Statistical Inference for Networks

Systems Biology Doctoral Training Centre

Theoretical Systems Biology Module

HILARY TERM 2008

PROF. GESINE REINERT

http://www.stats.ox.ac.uk/ ~reinert

WITH PAO-YANG CHEN

http://www.stats.ox.ac.uk/ ~chen

AND WAQAR ALI

Overview Lecture 1: *Network summaries.* What are networks? Some examples from social science and from biology. The need to summarise networks. Clustering coefficient, degree distribution, shortest path length, motifs, between-ness, second-order summaries. Roles in networks, derived from these summary statistics, and modules in networks. Directed and weighted networks. The choice of summary should depend on the research question.

Lecture 2: *Models of random networks.* Models would provide further insight into the network structure. Classical Erds-Renyi (Bernoulli) random graphs and their random mixtures, Watts-Strogatz small worlds and the modification by Newman, Barabasi-Albert scale-free networks, exponential random graph models.

Lecture 3: *Fitting a model: parametric methods.* Deriving the distribution of summary statistics. Parametric tests based on the theoretical distribution of the summary statistics (only available for some of the models).

Lecture 4: *Statistical tests for model fit: nonparametric methods.* Quantile-quantile plots and other visual methods. Monte-Carlo tests based on shuffling edges with the number of edges fixed, or fixing the node degree distribution, or fixing some other summary. The particular issue of testing for power-law dependence. Subsampling issues. Tests carried out on the same network are not independent.

Lecture 5: *Statistical inference for networks: local properties.* Inferring characteristics for a missing node from the existing network. Log-linear regression models. Inferring missing edges and identifying false-positive edges. Logistic regression models.

Lecture 6: *Statistical inference for networks: modules, motifs and roles.* Identifying similar edges in networks. Clustering algorithms. Comparison of networks: two networks on the same set of nodes. Regression models.

Lecture 7: *Further topics.* Hierarchical networks. Dynamics on networks.

Suggested reading

1. U. Alon: *An Introduction to Systems Biology Design Principles of Biological Circuits.* Chapman and Hall 2007.
2. S.N. Dorogovtsev and J.F.F. Mendes: *Evolution of Networks*. Oxford University Press 2003.
3. R. Durrett: *Random Graph Dynamics*. Cambridge University Press 2007.
4. R. Gentleman, V. Carey, W. Huber, R. Irizarry, S. Dutoit (eds). *Bioinformatics and Computational Biology Solutions using R and Bioconductor*. Springer 2005.
5. W. de Nooy, A. Mrvar and V. Bagatelj. *Exploratory Social Network Analysis with Pajek.* Cambridge University Press 2005.
6. S. Wasserman and K. Faust: *Social Network Analysis*. Cambridge

University Press 1994.

7. D. Watts: *Small Worlds*. Princeton University Press 1999.

This part of the module will take place Wednesday 27 February, Monday 10 March, and Friday 14 March, from 9:30 - 12 and 2-5 in the DTC.

The teaching will be a mixture of lectures, worked examples, and computer exercises.

We shall use the `R` language in connection with `Bioconductor`. Both of these are open source.

Lecture notes will be published at

`http://www.stats.ox.ac.uk/ reinert/dtc/networks.html`.

The notes may cover more material than the lectures. The notes may be updated throughout the course.

The statistical analysis of networks is a very complex topic, far beyond what could be covered in 3-day course. Hence the goal of the class is to give a brief overview of the basics, highlighting some of the issues to be addressed.

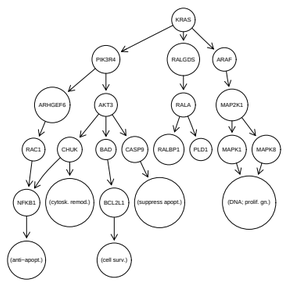
1 Network summaries

1.1 What are networks?

Networks are just graphs. Often one would think of a network as a connected graph, but not always. In this lecture course we shall use *network* and *graph* interchangeably.

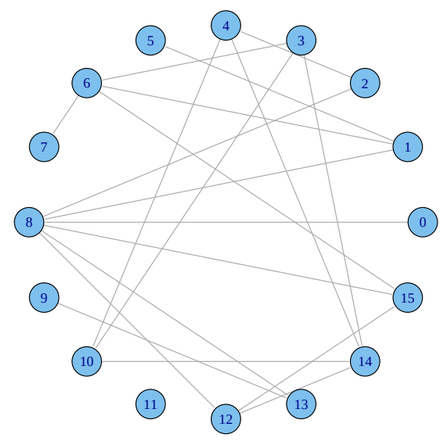
Here are some examples of networks (graphs).

`MAPK`: Pancreatic pathway



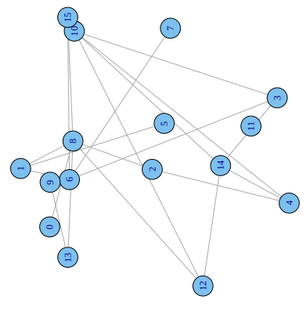
This graph shows part of the KEGG pancreatic cancer model, surrounding the MAPK signaling pathway.

`Flo`: Florentine Families



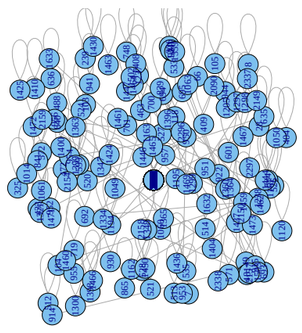
Marriage relations between Florentine families, in the order
0 ACCIAIUOL,
1 ALBIZZI,
2 BARBADORI,
3 BISCHERI,
4 CASTELLAN,
5 GINORI,
6 GUADAGNI,
7 LAMBERTES,
8 MEDICI,
9 PAZZI,
10 PERUZZI,
11 PUCCI,
12 RIDOLFI,
13 SALVIATI,

14 STROZZI,
15 TORNABUON.
The Medici beat their arch-rivals, the Strozzi.



Marriage relations between Florentine families: different drawing program.

`Yeast`: A plot of a connected subset of Yeast protein interactions.



Networks arise in a multitude of contexts, such as

- metabolic networks
- protein-protein interaction networks
- spread of epidemics
- neural network of *C. elegans*
- social networks
- collaboration networks (Erdös numbers ...)
- Membership of management boards
- World Wide Web
- power grid of the Western US

The study of networks has a long tradition in social science, where it is called *Social Network Analysis*. The networks under consideration are typically fairly small. In contrast, starting at around 1997, statistical physicists have turned their attention to large-scale properties of networks. Our lectures will try to get a glimpse on both approaches.

Typical networks in systems biology are

- Metabolic networks
- Gene interaction networks
- Protein interaction networks.

Research questions include

- How do these networks work? Where could we best manipulate a network in order to prevent, say, tumor growth?
- How did these biological networks evolve? Could mutation affect whole parts of the network at once?
- How similar are these networks? If we study some organisms very well, how much does that tell us about other organisms?
- How are these networks interlinked? Can we infer information from gene interaction networks that would be helpful for protein interaction networks?
- What are the building principles of these networks? How is resilience achieved, and how is flexibility achieved? Could we learn from biological networks to build man-made efficient networks?

From a statistical viewpoint, questions include

- How to best describe networks?
- How to infer characteristics of nodes in the network?
- How to infer missing links, and how to check whether existing links are not false positives
- How to compare networks from related organisms?
- How to predict functions from networks?
- How to find relevant sub-structures of a network?

Statistical inference relies on the assumption that there is some randomness in the data. Before we turn our attention to modelling such randomness, let’s look at how to describe networks, or graphs, in general.

1.2 What are graphs?

A *graph* consists of *nodes* (sometimes also called *vertices*) and *edges* (sometimes also called *links*). We typically think of the nodes as actors, or proteins, or genes, or metabolites, and we think of an edge as an interaction between the two nodes at either end of the edge. Sometimes nodes may possess characteristics which are of interest (such as structure of a protein, or function of a protein). Edges may possess different weights, depending on the strength of the interaction. For now we just assume that all edges have the same weight, which we set as 1.

Mathematically, we abbreviate a graph G as $G = (V, E)$, where V is the set of nodes and E is the set of edges. We use the notation $|S|$ to denote the number of elements in the set S. Then $|V|$ is the number of nodes, and $|E|$ is the number of edges in the graph G. If u and v are two nodes and there is an edge from u to v, then we write that $(u, v) \in E$, and we say that v is a *neighbour* of u.

If both endpoints of an edge are the same, then the edge is a *loop*. For now we exclude self-loops, as well as multiple edges between two nodes.

Edges may be *directed* or *undirected*. A *directed graph*, or *digraph*, is a graph where all edges are directed. The *underlying* graph of a digraph is the graph that results from turning all directed edges into undirected edges. Here we shall mainly deal with undirected graphs.

Two nodes are called *adjacent* if they are joined by an edge. A graph can be described by its *adjacency matrix* $A = (a_{u,v})$. This is a square $|V| \times |V|$ matrix. Each entry is either 0 or 1;

$$a_{u,v} = 1 \text{ if and only if } (u, v) \in E.$$

As we assume that there are no self-loops, all elements on the diagonal of the adjacency matrix are 0. If the edges of the graph are undirected, then the adjacency matrix will be symmetric.

The adjacency matrix entries tell us for every node v which nodes are within distance 1 of v. If we take the matrix produce $A^2 = A \times A$, the entry for (u, v) with $u \neq v$ would be

$$a^{(2)}(u, v) = \sum_{w \in V} a_{u,w} a_{w,v}.$$

If $a^{(2)}(u, v) \neq 0$ then u can be reached from v within two steps; u is within distance 2 of v. Higher powers can be interpreted similarly.

Example: Adjacency matrix for Florentine marriages

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0
\end{bmatrix}$$

A *complete* graph is a graph such that every pair of nodes is joined by an edge. The adjacency matrix has entry 0 on the diagonal, and 1 everywhere else.

A *bipartite* graph is a graph where the node set V is decomposed into two disjoint subsets, U and W, say, such that there are no edges between any two nodes in U, and also there are no edges between any two nodes in W; all edges have one endpoint in U and the other endpoint in W. An example is a network of co-authorship and articles; U could be the set of authors, W the set of articles, and an author is connected to an article by an edge if the author is a co-author of that article. The adjacency matrix A can then be arranged such that it is of the form

$$\begin{bmatrix} 0 & A_1 \\ A_2 & 0 \end{bmatrix}.$$

1.3 Network summaries

The *degree* $deg(v)$ of a node v is the number of edges which involve v as an endpoint. The degree is easily calculated from the adjacency matrix A;

$$deg(v) = \sum_u a_{u,v}.$$

The *average degree* of a graph is then the average of its node degrees.

(For directed graphs we would define the *in-degree* as the number of edges directed at the node, and the *out-degree* as the number of edges that go out from that node.)

The *clustering coefficient* of a node v is, intuitively, the proportion of its ”friends” who are friends themselves. Mathematically, it is the proportion of neighbours of v which are neighbours themselves. In adjacency matrix notation,

$$C(v) = \frac{\sum_{u,w \in V} a_{u,v} a_{w,v} a_{u,w}}{\sum_{u,w \in V} a_{u,v} a_{w,v}}.$$

The *(average) clustering coefficient* is defined as

$$C = \frac{1}{|V|} \sum_{v \in V} C(v).$$

Note that

$$\sum_{u,w \in V} a_{u,v} a_{w,v} a_{u,w}$$

is the number of triangles involving v in the graph. Similarly,

$$\sum_{u,w \in V} a_{u,v} a_{w,v}$$

is the number of *2-stars* centred around v in the graph. The clustering coefficient is thus the ratio between the number of triangles and the number of 2-stars. The clustering coefficient describes how ”locally dense” a graph is. Sometimes the clustering coefficient is also called the *transitivity*.

The clustering coefficient in the Florentine family example is 0.1914894; the average clustering coefficient in the Yeast data is 0.1023149.

Exercise 1:

- Draw an undirected complete graph on 6 nodes, and write down its adjacency matrix. Determine the degrees of the 6 nodes. What is the clustering coeffient?
- Draw two different undirected graphs on 6 nodes where each node has degree 2, and write down their adjacency matrices. What are their clustering coefficients? *Hint: a graph does not have to be connected.*

In a graph a *path* from node v_0 to node v_n is an alternating sequence of nodes and edges, $(v_0, e_1, v_1, e_2, \ldots, v_{n-1}, e_n, v_n)$ such that the endpoints of e_i are v_{i-1} and v_i, for $i = 1, \ldots, n$. A graph is called *connected* if there is a walk between any pair of nodes in the graph, otherwise it is called *disconnected*. The *distance* $\ell(u, v)$ between two nodes u and v is the length of the shortest path joining them. This path does not have to be unique.

We can calculate the distance $\ell(u, v)$ from the adjacency matrix A as the smallest power p of A such that the (u, v)-element of A^p is not zero.

In a connected graph, the *average shortest path length* is defined as

$$\ell = \frac{1}{|V|(|V| - 1)} \sum_{u \neq v \in V} \ell(u, v).$$

The average shortest path length describes how ”globally connected” a graph is.

Example: *H. Pylori* and Yeast protein interaction network comparison:

	n	ℓ	C
H.Pylori	686	4.137637	0.016
Yeast	2361	4.376182	0.1023149

Node degree, clustering coefficient, and shortest path length are the most common summaries of networks. Other popular summaries, to name but a few, are: the *between-ness of an edge* counts the proportion of shortest paths between any two nodes which pass through this edge. Similarly, the *between-ness of a node* is the proportion of shortest paths between any two nodes which pass through this node. The *connectivity* of a connected graph is the smallest number of edges whose removal results in a disconnected graph.

