1 Network data incompleteness and sampling

Sometimes in network analysis and modeling, we consider only a subsample of a given network, rather than all of the vertices and edges. The particular reasons why we may need to work with a sample of a network rather than the whole network are varied, but largely fall into two categories:

- *computational cost*: the network analysis we are conducting is too expensive (time, memory, money) to run on the full network; and
- *limited access*: we do not have access to or cannot obtain the full network, and must instead work with what we can get (e.g., through an API, as with Twitter, or from scraping, as with the WWW).

For instance, the World Wide Web, which has been estimated to contain at least 10^{10} vertices, is too large to store for anyone except the largest technology companies, and even they have difficulties running sophisticated network analyses on such large networks.[1] The best strategy one can employ in such situations is to only run calculations that are linear in the number of edges $O(m)$, as these only require scanning the adjacency list a small number of times. For example, computing the mean degree or, if the graph is sparse $\langle k \rangle = O(1)$, various node-level statistics like the local clustering coefficient or degree-degree correlations. In contrast, most calculations that require looking at all pairs of nodes will be prohibitive, because they run in $\Omega(n^2)$ time. For example, calculating the diameter or diagonalizing the adjacency matrix. Calculations that lay in-between may or may not be tractable, depending on the graph, such as counting triangles or other small motifs.

In all cases where we cannot analyze the full network $G = (V, E)$, we instead work with a network sample, i.e., a subset of vertices $V' \subset V$ and a subset of the edges $E' \subset E$, defining the subgraph $G' = (V', E')$. But, *how* we obtain G' from G can strongly impact the analysis we conduct, and hence the conclusions we draw, because it determines which edges we observe and don't observe.

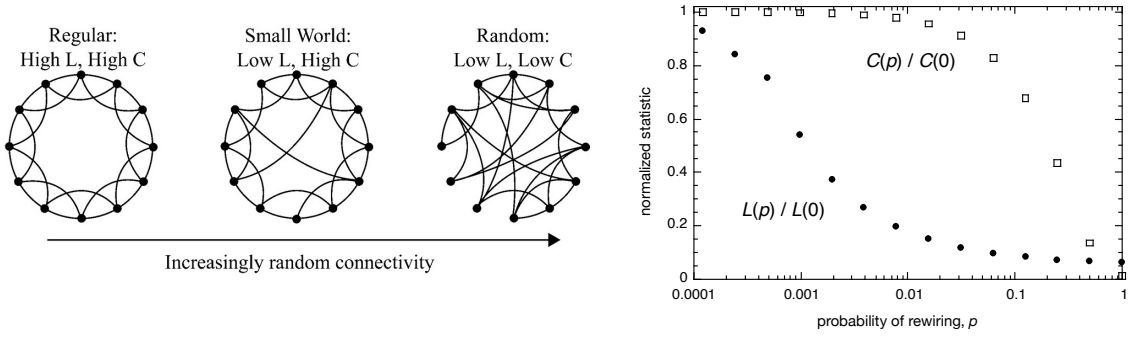
The key point : Working with a sampled network G', rather than the full network G, can be fine if the function f you are computing with the data would yield equivalent results in both cases, i.e., if $f(G) = f(G')$, or, more generally, if the distribution of outputs is the same: $\Pr(f(G)) = \Pr(f(G'))$. Whether this is true depends on your function f, and what it does with its input network. For instance, the mean degree function is relatively robust to certain types of sampling, but not others, while functions like estimating the degree distribution $\Pr(k)$ or the diameter $\ell_{\max}$ are very fragile.

[1] For instance, in 2011, Facebook undertook two such studies of its friendship network, which at the time contained $n = 7.21 \times 10^8$ nodes and $m = 6.9 \times 10^{10}$ undirected friendship edges worldwide. In Backstrom et al., "Four Degrees of Separation." Preprint, arXiv:1111.4570 (2011), they studied the path-length distribution using a clever sampling approach, while in Ugander et al., "The Anatomy of the Facebook Social Graph." Preprint, arXiv:1111.4503 (2011), they analyzed other, mainly node-level patterns. Both analyses used a mind-boggling amount of computational resources to estimate basic descriptive statistics.

