Random graphs, social networks and the internet

Lecture 4

Mariana Olvera-Cravioto

UNC Chapel Hill

molvera@unc.edu

October 14th, 2021

Google

- What do you envision when you think about Google?

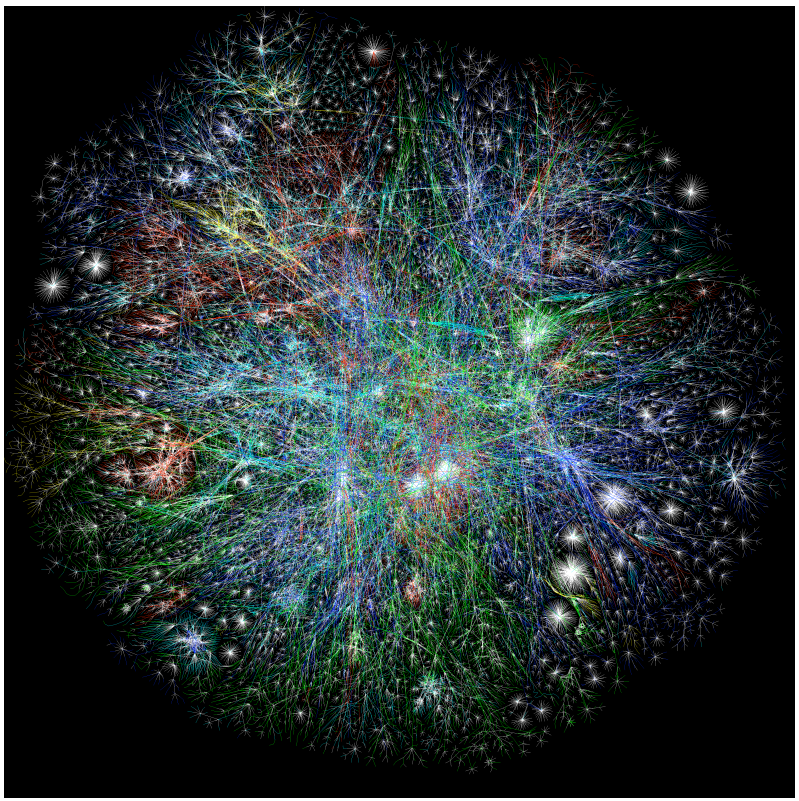
Google

- What do you envision when you think about Google?
- Google, the company we know today, started with a search engine, i.e., a computer program that was designed to help us find specific webpages in the World Wide Web.

The Internet

- The Internet is a giant network of computers around the world connected through "wires".
- Think of the Internet as a giant graph consisting of vertices and edges:

vertices = servers/computers
edges = a wired connection between them



The World Wide Web

- The WWW is a "virtual" network connecting webpages through links.
- It defines a **directed** graph where:

  vertices = webpages

  edges = directed links from one webpage to another

- The WWW was officially born in 1991 with the creation of the first *browser*, a software interface that allowed users to access many different types of files stored in many different computers.

The World Wide Web... cont.

- With the creation of the first *free* web browser, Mosaic, the WWW quickly started growing.
- Many other web browsers quickly followed: Internet Explorer, Netscape Navigator, Mozilla, Firefox, Safari, Google Chrome, etc.
- As the WWW grew, it became more and more important to be able to search for information quickly and efficiently.

Search Engines

- Search engines started being developed since the start of the WWW.
- Their goal was to find and organize the vast amount of information stored on the Internet.
- In the early ages of the WWW, many search engines were competing for our attention.

What happened?

- Most of the early search engines were based on *simple searches* within large databases:

  keyword $\longrightarrow$ list of webpages containing the keyword

- As the WWW grew larger, simple searches started returning too many results... ordering them became important.
- Google was the first search engine to attempt a **universal ranking** of all the webpages in the WWW which would determine in which order the search results should be displayed.
- Page and Brin, the creators of Google, proposed an algorithm, called **PageRank**, that would capture the **relevance** of each webpage into a single score, which could then be used to order the search results.