In addition to considering these general summary statistics, it has proven fruitful to describe networks in terms of *motifs*; these are building- block patterns of networks such as a feed-forward loop, see the book by *Alon*. Here we think of a motif as a subgraph with a fixed number of nodes and with a given topology. In biological networks, it turns out that motifs seem to be conserved across species. They seem to reflect functional units which combine to regulate the cellular behaviour as a whole.

The decomposition of *communities* in networks, small subgraphs which are highly connected but not so highly connected to the remaining graph, can reveal some structure of the network. Identifying *roles* in networks singles out specific nodes with special properties, such as hub nodes, which are nodes with high degree.

Looking at the ”spectral decomposition”, i.e. at eigenvectors and eigenvalues, of the adjacency matrix, provides another set of summaries, such as *centrality*.

The above network summaries provide an initial go at networks. Specific networks may require specific concepts. In protein interaction networks, for example, there is a difference whether a protein can interact with two other proteins simultaneously (party hub) or sequentially (date hub). In addition, the research question may suggest other summaries. For example, in fungal networks, there are hardly any triangles, so the clustering coefficient does not make much sense for these networks.

Excursion: Milgram and the *small world effect*.

In 1967 the American sociologist Milgram reported a series of experiments of the following type. A number of people from a remote US state (Nebraska, say) are asked to have a letter (or package) delivered to a certain person in Boston, Massachusetts (such as the wife of a divinity student). The catch is that the letter can only be sent to someone whom the current holder knew on a first-name basis. Milgram kept track of how many intermediaries were required until the letters arrived; he reported a median of six; see for example $http://www.uaf.edu/northern/big_world.html$. This made him coin the notion of *six degrees of separation*, often interpreted as everyone being six handshakes away from the President. While the experiments were somewhat flawed (in the first experiment only 3 letters arrived), the concept of *six degrees of separation* has stuck.

2 Models of random networks

In order to judge whether a network summary is ”unusual” or whether a motif is ”frequent”, there is an underlying assumption of randomness in the network.

Network data are subject to various errors, which can create randomness, such as

- There may be missing edges in the network. Perhaps a node was absent (social network) or has not been studied yet (protein interaction network).
- Some edges may be reported to be present, but that recording is a mistake. Depending on the method of determining protein interactions, the number of such *false positive* interactions can be substantial, of around 1/3 of all interactions.
- There may be transcription errors in the data.
- There may be bias in the data, some part of the network may have received higher attention than another part of the network.

Often network data are snapshots in time, while the network might undergo dynamical changes.

In order to understand mechanisms which could explain the formation of networks, mathematical models have been suggested.

2.1 Bernoulli (Erdös-Renyi) random graphs

The most standard random graph model is that of Erdös and Renyi (1959). The (finite) node set V is given, say $|V| = n$, and an edge between two nodes is present with probability p, independently of all other edges. As there are

$$\binom{n}{2} = \frac{n(n-1)}{2}$$

potential edges, the expected number of edges is then

$$\binom{n}{2} p.$$

Each node has $n-1$ potential neighbours, and each of these $n-1$ edges is present with probability p, and so the expected degree of a node is $(n-1)p$. As the expected degree of a node is the same for all nodes, the average degree is $(n-1)p$.

Similarly, the average number of triangles in the graph is

$$\binom{n}{3}p^3 = \frac{n(n-1)(n-2)}{6}p^3,$$

and the average number of 2-stars is

$$\binom{n}{3}p^2.$$

Thus, with a bit of handwaving, we would expect an average clustering coefficient of about

$$\frac{\binom{n}{3}p^3}{\binom{n}{3}p^2} = p.$$

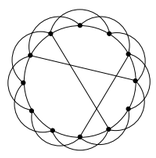
In a Bernoulli random graphs, your friends are no more likely to be friends themselves than would be a two complete strangers. This model is clearly not a good one for social networks. Below is an example from scientific collaboration networks (*N. Boccara, Modeling Complex Systems, Springer 2004, p.283*. We can estimate p as the fraction of average node degree and $n-1$; this estimate would also be an estimate of the clustering coefficient in a Bernoulli random graph.

Network	n	ave degree	C	$C_{Bernoulli}$
Los Alamos archive	52,909	9.7	0.43	0.00018
MEDLINE	1,520,251	18.1	0.066	0.000011
NCSTRL	11,994	3.59	0.496	0.0003

Also in real-world graphs often the shortest path length is much shorter than expected from a Bernoulli random graph with the same average node degree. The phenomenon of short paths, often coupled with high clustering coefficient, is called the *small world phenomenon*. Remember the Milgram experiments!

2.2 The Watts-Strogatz model

Watts and Strogatz (1998) published a ground-breaking paper with a new model for small worlds; the version currently most used is as follows. Arrange the n nodes of V on a lattice. Then hard-wire each node to its k nearest neighbours on each side on the lattice, where k is small. Thus there are nk edges in this hard-wired lattice. Now introduce random shortcuts between nodes which are not hard-wired; the shortcuts are chosen independently, all with the same probability.



If there are no shortcuts, then the average distance between two randomly chosen nodes is of the order n, the number of nodes. But as soon as there are just a few shortcuts, then the average distance between two randomly chosen nodes has an expectation of order $\log n$. Thinking of an epidemic on a graph - just a few shortcuts dramatically increase the speed at which the disease is spread.

It is possible to approximate the node degree distribution, the clustering coefficient, and the shortest path length reasonably well mathematically; we may come back to these approximations later.

While the Watts-Strogatz model is able to replicate a wide range of clustering coefficient and shortest path length simultaneously, it falls short of producing the observed types of node degree distributions. It is often observed that nodes tend to attach to ”popular” nodes; popularity is attractive.

2.3 ”The” Barabasi-Albert model

In 1999, Barabasi and Albert noticed that the actor collaboration graph adn the World Wide Web had degree distributions that were of the type

$$Prob(degree = k) \sim Ck^{-\gamma}$$

for $k \to \infty$. Such behaviour is called *power-law behaviour*; the constant γ is called the *power-law exponent*. Subsequently a number of networks have been identified which show this type of behaviour. They are also called *scale-free random graphs*. To explain this behaviour, Barabasi and Albert introduced the *preferential attachment* model for network growth.

Suppose that the process starts at time 1 with 2 nodes linked by m (parallel) edges. At every time $t \geq 2$ we add a new node with m edges that link the new node to nodes already present in the network. We assume that the probability π_i that the new node will be connected to a node i depends on the degree $deg(i)$ of i so that

$$\pi_i = \frac{deg(i)}{\sum_j deg(j)}.$$

To be precise, when we add a new node we will add edges one at a time, with the second and subsequent edges doing preferential attachment using the updated degrees.

This model has indeed the property that the degree distribution is approximately power law with exponent $\gamma = 3$. Other exponents can be achieved by varying the probability for choosing a given node.

Unfortunately the above construction will not result in any triangles at all. It is possible to modify the construction, adding more than one edge at a time, so that *any* distribution of triangles can be achieved.

2.4 Erdös-Renyi Mixture Graphs

An intermediate model with not quite so many degrees of freedom is given by the Erdös-Renyi mixture model, also known as *latent block models* in social science (*Nowicky and Snijders (2001)*). Here we assume that nodes are of different types, say, there are L different types. Then edges are constructed independently, such that the probability for an edge varies only depending on the type of the nodes at the endpoints of the edge. *Robin et al* have shown that this model is very flexible and is able to fit many real-world networks reasonably well. It does not produce a power-law degree distribution however.

Exercise 2: Consider an Erdös-Renyi mixture model with two types of nodes. Type 1, of which there are n_1 nodes, has edge probability p_1; whereas Type 2, of which there are n_2 nodes, has edge probability p_2; the edge probability for an edge between a Type-1 node and a Type-2 node is $p_{1,2}$. We are interested in the average node degree and the clustering coefficient.

- What should average node degree and the clustering coefficient be if $p_{1,2} = 0$? What if $p_{1,2} \neq 0$ but $p_1 = p_2 = 0$?
- What would the average node degree be in general?
- For the clustering coefficient, which types of triangles would you need to consider? What would you expect for the case $p_{1,2} = 0$? What if $p_{1,2} \neq 0$ but $p_1 = p_2 = 0$?

2.5 Exponential random graph (p^*) models

Networks have been analysed for ”ages” in the social science literature, see for example the book by *Wasserman and Faust*. Here usually digraphs are studied, and the research questions are different from biological networks. Typical research questions could be

- Is there a tendency in friendship towards transitivity; are friends of friends my friends?
- What is the role of explanatory variables such as income on the position in the network?
- What is the role of friendship in creating behaviour (such as smoking)?
- Is there a hierarchy in teh network?
- Is the network influenced by other networks for which the membership overlaps?

Exponential random graph (p^*) models model the whole adjacency matrix of a graph simultaneously, making it easy to incorporate dependence. Suppose that $\mathbf{X}$ is our random adjacency matrix. The general form of the model is

$$Prob(\mathbf{X} = \mathbf{x}) = \frac{1}{\kappa} \exp\{\sum_{B} \lambda_B z_B(\mathbf{x})\},$$

where the summation is over all subsets B of the set of potential edges,

$$z_B(\mathbf{x}) = \prod_{(i,j) \in B} x_{i,j}$$

is the network statistic corresponding to the subset B, κ is a normalising quantity so that the probabilities sum to 1, and the parameter $\lambda_B = 0$ for all $\mathbf{x}$ unless all the the variables in B are mutually dependent.

The simplest such model is that the probability of any edge is constant across all possible edges, i.e. the Bernoulli graph, for twhich

$$Prob(\mathbf{X} = \mathbf{x}) = \frac{1}{\kappa} \exp\{\lambda L(\mathbf{x})\},$$

where $L(\mathbf{x})$ is the number of edges in the network $\mathbf{x}$ and λ is a parameter.

For social networks, Frank and Strauss (1986) introduced *Markov dependence*, whereby two possible edges are assumed to be conditionally dependent if they share a node. For non-directed networks, the resulting model has parameters relating only to the configurations *stars of various types, and triangles*. If the number $L(\mathbf{x})$ of edges, the number $S_2(\mathbf{x})$ of two-stars, the number $S_3(\mathbf{x})$ of three-stars, and the number $T(\mathbf{x})$ of triangles are included, then the model reads

$$Prob(\mathbf{X} = \mathbf{x}) = \frac{1}{\kappa} \exp\{\lambda_1 L(\mathbf{x}) + \lambda_2 S_2(\mathbf{x}) + \lambda_3 S_3(\mathbf{x}) + \lambda_4 T(\mathbf{x})\}.$$

By setting the parameters to particular values and then simulating the distribution, we can examine global properties of the network.

2.6 Specific models for specific networks

Depending on the research question, it may make sense to build a specific network model. For example, a gene duplication model has been suggested which would result in a power-law like node degree distribution. For metabolic pathways, a number of Markov models have been introduced. When thinking of flows through networks, it may be a good idea to use weighted networks; the weights could themselves be random.

Further references

J.-J. Daudin, F. Picard, S. Robin (2006). A mixture model for random graphs. Preprint.

O. Frank and D. Strauss (1986). Markov graphs. *Journal of the American Statistical Association* **81**, 832-842.

K. Nowicky and T. Snijders (2001). Estimation and Prediction for Stochastic Blockstructures. *Journal of the American Statistical Association* **455**, Vol. 96, pp. 1077-1087.

S. Milgram (1967). The small world problem. *Psychology Today* **2**, 60–67.

Recap and additions

Networks are complex, hence the need to find good summaries. The most common ones are *node degrees*, *clustering coefficient*, and *shortest path length*.

Addendum: summaries based on spectral properties of the adjacency matrix.

If λ_i are the eigenvalues of the adjacency matrix A, then the spectral density of the graph is defined as

$$\rho(\lambda) = \frac{1}{n} \sum_i \delta(\lambda - \lambda_i),$$

where $\delta(x)$ is the delta function. For Bernoulli random graphs, if p is constant as $n \to \infty$, then $\rho(\lambda)$ converges to a semicircle.

The eigenvalues can be used to compute the kth moments,

$$M_k = \frac{1}{n}\sum_i (\lambda_i)^k = \frac{1}{n}\sum_{i_1,i_2,\ldots,i_k} a_{i_1,i_2}a_{i_2,i_3}\cdots a_{i_{k-1},i_k}.$$

The quantity nM_k is the number of paths returning to the same node in the graph, passing through k edges, where these paths may contain nodes that were already visited.

Because in a tree-like graph a return path is only possible going back through already visited nodes, the presence of odd moments is an indicator for the presence of cycles in the graph.

The *subgraph centrality*

$$Sc_i = \sum_{k=0}^{\infty} \frac{(A^k)_{i,i}}{k!}$$

measures the ”centrality” of a node based on the number of subgraphs in which the node takes part. It can be computed as

$$Sc_i = \sum_{j=1}^{n} v_j(i)^2 e^{\lambda_i},$$

where $v_j(i)$ is the ith element of the jth eigenvector.