Consider the other functions $f(G)$ that have we encountered in the class so far: which seem more likely to be robust vs. more likely to be fragile? In this lecture, we'll explore several different concrete network sampling algorithms, and how they induce biases in something as simple as the sampled degree distribution.

1.1 Implications from Watts-Strogatz for subsampled networks

Recall the Watts-Strogatz model of "small-world networks,"[2] in which we take a "ring" network where every node has degree $2 + k$, where the k extra links attached to node i connect to the nearest k nodes to i along either direction of the ring, and then randomly rewire each edge with some probability p. As p varies from 0 to 1, we rewire a greater proportion of links randomly, and so p interpolates between a regular-graph or lattice structure ($p = 0$) and an Erdős-Rényi random-graph structure ($p = 1$). In response to this variation, the clustering coefficient C and the mean geodesic distance ℓ behave differently, as illustrated in the figure below (adapted from Watts & Strogatz (1998)).



But, this model *also* has a lesson to teach us about subsampling networks. Suppose that we are in the low-p regime, in which only a very small fraction of edges have been rewired, e.g., $p = 0.01$. Here, the path length structure of the network, measured by the mean geodesic distance $\langle \ell \rangle$, has already "collapsed", and that property depends on the presence of the small number of rewired links. That is, the rewired links act like major thoroughfares in the All-Pairs-Shortest-Paths calculation, so that the vast majority of geodesic paths run over these few "long-range" links.

Now, imagine G is very large, and that to produce a more reasonably sized G', we are sampling edges uniformly at random. If the probability q that we sample any particular edge is not high, then we are likely to miss the existence of most or all of the few long-range links. Hence, we are likely to dramatically misestimate the mean geodesic distance $\langle \ell \rangle$, because we don't sample the few critical edges that support it. (Effectively, subsampling here acts to effectively lower the p, thereby moving our perception of the network to the left in the figure above.) In contrast, our estimate of the clustering coefficient C will be highly robust to these errors.

[2] Watts and Strogatz, "Collective dynamics of small-world networks." *Nature* **393**, 440–442 (1998).

1.2 General approaches to sampling a network

There are many ways to derive a network sample G' from a larger network G, and these largely divide into two classes, depending on whether or not we have access to the full network. If we can store the full network, then we can, in principle, choose any vertex or edge to include. Otherwise, we must sample edges and vertices by exploring the network starting from one or several known "seed" vertices. (Can you think of some real-world networks in which the former or latter would be the case?)

In this lecture, we will consider examples of both types of sampling approaches. The following is a rough taxonomy:

1. *Probabilistic sampling*: assumes access to the full network
    - Uniform vertex: include each vertex i (and its neighbors) with probability p
    - Uniform edge: include each edge (i, j) with probability p
    - Degree-proportional: include each vertex i (and its neighbors) with probability $p \propto k_i$
    - Attribute-proportional: include each vertex i (and its neighbors) with probability $p \propto x_i$
2. *Seed-based sampling*: assumes access to one or more seed vertices only
    - Snowball sampling: for each seed vertex i, and distance ℓ, include all vertices (and their neighbors) for an ℓ-step breadth-first search tree rooted at i
    - BFS edge sampling: for each seed vertex i, and distance ℓ, include all edges included in an ℓ-step breadth-first search tree rooted at i
    - Adaptive sampling: for each seed vertex i, and integer s, include all vertices (and their neighbors), or include all edges, in an adaptively-grown tree containing s vertices rooted at i
    - Random walks: for each seed vertex i, and length ℓ, include all nodes (and edges) in an ℓ-step random walk on the graph
3. *Still other approaches*
    - Degree sampling: include all vertices with degree above $k_{\min}$ or include the top ℓ vertices by degree

