1. Lecture 2: Random Network Growth Models

1.1 Biological Networks

- Many biological systems can be idealized as graphs: signaling pathways, metabolic pathways.
- High-throughput pairwise data is also often available, and this data can be encoded into graphs.
- Graphs summarizing several sources of information are also frequently built, where $V$ are, say, genes, and $E$ are edges (or multiedges) where the weights on edge $(u, v)$ is a function of how functionally related genes $u$ and $v$ are.

Common types of biological graphs:

1. Protein-protein interaction graphs ($V =$ proteins, $E =$ physical interactions)
2. Transcription networks: directed graph ($V =$ genes/proteins, $(u, v) \in E$ indicates that $u$ regulates the expression (on/off state) of gene $v$)
3. Synthetic Sick-or-lethal network (SSL) [newer versions: EMAP]: $V =$ genes $(u, v) \in E$ indicates that the fitness of an organism when knocking out both $u$ and $v$ is much different than you would expect given the fitness when knocking out $u$ or $v$ alone.
4. Metabolic networks: $V =$ metabolites (small molecules); $E =$ chemical reactions; sometimes $E$ is a “directed hypergraph” to handle the case where $> 1$ reactant yields $> 1$ resultants.
5. Social networks: Facebook, twitter, etc.

1.2 Models of random graph growth

- Networks in many contexts are changing over time: Internet grows in popularity; people join facebook; simple organisms become more complex.
- Last lecture: saw some models of population growth that mostly ignored structure / interactions between members of the population, or they had everyone of the same “species” have the same interactions. Here: entire focus is on the evolution of that structure.
- Aside: Network alignment: the computational problem of finding “homologous” (aka “corresponding”) pathways shared between $> 1$ network. Lots of algorithmic work on this problem: PathBLAST was an early attempt. QNet, Isorank, and many others. [Subgraph and graph isomorphism problems.]
- Study random models of network growth to help answer many of the questions we brought up last time, e.g.:

1. Does the simple mechanism embedded in the model lead to the salient features of real networks?
2. Could observed features have arisen simply by chance [use the random growth model as a null model]
3. Predict the future growth of the network. [How will the diameter of the network change over time? Will dense communities develop or dissolve over time? How will the growth of the network affect the spread of disease or rumors?]
4. Infer unobserved, past states of the networks.
5. Also: algorithm testing ("will my algorithm work well on graphs of this type?"; “average case” complexity)

6. Graph sampling: if you network is too large (e.g. internet), you sample from it. If you have a good model, you can test whether your sample obeys the properties implied by the model (e.g. shrinking diameter).

7. Abnormality detection: if low probability events start happening a lot, something might be wrong.

1.3 Erdös-Renyi (ER) Random Graphs

- Very early type of random graph.
- Not a growth model
- Parameters: $n =$ number of nodes; $p =$ probability there is an edge between each pair.
- Expected degree is $(n - 1)p$
- Every vertex “looks the same” in expectation.

(Left): ER graph with 20 vertices, $p = 0.1$; (Right) ER graph with 200 vertices, $p = 0.1$.

1.4 Preferential Attachment (PA)

- Studied in context of social networks for a long time
- Re-popularized in late 1990s by Albert & Barabasi.
- Parameters: none
- At time step $t$: node $u_t$ is added; edge $\{v, u_t\}$ added independently with prob. $\text{deg}(v)/\sum_w \text{deg}(w)$.
- Rich-get-richer: popular websites (or people) are more likely to accumulate additional links. [Google or Yahoo! have a lot of links, and new websites are more likely to link to them than to your homepage.]
- Degree distribution is one feature that can be used to distinguish between graphs derived from different models:
  1. ER degree distribution is exponentially distributed around the expected degree.
  2. $\Pr(\text{deg}(u) = k) = \binom{n-1}{k} p^k (1-p)^{n-k-1}$. $\mathbb{E}\text{deg}(u) = (n - 1)p$: 

...
3. PA degree distributions has “scale-free” distribution: lots of low degree nodes; fewer high-degree nodes, but more than you would expect by chance.

- **Variant:** Linear preferential attachment.
  - **Parameters:** \(k\) = initial degree of entering node; \(a\) = smoothing parameter.
  - **Dynamics:** Node enters, \(k\) edges are added at random according to the distribution given by \(\text{deg}(u_j) + a\) (after normalization).
  - When \(a = 0\) → similar to preferential attachment (differences remain: growing & \(k\) fixed)
  - When \(a\) is very large, the effect of “rich-get-richer” is reduced.

(left) PA graph with 20 nodes and every new node adding 3 connections to existing nodes. (right) PA graph with 200 nodes and every new node adding 6 connections to existing nodes.