Addendum: entropy-type summaries

The structure of a network is related to its reliability and speed of information propagation. If a random walk starts on node i going to node j, the probability that it goes through a given shortest path $\pi(i,j)$ between these vertices is

$$\mathcal{P}(\pi(i,j)) = \frac{1}{d(i)} \sum_{b \in \mathcal{N}(\pi(i,j))} \frac{1}{d(b)-1},$$

where $d(i)$ is the degree of node i, and $\mathcal{N}(\pi(i,j))$ is the set of nodes in the path $\pi(i,j)$ excluding i and j. The *search information* is the total information needed to identify one of all the shortest paths between i and j,

$$S(i,j) = -\log_2 \sum_{\pi(i,j)} \mathcal{P}(\pi(i,j)).$$

Similarly, an entropy can be defined based on the predictability of a

message flow.

Further reading:

L. da F. Costa, F.A. Rodrigues, P.R. Villas Boas, G. Travieso (2007).
Characterization of complex networks: a survey of measurements.
Advances in Physics 56, Issue 1 January 2007, 167 - 242.

Recap: network models

We looked at Bernoulli random graphs and their mixtures, Watts-Strogatz small worlds, Barabasi-Albert scale-free networks, and exponential random graphs. We saw that in these models the summaries are dependent. As an extreme case, knowing the degree sequence may already completely specify the network.

Specific networks may allow for specific modelling, and summaries may be chosen to best reflect the main features of the network.

3 Fitting a model: parametric methods

Parametric just means that we have a finite set of parameters which fully specify the model. For example:

Bernoulli (Erdös-Renyi) random graphs

In the random graph model of Erdös and Renyi (1959), the (finite) node set V is given, say $|V| = n$. We denote the set of all potential edges by E; thus $|E| = \binom{n}{2}$. An edge between two nodes is present with probability p, independently of all other edges. Here p is an unknown parameter.

3.1 Parameter estimation

In classical (frequentist) statistics we often estimate unknown parameters via the method of maximum likelihood.

The *likelihood* of the parameter given the data is just the probability of seeing the data we see, given the parameter.

Example: Bernoulli random graphs.

Our data is the network we see. We describe the data using the adjacency matrix, denote it by $\mathbf{x}$ here because it is the realisation of a random adjacency matrix $\mathbf{X}$. Recall that the adjacency matrix is the square $|V| \times |V|$ matrix where each entry is either 0 or 1;

$$x_{u,v} = 1 \text{ if and only if there is an edge between } u \text{ and } v.$$

The likelihood of p being the true value of the edge probability if we see $\mathbf{x}$ is

$$\mathcal{L}(p; \mathbf{x}) = \prod_{(i,j) \in E} \left\{ p^{x_{i,j}} (1-p)^{1-x_{i,j}} \right\}.$$

For example,

$$\begin{aligned}\mathcal{L}(0.5;\mathbf{x}) &= \prod_{(i,j)\in E}\left\{(0.5)^{x_{i,j}}(1-0.5)^{1-x_{i,j}}\right\} \\ &= \prod_{(i,j)\in E} 0.5 = 0.5^{|E|}.\end{aligned}$$

In general we can simplify

$$\begin{aligned}\mathcal{L}(p;\mathbf{x}) &= (1-p)^{|E|}\prod_{(i,j)\in E}\left(\frac{p}{1-p}\right)^{x_{i,j}} \\ &= (1-p)^{|E|}\left(\frac{p}{1-p}\right)^{\sum_{(i,j)\in E} x_{i,j}}.\end{aligned}$$

Note that $t = \sum_{(i,j)\in E} x_{i,j}$ is the total number of edges in the random graph.

To maximise the likelihood, we often take logs, and then differentiate. Here this would give

$$\begin{aligned} \ell(p;\mathbf{x}) &= \log \mathcal{L}(p;\mathbf{x}) \\ &= |E| \log(1-p) + t \log p - t \log(1-p); \end{aligned}$$

and

$$\frac{\partial \ell(p;\mathbf{x})}{\partial p} = -\frac{1}{1-p}(|E| - t) + \frac{t}{p}.$$

To find a maximum we equate this to zero and solve for p,

$$\begin{aligned} \frac{t}{p} &= \frac{1}{1-p}(|E|-t) \iff t(1-p) = p(|E|-t) \\ \iff t &= p|E| \iff p = \frac{t}{|E|}. \end{aligned}$$

We can check that the second derivative of ℓ is less than zero, so the fraction of edges that are present in the network,

$$\hat{p} = \frac{t}{|E|},$$

is our maximum-likelihood estimator.

Maximum-likelihood estimators have attractive properties; under some regularity conditions they would not only converge to the true parameter as the sample size tends to infinity, but it would also be approximately normally distributed if suitably standardized, and we can approximate the asymptotic variance.

Maximum-likelihood estimation also works well in Erdös-Renyi Mixture graphs when the number of types is known, and it works well in Watts-Strogatz small world networks when the number k of nearest neighbours we connect to is known. When the number of types, or the number of nearest neighbours, is unknown, then things become messy.

In Barabasi-Albert models, the parameter would be the power exponent for the node degree, as occurring in the probability for an incoming node to connect to some node i already in the network.

In exponential random graphs, unless the network is very small, maximum-likelihood estimation quickly becomes numerically unfeasible. Even in a simple model like

$$Prob(\mathbf{X} = \mathbf{x}) = \frac{1}{\kappa} \exp\{\lambda_1 L(\mathbf{x}) + \lambda_2 S_2(\mathbf{x}) + \lambda_3 S_3(\mathbf{x}) + \lambda_4 T(\mathbf{x})\}$$

the calculation of the normalising constant κ becomes numerically impossible very quickly.

3.2 Markov Chain Monte Carlo estimation

A Markov chain is a stochastic process where the state at time n only depends on the state at time $n-1$, plus some independent randomness; a random walk is an example. A Markov chain is *irreducible* if any set of states can be reached from any other state in a finite number of moves. The Markov chain is *reversible* if you cannot tell whether it is running forwards in time or backwards in time. A distribution is *stationary* for the Markov chain if, when you start in the stationary distribution, one step after you cannot tell whether you made any step or not; the distribution of the chain looks just the same.

There are mathematical definitions for these concepts, but we only need the main result here:

If a Markov chain is irreducible and reversible, then it will have a unique stationary distribution, and no matter in which state you start the chain, it will eventually converge to this stationary distribution.

We make use of this fact by looking at our target distribution, such as the distribution for $\mathbf{X}$ in an exponential random graph model, as the stationary distribution of a Markov chain.

This Markov chain lives on graphs, and moves are adding or deleting edges, as well as adding types or reducing types. Finding suitable Markov chains is an active area of research.

The `ergm` package has MCMC implemented for parameter estimation. We need to be aware that there is no guarantee that the Markov chain has reached its stationary distribution. Also, if the stationary distribution is not unique, then the results can be misleading. Unfortunately in exponential random graph models it is known that in some small parameter regions the stationary distribution is not unique. Another very active area of research.

3.3 Assessing the model fit

Suppose that we have estimated our parameters in our model of interest. We can now use this model to see whether it does actually fit the data.

To that purpose we study the (asymptotic) distributions of our summary statistics *node degree*, *clustering coefficient*, and *shortest path length*. Then we see whether our observed values are plausible under the estimated model.

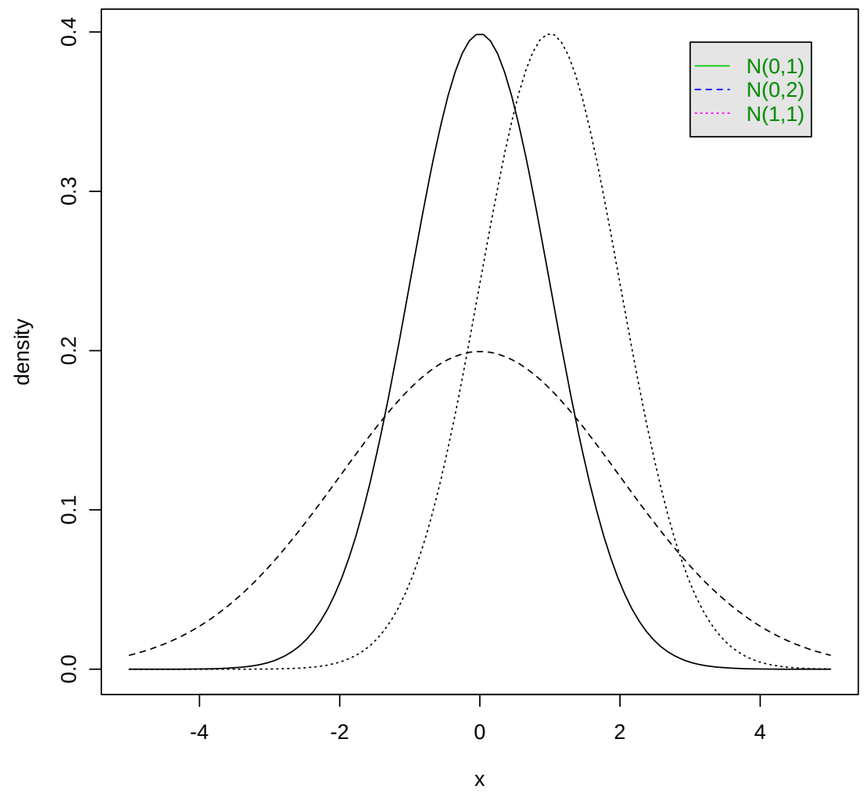
Often, secretly we would like to find that they are not plausible! Because then we can reject, say, the simple random graph model, and conclude that something more complicated is going on.

3.4 A quick review of distributions

Just a quick reminder of some classical distributions which often appear as limiting distributions.

The normal distribution $\mathcal{N}(\mu, \sigma^2)$

This distribution has mean μ and variance σ^2. Its shape is given by the Bell curve. Its density is awkward to write down, but probabilities can be calculated numerically.


Normal Probability Densities
density
0.0
0.1
0.2
0.3
0.4
N(0,1)
N(0,2)
N(1,1)
-4
-2
0
2
4
x

For a normally distributed random variable, around 2/3 of the time it will be within σ (the standard deviation, square root of the variance) of the mean.

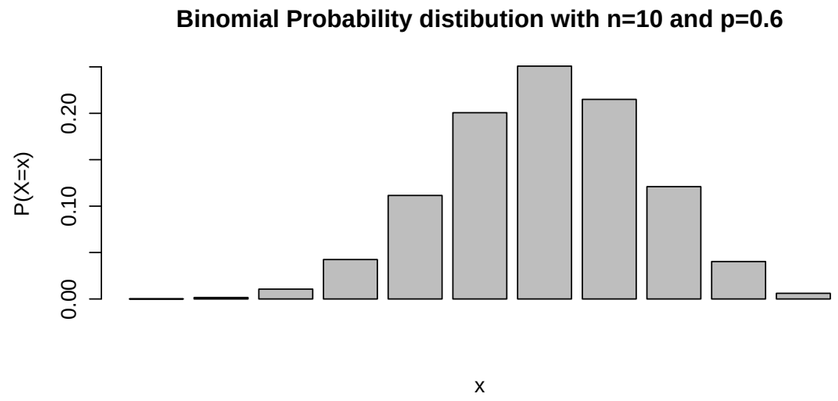
Around 95% of the time it will be within 2σ of the mean.

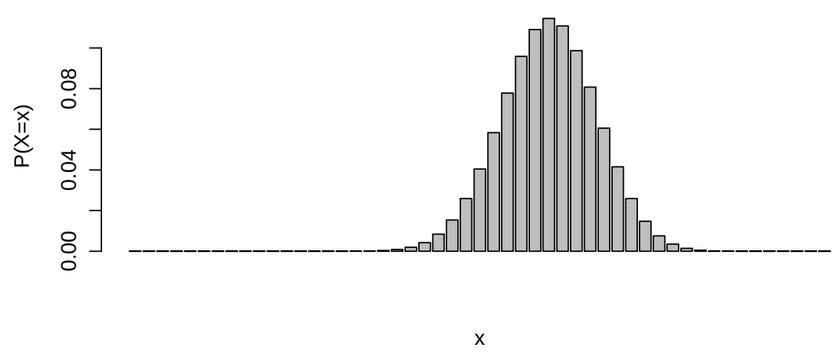
Around 99% of the time it will be within 3σ of the mean.

Thus if an observed value is further than 3σ away from μ, we would find that rather unusual; we would reject the null hypothesis that the data is normally $\mathcal{N}(\mu, \sigma^2)$ distributed at the level 1%.

The *Central Limit Theorem* tells us that, in a sequence of independent identically distributed observations with finite variance, the sample mean will converge to a normal distribution, and the standardised sample mean will be approximately standard normal.

Fact: If the observations are dependent, but only ”weakly” dependent, then the Central Limit Theorem still holds. (Another area of research.)