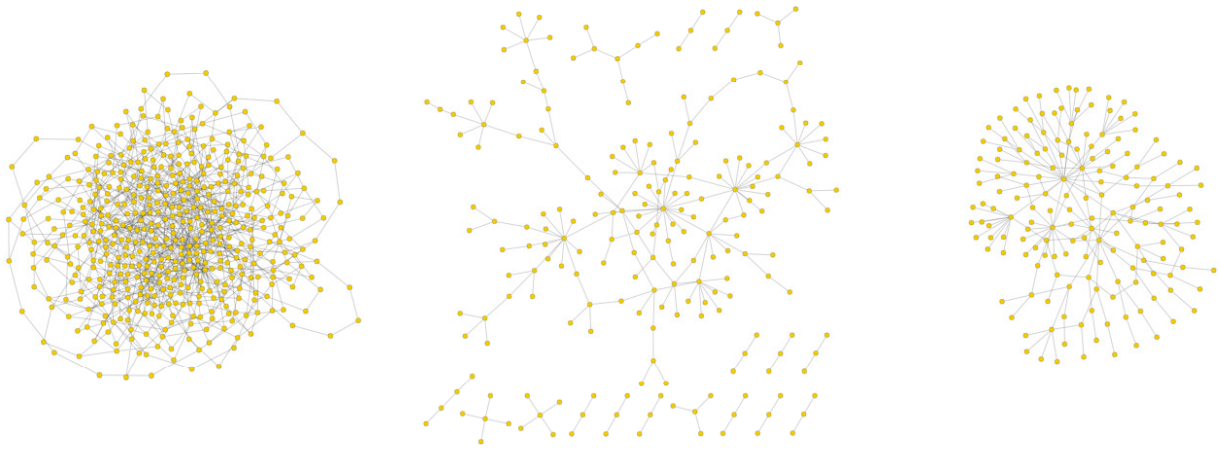
Mathematically, every network sample, regardless of how it is constructed, can be viewed as a particular ordering of the network's edges in an edge list representation, where we then only get to observe the edges (and associated nodes) up to some specified "depth" in that list. For instance, in vertex-sampling approaches, we observe some vertex i and then we observe all of its neighbors; this is equivalent to an ordering in which all the edges (i, x) for any x appear contiguously.

1.3 Sampling induces patterns

Any time we discard some edges or some vertices (and all edges incident to them) from a network, we are changing the distribution of edges in the network, and this can change the resulting statistics that we compute on them.

There are three general patterns that sampling produces.

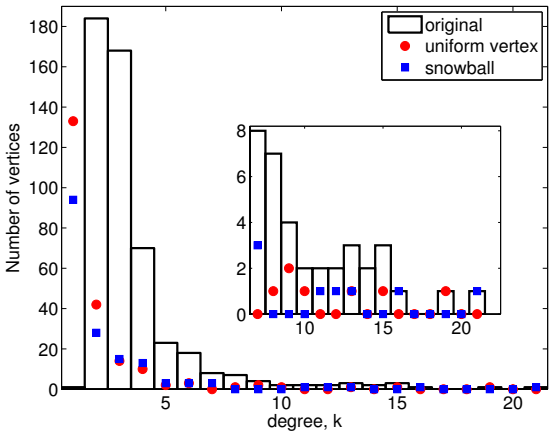
- *Extreme sparsity*, which appears when we sample a modest number of vertices or edges semi-independently, e.g., in uniform sampling approaches, and occurs because the probability is very small that the neighbor sets of two such vertices overlap.
- *Compact but biased subgraphs*, which appears when we preferentially sample vertices and edges that are close to each other in the network, e.g., in seed-based sampling.
- *Overabundance of low-degree vertices* (often $k = 1$), which is caused by including the neighbors of some vertices, but not those neighbors' neighbors.



For example, above are three versions of the same network. The first is the full network, generated using the configuration model with a power-law degree distribution, where $p_k \propto k^{-3}$ with $k_{\min} = 2$ for $n = 500$ nodes and $m = 880$ edges. The other two show the observed graph after sampling.