### 1.5 Random Growing

- Close to ER networks but models growth.
- **Parameters:** \(k\) = initial number of neighbors of incoming node;
- **Dynamics:** When node \(u_t\) enters, \(k\) random edges are added, chosen uniformly. These edges need to be adjacent to \(u_t\).
- Early nodes have more chances to be connected.
- Duncan S. Callaway, John E. Hopcroft, Jon M. Kleinberg, M. E. J. Newman, and Steven H. Strogatz, Phys Rev. E is one such example.

### 1.6 Aging Vertices

- Start with a clique on \(m\) nodes.
- New node enters and links to all active nodes and the new node becomes active.
- Choose some active node \(j\) according to
  \[ \Pr(j) = \gamma/(a + \text{deg}(j)) \]  
  where \(\gamma\) is a normalization factor to turn this into a probability distribution, and \(a\) is some uniforming constant.
- Idea: there are some \(m\) nodes that are “current” or highly relevant; eventually (inversely proportionally to their popularity) nodes become inactive or are “forgotten” about.
1.7 Watts-Strogatz Small World

- **Parameters:** \( k \) = the dimension of the ring lattice; rewiring probability \( q \).
- Start with a ring lattice: \( G = (V, E) \) where \( V = \{1, \ldots, n\} \) and \( E = \{u, v\} \) when \( |u - v| < \lfloor k/2 \rfloor \)
- For each edge \( \{i, j\} \), with probability \( q \) delete it and add edge \( \{i, v\} \) for a random vertex \( v \).
- Idea: most connections between people are local (where local is governed by parameter \( k \)), but there are a few long range links.
- Models the idea of “small world” (small diameter).
- “Require”: \( n >> k >> \log(n) >> 1 \); the constraint \( k >> \log(n) \) implies that the resulting graph is likely to be connected.
- “Neighborhood”-ness (clustering coefficient) drops much slower than average distance between nodes as \( q \) increases: each rewired edge gives a long-range shortcut that drops the average distance a lot, but doesn’t impact the clustering coefficient much.

(Left) Watts-Strogatz random graph with 20 nodes, \( k = 3 \), \( q = 0.1 \); (right) WS random graph with 200 nodes, \( k = 6 \), \( q = 0.1 \).

1.8 Duplication

- “Duplication with mutation” (or imperfect replication) is a process by which many networks grow:
  1. Genes are duplicated when sections of genomes are duplicated
  2. When I join facebook, I might copy the friends of one of my friends — then I look like a duplicate of my friend.
  3. When I write a paper, I might cite the same collection of papers cited in a previous paper.
- Simplest model: choose node \( v \) and random, connect new node \( u_i \) to the neighbors of \( v \) at random.
- Duplication with Mutation:
  1. Capturing imperfect duplication: remove each new edge with probability \( q_{\text{del}} \).
  2. Capturing changes unrelated to the duplication: for every node \( w \) that is not a neighbor of \( u_i \) add edge \( (u_i, w) \) with probability \( q_{\text{new}}/t \). (The probability of adding new edges decays over time).
- Capturing the preservation of function: **Parameters:** probabilities \( q_{\text{con}} \), \( q_{\text{mod}} \).
  1. Choose a random anchor node \( v \) and create its duplicate, \( u_i \), by connecting \( u_i \) to all of \( v \)’s neighbors.
  2. For each neighbor \( x \) of \( u_i \), decide to modify the edge or its compliment with probability \( q_{\text{mod}} \). If the edge is to be modified, delete either edge \( (v, x) \) or \( (u_i, x) \) by the flip of a fair coin.
  3. Add edge \( (u, v) \) with probability \( q_{\text{con}} \).
4. **Idea:** Each edge is performing some important function. After a node is duplicated, the duplicate can take over the function (or both can continue to do it), but you can’t have both a node and its duplicate decide to abandon the function.

1.9 **Kronecker**

- Tensor (Kronecker) product $A \otimes B$ between two matrices $A = (a_{ij})$ and $B = (b_{ij})$ is a matrix where each entry $a_{ij}$ in $A$ is replaced by a matrix $a_{ij}B$.
- Very “fractal” construction.
- **Parameters:** starting graph $H$; number of iterations $k$.