Binomial Probability distibution with n=10 and p=0.6
P(X=x)
0.00
0.10
0.20
x


Binomial Probability distibution with n=50 and p=0.6
P(X=x)
0.00
0.04
0.08
x

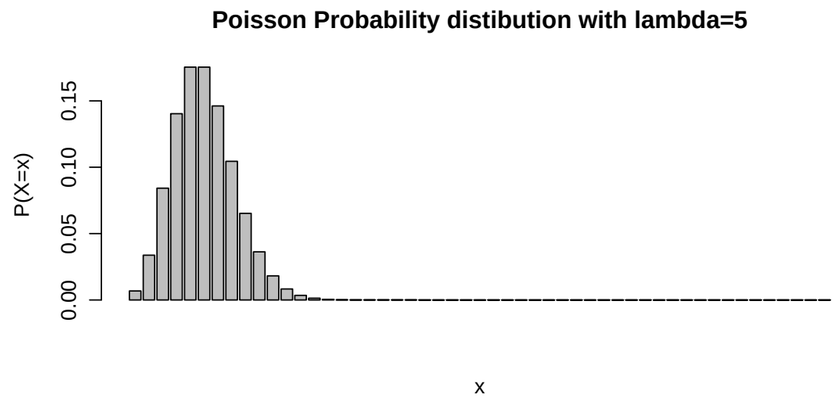
The Poisson distribution

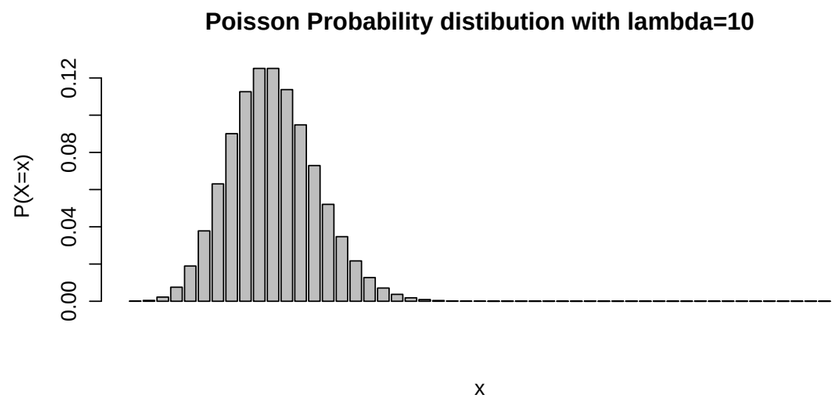
When considering the occurrence of ”rare” events, an approximation with a Poisson distribution is often more appropriate than a normal approximation.

The *Poisson distribution* lives on the non-negative integers; it has a parameter λ and X has Poisson distribution with parameter λ if

$$P(X = k) = e^{-\lambda} \frac{\lambda^k}{k!}, \quad k = 0, 1, 2, \ldots$$

It is relatively easy to calculate that mean and variance both equal to λ.

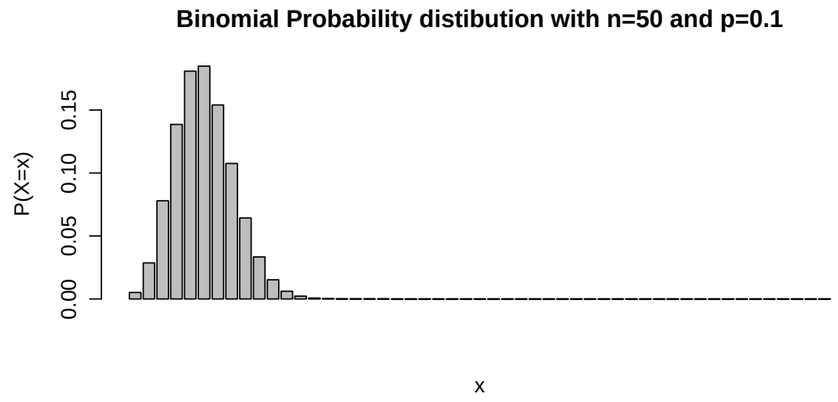

Poisson Probability distibution with lambda=5
P(X=x)
0.00
0.05
0.10
0.15
x

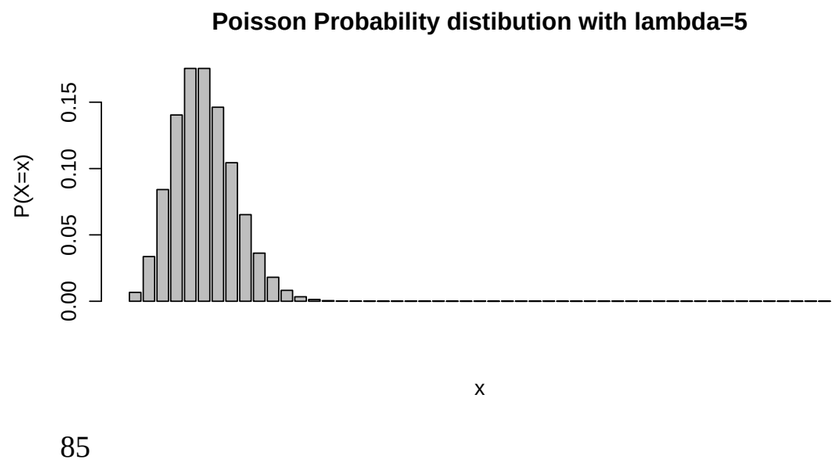

Poisson Probability distibution with lambda=10
P(X=x)
0.00
0.04
0.08
0.12
x

A famous result result states that if the expected number np of successes in n independent Bernoulli trials with probability of success p each is such that $np \to \lambda$ as $n \to \infty$, then the number of successes in these trials is approximately Poisson distributed with parameter λ.

Note: $np \to \lambda$ as $n \to \infty$ means that in each trial the probability of success is very small. A Poisson approximation is used for *rare events*.

The Poisson approximation is also good if the trials are only ”weakly” dependent.


Binomial Probability distibution with n=50 and p=0.1
P(X=x)
0.00
0.05
0.10
0.15
x


Poisson Probability distibution with lambda=5
P(X=x)
0.00
0.05
0.10
0.15
x

3.5 The distribution of summary statistics in Bernoulli random graphs

In a Bernoulli random graph on n nodes, with edge probability p, the network summaries are pretty well understood.

3.5.1 The degree of a random node

Pick a node v, and denote its degree by $D(v)$, say. The degree is calculated as the number of neighbours of this node. Each of the other $(n-1)$ nodes is connected to our node v with probability p, independently of all other nodes. Thus the distribution of $D(v)$ is Binomial with parameters n and p, for each node v.

Typically we look at relatively sparse graphs, and so a Poisson approximation applies. If $\mathbf{X}$ denotes the random adjacency matrix, then, in distribution,

$$D(v) = \sum_{u:u \neq v} X_{u,v} \approx Poisson((n-1)p).$$

Note that the node degrees in a graph are not independent. We have seen last time that there is **no** graph on 6 nodes which has 5 nodes of degree 5 and 1 node of degree 1. So $D(v)$ does **not** stand for the average node degree.

Computer exercise: Simulate many Bernoulli random graphs and look at their average node degree distribution. Also: pick node at random in each of the simulated graph, find its degree, and look at the distribution of these degrees.

How about the average degree of a node? Denote it by $\bar{D}$. Note that the average does not only take integer values, so would certainly not be Poisson distributed. But

$$\bar{D} \quad = \quad \frac{1}{n}\sum_{v=1}^{n} D(v) = \frac{2}{n}\sum_{v=1}^{n}\sum_{u<v} X_{u,v},$$

noting that each edge gets counted twice. As the $X_{u,v}$ are independent, we can use a Poisson approximation again, giving that

$$\sum_{v=1}^{n}\sum_{u<v} X_{u,v} \approx Poisson\left(\frac{n(n-1)}{2}p\right)$$

and so, in distribution,

$$\bar{D} \approx \frac{2}{n}Z,$$

where $Z \sim Poisson\left(\frac{n(n-1)}{2}p\right)$.

3.5.2 The clustering coefficient of a random node

Here it gets a little tricky already. Recall that the *clustering coefficient* of a node v is,

$$C(v) = \frac{\sum_{u,w \in V} X_{u,v} X_{w,v} X_{u,w}}{\sum_{u,w \in V} X_{u,v} X_{w,v}}.$$

The ratio of two random sums is not easy to evaluate. If we just look at

$$\sum_{u,w \in V} X_{u,v} X_{w,v} X_{u,w}$$

then we see that we have a sum of dependent random variables.

Exercise: Calculate the covariance of $X_{u,v}X_{w,v}X_{u,w}$ and $X_{u,v}X_{t,v}X_{u,t}$, where $w \neq t$.

Recall: for random variables X and Y, the covariance is defined as

$$Cov(X, Y) = E(XY) - E(X)E(Y).$$

Most 3-tuples (u, w, v) and (r, s, t), though, will not share an index, and hence $X_{u,v}X_{w,v}X_{u,w}$ and $X_{r,s}X_{s,t}X_{r,t}$ will be independent. The dependence among the random variables overall is hence weak, so that a Poisson approximation applies. As

$$E \sum_{u,w,v \in V} X_{u,v}X_{w,v}X_{u,w} = \binom{n}{3} p^3,$$

we obtain that, in distribution,

$$\sum_{u,w.v \in V} X_{u,v}X_{w,v}X_{u,w} \approx Poisson\left(\binom{n}{3} p^3\right).$$

Similarly,

$$E \sum_{u,w \in V} X_{u,v}X_{w,v} = \binom{n}{2} p^2.$$

K. Lin (2007) showed that, for the average clustering coefficient

$$C = \frac{1}{n}\sum_{v} C(v)$$

it is also true that, in distribution,

$$C \approx \frac{1}{n\binom{n}{2}p^2} Z,$$

where $Z \sim Poisson\left(\binom{n}{3}p^3\right)$.

Example. In the Florentine family data, we observe a total number of 20 edges, an average node degree of 2.5, and an average clustering coefficient of 0.1914894, with 16 nodes in total. Assess the null hypothesis that the data come from a Bernoulli random graph.

Let us assume that the null hypothesis is true, the data come from a Bernoulli random graph. Then we estimate

$$\hat{p} = \frac{20}{\binom{16}{2}} = \frac{20 \times 2}{16 \times 15} = \frac{1}{6}.$$

As in a Bernoulli random graph $\bar{D} \approx \frac{2}{n} Z$, where $Z \sim Poisson\left(\frac{n(n-1)}{2} p\right)$, under the null hypothesis the average node degree would be

$$\bar{D} \approx \frac{1}{8} Z,$$

where $Z \sim Poisson(20)$.

The probability under the null hypothesis that $\bar{D} \geq 2.5$ would then be

$$P(Z \geq 2.5 \times 8) = P(Z \geq 20) \approx 0.55,$$

so no reason to reject the null hypothesis.

Exercise: Test the null hypothesis that the Florentine family data come from a Bernoulli random graph using a test based on the average clustering coefficient.

3.5.3 Shortest paths: Connectivity in Bernoulli random graph

Erdös and Renyi (1960) showed the following ”phase transition” for the connectedness of a Bernoulli random graph.

If $p = p(n) = \frac{\log n}{n} + \frac{c}{n} + 0\left(\frac{1}{n}\right)$ then the probability that a Bernoulli graph, denoted by $\mathcal{G}(n, p)$ on n nodes with edge probability p is connected converges to $e^{-e^{-c}}$.

Recall the O and o notation: $f(n) = O(g(n))$ as $n \to \infty$ if the fraction $\frac{f(n)}{g(n)}$ is bounded away from ∞. If $f(n) = o(g(n))$ as $n \to \infty$ then the fraction $\frac{f(n)}{g(n)}$ tends to zero as $n \to \infty$.

The *diameter* of a graph is the maximum diameter of its connected components; the diameter of a connected component is the longest shortest path length in that component.

Chung and Lu (2001) showed that, if $np \geq 1$ then, asympotically, the ratio between the diameter and $\frac{\log n}{\log(np)}$ is at least 1, and remains bounded above as $n \to \infty$.

If $np \to \infty$ then the diameter of the graph is $(1 + o(1))\frac{\log n}{\log(np)}$. If $\frac{np}{\log n} \to \infty$, then the diameter is concentrated on at most two values.

In the Physics literature, the value $\frac{\log n}{\log(np)}$ is used for the average shortest path length in a Bernoulli random graph. This has hence to be taken with a lot of grains of salt.

While we have some idea about how the diameter (and, relatedly, the shortest path length) behaves, it is an inconvenient statistics for Bernoulli random graphs, because the graph need not be connected.

3.6 The distribution of summary statistics in Watts-Strogatz small worlds

Recall that in this model we arrange the n nodes of V on a lattice. Then hard-wire each node to its k nearest neighbours on each side on the lattice, where k is small. Thus there are nk edges in this hard-wired lattice. Now introduce random shortcuts between nodes which are not hard-wired; the shortcuts are chosen independently, all with the same probability ϕ.

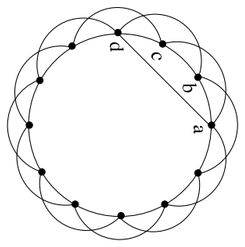
Thus the shortcuts behave like a Bernoulli random graph, but the graph will necessarily be connected. The degree $D(v)$ of a node v in the Watts-Strogatz small world is hence distributed as

$$D(v) = 2k + Binomial(n - 2k - 1, \phi),$$

taking the fixed lattice into account. Again we can derive a Poisson approximation when p is small; see K.Lin (2007) for the details.

For the clustering coefficient there is a problem - triangles in the graph may now appear in clusters. Each shortcut between nodes u and v which are a distance of $k + a \leq 2k$ apart on the circle creates $k - a - 1$ triangles automatically.

Thus a Poisson approximation will not be suitable; instead we use a *compound Poisson distribution*. A compound Poisson distribution arises as the distribution of a Poisson number of clusters, where the cluster sizes are independent and have some distribution themselves. In general there is no closed form for a compound Poisson distribution.


d
c
b
a

The compound Poisson distribution also has to be used when approximating the number of 4-cycles in the graph, or the number of other small subgraphs which have the clumping property.