What happened?

- Most of the early search engines were based on *simple searches* within large databases:

  keyword $\longrightarrow$ list of webpages containing the keyword

- As the WWW grew larger, simple searches started returning too many results... ordering them became important.
- Google was the first search engine to attempt a **universal ranking** of all the webpages in the WWW which would determine in which order the search results should be displayed.
- Page and Brin, the creators of Google, proposed an algorithm, called **PageRank**, that would capture the **relevance** of each webpage into a single score, which could then be used to order the search results.

  Their idea worked!

Recap on graphs

- A graph consists of a set of vertices and edges that connect them.
- Graphs can be directed or undirected, as we already saw.
- The structure of a graph matters quite a bit when we consider questions like:
  - Is there a path of edges connecting any pair of vertices?
  - What is the length of the longest path connecting two vertices?
  - What is the typical distance between vertices?
  - What is the average number of (inbound/outbound) neighbors a vertex has?
- Not all graphs have the same structure!
- Most of the interesting *large* graphs you can think of are rather special...

Modeling complex networks

- Many real-world graphs are extraordinarily big, e.g., millions or billions of vertices.
- Most of them are fairly sparse, i.e., the ratio

$$\frac{\#\text{ edges}}{\#\text{ vertices}}$$

  is not too big.
- Many share two key properties:
  - **Small world:** the typical distance between vertices is small compared to the total number of vertices.
  - **Scale-free:** the proportion of vertices with k (inbound/outbound) neighbors decays as a power of k, e.g.,

$$\frac{\#\text{ vertices with } k \text{ neighbors}}{\text{total }\#\text{ vertices}} \approx Ck^{-\alpha}$$

Other properties of complex networks

- Many interesting graphs are disconnected, but may have a large subset of vertices that are connected.
- Some graphs have many "clusters" (groups of vertices that have more connections among themselves than with the rest of the graph).
- Some directed graphs exhibit high levels of correlation between the number of inbound neighbors and the number of outbound neighbors of a given vertex.

Other properties of complex networks

- Many interesting graphs are disconnected, but may have a large subset of vertices that are connected.
- Some graphs have many "clusters" (groups of vertices that have more connections among themselves than with the rest of the graph).
- Some directed graphs exhibit high levels of correlation between the number of inbound neighbors and the number of outbound neighbors of a given vertex.
- These properties influence how fast a message can spread through a network and/or how many vertices it can reach.
- **They also influence which vertices are more "central" to the network.**

Relevance and centrality

- Back to the topic of search engines....

Relevance and centrality

- Back to the topic of search engines....
- Intuitively, a vertex in a graph is **central** if many paths go through it.
- The idea behind Google's PageRank algorithm is that **relevant** webpages should be those that are **central** to the network.
- **Why?**

Relevance and centrality

- Back to the topic of search engines....
- Intuitively, a vertex in a graph is **central** if many paths go through it.
- The idea behind Google's PageRank algorithm is that **relevant** webpages should be those that are **central** to the network.
- **Why?** Links are created by people, and people will tend to create links to webpages that have relevant/interesting content.
- How does PageRank find "central" vertices?

The PageRank algorithm

- Let n denote the number of vertices in the WWW.
- The PageRank of webpage i, denoted r_i, is a number in $[0, 1]$ that measures its "centrality" within the network.
- r_i is a "universal" rank, i.e., it does not change from one search to another, and it has nothing to do with the content of webpage i.
- r_i depends only on the topology of the graph, i.e., on the structure determined by the edges connecting the vertices.
- **Relevance is contagious:** If a relevant webpage has a link pointing to another webpage, it makes it relevant too, but if it points to too many webpages this effect is reduced.

