E}(Y_n)\right) \leq n^{-\delta^2/2} \to 0$$

5.4 Clustering Unweighted Graphs

We paraphrase the axioms a clustering algorithm $\mathcal{A}$ should satisfy, based on Willson and Warnow (2024):

1. Richness: $\forall \mathcal{C}$ clustering of $[\![n]\!]$, $\exists E$ s.t. $\mathcal{A}(V,E) = \mathcal{C}$
2. Consistency[8]: E' removes inter-cluster/adds intra-cluster edges, $\mathcal{A}(V,E')$ refines $\mathcal{A}(V,E')$.
3. Connectivity: $\exists f : \mathbb{R}^+ \to \mathbb{R}^+$ unbounded and non-decreasing s.t. min-cut of all $C \in \mathcal{C}$ exceeds $f(|C|)$.
4. Fixed point: applying $\mathcal{A}$ to any induced subgraph by clusters would return same clusters.
5. Pair-of-cliques: $\exists n_0$ such that two cliques K_{n_0} connected by edge are distinct clusters.

[7] The specific form I used is given by Lemma 13.2.5 in `https://sarielhp.org/p/notes/16/chernoff/cheat_sheet.pdf`

[8] Standard consistency refines for strict equality of clusterings, but is unintuitive, e.g. a ring lattice.

References

Kamiński, Bogumił, Paweł Prałat, and François Théberge (Nov. 2021). *Mining complex networks*. Chapman and Hall/CRC. DOI: https://doi.org/10.1201/9781003218869.

Kamiński, Bogumił, Paweł Prałat, and François Théberge (Nov. 2019). "An unsupervised framework for comparing graph embeddings". en. In: *J. Complex Netw.* DOI: https://doi.org/10.1093/comnet/cnz043.

Peel, Leto, Daniel B Larremore, and Aaron Clauset (May 2017). "The ground truth about metadata and community detection in networks". en. In: *Sci. Adv.* 3.5, e1602548. DOI: https://doi.org/10.1126/sciadv.1602548.

Poulin, Valérie and François Théberge (Dec. 2019). "Ensemble clustering for graphs: comparisons and applications". en. In: *Appl. Netw. Sci.* 4.1. DOI: https://doi.org/10.1007/s41109-019-0162-z.

Willson, James and Tandy Warnow (Oct. 2024). "Axioms for clustering simple unweighted graphs: No impossibility result". en. In: *PLOS Complex Syst* 1.2, e0000011. DOI: https://doi.org/10.1371/journal.pcsy.0000011.