It is also worth noting that when counting the joint distribution of the number of triangles and the number of 4-cycles, these counts are not independent, not even in the limit; a bivariate compound Poisson approximation with dependent components is required. See Lin (2007) for details.

3.6.1 The shortest path length

Let $\mathcal{D}$ denote shortest distance between two randomly chosen points, and abbreviate $\rho = 2k\phi$. Then (Barbour + Reinert) show that uniformly in $|x| \leq \frac{1}{4}\log(n\rho)$,

$$\begin{aligned} &\mathbf{P}\left(\mathcal{D} > \frac{1}{\rho}\left(\frac{1}{2}\log(n\rho) + x\right)\right) \\ &= \int_0^\infty \frac{e^{-y}}{1 + e^{2x}y}\, dy + O\left((n\rho)^{-\frac{1}{5}}\log^2(n\rho)\right) \end{aligned}$$

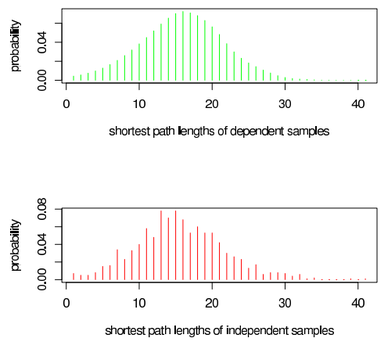
if the probability of shortcuts is small. If the probability of shortcuts is relatively large, then $\mathcal{D}$ will be concentrated on one or two points.

Note that $\mathcal{D}$ is the shortest distance between two randomly chosen points, **not** the average shortest path. Again the difference can be considerable (Computer exercise).

Dependent sampling

Our data are usually just one graph, and we calculate all shortest paths. But there is much overlap between shortest paths possible, creating dependence

Simulation: $n = 500, k = 1, \phi = 0.01$



Simulation: dependent vs independent

We simulate 100 replicas, and calculate the average shortest path length in each network. We compare this distribution to the theoretical approximate distribution; we carry out 100 chi-square tests:

n	k	ϕ	$E.no$	mean p-value	max p-value
300	1	0.01	3	1.74 E-09	8.97 E-08
		0.167	50	0.1978	0.8913
	2	0.01	6	0	0
1000	1	0.003	3	1.65E-13	3.30 E-12
		0.05	50	0.0101	0.1124
	2	0.03	60	0.0146	0.2840

Thus the two statistics are close if the expected number $E.no$ of shortcuts is large (or very small); otherwise they are significantly different.

Recall: chi-square test of goodness-of-fit

We want to test whether a data set comes from a conjectured (*null*) distribution. We group our data into cells such that under the null distribution, the count in each cell is expected to be at least 5.

Then we take the sum

$$X^2 = \sum_{\text{cells } i} \frac{(Observed_i - Expected_i)^2}{Expected_i}.$$

If the data come indeed from the null distribution, then X^2 will be approximately chi-squared distributed, with degrees of freedom ”number(cells) - number(fitted parameters) - 1”.

The p-value is the probability of seeing X^2 as least as large as the observed value if the null distribution is the correct distribution.

Aside: When comparing continuous distributions, the *Kolmogorov-Smirnov test* is another nonparametric alternative, as are *Wilcoxon tests*.

3.7 The distribution of summary statistics in Barabasi-Albert models

The node degree distribution is given by the model directly, as that is how it is designed.

The clustering coefficient depends highly on the chosen model. In the original Barabasi-Albert model, when only one new edge is created at any single time, there will be no triangles (beyond those from the initial graph). The model can be extended to match any clustering coefficient, but even if only two edges are attached at the same time, the distribution of the number of the clustering coefficient is unknown to date.

The expected value, however, can be approximated. Fronczak *et al.* (2003) studied the models where the network starts to grwo from an initial cluster of m fully connected nodes. Each new node that is added to the network created m edges which connect it to previously added nodes. The probability of a new edge to be connected to a node v is proportional to the degree $d(v)$ of this node. If both the number of nodes, n, and m are large, then the expected average clustering coefficient is

$$EC = \frac{m-1}{8}\frac{(\log n)^2}{n}.$$

The average pathlengh ℓ increases approximately logarithmically with network size. If $\gamma = 0.5772$ denotes the Euler constant, then Fronczak *et al.* (2004) show for the mean average shortest path length that

$$E\ell \sim \frac{\log n - \log(m/2) - 1 - \gamma}{\log\log n + log(m/2)} + \frac{3}{2}.$$

The asymptotic distribution is not understood.

3.8 The distribution of summary statistics in exponential random graph models

The distribution of the node degree, clustering coefficient, and the shortest path length is poorly studied in these models. One reason is that these models are designed to predict missing edges, and to infer characteristics of nodes, but their topology itself has not often been of interest.

The summary statistics appearing in the model try to push the random networks towards certain behaviour with respect to these statistics, depending on the sign and the size of their factors θ.

When only the average node degree and the clustering coefficient are included in the model, then a strange phenomenon happens. For many combinations of parameter values the model produces networks that are either full (every edge exists) or empty (no edge exists) with probability close to 1. Even for parameters which do not produce this phenomenon, the distribution of networks produced by the model is often bimodal: one mode is sparsely connected and has a high number of triangles, while the other mode is densely connected but with a low number of triangles. Again: active research.

4 Statistical tests for model fit: nonparametric methods

What if we do not have a suitable test statistic for which we know the distribution? We need some handle on the distribution, so here we assume that we can simulate random samples from our null distribution. There are a number of methods available.

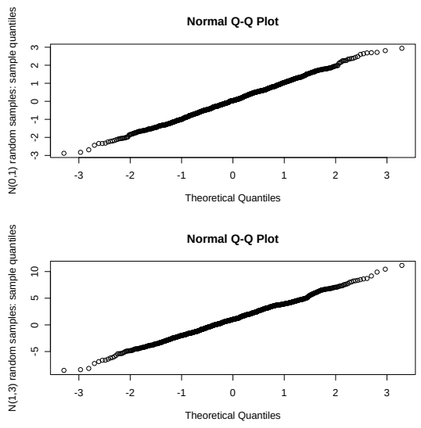
4.1 Quantile-quantile plots

It is often a good idea to use plots to visually assess the fit. A much used plot in Statistics a *quantile-quantile plot*.

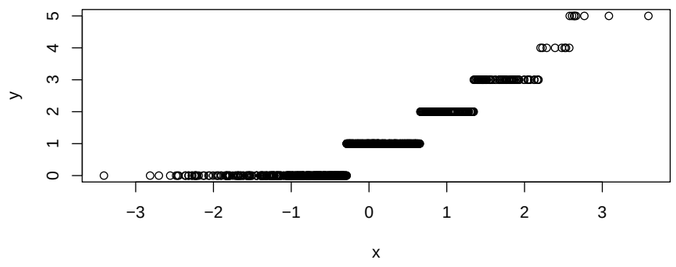
The *quantiles* of a distribution are its ”percent points”; for example the 0.5 quantile is the 50 % point, i.e. the median. Mathematically, the *(sample) quantiles* q_α, are defined for $0 \leq \alpha \leq 1$ so that a proportion of at least α of the data are less or equal to q_α and a proportion of at least $1 - \alpha$ is greater or equal to q_α. There are many (at least 8) definitions of q_α if αn is not an integer.

We plot the quantiles of our observed (empirical) distribution against the quantiles of our hypothesised (null) distribution; if the two distributions agree, then the plot should result in a roughly diagonal line.

Example: Simulate 1,000 random variables from a normal distribution. Firstly: mean zero, variance 1; secondly: mean 1, variance 3. Both QQ-plots are satisfactory.



We can also use a quantile-quantile plot for two sets of simulated data, or for one set of simulated data and one set of observed data. The interpretation is always the same: if the data come from the same distribution, then we should see a diagonal line; otherwise not. Here we compare 1000 Normal (0,1) variables with 1000 Poisson (1) variables - clearly not a good fit.


y
0
1
2
3
4
5
−3
−2
−1
0
1
2
3
x

4.2 Monte-Carlo tests

The Monte Carlo test, attributed to Dwass (1957) and Barnard (1963), is an exact procedure of virtually universal application and correspondingly widely used.

Suppose that we would like to base our test on the statistic T_0. We only need to be able to simulate a random sample $T_{01}, T_{02}, \ldots$ from the distribution, call it F_0, determined by the null hypothesis. We assume that F_0 is continuous, and, without loss of generality, that we reject the null hypothesis H_0 for large values of T_0. Then, provided that $\alpha = \frac{m}{n+1}$ is rational, we can proceed as follows.

1. Observe the actual value t^* for T_0, calculated from the data
2. Simulate a random sample of size n from F_0
3. Order the set $\{t^*, t_{01}, \ldots, t_{0n}\}$
4. Reject H_0 if the *rank* of t^* in this set (in decreasing order) is $\geqslant m$.

The basis of this test is that, under H_0, the random variable T^* has the same distribution as the remainder of the set and so, by symmetry,

$$\mathbf{P}(t^* \text{ is among the largest } m \text{ values }) = \frac{m}{n+1}.$$

The procedure is exact however small n might be. However, increasing n increases the power of the test. The question of how large n should be is discussed by Marriott (1979), see also Hall and Titterington (1989). A reasonable rule is to choose n such that $m \geqslant 5$. Note that we will need more simulations to test at smaller values of α.

An alternative view of the procedure is to count the number M of simulated values $> t^*$. Then $\hat{P} = \frac{M}{n}$ estimates the true significance level P achieved by the data, i.e.

$$P = \mathbf{P}(T_0 > t^* | H_0).$$

In discrete data, we will typically observe ties. We can break ties randomly, then the above procedure will still be valid.

Unfortunately this test does not lead directly to confidence intervals.

For random graphs, Monte Carlo tests often use shuffling edges with the number of edges fixed, or fixing the node degree distribution, or fixing some other summary.

Suppose we want to see whether our observed clustering coefficient is ”unusual” for the type of network we would like to consider. Then we may draw many networks uniformly at random from all networks having the same node degree sequence, say. We count how often a clustering coefficient at least as extreme as ours occurs, and we use that to test the hypothesis.

In practice these types of test are the most used tests in network analysis. They are called *conditional uniform graph tests*.

Some caveats:

In Bernoulli random graphs, the number of edges asymptotically determines the number of triangles when the number of edges is moderately large. Thus conditioning on the number of edges (or the node degrees, which determine the number of edges) gives degenerate results. More generally, we have seen that node degrees and clustering coefficient (and other subgraph counts) are not independent, nor are they independent of the shortest path length. By fixing one summary we may not know exactly what we are testing against.

”Drawing uniformly at random” from complex networks is not as easy as it sounds. Algorithms may not explore the whole data set.

”Drawing uniformly at random”, conditional on some summaries being fixed, is related to sampling from exponential random graphs. We have seen already that in exponential random graphs there may be more than one stationary distribution for the Markov chain Monte Carlo algorithm; this algorithm is similar to the one used for drawing at random, and so we may have to expect similar phenomena.

4.3 Scale-free networks

Barabasi and Albert introduced networks such that the distribution of node degrees the type

$$Prob(degree = k) \sim Ck^{-\gamma}$$

for $k \to \infty$. Such behaviour is called *power-law behaviour*; the constant γ is called the *power-law exponent*. The networks are also called *scale-free*:

If $\alpha > 0$ is a constant, then

$$Prob(degree = \alpha k) \sim C(\alpha k)^{-\gamma} \sim C'k^{-\gamma},$$

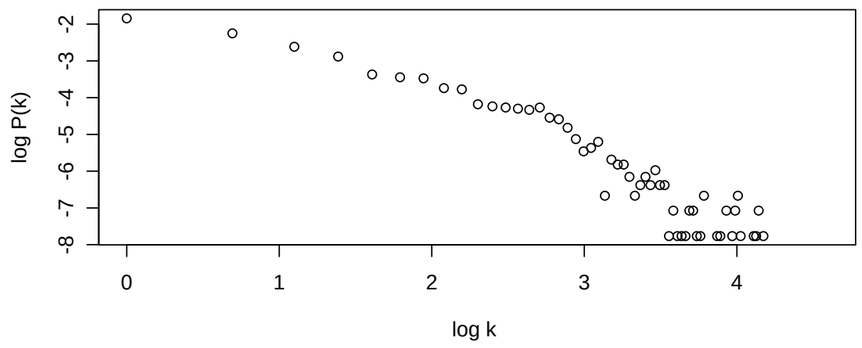
where C' is just a new constant. That is, scaling the argument in the distribution changes the constant of proportionality as a function of the scale change, but preserves the shape of the distribution itself.

