Great Ideas in Computing

University of Toronto CSC196
Winter/Spring 2019

Week 8: November 2-6 (2020)

Announcements

- Assignment 3 (A3) is now available on the course web page. It is due Wednesday, November 18 at 11 AM. I clarified the second part of question 2. Please submit on Markus pdf files for the assignment. We need pdf to annotate comments about your assignment.
- The third and last question on this assignment concerns neural nets. This will motivate us to briefly comment on the general topic of machine learning (ML) and how deep learning fits into this field. We will discuss this at the start of today.
- Roger Grosse's slides are available on Quercus. I also posted a link to the recording of that class.
- Next week is reading week. No classes.
- On November 18 we will have our final guest presentation by Professor Aleksandar Nikolov who will be discussing differential privacy. In this field of study, we are concerned with how to extract useful aggregate information from a large data base without compromising individual information.

Agenda and the road ahead

- Some quick comnments on ML.
- We will continue with some graph theory concepts and social networks
- With regard to social networks, we will discuss influence spread in a social network and also how the network stucture by itself can yield infomation about people and activity in a social network.
- We will then discuss complexity theory and the great idea of NP completeness. As in deep learning, our department has been and continues to be leaders in this field. Steve Cook won the Turing award (in 1982) for his seminal work in complexity theory and proof complexity.
- This will naturally lead us to the topic of cryptography and in particular complexity based cryptography. Here again, our department has played a seminal role led by the work of Charles Rackoff.

Of course, no one is an island in the world of research. Results take place in a context but still we can try to identify seminal ideas and papers and we are entitled to be a little parochial in identifying work that has been and continues to be done in our department and the Univeristy of Toronto.

Brief Comments on ML and deep learning

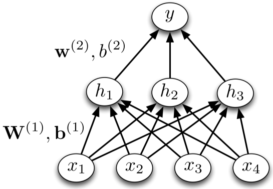
Of course, a one hour discussion on ML/deep learning can hardly do justice to the activity in ML and its impact. The topic of machine learning generally considers the following problem:

- We are given a labelled set of training data (e.g. images that we want to classify and the label for each image is a correct label).
- We train a learning algorithm on this training data and in general our learned algorithm may not even correctly label all the data in the training set. The hope is that any training error is small.
- Note: In question 3 of the assignment A3, we don't really have to train the neural net and can obtain an correct answer for all inputs. But this question was just meant to be an introduction to the topic.
- Neural nets now have available known methods (i.e., back propagation) to train the net, that is, to learn weights and biases in the network for a given *network architecture*.
- Once an algorithm has been developed on the training data, it is tested on a new set of data, called the test data.

Comments on ML and deep learning continued

- In terms of ML theory, one tries to obtain generalization bounds, namely bounding the test error in terms of the training error and quantifiable properties about the problem under consideration.
- It may be that one has to change the architecture and/or add new features so as to better describe the data and thereby improve upon the results.
- Beyond being able to classify data (eg images) we may want to reproduce similar images (with the possible abuse of *deep fake*).
- There are other ML methods (e,g, decision trees, logistic regression, support vector machines) are other methods. However, deep learning (meaning neural nets with many hidden layers) is currently what works significantly better than other methods for most applications involving lots of data.
- Indeed, along with the algorithmic ideas in deep learning, it is the availability iof large amounts of training data, and improved hardware (e.g., GPUs initially designed for computer games) that has led to the current success. **However, there is much to still be understood.**

Brief discussion on question 3 in Assignment A3

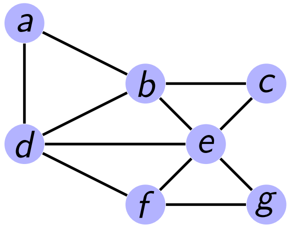

y
$\mathbf{w}^{(2)}, b^{(2)}$
h_1
h_2
h_3
$\mathbf{W}^{(1)}, \mathbf{b}^{(1)}$
x_1
x_2
x_3
x_4

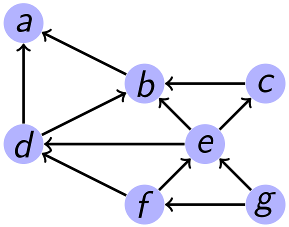
You need to find the weights and biases for each of the three "neurons" h_1, h_2, h_3 in the hidden layer and the one output neuron y. For example, we want $h_1 = \phi[w_1^1 x_1 + w_2^1 x_2 + w_3^1 x_3 + w_4^1 x_4 + b^1]$ and you have to determine the weights $w_1^1, w_2^1, w_3^1, x_4^1$ and the bias b^1 for h_1).

