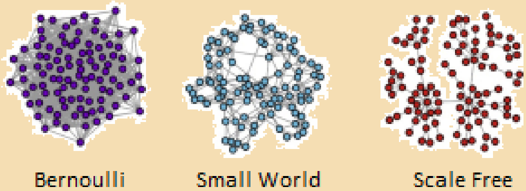
Network models

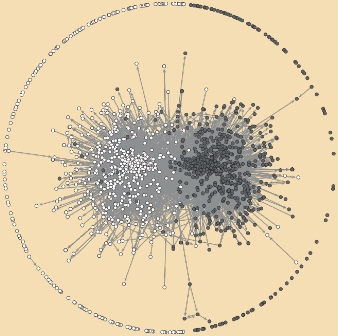


Andrej Mrvar

Example Network...

Political blogosphere, United States, February 8, 2005

Network of hyperlinks between 1,490 political blogs in February 2005 discussing the 2004 election in the United States, compiled by L.A. Adamic and N. Glance. An arc between two blogs represents a reference in a blogroll or in a post on the blog's front page.



We will use a simplified - undirected version of this network and a partition into liberal and conservative blogs (**Political-blogs.paj**).

Partition **Political_leaning** indicates political leaning of the blog: 0 (left or liberal), 1 (right or conservative).

...Example Network...

Political blogosphere, United States, February 8, 2005

Draw network with **Pivot MDS** (the fastest automatic drawing), compute $E - I$ index (network structure can be explained by left-right division).

Some properties of the blogosphere network:

- number of weakly connected components (WCC) = 268
- size of the largest WCC = 1222 vertices (82%)
- average degree = 22.436
- diameter = 8
- average distance among reachable pairs = 2.738
- clustering coefficient (transitivity) = 0.226
- degree centralization = 0.221
- betweenness centralization = 0.065
- E-I Index = -0.81

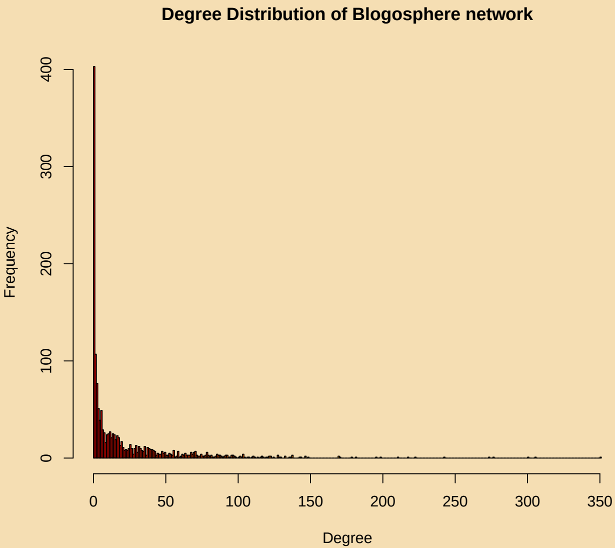
Average distance is computed using the following command:

Network / Create Vector / Distribution of Distances*

It gives distribution of lengths of the shortest paths (distances) and average path length (average distance) among all reachable pairs of vertices.

...Example Network...

Political blogosphere, United States, February 8, 2005



...Example Network

Histogram of degree distribution in R

After computing degrees: **Network** / **Create Vector** / **Centrality** / **Degree** we can draw degree distrubution in R:

Tools / **R** / **Locate R** // R needs to be located only before the first call
Tools / **R** / **Send to R** / **Current Vector**
R: hist(v??, breaks= 0:???)

Details: Picture of degree distrubution of the Blogosphere network was obtained using some additional parameters for drawing histograms in R:

```
par(bg="wheat")
hist(v1, breaks=0:max(v1),
col=terrain.colors(max(v1)),
main="Degree Distribution of Blogosphere network",
xlab="Degree", right=FALSE)
```

Similarly we can export Pajek objects to *Excel* (select *Histogram* as the "*Chart Type*") and *SPSS*.