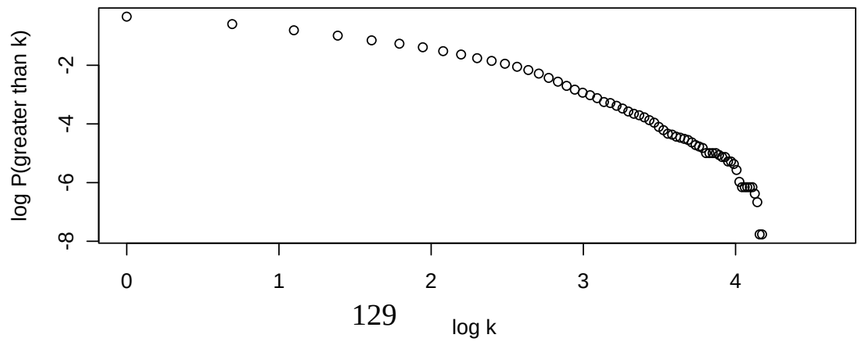
If we take logarithms on both sides:

$$\begin{aligned} \log Prob(degree = k) &\sim \log C - \gamma \log k \\ \log Prob(degree = \alpha k) &\sim \log C - \gamma \log \alpha - \gamma \log k; \end{aligned}$$

scaling the argument results in a linear shift of the $\log$ probabilities only. This equation also leads to the suggestion to plot the $\log relfreq(degree = \alpha k)$ of the empirical relative degree frequencies against $\log k$. Such a plot is called a *log-log plot*. If the model is correct, then we should see a straight line; the slope would be our estimate of γ.

Example: Yeast data.





These plots have a lot of noise in the tails. As an alternative, *Newman (2005)* suggests to plot the $\log$ of the empirical cumulative distribution function instead, or, equivalently, our estimate for

$$\log Prob(degree \geq k).$$

If the model is correct, then one can calculate that

$$\log Prob(degree \geq k) \sim C'' - (\gamma - 1) \log k.$$

Thus a log-log plot should again give a straight line, but with a shallower slope. The tails are somewhat less noisy in this plot.

In both cases, the slope is estimated by least-squares regression: for our observations, $y(k)$ (which could be log probabilities or log cumulative probabilities, for example) we find the line $a + bk$ such that

$$\sum (y(k) - a - bk)^2$$

is as small as possible.

As a measure of fit, the sample correlation R^2 is computed. For general observations $y(k)$ and $x(k)$, for $k = 0, 1, \ldots, n$, with averages $\bar{y}$ and $\bar{x}$, it is defined as

$$R = \frac{\sum_k (x(k) - \bar{x})(y(k) - \bar{y})}{\sqrt{(\sum_k (x(k) - \bar{x})^2)(\sum_k (y(k) - \bar{y})^2)}}.$$

It measures the strength of the linear relationship.

In linear regression, $R^2 > 0.9$ would be rather impressive. However, the rule of thumb for log-log plots is that

1. $R^2 > 0.99$
2. The observed data (degrees) should cover at least 3 orders of magnitude.

Examples include the World Wide Web at some stage, when it had around 10^9 nodes. The criteria are not often matched.

Computer exercise: Generate random samples from your favourite probability distribution, make a log-log plot.

A final issue for scale-free networks: It has been shown (*Stumpf et al. (2005)* that when the underlying real network is scale-free, then a subsample on fewer nodes from the network will not be scale-free. Thus if our subsample looks scale-free, the underlying real network will not be scale-free.

In biological network analysis, is is debated how useful the concept of ”scale-free” behaviour is, as many biological networks contain relatively few nodes.

Further references

A.D. Barbour and G. Reinert (2001). Small Worlds. Random Structures and algorithms 19, 54 - 74.

A.D. Barbour and G. Reinert (2006). Discrete small world networks. Electronic Journal of Probability 11, 12341283.

G. Barnard (1963). Contribution to the discussion of Bartlett’s paper. J. Roy. Statist. Soc. B, 294.

F. Chung and L. Lu (2001). The diameter of sparse random graphs. Advances in Applied Math. 26, 257–279.

M. Dwass (1957). Modified randomization tests for nonparametric hypotheses. *Ann. Math. Stat.* **28**, 181–187.

A. Fronczak, P. Fronczak and Janusz A. Holyst (2003). Mean-field theory for clustering coefficients in Barabasi-Albert networks Phys. Rev. E 68, 046126.

A. Fronczak, P. Fronczak, Janusz A. Holyst (2004). Average path length in random networks Phys. Rev. E 70, 056110.

P. Hall and D.M. Titterington (1989). The effect of simulation order on level accuracy and power of Monte Carlo tests. J. Roy. Statist. Soc. B, 459.

K. Lin (2007). Motif counts, clustering coefficients, and vertex degrees in

models of random networks. D.Phil. dissertation, Oxford.

F. Marriott (1979). Barnard’s Monte Carlo tests: how many simulations? Appl. Statist.
28, 75–77.

M.E.J. Newman (2005). Power laws, Pareto distributions and Zipf’s law. Contemporary Physics 46, 323351.

M. P. H. Stumpf, C.Wiuf and R. M. May (2005). Subnets of scale-free networks are not scale-free: Sampling properties of networks. Proc Natl Acad Sci U S A. 2005 March 22; 102(12): 4221 - 4224.

Recap

We looked at Bernoulli random graphs and their mixtures, Watts-Strogatz small worlds, Barabasi-Albert scale-free networks, and exponential random graphs. We saw that in these models the summaries are dependent. As an extreme case, knowing the degree sequence may already completely specify the network.

Specific networks may allow for specific modelling, and summaries may be chosen to best reflect the main features of the network.

For Bernoulli random graphs we have a reasonable grasp on the distribution of our network summaries, we can use maximum-likelihood estimation and we can use the distribution of the summaries for testing hypotheses. For Watts-Strogatz small worlds some results are available. A main observation is that, in contrast to Bernoulli random graphs, even for counting triangles a compound Poisson approximation is needed, rather than a Poisson approximation. The underlying issue is that triangles (and also other motifs) occur in clumps.

For random graphs, Monte Carlo tests often use shuffling edges with the number of edges fixed, or fixing the node degree distribution, or fixing some other summary. Then we use a Monte Carlo test to see whether our test statistic is unusual, compared to graphs drawn at random with the same number of edges, or the same node degree distribution, say. The results have to be interpreted carefully.

5 Statistical inference for networks: local properties.

In the statistical analysis of networks we are often interested in inferring ”local” properties, such as the existence of an edge or the characteristics of a node, from the position of the node, or the potential edge, in the network. Such inference has a long tradition in social network analysis. Let us see how exponential random graphs lend themselves to statistical analysis.

Recall that exponential random graph (p^*) models model the whole adjacency matrix of a graph simultaneously. If $\mathbf{X}$ is our random adjacency matrix, then the general form of the model is

$$Prob(\mathbf{X} = \mathbf{x}) = \frac{1}{\kappa} \exp\{\sum_{B=1}^{N} \lambda_B z_B(\mathbf{x})\},$$

where the $z_B(\mathbf{x})$ are network statistics, κ is a normalising quantity so that the probabilities sum to 1, and $\lambda = (\lambda_1, \ldots, \lambda_N)$ is a vector of unknown parameters. See also *Anderson, Wasserman, Crouch (1999)*.

For social networks, *Markov graphs* are of particular interests, where two possible edges are assumed to be conditionally dependent if they share a node. Typically social network analysis focuses on directed networks, but let's stay with undirected networks for simplicity.

The celebrated *Hammersley-Clifford Theorem* tells us that non-directed Markov graphs have parameters relating only to the configurations *stars of various types, and triangles*. This is why usually only the number of edges, the number of k-stars for $k = 2, 3$, and the number of triangles are included in social network models.

5.1 Log-linear models

The response variable is the probability of the observed $\mathbf{x}$, which is $Prob(\mathbf{X} = \mathbf{x})$; this is a value between 0 and 1. For regression analysis we would usually like to assume that our errors are approximately normally distributed. Hence we usually model not the probability itself, but a logarithmic transformation of it. The p^*-model can be read as

$$\log Prob(\mathbf{X} = \mathbf{x}) \propto \lambda_0 + \sum_{B=1}^{N} \lambda_B z_B(\mathbf{x}).$$

For identifiability of parameters we use the constraint that $\sum_{B=1}^{N} \lambda_B = 0$. Thus the model parameters, the elements of the vector $\lambda = (\lambda_0, \lambda_1, \ldots, \lambda_N)$, will be the coefficients of a linear function of these explanatory variables, just as in standard linear models:

$$\lambda_0 + \lambda_1 z_1(\mathbf{x}) + \lambda_2 z_2(\mathbf{x}) + \ldots + \lambda_N z_N(\mathbf{x}).$$

Usually in log-linear models the maximum-likelihood, or a version of it, is used to estimate the parameters. For a tractable likelihood it is usually assumed that the edges are independent. The model for the error depends on the explanatory variables; often we assume normally distributed errors, but we could also assume Poisson distributed errors, for example.

While the p^*-model looks like a multivariate *log-linear model*, it is not. Loglinear models assume independent observations, an assumption we explicitly do not make with Markov and higher order models. So the parameter estimates may be biased; and the standard errors are approximate at best, and may be too small. Whenever possible, the preferred option is to use Monte Carlo estimation procedures.

Often we are interested in a one-dimensional output measure. For example, our nodes may have certain properties (for example age, or protein function) and we would like to predict the property of a node in the network, using the information from the network.

Let us suppose that we have L covariate classes denoted by $\mathbf{c}_1, \ldots, \mathbf{c}_L$, class $\mathbf{c}_j$ containing m_j individuals, and we have K response categories. The resulting observations are $\mathbf{y}_1, \ldots, \mathbf{y}_K$, where $\mathbf{y}_j = (y_{j,1}, \ldots, y_{j,r})$ denotes the number in each response category for the jth covariate class. Then the $\mathbf{Y}_j$ follow independent multinomial distributions with probability vector

$$\pi_j = \pi(\mathbf{c}_j) = (\pi_{j,1}, \ldots, \pi_{j,r}) = (\pi_1(\mathbf{c}_j), \ldots, \pi_r(\mathbf{c}_j))$$

Here $\pi_{j,i} = \pi_i(\mathbf{c}_j)$ denotes the probability for an observation in the $\mathbf{c}_j$ covariate class to yield the response i.

We can hence write down the log-likelihood for the whole data as

$$\ell(\pi, \mathbf{y}) = \sum_{i,j} y_{i,j} \log \pi_{j,i}$$

with constraints $\sum_j y_{i,j} = m_i$ and $\sum_{i,j} \pi_{j,i} = 1$.

If we do not model $\pi_{j,i}$ then the maximum-likelihood estimate for $\pi_{j,i}$ is just the proportion of observations in the $\mathbf{c}_j$ covariate class which yield the response i.

But often we would like to model the $\pi_{j,i}$ as depending on certain covariates, by assuming that

$$\log \pi_{j,i} = \theta_{0,i} + \theta_{1,i} x_{1,j}(\mathbf{c}_j) + \theta_{2,i} x_{2,j}(\mathbf{c}_j) + \ldots + \theta_{K,i} x_{K,j}(\mathbf{c}_j).$$

Here the vector θ are unknown parameters, and the x_j are summaries of the network. Any summary of the network, for example, the number of edges, and any characteristics of the nodes themselves (for example, gender) can be used as covariates.

We can replace $\pi_{j,i}$ by our chosen model and estimate the unknown parameters by maximizing this likelihood. This is done numerically. Once we have obtained (maximum-likelihood) estimates for θ, we can use it to infer properties of a specific node v. Suppose that we know the position of node v in the network, and we know its covariate class c_v, then we would estimate that node v falls into category k by $\hat{\pi}_k(\mathbf{c}_v)$.

5.2 Example: Inferring characteristics of proteins from their position in a protein interaction network

The fairly rigid independence assumptions in the loglinear model are usually not satisfied. Nevertheless we can use ideas from loglinear models to develop a *score* which we can use to classify proteins, say. Here is an elaborate example from *Chen et al. (2007)*.

As characteristics we consider structure (7 categories) and function (24 categories). From the protein-protein interaction network, we build an upcast set of category-category interactions. A category-category interaction is constructed by two characteristic categories from two interacting proteins.

Consider a protein x, within the set of all characteristic categories S, $S(x)$ includes the categories that protein x is classified into. If two proteins x and y interact, the category-category interaction is the edge between two characteristic categories, a and b ($a \in S(x)$, $b \in S(y)$), from each of two proteins (denoted by $a \sim b$). The upcast set of category-category interactions is a collection of all category-category interactions extracted from the protein-protein interaction network, which may be from one or multiple organisms.

Our scoring method is based on the heuristic assumption that the likelihood for a specific category to be observed in the query protein is roughly proportional to the product of the relative frequencies of observing this category in all pairs around the neighbours of a query protein.

The score for the query protein x with annotated neighbours $B(x)$, to be in a specific category a is proportional to the product $C(a, x)$. This is the product of the relative frequencies f of observing category a for all category-category interactions of x's neighbours in the prior data base;

$$C(a, x) = \prod_{\substack{b \in S(n) \\ n \in B(x)}} f(a \sim b), \tag{1}$$

where $f(\, a \sim b)$ is the relative frequency of category-category interaction $\{\, a \sim b\}$ among all category-category interactions. This would be the maximum-likelihood estimate in the loglinear model. But we do not dare to assume that the edges are independent. Thus we use the maximum-likelihood estimate as a guidance to derive a score.

We define our score $F(a, S(x))$ by

$$F(a, S(x)) := \frac{C(a, x)}{\sum_{k \in S} C(k, x)}.$$

The protein is then predicted to possess the characteristic category, or categories, with the highest score.

This score is derived as an analogy of the likelihood of observing category a in $S(x)$ if all edges in the category interaction network occurred independently. Heuristically this score serves as a measure for the chance of protein x having characteristic a.