As just explained, with a number of positive and negative instance of inputs, the neural net could be learned so as to achieve the desired function f. But you do not have to emulate a learning algorithm but rather think about how you would want to set each of the h_i so that there would be any easy way to set y.

Back to graph theory concepts

Recall: undirected graphs vs. directed graphs





More definitions and terminology

- In order to refer to the nodes and edges of a graph, we define graph $G = (V, E)$, where
  - V is the set of nodes (often called vertices)
  - E is the set of edges (sometimes called links or arcs)

More definitions and terminology

- In order to refer to the nodes and edges of a graph, we define graph $G = (V, E)$, where
  - V is the set of nodes (often called vertices)
  - E is the set of edges (sometimes called links or arcs)

- Undirected graph: an edge (u, v) is an unordered pair of nodes.

More definitions and terminology

- In order to refer to the nodes and edges of a graph, we define graph $G = (V, E)$, where
  - V is the set of nodes (often called vertices)
  - E is the set of edges (sometimes called links or arcs)

- Undirected graph: an edge (u, v) is an unordered pair of nodes.

- Directed graph: a directed edge (u, v) is an ordered pair of nodes $\langle u, v \rangle$.
  - However, we usually know when we have a directed graph and just write (u, v).

Basic definitions continued

- First start with undirected graphs $G = (V,E)$.

Basic definitions continued

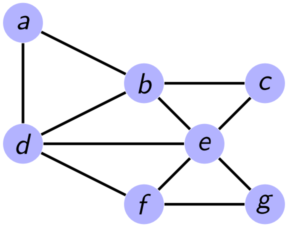
- First start with undirected graphs G = (V,E).
- A path between two nodes, say u and v is a sequence of nodes, say $u_1, u_2, \ldots, u_k$, where for every $1 \leq i \leq k-1$,
  - the pair (u_i, u_{i+1}) is an edge in E,
  - $u = u_1$ and $v = u_k$

Basic definitions continued

- First start with undirected graphs G = (V,E).
- A path between two nodes, say u and v is a sequence of nodes, say $u_1, u_2, \ldots, u_k$, where for every $1 \leq i \leq k - 1$,
  - the pair (u_i, u_{i+1}) is an edge in E,
  - $u = u_1$ and $v = u_k$
- The length of a path is the number of edges on that path.

Basic definitions continued

- First start with undirected graphs G = (V,E).
- A path between two nodes, say u and v is a sequence of nodes, say $u_1, u_2, \ldots, u_k$, where for every $1 \leq i \leq k-1$,
  - the pair (u_i, u_{i+1}) is an edge in E,
  - $u = u_1$ and $v = u_k$
- The length of a path is the number of edges on that path.
- A graph is a connected if there is a path between every pair of nodes. For example, the following graph is connected.



Romantic Relationships [Bearman et al, 2004]