In the full network, the mean degree is $\langle k \rangle = 3.5$, which is large enough that the network is a single connected component. The second shows a uniform vertex sample of this network with $p = 0.1107$, producing $n_{\text{sample}} = 211$ and $m_{\text{sample}} = 205$. Although the average degree here is $\langle k \rangle = 1.9$, the

network is not connected. The third shows an $\ell = 2$ snowball sample from a randomly chosen seed, producing $n_{\text{sample}} = 164$ and $m_{\text{sample}} = 187$, with mean degree $\langle k \rangle = 2.3$. Here are the degree distributions for all three networks, with the right-tail show in the inset.



Several things can be seen in these figures. First, while the full network is connected, the uniform sample is not, and it instead contains many small components. The largest component also does not really resemble the full network, and instead has several high-degree vertices loosely connected in the core with many long branches extending from them. Second, the snowball sample is a connected graph, but that is always true, since we selected vertices by following paths outward from the seed.

Both networks include a great many vertices with degree $k = 1$, while the full network contains none, by construction (recall that $k_{\min} = 2$). In the uniform sample, these vertices are distributed somewhat evenly across the entire network, while in the snowball sample, they are all neighbors of the vertices at $\ell = 2$ away from the seed, i.e., they are vertices that neighbor the vertices in our tree and in particular, they are neighbors of nodes at the edge of our snowball sample.

In the full network, the underlying degree distribution is very right-skewed, with very many vertices in the degree 2–4 range. Most of the new $k = 1$ vertices are drawn from this population because they are so abundant. But the sampling also reduces the degree of some higher-degree vertices, as shown in the inset. In general, only the highest-degree vertices are faithfully captured via sampling, because these vertices are the more likely to be neighbors of a vertex we do sample, and thus more likely to themselves be sampled. (Do you see why this is true for both sampling approaches?)

1.4 Uniform vertex sampling

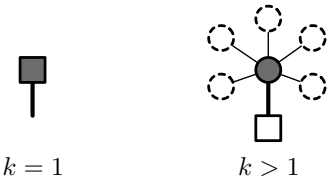
In the uniform sampling situation, we have access to the entire network and choose a subset of vertices, and their neighbors, to work with. For a target sample size of s vertices, we include each vertex i, along with all of its neighbors, independently with some probability p. How should we choose p so that the size of the sampled graph is close to s?

Because vertices are chosen independently and with equal probability, each time we select some i, in expectation we add $1+\langle k\rangle$ vertices to the sampled network. Thus, if we choose each vertex with probability $p = s/(1+\langle k\rangle)n$ we obtain s vertices in the sample. Substituting $\langle k\rangle = 2m/n$ into this expression yields

$$p = \frac{s}{\left(1+\frac{2m}{n}\right)n} = \frac{s}{n+2m} \ ,$$

which is proportional to s/n in a sparse graph.

How will such sampling change the degree distribution of a network? Or, more specifically, how many of our sampled vertices will have degree $k' = 1$? There are two ways a vertex i could have degree 1 in the sample: either i had degree $k = 1$ in the full network and was sampled *or* i had degree $k > 1$ in the full network and exactly one of its neighbors was sampled:



$k = 1$ $k > 1$

where the boxes indicates a sampled vertex. The first possibility occurs with probability p for each of the n_1 vertices with degree 1, while the second occurs with probability $p(1-p)^{k-1}$ for each of the n_k vertices with degree $k > 1$. A similar argument holds for general k': either a vertex with degree $k = k'$ was sampled directly, or it had degree $k > k'$ and exactly k' of its neighbors were sampled. Thus, the expected number of degree k' vertices in the sampled network is

$$\mathrm{E}[n'_k] = p\, n_{k'} + p^{k'} \left(\sum_{k=2}^{n} (1-p)^{k-k'}\, n_k \right) \ .$$

Without knowing the number of vertices of different degrees n_k in the original network, this expression cannot be further simplified. But, if we know the degree distribution and the size of the network, then we can obtain estimates of the n_k and numerically estimate the sampled degree distribution.