The method can be extended to include two or more protein characteristics in the prediction of a specific protein characteristic. Then the category in a category-category interaction is now a vector containing all characteristics of the protein. In the case of two protein characteristics, S_1 and S_2, a characteristic vector is a 2-vector with 2 characteristic categories from S_1 and S_2. While S is now the set of all characteristic vectors, $S(x)$ is the subset of S of the characteristic vectors of protein x,

$$S(x) = \left\{ [s_1, s_2] \,\middle|\, s_1 \in S_1(x), s_2 \in S_2(x) \right\}.$$

Given the characteristics of the neighbours, the product of the frequencies of category-category interactions $C(a, x; S_i)$ for a protein x to be in the characteristic category a of S_i is defined, in a similar way to (1), as

$$C(a, x; S_i) = \prod_{\substack{v_b \in S(n) \\ n \in B(x)}} f(v_a \sim v_b, v_{ai} = a),$$

where $v_a = [v_{a1}, v_{a2}]$, $v_{aj} \in S_j(x)$ $(j \neq i)$. We add 1 to the relative frequency $f(v_a \sim v_b)$ to avoid the case when $v_a \sim v_b$ exists in unobserved interactions.

The accuracies of structure and function prediction using different methods: The methods for predictions are ”majority vote” (M.V.), our basic method (F.), and our enhanced method (E.F.). For structure, the protein structure is predicted the class with the highest probability. For function, a function prediction is counted as correct if one of the best three predicted categories is correct. We underline a score where the result outperforms M.V. with statistical significance (5%).

Organism (DIP)	Predicted proteins	M.V.	F.	E. F.
Structure				
D.Melanogaster	1262	0.35	0.17	**0.44**
C.Elegans	78	0.36	0.37	0.49
S.Cerevisiae	1608	0.39	0.31	**0.54**
E.Coli	150	0.57	0.70	**0.71**
M.Musculus	32	0.72	0.50	0.69
H.Sapiens	273	0.44	0.47	0.71
Function				
D.M	1275	0.53	0.67	**0.69**
C.E	85	0.38	0.55	0.71
S.C	1618	0.67	0.61	0.67
E.C	154	0.69	0.69	0.70
M.M	32	0.59	0.88	0.81
H.S	274	0.79	0.90	0.89

For maximum-likelihood inference in our log-linear model we need to calculate the normalising constant κ, which causes numerical difficulties.

5.3 Logistic models

Often we are only interested in a binary outcome, say $Y \in \{0, 1\}$, which indicates, say, whether or not an edge between two nodes is present in the network. Then we can use the so-called *logit* transformation; the log odds that an event occurs. For any $p \in [0, 1]$ we define

$$logit(p) = \log\left(\frac{p}{1-p}\right).$$

This function takes values on the whole real line. We can transform back: If $\theta = logit(p)$ then

$$p = \frac{e^\theta}{1 + e^\theta}.$$

Typically our model is

$$logit(y) = \lambda_0 + \lambda_1 x_1 + \lambda_2 x_2 + \ldots + \lambda_N x_N.$$

In the Statistics module, you phrased this model slightly differently; for binary data $y_1, \ldots, y_n$ you modelled

$$Y_i \sim Binomial(1, \alpha_i)$$

with

$$\alpha_i = (1 + exp(-\lambda_0 + \sum_j \lambda_j x_j))^{-1}.$$

Then

$$logit(\alpha_i) = \log\left(\frac{\alpha_i}{1-\alpha_i}\right)$$

and

$$1 - \alpha_i = \frac{exp(-\lambda_0 + \sum_j \lambda_j x_j)}{1 + exp(-\lambda_0 + \sum_j \lambda_j x_j)},$$

so that

$$\begin{aligned}\frac{\alpha_i}{1-\alpha_i} &= (1+exp(-\lambda_0+\sum_j \lambda_j x_j))^{-1}\frac{1+exp(-\lambda_0+\sum_j \lambda_j x_j)}{exp(-\lambda_0+\sum_j \lambda_j x_j)} \\ &= exp(\lambda_0+\sum_j \lambda_j x_j),\end{aligned}$$

giving

$$logit(\alpha_i) = \lambda_0+\sum_j \lambda_j x_j,$$

so this is the same model as the one you have seen in the previous module.

To interpret the parameters, the most natural way uses odds and odds ratios. Taking the exponential of our model gives us a multiplicative model for the odds of the binary response variable; that is, with $x_0 = 1$,

$$\frac{Prob(Y = 1)}{Prob(Y = 0)} = e^{\sum_B \lambda_B x_B}.$$

Thus, the odds that an event occurs changes multiplicatively with changes in the explanatory variables. For example, holding all variables except x_k constant, the odds that $Y = 1$ when the explanatory variable $x_k = x + 1$ is e^{λ_k} times the odds when $x_k = x$. In other words, the ratio of the odds or odds ratio for a one unit increase in x_k equals e^{λ_k}.

For networks, we can set up a model for the conditional log odds that an edge is present between nodes u and v, conditional on the network $\mathbf{X}^{u,v}$, which is the whole network *except* the edge indicator $X_{u,v}$. Then we model

$$logitProb(X_{u,v} = x_{u,v}|\mathbf{X}^{u,v} = \mathbf{x}^{u,v}) \propto \lambda_0 + \sum_{B=1}^{N} \lambda_B z_B(\mathbf{x}^{u,v}).$$

Here the summary statistics may of course be different to the ones used in the loglinear model.

Warning: This looks like a logistic regression but it is not. Same reason as before - logistic regression assumes independent observations, an assumption we explicitly do not make with Markov and higher order models. Whenever possible, the preferred option is to use Monte Carlo estimation procedures.

Example: The Florentine marriage network

In the Florentine marriage network we do not have any attributes available, but we have the network itself. So we can fit an exponential random graph model to the data. Using `ergm` for the model

$$\log network \propto 1stars + 2stars + triangles$$

we obtain as Monte Carlo MLE coefficients

kstar1	kstar2	triangle
-0.7855	-0.0277	0.2192

We could use this model to predict whether or not an edge is present. We first remove the edge, calculate the model, and then simulate from the model and count how often the edge is present in the simulated networks.

Example: Faux Mesa High

Here the data are made-up but realistic High school interaction data. Each node represents a student in grades 7 to 12 at a hypothetical school in the US, and each edge indicates a mutual friendship in which each node names the other as one of his or her top five male or top five female friends. The nodes have attributes such as grade and gender.

We fit a logistic model with covariates “nodematch(“Grade”)”, telling us whether the two nodes are in the same grade or not, and gender. Not taking any network structure into account, we obtain as results:

	Estimate	Std. Error	MCMC s.e.	p-value
nodematch.Grade.7	-1.97558	0.12297	NA	<1e-04 ***
nodematch.Grade.8	-0.91261	0.20281	NA	<1e-04 ***
nodematch.Grade.9	-2.24637	0.21885	NA	<1e-04 ***
nodematch.Grade.10	-1.48103	0.36637	NA	<1e-04 ***
nodematch.Grade.11	-0.63663	0.29378	NA	0.0302 *
nodematch.Grade.12	-0.97621	0.47536	NA	0.0400 *
nodefactor.Sex.M	-3.95003	0.07652	NA	<1e-04 ***

The fit is measured in terms of the deviance: twice the log likelihood difference between the fitted model and the null model which assumes no linear relationship. The deviance is approximately chi-square distributed. Here the deviance is 21152.7 on 7 degrees of freedom, highly significant.

What happens if we include network structure in the model? In addition to nodematch and gender, let's use the number of edges in the model as well. Then the results are:

	Estimate	Std. Error	MCMC s.e.	p-value
edges	-5.6784	0.1820	NA	< 1e-04 ***
nodematch.Grade.7	2.7869	0.1981	NA	< 1e-04 ***
nodematch.Grade.8	2.9969	0.2395	NA	<1e-04 ***
nodematch.Grade.9	2.4074	0.2643	NA	< 1e-04 ***
nodematch.Grade.10	2.6208	0.3744	NA	< 1e-04 ***
nodematch.Grade.11	3.3627	0.2971	NA	< 1e-04 ***
nodematch.Grade.12	3.6628	0.4578	NA	< 1e-04 ***
nodefactor.Sex.M	-0.3743	0.1047	NA	0.000351 ***

Now the deviance is 27072.4 on 8 degrees of freedom. As the first model is a sub-model of the second model, we can test whether the second model is a better fit, by looking at the deviance difference:

$$27072.4 - 21152.7 = 5919.7,$$

with 1 degree of freedom: highly significant. Taking the number of edges into account, the signs of the coefficients for the nodematches change!

5.4 Example: Inferring protein interactions from protein characteristics using the protein interaction network

Again, the assumptions of a logistic model are usually not satisfied, but we can use these ideas to develop a score which we can use to predict and to validate protein interactions, based on the protein characteristics and the protein interaction network.

Here, network structure comes into play. We observe that the protein interaction network has a tendency to form triangles.

5.4.1 Tendency to form triangles

Let a, b, $c \in S$ be three characteristic vectors, and let N be the set of proteins in the protein interaction network. Assume that all of a, b and c are indeed observed in the proteins. For each type of category-category pair $\{a, c\}$ with a fixed category b, the ratio of conditional probabilities $r_{abc} = \frac{P(a \sim c | a \sim b \sim c)}{P(a \sim c)}$ is then estimated by

$$\hat{r}_{abc} = \frac{\hat{P}(a \sim c | a \sim b \sim c)}{\hat{P}(a \sim c)} = \frac{\hat{P}(a \sim c,\ a \sim b \sim c)}{\hat{P}(a \sim b \sim c \sim a) + \hat{P}(a \sim b \sim c \not\sim a)},$$

where $\hat{P}(a \sim c)$ is the proportion of pairs of proteins x, y in N, with characteristics such that $a \in S(x), b \in S(y)$, which interact, relative to all pairs of proteins with such characteristics. Note that, in contrast to the triangle rate score, we sum over proteins and keep characteristics fixed. Similarly, $\hat{P}(a \sim b \sim c \sim a)$ is the proportion of protein triplets, with given characteristics, which form a triangle, and $\hat{P}(a \sim b \sim c \not\sim a)$ is the proportion of protein triplets, with given characteristics, which form a line (but not a triangle).

For each organism (protein interaction network), $\bar{r}$ is the average of $\hat{r}_{abc}$ for all $a, b, c \in S$,

$$\bar{r} = \frac{\sum_{a,b,c \in S} \hat{r}_{abc}}{\frac{1}{2}|S|^2(|S|+1)}.$$

If $\bar{r} \ll 1$, the existence of the interacting partner tends to decrease the chance of interaction. If $\bar{r} \gg 1$, the interaction is more likely if two protein have an common interacting partner. The average ratios of conditional probabilities from different organisms are estimated in Table 1.

Table 1: Estimates of $\bar{r}$ from characteristic triplets

Organisms	#obs. pairs†	#obs. triples‡	$\bar{r}$	S.E.	$\bar{r} > 1$§
triangles/lines (structure)					
D.M.	23	94	5.6	2.76	$\star$
S.C.	26	157	25.2	8.28	$\star$
E.C.	20	105	9.6	4.14	$\star$
H.S.	19	74	26.7	20.44	
triangles/lines (function)					
D.M.	110	534	48.3	67.6	
S.C.	214	1850	55.9	91.36	
E.C.	76	494	16.1	9.91	$\star$
H.S.	60	350	76.8	125.25	

† number of different pairs $\{a, c\}$ forming triples $\{a \sim b \sim c\}$

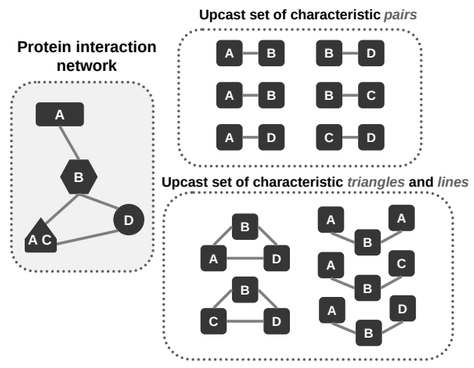
‡ total number of different triples $\{a \sim^{171} b \sim c\}$

§ 5% level of significance

$\star$ organism showing tendency of formation of triangles

The standard errors (S.E.) in Table 1 are much larger in the estimates based on enhanced lines and triangles than in non-enhanced ones. The reasons may be that, firstly, many types of triples have only a few counts, and secondly, missing data are likely to lead to biological triangles counted as lines.

We now build an upcast set of triangles and lines as follows.



Here, A, B, C and D denote protein characteristics, whereas different shapes indicate different proteins. A protein may possess more than one characteristic. Our triplets are triangles and lines of three characteristic vectors according to their interacting patterns. A characteristic line is a specific pattern constructed by three vectors with two vector interactions among them. A characteristic triangle is formed by three vectors interacting with each other.

Within the triplet interactions, we assess the odds to observe triangles versus lines around the query protein pair. More formally, let t_{xy} be the total frequency of all characteristic triangles around the query protein pair $\{x, y\}$; denoting by $z \in B(x, y)$ the set of all common neighbours of x and y in the protein interaction network,

$$t_{xy} = \sum_{z \in B(x,y)} \left[\sum_{v_a \in S(x), v_b \in S(y), v_c \in S(z)} f(v_a \sim v_c \sim v_b \sim v_a) \right],$$

where $f(v_a \sim v_c \sim v_b \sim v_a)$ is the frequency of triangle $\{v_a \sim v_c \sim v_b \sim v_a\}$ among all characteristic triangles in the prior data base. Similarly, l_{xy} is the total frequency of all characteristic lines around the query protein pair $\{x, y\}$.