Computing the PageRank vector

- To compute the PageRank vector $\mathbf{r} = (r_1, \ldots, r_n)$ we solve the system of linear equations:

$$r_i = \frac{1-\alpha}{n} + \alpha \sum_{j \to i} \frac{r_j}{d_j^+},$$

where the sum is taken over all the inbound neighbors to webpage i, d_j^+ is the number of outbound neighbors of webpage j, and $\alpha \in (0, 1)$ is a constant known as the *damping factor*, usually $\alpha = 0.85$.

Computing the PageRank vector

- To compute the PageRank vector $\mathbf{r} = (r_1, \ldots, r_n)$ we solve the system of linear equations:

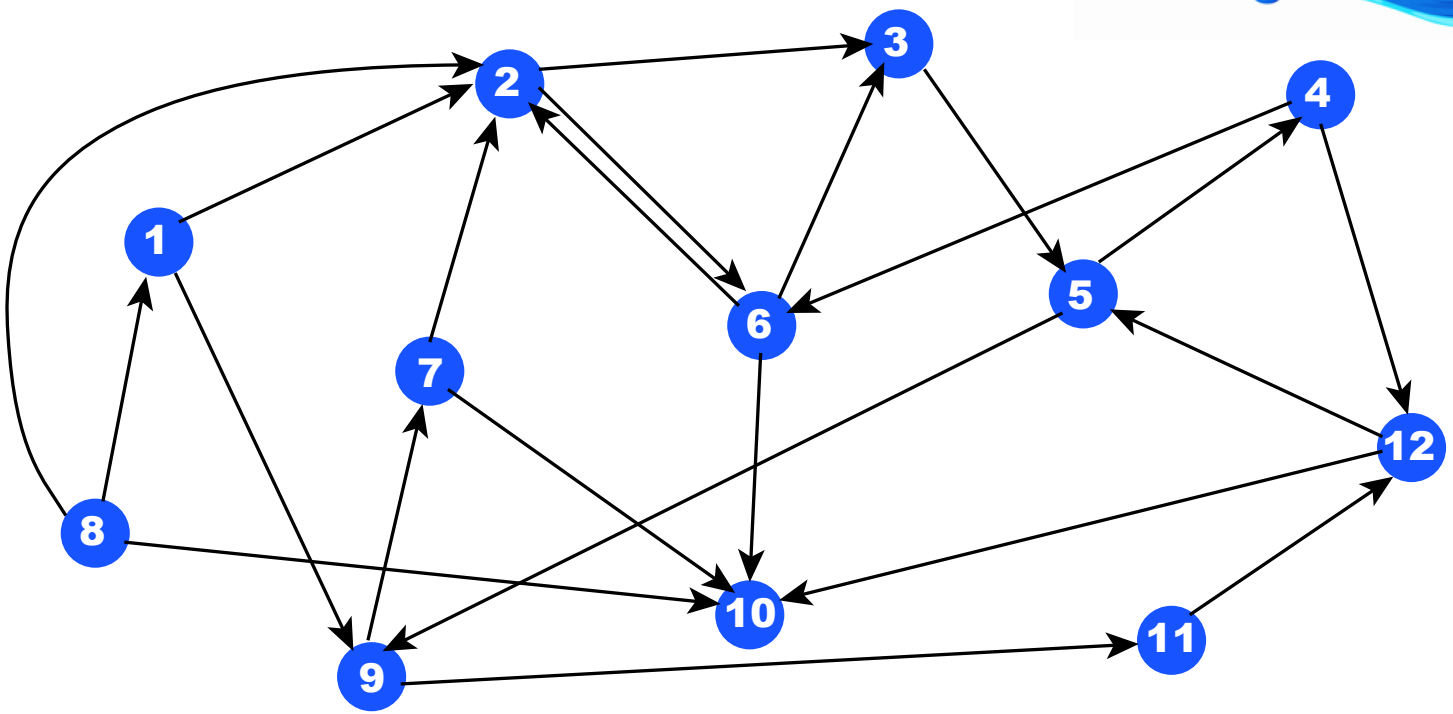
$$r_i = \frac{1-\alpha}{n} + \alpha \sum_{j \to i} \frac{r_j}{d_j^+},$$

  where the sum is taken over all the inbound neighbors to webpage i, d_j^+ is the number of outbound neighbors of webpage j, and $\alpha \in (0, 1)$ is a constant known as the *damping factor*, usually $\alpha = 0.85$.