1.5 Uniform edge sampling

Another approach is to choose edges uniformly at random, with some probability p. Unlike in uniform vertex sampling, here we do not add the neighbors of the vertices in the edge we sample.[3]

Another difference with uniform vertex sampling is that uniform edge sample does not alter the relative distribution of edges, because we sample the edges attached to a vertex i with probability proportional to its degree k_i. A vertex with degree k in the full network will have degree $p\,k$ in the sampled network. As a result, the expected sampled degree distribution is the same as the observed degree distribution in the full graph (but with Poisson noise around each value of $\Pr(k)$).

The sampled network will still be extremely sparse, as the $p\,m$ edges are distributed across n vertices, meaning that the average degree of the sampled network will be proportionally lower than in the full network, $\langle k' \rangle = p\langle k \rangle$. Roughly speaking, if the average degree approaches or falls below the critical value of $\langle k' \rangle = 1$, then the sampled graph may not have a large connected component, and the graph is likely to be highly disconnected.

1.6 Snowball sampling

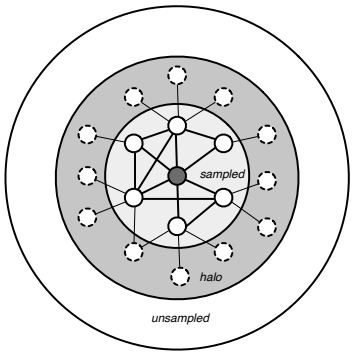
The most popular form of seed-based sampling is "snowball sampling," in which we choose some seed i and a distance ℓ. We then include all the vertices, and their connections, that are within a distance ℓ from i (as in a BFS tree). This process divides vertices into three classes:

- sampled vertices, which are a geodesic distance $d_{ij} \leq \ell$ from vertex i,
- partially sampled vertices, which are a distance $d_{ij} = \ell + 1$ from vertex i, i.e., vertices that neighbor those vertices with $d_{ij} = \ell$ exactly, and
- unsampled vertices, which have $d_{ij} > \ell + 1$.

Under this sampling approach, all vertices that are sampled directly have their entire neighbor sets included and all vertices that are unsampled are omitted entirely. It is the partially sampled vertices, however, that we see a skewed view of, because we only see them because they have at least one neighbor at distance ℓ from the seed vertex. The degree of a partially sampled vertex is equal to the number of neighbors it has at distance ℓ to i, which is typically 1. Thus, snowball sampling may give a reasonably accurate representation of the local structure around i, but it includes a large "halo" of degree 1 vertices that can complicate any subsequent analyses.

This situation may sound fairly reasonable, because we get a good view of the local neighborhood of the seed, and perhaps we just throw out the vertices in the halo. Snowball samples, however, are not exactly unbiased.

[3]Sampling edges and then adding the neighbors of the endpoints can also be done. Its bias is similar to that of uniform vertex sampling: instead of choosing a single vertex, we instead choose two, which happen to be connected.



Recall that the mean degree of a vertex's neighbor is usually greater than the vertex's degree itself. Now consider how the observed degrees change as we grow the snowball sample outward from i. Because the neighbors of i have, on average, larger degrees than i, and their neighbors have, on average, larger degrees than them, a snowball sample will tend to touch high degree vertices very quickly. Put another way, high degree vertices, by virtue of their having many connections, have many paths into them, and thus are more likely to be included in geodesic paths emanating from some seed vertex and therefore have a relatively greater probability of being included in a sample.

Returning to our example network from Section 1.3, we can illustrate the over-representation of high-degree vertices in snowball samples by counting the fraction of times each vertex j appears in an $\ell = 2$ snowball sample with seed vertex i, for all i. The figures below show the results; on the left, the fraction of times a vertex is fully sampled (i.e., has $d_{ij} \leq \ell$), while on the right is the fraction it is partially or fully sampled ($d_{ij} \leq \ell + 1$). To place these numbers in context, the diameter of this network is 11, and the mean geodesic path length is 4.9.