We define the *triangle rate score*, $tri(x, y)$ for the protein pair $\{x, y\}$ as the odds of observing triangles versus lines among triangles and lines in its neighbourhood,

$$tri(x, y) = \frac{t_{xy}}{t_{xy} + l_{xy}}.$$

Heuristically, the higher the triangle rate score is, the higher the chance one would observe an interaction between the query protein pair.

5.4.2 The Receiver Operating Characteristic (ROC) curve

In order to put our scores to work we choose a threshold; all pairs with scores above that threshold would be classified as interacting, while all pairs below that threshold would be classified as non-interacting.

The choice of threshold depends on the desired sensitivity and specificity. The *sensitivity* is the fraction of correct predictions among all predicted positive pairs and the *specificity* is the fraction of correct predictions among all predicted negative pairs.

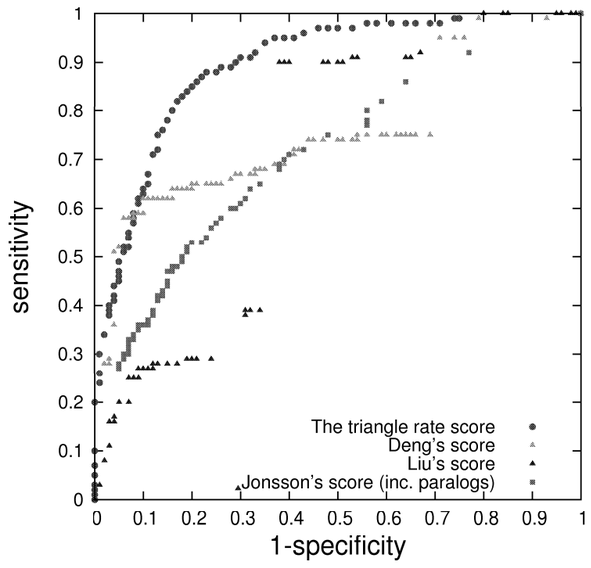
To assess our scores we use a Receiver Operating Characteristic (ROC) curve, which is a useful technique for examining the performance of a classifier; in our case the score “interacting” or “non-interacting” for a pair of proteins. The curve plots sensitivity against (1 minus specificity). Each point on a ROC curve is generated by selecting a score threshold for a method. We move the cutoff along the range of the score and record different sensitivities and specificities of a method. The closer the curve is to the upper left hand corner (i.e., the larger the area under curve), indicating that sensitivity and specificity are both high, the better the predictive score.

Validation procedure

While we are never completely certain that a prediction is correct, we assume that a positive prediction is correct if it is contained in our gold-standard positive (GSP) set, and that a negative prediction is correct if it is contained in our gold-standard negative (GSN) set. The GSP set is based on 8,250 hand-curated interactions in MIPS complexes catalog (MIPS-GSP). These positive interactions are identified if two proteins are within the same complex and if the interactions are confirmed by various experimental techniques. The set of gold-standard negatives (GSN) are random protein pairs which neither share protein localisation, nor expression nor homologous interaction data.

We have many more gold-standard negatives than positives. The unequal sizes of gold-standard sets may affect the ROC curve; when the cutoff is high, too many gold-standard negatives would cause a rapid increase in true negatives, which would result in artificially high specificity. To avoid this bias, we collect 300 samples of randomly selected pairs from the extensive GSN. Each sample is the same size as our GSP set. Predictions are verified against these 300 reference sets obtained by combining the GSP set and the sample from the GSN set. We test the difference between two ROC curves through a z-test for differences, at 5% significance level.

Here are the ROC curves, 1 minus specificity vs. sensitivity, for predicting yeast protein interactions using domain interaction based approaches (Deng's score and Liu's score), a homology-based approach (Jonsson's score plus paralogs) and our network-based approach (the triangle rate score).


sensitivity
1-specificity
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
The triangle rate score
Deng's score
Liu's score
Jonsson's score (inc. paralogs)

6 Statistical inference for networks: modules, motifs and roles.

6.1 Finding Modules

Finding modules, or communities, which make up the network, has been of interest not only for social networks. Statistically speaking, we would like to apply a *clustering method* to find out more about the structure of the network. There is an abundance of clustering methods available.

A much used algorithmic approach is the algorithm by Newman and Girvan, see for example *Newman and Girvan (2004)*. Recall that the betweenness of an edge is defined to be the number of shortest paths between node pairs that run along the edge in question, summed over all node pairs. The algorithm of Girvan and Newman then involves simply calculating the betweenness of all edges in the network and removing the one with highest betweenness, and repeating this process until no edges remain. If two or more edges tie for highest betweenness then one can either choose one at random to remove, or simultaneously remove all of them.

As a guidance to how many communities a network should be split into, they use the *modularity*. For a division with g groups, define a $g \times g$ matrix $\mathbf{e}$ whose component e_{ij} is the fraction of edges in the original network that connect nodes in group i to those in group j. Then the modularity is defined to be

$$Q = \sum_i e_{i,i} - \sum_{i,j,k} e_{i,j} e_{k,i},$$

the fraction of all edges that lie within communities minus the expected value of the same quantity in a graph where the nodes have the same degrees but edges are placed at random. A value of $Q = 0$ indicates that the community is no stronger than would be expected by random shuffling.

Unfortunately the Newman-Girvan algorithm does not provide a measure for statistical significance. The approach by *Handcock et al (2007)* in contrast can assess statistical significance of the clusters, for an Erdös-Renyi mixture model.

They propose a new model, the latent position cluster model, under which the probability of a edge between two nodes depends on the distance between them in an unobserved Euclidean space, and the nodes' locations in the latent space arise from a mixture of distributions, each corresponding to a cluster. They propose two estimation methods: a two-stage maximum likelihood method and a fully Bayesian method that uses Markov chain Monte Carlo sampling. The former is quicker and simpler, but the latter performs better.

6.2 Motifs

Network *motifs* are small subgraphs with a fixed number of nodes and with a given topology. Motifs seem to be conserved across species, suggesting a link between protein evolution and topological features of the protein interaction network. Recently much attention has been devoted to finding motifs which occur more frequently than ”expected”.

Most commonly, significantly over-represented motifs are detected based on a conditional uniform graph test which preserve some characteristics of the network. Instead of a Monte-Carlo test, some authors employ a normal approximation, which is questionable when the motifs are rare or if small p-values are required. For a p-value of 10^{-5}, around 10^{7} simulations would be necessary. Instead a compound Poisson approximation is more adequate in many models.

Picard et al. calculate the mean and the variance for motifs on 3 and 4 nodes in undirected graphs under the models

1. Bernoulli random graph (ER)

2. Random graphs with fixed degree sequence (FDD) (here they estimate the mean and standard deviation from simulations; and they also consider a version with fixed expected degrees)

3. Erdös-Renyi mixture models (ERMG).

Here are some examples of their results. For the mean:

motif	obs	ER	FDD	ERMG
H.Pylo				
2-stars	14,113	5704.08	14,113	13,602.97
triangles	75	10.85	66.91	52.82
3-stars	112,490	7676.83	112,490	93,741.08
E.coli				
2-stars	248,093	52,774.79	248,093	243,846.93
triangles	11,368	72.47	3579.49	10,221.17
3-stars	6,425.495	133,050.00	5,772,005.15	1,537,740.00

Note that in FDD the degree distribution is fixed to equal the degree sequence in the network.

For the standard deviation:

motif	ER	FDD	ERMG
H.Pylo			
2-stars	311.08	0	2659.18
triangles	3.40	7.80	20.41
3-stars	681.76	0	27,039.88
E.coli			
2-stars	1281.87	0	51,676.68
triangles	8.90	68.58	3041.98
3-stars	5089.62	0	1,672.086.51

The expectation and variance strongly depend on the model we choose for comparison. Any ”significance” has to include the model which was used to assess the significance.

We see that the variance is very different from the mean, so a Poisson approximation is not adequate. Also the counts are relatively low compared to the potential number of structures on the networks, thus a normal approximation (z-score) is not appropriate. Instead we use a *Polya-Aeppli distribution*, which is a special case of a compound Poisson distribution. It is obtained when the clump size has a geometric distribution. If the probability that a clump has size k is

$$Prob(k) = a^{k-1}(1-a),$$

and if the number of clumps is Poisson distributed with parameter λ, then we can write down the Polya-Aeppli distribution for the count W,

$$Prob(W = w) = e^{-\lambda}a^w \sum_{c=1}^{w} \frac{1}{c!}\binom{w-1}{c-1}\left(\frac{\lambda(1-a)}{a}\right)^c \text{ if } w = 1, 2, \ldots,$$

and

$$Prob(W = 0) = e^{-\lambda}.$$

The parameters of the Polya-Aeppli distribution can be calculated from the first two moments:

$$a = \frac{var - mean}{var + mean}$$

and

$$\lambda = (1 - a)mean.$$

The Polya-Aeppli distribution is a better fit to the count distribution, compared to the normal distribution; this result is fairly consistent across motifs and across networks. In particular the normal approximation can lead to false positive results: some motifs may be thought as being exceptional while they are not.

When assessing the significance of two or more motifs, their dependence has to be taken into account, or else we resort to the Bonferroni correction, dividing the significance level of our tests by the number of tests carried out and using this as new significance level for each individual test.

6.3 Roles in networks

Following up from network motifs, it is interesting to look at nodes which are ”special” in networks. These could be nodes with high degree, so-called *hubs*, or these could be nodes which have low degree but high between-ness, indicating that they may link fairly separate part of the network, or modules. Depending on the scientific question one may like to identify other roles.

From a statistical viewpoint, we would apply our array of tests and approaches to identify, say, nodes with particularly high degree.

7 Further topics.

There are many further topics, and many more open questions. Here are just two.

7.1 Hierarchical networks

Many networks, such as food webs, have a hierarchical structure. *Ravasz et al (2002)* show that the metabolic networks of 43 distinct organisms are organized into many small, highly connected topologic modules that combine in a hierarchical manner into larger, less cohesive units, with their number and degree of clustering following a power law. Within Escherichia coli, the uncovered hierarchical modularity closely overlaps with known metabolic functions.

Hierarchical structures can in principle be recovered via hierarchical clustering methods. Clustering algorithms return different layers of clusters.

7.2 Dynamics on networks

The yeast interactome exhibits organized modularity where a small proportion of proteins the ’hubs’ interact with many partners. These hubs fall into one of two categories: ’party’ hubs, which interact with most of their partners simultaneously, and ’date’ hubs, which bind different partners at different locations and times. The biological role of topological hubs may vary depending upon the timing and location of the interactions they mediate, see *Han et al. (2004)*. This is just one example where time evolution of networks should be taken into account.

Palla et al (2007) look at the time evolution of overlapping communities in social networks. They define autocorrelation function for the communities as the relative node overlap between communities at time t and $t + 1$. Communities may overlap with several other communities at the same time. But this is just one of many ideas. See Nick Jones' lectures for time series on networks.

A recent good compilations of essays on biological networks is *Kepes (2007)*. Note that the field is very active, googling can often yield interesting result. Enjoy your studies!

Further references

C. J. Anderson, S. Wasserman, B.Crouch (1999). A p* primer: logit models for social networks. Social Networks 21, 37–66.

P. Chen, C. Deane, G. Reinert (2007) A statistical approach using network structure in the prediction of protein characteristics. Bioinformatics, Vol 23, 17, 2314-2321.

P. Chen, C. Deane, G. Reinert (2008) Predicting and validating protein interactions using network structure. Submitted.

J-D. J. Han, N. Bertin, T. Hao, D. S. Goldberg, G.F. Berriz, L. V. Zhang, D. Dupuy,A. J. M Walhout, M. E. Cusick, F. P. Roth, and M. Vidal (2004). Evidence for dynamically organized modularity in the yeast protein-protein interaction network. Nature, 430, 8893.

Handcock, M.S., Raftery, A.E., and Tantrum, J.M. 2007. Model-based

clustering for social networks. Journal of the Royal Statistical Society. **170**:122.

P. W. Holland, S. Leinhardt (1981). An exponential family of probability distributions for directed graphs (with discussion). Journal of the American Statistical Association, 76, 33-65.

F. Kepes ed. (2007). Biological Networks. Complex Systems and Interdisciplinary Science Vol. 3. World Scientific, Singapore.

Newman, M.E.J. and Girvan, M. 2004. Finding and evaluating community structure in networks. Physical Review E. 69 026113.

G. Palla, A.-L. Barabasi, T. Vicsek (2007). Quantifying social group evolution. Nature 446, 664–667.

F. Picard, J.-J. Daudin, M. Koskas, S. Schbath, S. Robin (2008). Assessing the exceptionality of network motifs. Journal of Computational Biology 15, 1–20.

E. Ravasz, A. L. Somera, D. A. Mongru, Z. N. Oltvai, A.-L. Barabsi (2002). Hierarchical Organization of Modularity in Metabolic Networks. Science 297, 1551.

G.L. Robins, T.A.B. Snijders, P. Wang, M. Handcock, P. Pattison (2007). Recent developments in exponential random graph (p*) models for social networks. Social Networks 29, 192-215 (2007). http://www.stats.ox.ac.uk/ snijders/siena/RobinsSnijdersWangHandcockPattison2007.pdf

G.L. Robins, P. Pattison, Y. Kalisha, D. Lushera (2007). An introduction to exponential random graph (p*) models for social networks. Social Networks 29, 173-191.