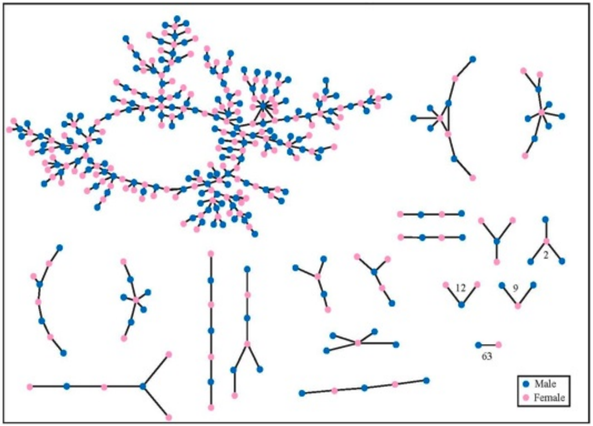
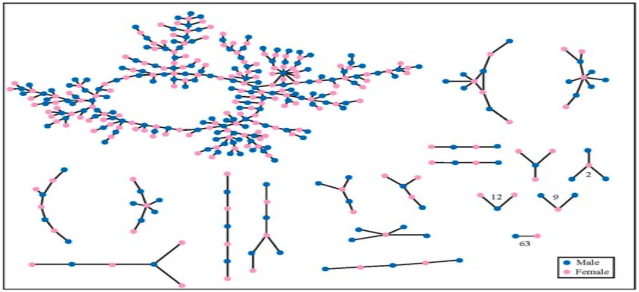


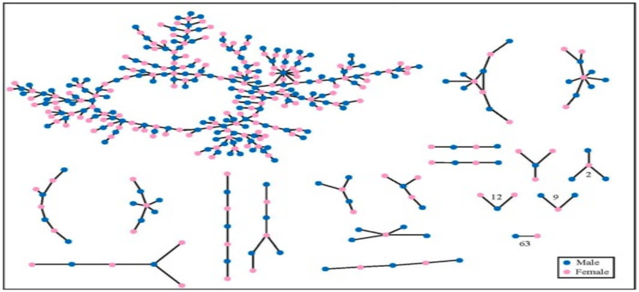
Figure: Dating network in US high school over 18 months.

- Illustrates common "structural" properties of many networks
- What predictions could you use this for?

More basic definitions



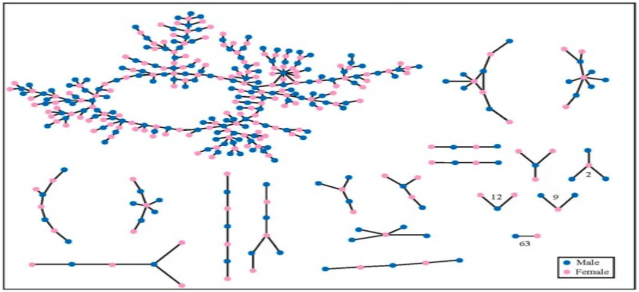
More basic definitions



Observation

Many connected components including one "giant component"

More basic definitions



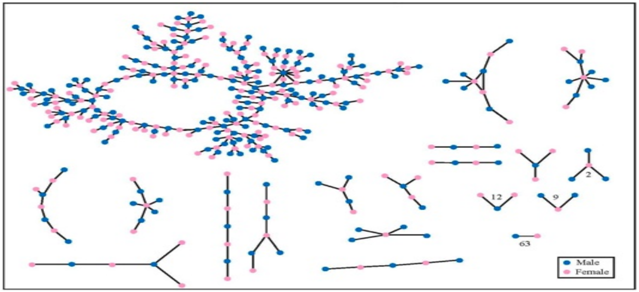
Observation

Many connected components including one "giant component"

- We will use this same graph to illustrate some other basic concepts.
- A cycle is path $u_1, u_2, \ldots, u_k$ such that $u_1 = u_k$; that is, the path starts and ends at the same node.

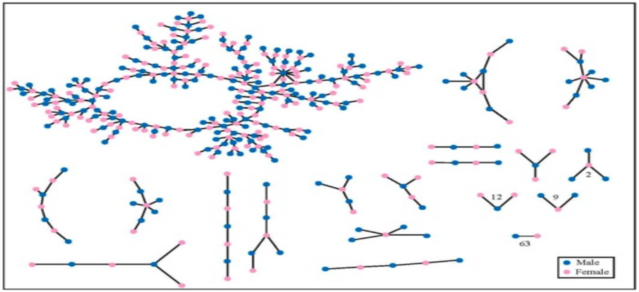
Simple paths and simple cycles

- Usually only consider simple paths and simple cycles: no repeated nodes (other than the start and end nodes in a simple cycle.)