What is immediately clear is that high degree vertices are strongly over-represented in either case, appearing in many more snowball samples than any particular low degree vertex. In this particular network, the three highest degree vertices are each directly sampled in more than 16% of snowball samples, and are each partially sampled in another 33%. Another point worth noting is the large variance in these numbers for low degree vertices, indicating that some are much more centrally located, and thus have many geodesic paths crossing them, than we might naïvely expect based on their degree alone.[4] The take-home message here is that snowball sampling is not a bad way to get a connected, locally accurate sample of a network, but it is far from an unbiased one.

[4]The distribution of betweenness centrality, or rather, the distribution of geodesic paths, across vertices should, in principle, tell us a great deal about what the structure of snowball samples should look like for any particular network. However, in general, if we are forced to do a snowball sample on some network, we generally don't know what the distribution of geodesic paths should be. One potential circumvention would be if the distribution of geodesic paths itself followed some predictable pattern across different general classes of networks.

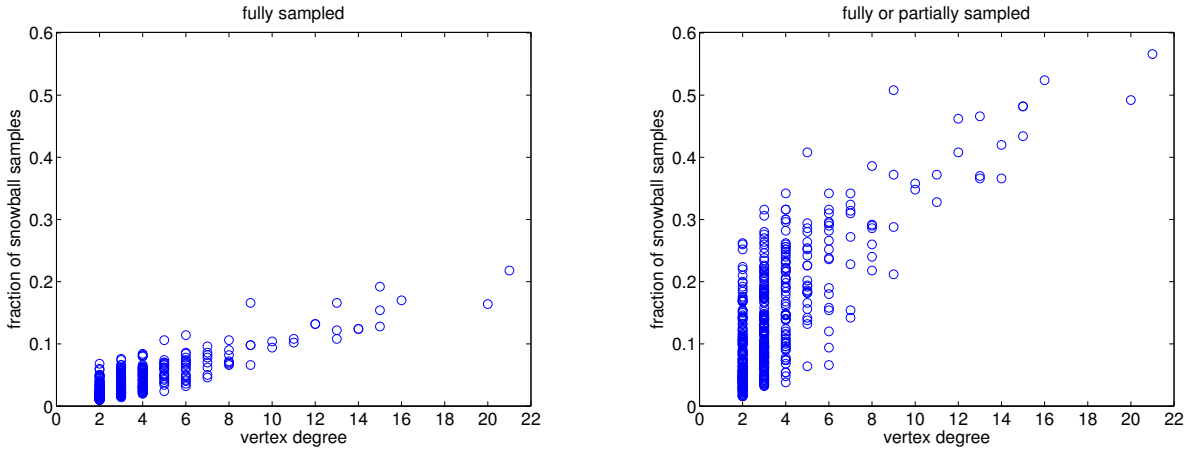


Figure 1: Fraction of times a node with degree k is (left) sampled (appears fully inside a snowball sample; not in the halo), or (right) partially or fully sampled (in halo or better) for uniformly randomly chosen seed nodes in the network in Section 1.3. Note that regardless of the criteria, higher-degree nodes are always better observed.