Bernoulli model

The most common implementation of random model assigns a line to each pair of vertices independently, with a fixed probability. Independence means that the probability that a pair of vertices is connected by a line is independent from the presence or absence of lines among other pairs.

Network / Create Random Network / Bernoulli/Poisson

The original Erdős–Rényi model fixes the exact number of arcs (instead of probability of an arc), so this characteristic does not vary among random graphs; it is a condition just like the number of vertices. Result is always a directed network.

Network / Create Random Network / Total No. of Arcs

The two models are very unlikely to appear in social networks. In the two models *individual does not care 'whom (s)he will select as a friend'*.

Some properties of Bernoulli networks

Weakly connected components:

- An average degree over 1 produces with high probability a graph containing one large weakly connected component (WCC) while all other components are about equally small.
- The size of the large component grows with both the size of the graph and the average degree; with average degree 1.5, it is already expected to contain over 50% of all vertices. We call it a *giant component*.
- In contrast, average degree below 1 is expected to create random graphs containing only small components.

Diameter of such networks is relativelly small, a rough estimate is:

$$\text{Diameter}_{\text{expected}} = \frac{\ln(n)}{\ln(c)}$$

n is number of vertices, and c average degree.

In a Bernoulli random graph **Clustering coefficient - transitivity** (the expected proportion of transitive triples over connected triples) is:

$$\mathrm{CC}_{\mathrm{expected}} = \frac{c}{n-1}$$

which tends toward zero in larger sparse networks.

An obvious shortcoming of the Bernoulli random graph model is its low and network-size-dependent clustering. We need models yielding random graphs of any size with clustering coefficients in the range usually found in social networks, roughly put between 0.05 and 0.50.

The next model which we will describe will solve this problem.

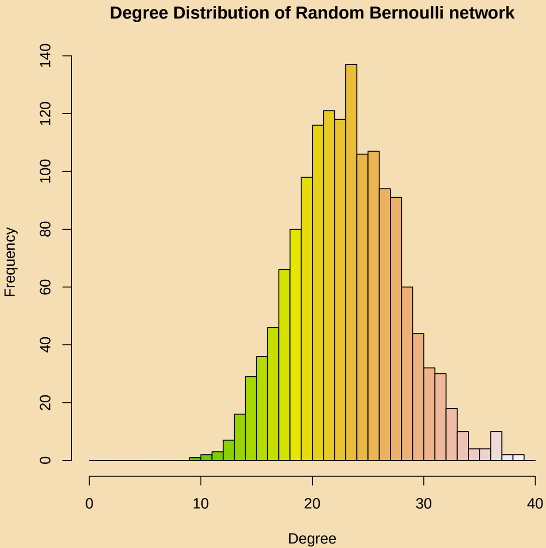
For undirected Bernoulli network with the same number of vertices (1490) and average degree (22.436) as we have in blogosphere network, we get the following expected values:

$$\text{Diameter}_{\text{expected}} = \frac{\ln(n)}{\ln(c)} = \frac{\ln(1490)}{\ln(22.436)} = 2.4$$

$$\text{CC}_{\text{expected}} = \frac{c}{n-1} = \frac{22.436}{1490-1} = 0.015$$

The observed diameter (8) is approx. three times larger than the one expected in Bernoulli network (2.4).
The observed transitivity (0.226) is more than ten times larger than the one expected in Bernoulli network (0.015).

Generate some random Bernoulli networks with the the same number of vertices (1490) and average degree (22.436) as we have in the blogs network and check diameter and clustering coefficients which you obtain. Are the obtained diameter and clustering coefficient close to expected values? What is the number of WCC and the size of the largest WCC?



Degree Distribution of Random Bernoulli network
Frequency
0
20
40
60
80
100
120
140
0
10
20
30
40
Degree

...Standard network models...