Simple paths and simple cycles

- Usually only consider simple paths and simple cycles: no repeated nodes (other than the start and end nodes in a simple cycle.)



Observation

- There is one big simple cycle and (as far as I can see) three small simple cycles in the "giant component".
- Only one other connected component has a cycle: a triangle having three nodes. Note: this graph is "almost" bipartite and "almost" acyclic.

Example of an acyclic bipartite graph

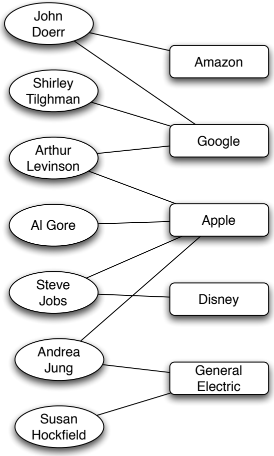


Figure: [E&K, Fig 4.4] One type of affiliation network that has been widely studied is the memberships of people on corporate boards of directors. A very small portion of this network (as of mid-2009) is shown here.

Florentine marriages and "centrality"

- Medici connected to more families, but not by much
- More importantly: lie between most pairs of families
  - shortest paths between two families: coordination, communication
  - Medici lie on 52% of all shortest paths; Guadagni 25%; Strozzi 10%

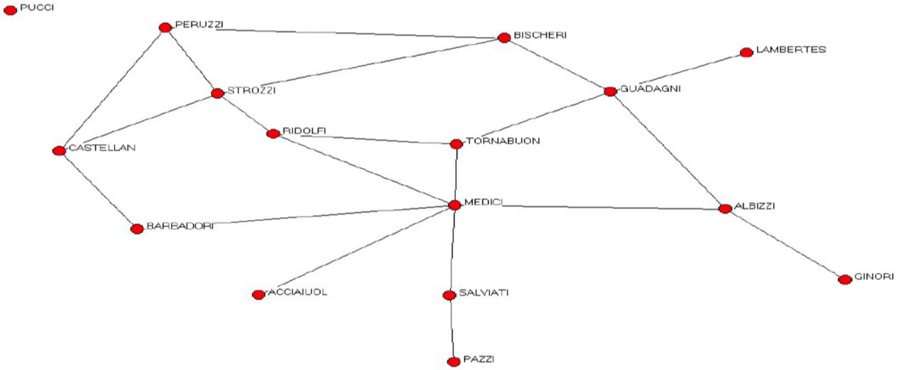


Figure: see [Jackson, Ch 1]

Breadth first search and path lengths [E&K, Fig 2.8]

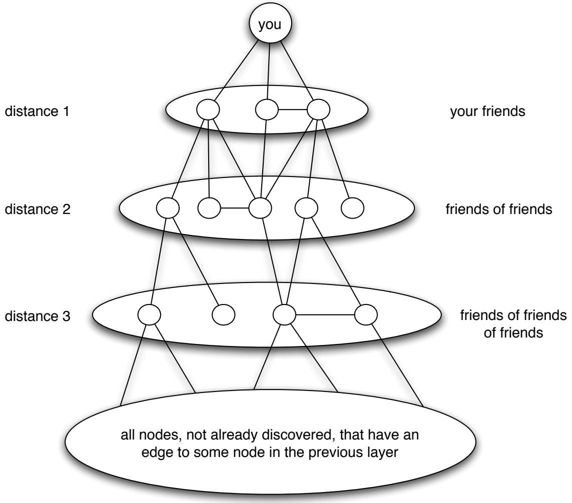

you
distance 1
your friends
distance 2
friends of friends
distance 3
friends of friends of friends
all nodes, not already discovered, that have an edge to some node in the previous layer

Figure: Breadth-first search discovers distances to nodes one “layer” at a time. Each layer is built of nodes adjacent to at least one node in the previous layer.

The Small World Phenomena

The small world phenomena suggests that in a connected social network any two individuals are likely to be connected (i.e. know each other indirectly) by a short path; that is, short in the graph theory sense of the number of edges.