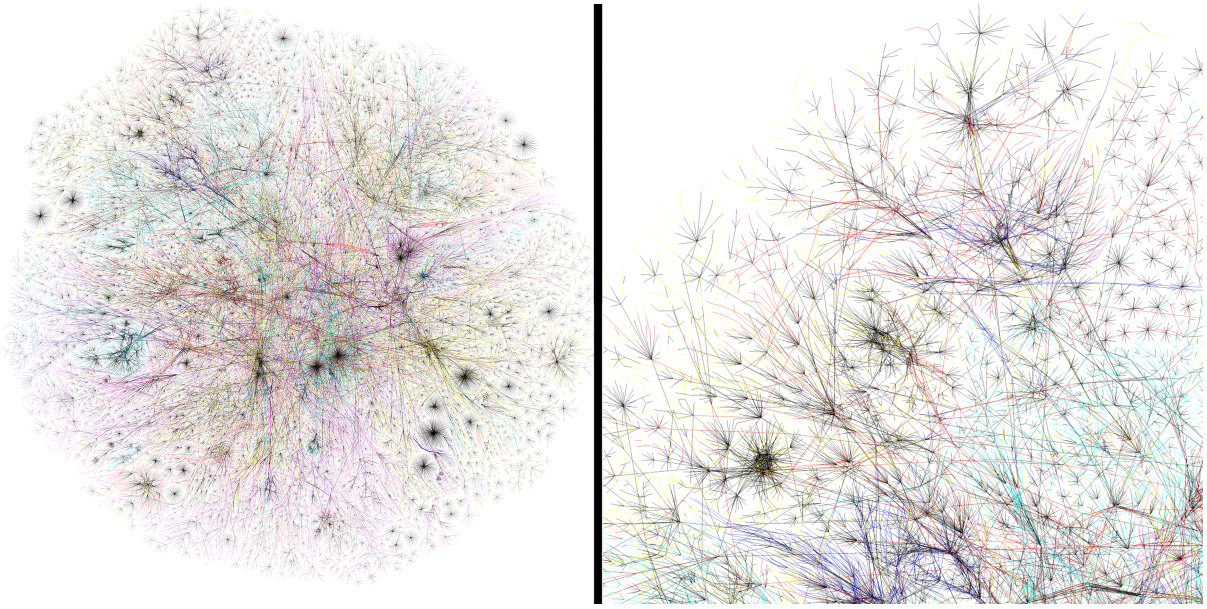
1.7 Edge sampling via trees

A fairly exotic form of sampling is to choose a seed vertex i, grow a tree T outward from it, usually a breadth-first search (BFS) tree, and then include in the sampled network only those edges contained in T. This form of sampling crops up whenever the method for exploring the network depends on following paths, such as geodesic ones, that emanate from some starting point.

This sampling approach is a reasonably good model for how the network utility `traceroute` works, which outputs the sequence of Internet Protocol (IP) addresses between your machine and some target IP address that represents a path on the IP network. If we run this procedure for all possible IP addresses, and take the union of these paths, the result is something very close to a BFS tree rooted at your machine. Because the Internet (at the IP level) is generally unknown and inaccessible, this is precisely how researchers have explored its structure. The following figure shows a nice visualization of the IP graph obtained by this procedure. On the left is the full Internet; the right shows a zoom of the top-left corner of the image.[5]

As it turns out, this type of sampling has a very well-known bias, which is that it tends to produce

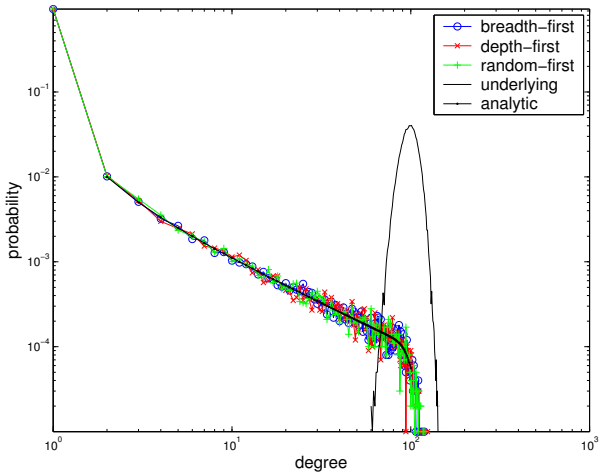
[5]Sadly, I used to know exactly who produced this figure, which appeared sometime around 2004, but both my memory and the website that hosted the project for producing this images has vanished. There is another, more famous but similar picture, due to Bill Cheswick and Hall Burch from Lumeta in the early 2000s.



sampled graphs that have power-law degree distributions, i.e., $p(k) \propto k^{-\alpha}$, or something very close to it, even when the degree distribution of the full graph is not remotely like a power law.[6] In fact, this behavior holds even for a k-regular random graph, where *every* vertex has degree k, as well as simple random graphs like the Erdős-Rényi model. For simple random graphs, the effect is even stronger: it doesn't matter whether you grow the tree breath-first, depth-first or even random-first, they all produce the same result, as the following figure illustrates. It shows simulation results on a $n = 10^5$ simple random graph, with $\langle k \rangle = 100$ so that the power law is completely obvious, since it only holds up to the mean degree.