Degree conditional Bernoulli model...

We can store degrees in a partition and request to generate a random Bernoulli network where vertices will have the same degrees as defined in a partition.
The partition also determines the number of vertices in the random graph, so the user need not specify this property.

Partition / Make Network / Random Network

After executing this command the obtained network might containe loops and multiple lines, so we usually remove them:

Network / Create New Network / Transform / Remove / Loops

Network / Create New Network / Transform / Remove / Multiple Lines / Single Line

Degrees can be provided manually for each vertex, or we can compute degrees for a given network:

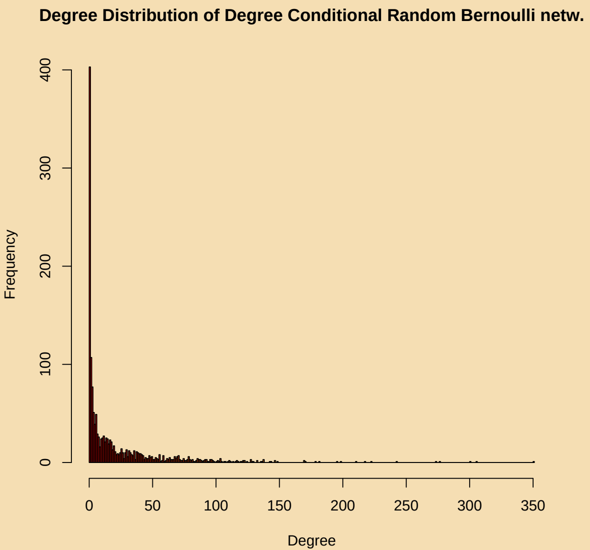
Network / Create Partition / Degree

and generate random network with the same degree distribution:

Partition / Make Network / Random Network

In the obtained network all vertices will have the same degrees as vertices in the original network, but lines will be distributed differently.

Generate degree conditional Bernoulli network with the same degree distribution as Blogoshpere network. Do not forget to remove loops and multiple lines. Compare structural features of the obtained random network with the Blogosphere network. Which structural features of both networks are equal, similar, which are different?


Degree Distribution of Degree Conditional Random Bernoulli netw.
Frequency
0
100
200
300
400
Degree
0
50
100
150
200
250
300
350

A psychologist Stanley Milgram made the following experiment with letters in 1967: The letter should reach a target person. The persons involved in experiment were asked to send the letter with these instructions to the target person (if they personally know him/her) or (if they do not know him/her personally) to their friend who was more likely to know the target. Letters were sent from Omaha (Nebraska) to target person in Boston (Massachusetts). The average length of the successful paths was 6: *six degrees of separation*.

`https://en.wikipedia.org/wiki/Small-world_experiment`

The average path length on the internet is 19 clicks.

Networks in which the *average shortest path length* is small are called **Small Worlds**.

Small world (Watts-Strogatz model): Indivuduals are linked to those who live in the neighbourhood and some others further away.

The original small-world model puts all vertices on a circle and connects each vertex to a fixed number of its neighbors in spatial sense: the vertices nearest in the plane. If the number of connected neighbors exceeds 2, triangles appear because each vertex is linked to its neighbor and its neighbor's neighbor.

The expected clustering coefficient (network transitivity) is determined only by the number of neighbors each vertex is linked to at each side (r) and it is easy to calculate:

$$\mathrm{CC}_{\mathrm{expected}} = \frac{3r-3}{4r-2}$$

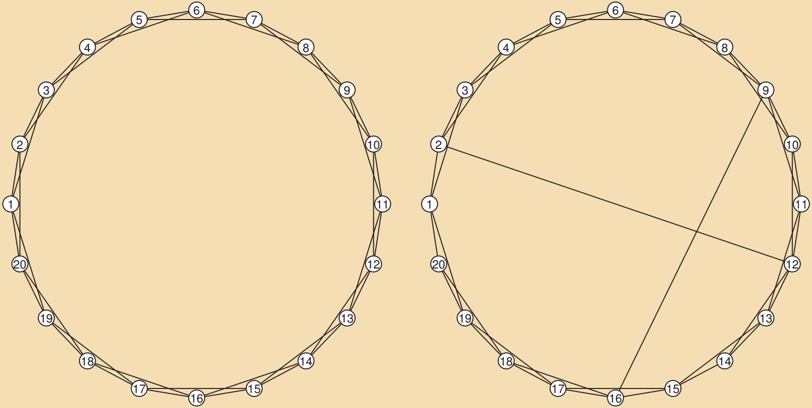
If the number of neighbors on either side is set to 1, the clustering coefficient reaches its minimum 0, whereas the coefficient tends toward .75 for a very large number of neighbors.

This approach, however, raises a new problem: A large graph containing only local lines has *an average path distance* much higher than the ones encountered in social networks, whereas the small-world phenomenon argues that even in the network containing the entire world population, people are reachable in a maximum of six steps.

This problem is solved by replacing one endpoint of a small proportion of the local lines by a random vertex (*rewiring*). Rewiring as little as 1 to 10 percent of the local lines suffices to obtain the small-world phenomenon of *low average path distance*. A low proportion of rewired lines does not change the *density* and *average degree* of the graph. It hardly changes its clustering, so the high *clustering characteristic* of social networks is preserved.

...Standard network models...

...Small world...



Small world model is built using:

Network / Create Random Network / Small World

The average degree in the blogs network is 22.4, so we can compare this network with a small-world random graph with vertices linked to the **eleven** nearest neighbors ($r = 22.436/2$), which has about the same average degree. The expected value of the clustering coefficient for this random graph is

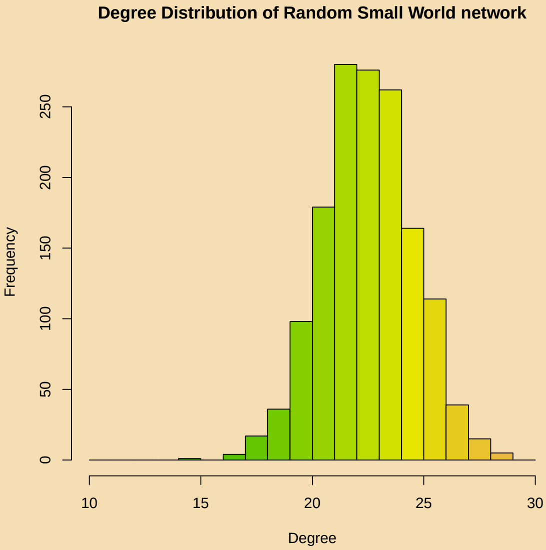
$$\mathrm{CC}_{\mathrm{expected}} = \frac{3r-3}{4r-2} = \frac{3*11-3}{4*11-2} = 0.71$$

This is in the limit of no rewiring (0.75). The observed value (0.226) is quite a bit lower, the reason probably being that vertex degree is not more or less equal for all vertices as assumed by the small-world random model; instead, it is highly skewed.

Generate some random Small world networks with the same number of vertices (1490) and **eleven** nearest neighbors as we have in blogosphere network. Set probability of rewiring somewhere in range 0.01 to 0.1. Check clustering coefficients which you obtain. Is the obtained clustering coefficient close to expected values?

...Standard network models...

...Small world



...Standard network models...

Preferential attachment - Scale free...

All models mentioned till now still do not have characteristics of real social networks. Such networks usually have *some vertices with very high degree*.
We solve this problem by simply assuming that vertices prefer to link to vertices with higher degree (*preferential attachment*).
This is a network variant of popular sayings such as "*the rich get richer*".