- **Why does this work?**

The random surfer interpretation

- Recall that the goal is to rank vertices according to their "centrality" within the network.
- Imagine you had a web surfer who navigates the WWW by choosing which links to follow at random.
- Specifically, when our surfer visits webpage i, she will choose where to go next with equal probability among all the outbound links of webpage i.
- In the field of probability we call this process a **random walk**.


1
2
3
4
5
6
7
8
9
10
11
12

Random walks on connected graphs

- Let $\{X_k : k \geq 0\}$ denote the stochastic process that tells us the identity of the vertex our surfer visits on the kth step.
- $\{X_k : k \geq 0\}$ is what we call in probability a Markov chain.
- If the underlying graph is connected, and we let $k \to \infty$, the proportion of visits to vertex i converges, i.e.,

$$\lim_{k \to \infty} \frac{\text{Number of visits to vertex } i \text{ in the first } k \text{ steps}}{k} = \pi_i$$

  exists, and is known as the **stationary probability** of vertex i.
- The **stationary probability** of vertex i has the interpretation of being the long-run proportion of time that our random surfer spends in vertex i.
- When the damping factor $\alpha = 0$, we have $r_i = \pi_i$!

Random walks on disconnected graphs

- The problem with the WWW is that it is a disconnected graph.
- On a disconnected graph our surfer can get "stuck".
- To fix this imagine our surfer has a coin that lands *heads* with probability α and *tails* with probability $1 - \alpha$.
- At each step, before choosing which link to follow next, she flips the coin:
  - If it lands *heads* she chooses with equal probability any of the outbound links if there is one, or chooses from all n webpages if there are no outbound links.
  - If it lands *tails* she chooses with equal probability any of the n webpages in the WWW.
- The **stationary probability** of vertex i is equal to its PageRank, i.e.,

$$\pi_i = r_i!$$

Other network centrality measures

- **Degree centrality:** for vertex i,

$$C_D(i) = D_i = \sum_{j \neq i} a_{ij}$$

  On directed graphs we define the in-degree and out-degree separately.

- **Closeness centrality:** let $d(i,j)$ denote the hop distance from vertex i to vertex j, and define

$$C_C(i) = \frac{n-1}{\sum_j d(i,j)}$$

  where n is the number of vertices in the graph.

- **Betweeness centrality:** let g_{jk} denote the number of paths connecting vertices j and k, and let $g_{jk}(i)$ denote the number of those paths that go through vertex i,

$$C_B(i) = \sum_j \sum_{k \neq j} \frac{g_{jk}(i)}{g_{jk}}$$

PageRank today

- The algorithm that Google uses today has greatly evolved since the original PageRank.
- Each website in the WWW still has a "universal" rank, although the way it is computed has become more sophisticated.
- The order in which the results of a search are displayed depends also on the user's computer, i.e., results are **personalized**.
- Personalized PageRank:

$$r_i = (1-\alpha)q_i + \alpha \sum_{j \to i} \frac{r_j}{d_j^+},$$

where $\mathbf{q} = (q_1, \ldots, q_n)$ is a probability vector that determines where to go after a *tail*.

PageRank today... cont.

- People have found ways to "cheat" Google and achieve high ranks artificially.
- Google keeps finding ways to identify cheaters.
- The PageRanks of all webpages in the WWW are constantly being updated.
- Given the sheer size of the WWW today, Google's evolved PageRank still performs remarkably well most of the times.

Thank you for your attention.