Intuitively, the reason for this profound sampling bias is straightforward: only the root vertex of the tree T has its degree sampled faithfully; every other vertex j is found at some distance ℓ from i. We can think about the growth of T in terms of "shells" of vertices, each containing all the vertices a distance ℓ from i. Only edges that cross from vertices at some ℓ to some $\ell + 1$ are included in the tree; all edges that cross within some shell are unobserved. As ℓ increases, a greater fraction of all the edges are hidden from us, until we reach the leaves of the tree, each of which has degree exactly 1.

[6] See Clauset and Moore, "Accuracy and Scaling Phenomena in Internet Mapping." *Phys. Rev. Lett.* **94**, 018701 (2005), and Achlioptas, Clauset, Kempe and Moore, "On the Bias of Traceroute Sampling." *Journal of the ACM* **56**, article 21 (2009).



1.8 Other sampling approaches

There are many other approaches to sampling. The approaches described above include most of the commonly-used methods. In future versions of these lecture notes, I'll expand this section to describe a few others, including respondent-driven sampling[7] (which is a form of adaptive sampling), and attribute-proportional sampling.

Another place sampling in networks crops up is in A/B testing in the online social networks like Facebook, where the goal is to choose two sample populations, and give each population a slightly different product experience. The difficulty here is that in networks, the behavior of one person may depend on how many of their friends are in the same treatment group as them, so a uniform partitioning is likely to produce poor results.[8]

1.9 Inverting a sampling

Suppose we obtain a network sample with degree sequence $n_{k'}$ and we know the manner in which it was sampled (e.g., a uniformly vertex sample) and we know the size n of the full network. A reasonable question is whether we can invert the sampling procedure to recover the true degree sequence, or some other property of the full network. Perhaps surprisingly, the answer to this question is not known for uniform vertex sampling. In fact, for exactly none of the sampling procedures described in these notes is the answer to this question known!

[7] For example, see Goel and Salganik "Assessing respondent-driven sampling." *PNAS*, **107**, 6743-6747 (2010).

[8] For more on this subject, see Ugander, Karrer, Backstrom, and Kleinberg, "Graph cluster randomization: network exposure to multiple universes." *KDD* (2013), available here `http://arxiv.org/abs/1305.6979`.

A general solution would imply a bijection between the degree sequence of a full network and that of its network sample for some particular sampling approach and parameters. Given the past work on sampling, it seems likely that no such bijection exists in general, and instead a particular sampled degree sequence can be produced by many full degree sequences.

Of course, because we have thrown out many edges as a result of any particular sampling approach, we have altered more than just the degree sequence. Sampling a network will also change the clustering coefficient, the geodesic path-length structure, etc. It is perhaps surprising here too that the impact of sampling on these vertex-level and network-level is not known in general. An interesting question is whether there exists a sufficient set of measures of network structure that we could solve the inversion problem, i.e., construct a bijection.

Supplemental readings

1. Thomas and Blitzstein, "Valued Ties Tell Fewer Lies: Why Not To Dichotomize Network Edges With Thresholds." Preprint, arXiv:1101.0788 (2011)
`http://arxiv.org/abs/1101.0788`

2. Maiya and Berger-Wolf, "Benefits of Bias: Towards Better Characterization of Network Sampling." *Proc. KDD '11*, 105–113 (2011)
`https://doi.org/10.1145/2020408.2020431`

3. Kurant et al., "Walking on a Graph with a Magnifying Glass." *Proc. SIGMETRICS '11*, 281–292 (2011)
`https://doi.org/10.1145/1993744.1993773`