![Kronecker Graph Example](image.png)

After 4 iterations starting from a path of length 3 (with self loops)

- Purpose of Kronecker graphs: generate large graphs with many properties of social networks.
- At every step, you can think of each node being *expanded* into $|H|$ nodes.
- If $H$ has self-loops on every edge, each step expands every node into a copy of $H$.
- Stochastic Kronecker Graphs: seed matrix is weighted in $[0, 1]$; take tensor products as usual to obtain a real-valued matrix; and choose each edge independently with the given probability.
- A plausible interpretation of Stochastic Kronecker Graphs: Let $G_k = (V_k, E_k)$ be the final graph after $k$ iterations.
  - We can assign every node $u \in V_k$ an *address* of length $k$ that specifies which nodes of $H$ were expanded to create it: $\text{addr}(u) = (v_{u}^1, \ldots, v_{u}^k)$. (Think of the nodes of $G_k$ being the leaves of a forest, which each node ultimately descending from one of the original nodes of $H$).
  - Let $H(a, b)$ be the weight in the seed graph between nodes $a$ and $b$. Then the $(u, v)$th entry of $G_k$ is:
    $$\prod_{i=1}^{k} H(\text{addr}(u)_i, \text{addr}(v)_i)$$
    because to get to the edge $(u, v)$ in $G_k$ you go to block $(\text{addr}(u)_1, \text{addr}(u)_1)$, and then within that to block $(\text{addr}(u)_2, \text{addr}(u)_2)$, etc.
  - In other words, each node of $G_k$ is associated with an attribute vector of length $k$, where each attribute can take on values represented by the node set of $H$. $H(a, b)$ gives the similarity of attribute values $a, b$. Finally, the probability two nodes are connected in $G_k$ is the product of the similarities of their attributes.
1.10 Forest Fire

- Designed to simulate the growth of social networks. In particular:
  1. get denser over time [average degree increases]
  2. average distance decreases over time

- Intuition: a node enters, finds some node in the network, and explores that nodes’ neighborhood until he “gets tired” and links to everyone that he finds in the neighborhood.

- The model:
  1. Node $v$ enters the network, randomly selects an anchor node $u$, and links to it.
  2. Node $v$ randomly chooses $x$ neighbors of $u$ and links to them, where $x$ is an integer chosen from a geometric distribution with mean $p/(1 - p)$. These vertices are flagged as active vertices.
  3. Set $u$ to each active vertex and recursively apply step 2. $u$ becomes non-active. Stop when no active vertices remain.

- Alternative, simpler, related model: **Random walk with restart growth**: Parameter: $r$ the restart probability.
  1. Node enters, chooses an existing node $v$ uniformly at random.
  2. Node does a “random walk with restart”, starting from $v$. A random walk with restart is: at each step, with probability $(1 - r)$ the walker choses an outgoing edge from its current node; with probability $r$, the walker jumps back to the starting point.

- Note: FF has “rich-get-richer” property too: nodes with high degree are likely to be reached no matter where you start.

1.11 Qualitative Differences Between Models


- ER and Watts-Strogatz networks have a degree distribution that peaks at some point, and the chance a node has a large degree decays exponentially.

- Other networks have a “scale-free” degree distribution: $P(k) \approx k^{-\gamma}$, where $P(k)$ is the fraction of nodes that have degree $k$, and $\gamma$ is some constant.

- These two types of networks have robustness to different types of disruptions. Types of disruptions:
  1. “Attack”: remove highest degree nodes in order of decreasing degree
  2. “Failure”: remove random nodes

- Average distance: average shortest-path distance between two nodes. [Several papers — erroneously — call this the diameter]

- for ER graphs: attack and failure have the same effect on average distance

- for “SF” graphs (e.g. preferential attachment): attack increases the average distance at a much faster rate than random failure.

- Scale-free graphs such as the WWW, Internet, and metabolic networks are much more robust to random failures than to targeted attack of high-degree nodes.

- Either type of attack tends to increase the average distance in ER graphs, while random failure of nodes in SF graphs has very little effect on the average distance.
• Metabolic networks (in some reasonably natural formalization) exhibit the same property: knocking out high-degree nodes leads to a large increase in average distance. (H. Jeong, B. Tombor, R. Albert, Z. Oltvai, A.-L. Barabsí The large-scale organization of metabolic networks Nature 407, 651-655 (2000).)

1.12 Choosing a model that fits observed networks


• For every model $\mathcal{M}$, generate a large number of graphs $G_1^\mathcal{M}, G_2^\mathcal{M}, \ldots, G_m^\mathcal{M}$.

• Generate a vector of graph features for each generated graph $f(G_i^\mathcal{M})$. [Diameter, count of subgraphs, degree distribution, etc.]

• Consider this vector of features as a labeled training example $(f(G_i^\mathcal{M}), \mathcal{M})$.

• Train a classifier (e.g. decision tree, SVM, etc.) to label a new graph $G$, represented by its vector of features $f(G)$ with one of the models.

• When this experiment is performed on fruit fly protein-protein interaction network, DMC is chosen as the label.