Some important properties of such networks:

- The **degree distribution** is *highly skewed to the right*. Since degree distribution of these networks has no natural scale they are called *scale free* networks.
- The **average distance** is very small: $O(\log n)$.
- Such networks are **robust** against random vertex or link removals (random attacks), but quickly become disconnected when vertices with large degree (Achilles' heel) are removed (targeted attacks).

...Standard network models...

...Preferential attachment - Scale free...

Most real life networks have characteristics of *scale free networks*. Some examples:

- social networks
  - collaboration networks: coauthorship networks, movie actors (playing in the same movie)...
  - personal networks: e-mails, phone calls, sexual contacts (drug users, AIDS)...
- computer networks (Internet, WWW);
- financial networks (interbank payment networks);
- protein-protein interaction networks;
- semantic networks;
- airline networks.

Network / Create Random Network / Scale Free

In each step of building scale free network we add a new vertex and some (average degree) lines.

Probability that vertex v is selected as the second endpoint of the new line is determined by ($|E|$ is number of lines):

$$\Pr(v) = \alpha \frac{\text{indeg}(v)}{|E|} + \beta \frac{\text{outdeg}(v)}{|E|} + \gamma \frac{1}{|V|}$$

where $\alpha + \beta + \gamma = 1$.

If *Adding=Free* is **checked** (free adding of lines) then both endpoints of the new line will be selected at random and vertices may remain isolated in the resulting random network.

If *Adding=Free* is **not checked** then in each step the new added vertex will be connected to one of already existing vertices.

For undirected network we select only α and set $\beta = \alpha$.

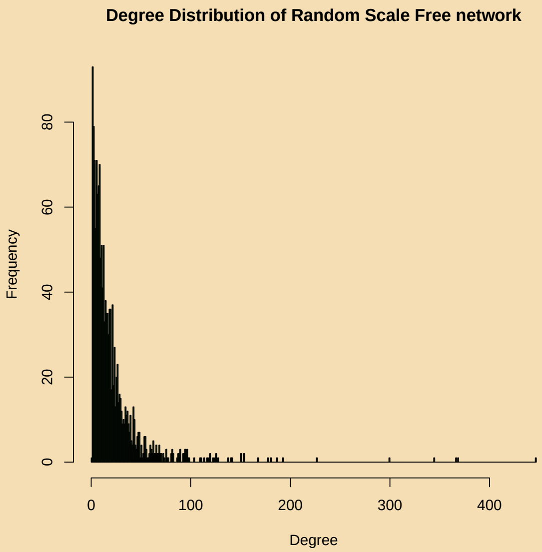
If $\alpha = 0.5$, then $\alpha + \beta = 1$: we always draw vertex proportional to to its degree: the higher the degree of vertex, the larger probability that this vertex will be selected as the second endpoint of the line.
If $\alpha = 0.25$ then $\alpha + \beta = 0.5$: a vertex is drawn with 0.5 probability to its degree and 0.5 uniformly from all vertices (like in Bernoulli model).

Random graphs generated by the Scale Free commands may contain multiple lines. If these are not desired – for example, because the clustering coefficient cannot handle multiple lines – they should be removed with: **Network** / **Create New Network** / **Transform** / **Remove** / **Multiple Lines** / **Single Line**.

Generate some random Scale free networks (and remove multiple lines) with the same number of vertices (1490) and average degree (22.436) as we have in blogosphere network.

For α (probability that vertex will be drawn proportionally to its degree) try different values in interval 0..0.5 (note that this number is multiplied by 2 for undirected networks).

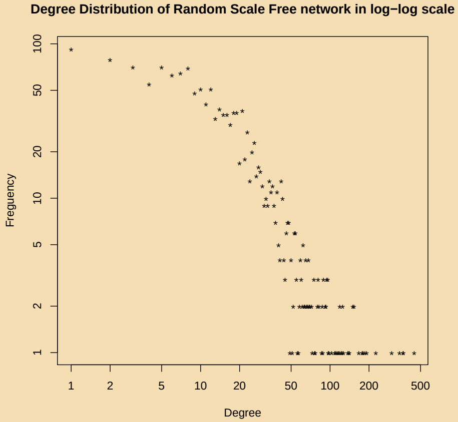
Try random generator with option:
Network / **Create Rand. Net.** / **Scale Free** / **Adding** / **Free**
once checked and once unchecked.
What difference do you observe (e.g. in number of weakly connected components)?