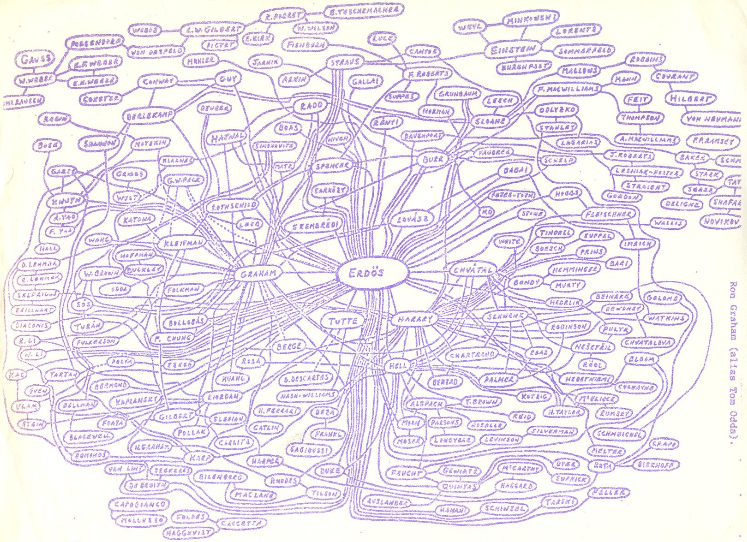
In Milgram's [1967] small world experiment, he asked 296 randomly chosen people in Omaha to forward a letter to a target person (a stockbroker) living in a Boston suburb. Of the 64 chains that succeeded the median length of the letter chain was 6, the motivation for the play and movie that came to popularize the phenomena known as six degrees of separation.

The interesting phenomena is not so much that there are short paths between people, but more that the successful letters made their way without any centralized algorithm but rather were guided by geographic distance (and perhaps occupation). There are interesting provable results and some recent computational studies helping to explain this phenomena. Distance need not be geographic distance but rather can be some concept of "social distance".

Small Collaboration Worlds

For now let us just consider collaboration networks like that of mathematicians or actors. For mathematicians (or more generally say scientists) we co-authorhsip on a published paper. For actors, we can form a collaboration network where an edge represents actors perfoming in the same movie. For mathematicians one considers their Erdos number which is the length of the shortest path ito Paul Erdos. For actors, a popular notion is ones Bacon number, the shortest path to Kevin Bacon.

Erdos collaboration graph drawn by Ron Graham [http:/www.oakland.edu/enp/cgraph.jpg]



Ron Graham (alias Tom Odda).

Figure 1
To appear in Topics in Graph Theory (F. Harary, ed.) New York Academy of Sciences (1979).

Announcements and agenda

- We ended at slide 18 on Wednesday. I do want to go over slide 16 and the Small Worlds Phenomena.
- Assignment 3 is due Wednesday, November 18. Also on November 18, we will have our last guest presentation by Aleksander Nikolov.
- Next week is reading week so no classes. We resume the week of November 16. Marta will do the tutorial on Monday, November 16. .
- After quickly going over the small worlds phenomena, we will return to graph concepts starting with directed graphs and then move on to discuss some properties of social networks. For those interested in the samll worlds phenomema, you can look at Chapter 20 of the Easley and Kleinberg (2010) textbook. The text is an excellent text for the study of social networks and more generally social the importance of networks. (You can also look at my CSC303 slides on my web page.)

Analogous concepts for directed graphs

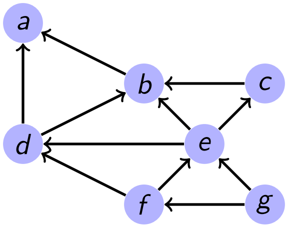
- We use the same notation for directed graphs, i.e. denoting a di-graph as $G = (V, E)$, where now the edges in E are directed.

Analogous concepts for directed graphs

- We use the same notation for directed graphs, i.e. denoting a di-graph as $G = (V, E)$, where now the edges in E are directed.
- Formally, an edge $\langle u, v \rangle \in E$ is now an ordered pair in contrast to an undirected edge (u, v) which is unordered pair.
    - However, it is usually clear from context if we are discussing undirected or directed graphs and in both cases most people just write (u, v).