Degree Distribution of Random Scale Free network
Frequency
0
20
40
60
80
0
100
200
300
400
Degree

...Standard network models...

...Preferential attachment - Scale free

Usually we draw degree distributions of scale free networks in a log-log scale where it looks linear.




```
deg <- tabulate(v1,nbins=max(v1))
plot(deg,log="xy", main="Degree Distribution of Scale Free
network in log-log scale", ylab="Freguency", xlab="Degree")
```

...Standard network models...

Overview

For the following networks:

- Blogosphere network
- Bernoulli network ($n = 1490, \overline{\deg} = 22.436$)
- Small world ($n = 1490, r = 11, p_{rew} = 0.1$)
- Small world ($n = 1490, r = 11, p_{rew} = 0.2$)
- Scale free ($n = 1490, \overline{\deg} = 22.436, \alpha = 0.25$)
- Scale free ($n = 1490, \overline{\deg} = 22.436, \alpha = 0.25, Add. = Free$)

compute the following properties:

- number of weakly connected components
- size of the largest weakly connected component
- diameter
- average distance
- clustering coefficient (transitivity)
- betweenness centralization

Write results in a table and compare them. For each property find out which of the random networks is the most similar to Blogosphere network.

Monte Carlo simulations...

Idea

To be sure about obtained parameters we have to generate more (e.g. 100) such random networks and take the average value:

First generate one network of selected type:
Network / **Create New Network** / **Random Network** / **...**

Next generate 99 more such networks (to get 100 networks):
Macro / **Repeat Last Command (F10)** 99

Examine info for the first obtained network:
Network / **Info** / **General**

Examine info for the rest 99:
Macro / **Repeat Last Command (F10)** 99

Result is also a new vector, which presents aggregate results for all 100 networks, e.g. vector of dimension 100, where each cell presents average degree for each individual network.
Using **Vector** / **Info** we can examine descriptive statistics (*arithmetic mean, median, standard deviation, quantiles*) for selected property (e.g. average degree). Similarly we can make an overview for other network properties (e.g. average distance, diameter, transitivity...).

Repeat the overview assigment but instead of generating only one random network of selected type, generate 100 such networks and provide the aggregate results. Report 2.5 and 97.5 percentile for each property.

Results:

	Blogs	Bernoulli		Small world $p_{rew} = 0.2$		Scale Free Adding = Free	
		2.5%	97.5%	2.5%	97.5%	2.5%	97.5%
# Components	268	1	1	1	1	96	134
% Largest com.	82	100	100	100	100	91	94
Diameter	8	4	4	4	5	7	9
Av. distance	2.74	2.61	2.63	2.98	3.01	3.07	3.14
Transitivity	.226	.017	.018	.355	.372	.095	.107
Betweenness	.065	.002	.003	.003	.004	.038	.064

...Monte Carlo simulations

Interpreting results

It is difficult to decide for one random graph model:

- With respect to *diameter*, the blogs network is most similar to the *preferential attachment* model.
- Its *average distance* among vertices is closest to the *Bernoulli* random graph model, although the observed average distance (2.74) is outside the 95% confidence interval (2.61; 2.63) for this model.
- The blogs network is more *transitive* than *Bernoulli* and *preferential attachment* models. It is closest to the *small-world* model. The small-world model that we used, however, is too transitive, so more lines must be rewired than 20% used here.
- According to the *betweenness centralization* of the network, the *preferential attachment* random graph model yields centralization scores that are closest to the centralization of the political blogs network, but centralization is still too low. We should use $\alpha + \beta$ larger than 0.5.