Analogous concepts for directed graphs

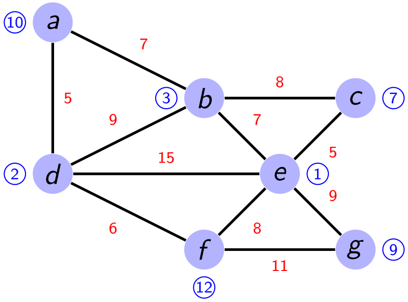
- We use the same notation for directed graphs, i.e. denoting a di-graph as $G = (V, E)$, where now the edges in E are directed.
- Formally, an edge $\langle u, v \rangle \in E$ is now an ordered pair in contrast to an undirected edge (u, v) which is unordered pair.
    - However, it is usually clear from context if we are discussing undirected or directed graphs and in both cases most people just write (u, v).
- We now have directed paths and directed cycles. Instead of connected components, we have strongly connected components.



The graph above is *not* strongly connectd (for example, node *a* is a sink and cannot reach any other node).

Weighted graphs

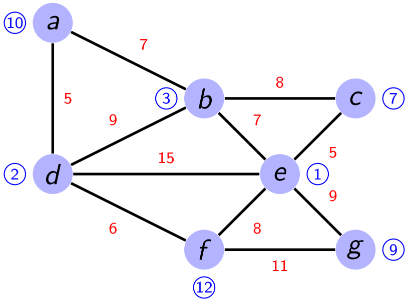
- We will often consider weighted graphs. Lets consider a (directed or undirected) graph $G = (V, E)$. Example:



  - red numbers: edge weights
  - blue numbers: vertex weights

Weighted graphs

- We will often consider weighted graphs. Lets consider a (directed or undirected) graph $G = (V, E)$. Example:

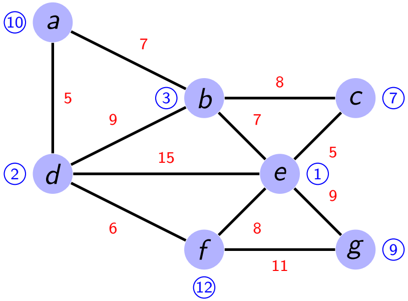


  - red numbers: edge weights
  - blue numbers: vertex weights

- We can have a weight $w(v)$ for each node $v \in V$ and/or a weight $w(e)$ for each edge $e \in E$.

Weighted graphs

- We will often consider weighted graphs. Lets consider a (directed or undirected) graph $G = (V, E)$. Example:

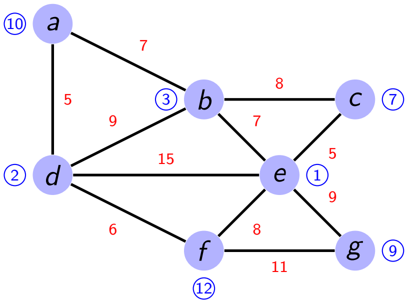


  - red numbers: edge weights
  - blue numbers: vertex weights

- We can have a weight $w(v)$ for each node $v \in V$ and/or a weight $w(e)$ for each edge $e \in E$.
- For example, in a social network whose nodes represent people, the weight $w(v)$ of node v might indicate the importance of this person.

Weighted graphs

- We will often consider weighted graphs. Lets consider a (directed or undirected) graph $G = (V, E)$. Example:



  - red numbers: edge weights
  - blue numbers: vertex weights

- We can have a weight $w(v)$ for each node $v \in V$ and/or a weight $w(e)$ for each edge $e \in E$.
- For example, in a social network whose nodes represent people, the weight $w(v)$ of node v might indicate the importance of this person.
- The weight $w(e)$ of edge e might reflect the strength of a friendship.

Edge weighted graphs

- When considering edge weighted graphs, we often have edge weights $w(e) = w(u, v)$ which are non negative (with $w(e) = 0$ or $w(e) = \infty$ meaning no edge depending on the context).

- In some cases, weights can be either positive or negative. A positive (resp. negative) weight reflects the intensity of connection (resp. repulsion) between two nodes (with $w(e) = 0$ being a neutral relation).

- Sometimes we will only have a qualitative (rather than quantitative) weight, say to reflect a strong or weak relation.

- Analogous to shortest paths in an unweighted graph, we often wish to compute least cost paths, where the cost of a path is the sum of weights of edges in the path.

Social networks

A social network is a network $G = (V, E)$ where the nodes in V are people or organizations. Social networks can be undirected or directed networks.

The edges can be relations between people (e.g. friendship) or membership of an individual in an organization.

Social networks can be of any size (e.g., a small network like the Karate Club on slide 14 in the week 7 slides) or enormous networks like Facebook and Twitter. We usually think of Facebook as an undirected graph (where *friendship* is an undirected edge) and Twitter as a directed graph (i.e., where *follows* is a directed edge).

Understanding how networks evolve, the resulting structure of social networks, and computational aspects for dealing with large networks is an active field of study in CS as well as in sociology, political science, economics, epidemiology, and any field that studies human behaviour. J. Kleinberg's 2000 analysis with regard to the six degrees of separation phenomena is an early result that sparked interest in algorithmic aspects of social networks.

The computational challenge presented by super large networks

The size of some modern networks such as the web and social networks such as Facebook are at an unprecedented scale.

As of October 10, 2020, Facebook has roughly 2.7 billion monthly active users worldwide. The average facebook user has 155 friends which then implies about $\frac{2.7 \cdot 155}{2}$ billion edges. It is interesting to note that 90% of daily active users are outside USA and Canada. See https://www.omnicoreagency.com/facebook-statistics/ // for lots of interesting demographic and other facts about Facebook.

What does this imply for the complexity of algorithms involving such super large networks?

Linear is the new exponential

In complexity theory (e.g. in the P vs NP issue that we will be discussing) we say (as an abstraction) that polynomial time algorithms are “efficient” and “exponential time” is infeasible. There are, of course, exceptions but as an abstraction this has led to invaluable fundamental insights.

As problem instances have grown, there was a common saying that “quadratic (time) is the new exponential”.

But with the emergence of networks such as the web graph and the Facebook network, we might now say that “linear is the new exponential” when it comes to extracting even the most basic facts about these networks. For example, how do we even estimate the average node degree in a giant network?

There are many facts about large networks that we would like to extract from the network. For example, how do we find “influential” or “interesting nodes” in a social network?

Sublinear time algorithms

What is sublinear time?

In general when we measure complexity, we do so as a funtion of the input/output size. For graphs $G = (V, E)$, the size of the input is usually the number of edges E. (An exception is that when the graph is presented say as an adjacency matrix, the size is n^2 where $n = |V|$.)

Since our interest is in massive information and social networks, we consider sparse graphs (e.g. average constant degree) so that $|E| = O(|V|)$ and hence we will mean sublinear time as a function of n. The desired goal will be time bounds of the form $O(n^\alpha)$ with $\alpha < 1$ and in some cases maybe even $O(\log n)$ or $polylog(n)$.

Given that optimal algorithms for almost any graph property will depend on the entire graph, we will have to settle for approximations to an optimum solution. Furthermore, we will need to sample the graph so as to avoid having to consider all nodes and edges. And we will need a way to efficiently access these massive graphs,

Coping with massive social graphs continued

One way to help coping with massive networks is to hope to utilize some substantial amount of parallelism. There is an area of corrent research concerning massive parallel computation (MPC) models where (in principle) we can achieve sublinear time by distributing computation amongst a large (i.e., non constant) number of processors.

But even if we could muster and organize thousands of machines, we will still need random samplng, approximation, and have highly efficient "local information algorithms".

Finally, in addition to random sampling and parallelism, we will have to hope that social networks have some nice structural properties that can be exploited to as to avoid complexity barriers that exist for arbitrary (sparse) graphs. These complexity barriers will be made clear when we discuss complexity theory, *NP completeness* and *NP hardness*.

Preferential attachment models

Preferential attachnment models (also called "rich get richer" models) are probabilistic generative models explaining how various networks can be generated. Namely, after starting with some small graph, when we add a new node u, we create a number of links between u to some number m of randomly chosen nodes $v_1, v_2, \ldots, v_m$. The probability of choosing a v_i is proportional to the current degree of v_i. More generally, the probability of choosing a node v_i can be an increasing function of the degree,

These models have been used to help explain the structure of the web as well as social networks. Furthermore, networks generated by such a process have some nice structural properties allowing for substantially more efficient algorithms than one can obtain for arbitrary graphs.

For such models, there are both provable analytic results as well as experimental evidence on synthetic and real networks that support provable results that follow from the model. (Remember, a model is just a model and is not "reality"; as models are implifications of real networks, they may not account for many aspects in a real network. For example, in this basic model, all the edges for a new node are set upon arrival.

Consequences for networks generated by a preferential attachment process

There are many properties, believed and sometimees proven. about preferential attachment network models that do not hold for uniformly generated random graphs (e.g., create sparse graphs with consstant average degree by choosing each possible edge with say probability proportional to $\frac{1}{n}$).

One of the most interesting and consequential proerties is that vertex degrees satisfy a *power law distribution* in expectation. Specifically, the expectation fraction $P(d)$ of nodes whose degree is d is proportional to $d^{-\gamma}$ for some $\gamma \geq 1$. Such a distribution is said to have a *fat tail*.

In a uniformaly random sparse graph (with average degree d_{avg}), with high probability , the fraction of nodes having a large degree $d > d_{avg}$ is proportional to c^{-d} for some $c > 1$.

The Barabasi and Albert preferential model

Barabasi and Albert [1999] specified a particular preferential attachment model and conjectured that the vertex degrees satisfy a power law in which the fraction of nodes having degree d is proportional to d^{-3}.

They obtained $\gamma \approx 2.9$ by experiments and gave a simple heuristic argument suggesting that $\gamma = 3$. That is, $P(d)$ is proportional to d^{-3}

Bollobas et al [2001] prove a result corresponding to this conjectured power law. Namely, they show for all $d \leq n^{1/15}$ that the *expected* degree distribution is a power law distribution with $\gamma = 3$ asymptotically (with n) where n is the number of vertices.

Note: It is known that an actual realized distribution may be far from its expectation, However, for small degree values, the degree distribution is close to expectation.

When we say that a distribution $P(d)$ is a power law distribtion this is often meant to be a "with high probaility" whereas results for networks generated by a preferential attachment process the power law is usually only in expectation.

Proven or observed properties of nodes in a social network generated by preferential attachment models

In addition to the power law phenomena suggesting many nodes with high degree, , other properies of social networks have been obseerved such as a relatively large number of nodes u having some or all of the following properties.

- high clustering coefficient defined as : $\frac{(u,v),(u,w),(v,w) \in E}{(u,v),(u,w) \in E}$. That is, mutual friend of u are likely to be friends.
- high centrality ; e,g, nodes on many pairs of shortest paths.

Brautbar and Kearns refer to such nodes as "interesting indiviudals" and these individuals might be candidates for being "highly influential individuals". Bonato et al [2015] refers to such nodes as the *elites* of a social network.

Other proven or observed properties of networks generated by preferentical attachment models

- correlation between the degree of a node u and the degrees of the neighboring nodes.
- graph has small diameter; suggesting “6 degrees of separation phenomena”
- relatively large dense subgraph communities.
- rapid mixing (for random walks to approach stationary distribution)
- relatively small (almost) *dominating sets* .

On my 303s20 web page, I posted a paper by Avin et al (2018) that shows that preferential attachment is the *only* “rational choice” for players (people) playing a simple natural network formation game. It is the rational choice in the sense that the strategy of the players will lead to a unique equilibrium (i.e. no player will want to deviate assumming other players do not deviate). For those intersted, I have posted (in my CSC303 webpage) a number of other papers on elites in a social network and preferential attachment.