Discussion Paper

Ensembles of networks

Léon Willenborg

20 January 2023

1 Introduction

The present paper[1)][2)] is about viewing a given network as a realization of a network which belongs to a certain set—or ensemble—of networks. This provides a kind of context for the network given. Of course, there is ususally a range of possible ensembles for a given network. Which one to choose depends on the network at hand and the context in which it is used. A network and an associated ensemble may be compared to a population and an associated superpopulation.

In case one has established an ensemble for a network one can use resampling methods to study the variation of graph parameters, such as their diameter, complexity, etc. This is similar as the goal of bootstrapping in case of a data set with records about individuals, etc.

A network to describe some system in reality can be seen as a kind of living organism, not created in a single step, but evolving in several, if not in many, steps over a period of time. And also, there are often many actors involved, each responsible for creating and developing only part of the network. Think of the Internet as a prime example.

So when we observe such a network at a certain point in time it is like watching a single frame in a film, consisting of a sequence of such pictures. Also, the development of a network is often not deterministic, but stochastic. This is due because its development is not planned, but it grows in a random fashion as the result of decisions of many persons working and deciding independently. So the network we see at some point in time can be regarded as a realization of some random process. It could just as well be a somewhat different network. So instead of focusing on one realization of a network, we would like to focus on a somewhat wider class of networks. We call such a wider class an ensemble, following a tradition in statistical mechanics, of which we shall also make use in the present paper.

An example of an ensemble associated with a network G is defined as those networks that have the same degree distribution as G (if it is a graph) or the same distribution of outdegree-indegree pairs as$G$ (if it is a directed graph, or digraph, for short). In terms of the adjacency matrix of a network the ensemble consists of all those networks whose adjacency matrices have the same (row and column) marginals. This formulation is more convenient, as it does not require to distinguish between graphs and digraphs.

In the present paper we consider ensembles, by taken certain characteristics (properties) Π of a given network G and consider the set of all networks with these properties, which we call the ensemble $\mathcal{E}_{G,\Pi}$.[3)] This association between G and $\mathcal{E}_{G,\Pi}$ is entirely deterministic.

Defining an ensemble is one thing. Another thing is producing elements from an ensemble. Usually it is too big to enumerate all of its elements. And for practical applications this is usually not required either. Instead one would like to generate a sample of elements from this set.

[1)] The views expressed in this paper are those of the author(s) and do not necessarily reflect the policies of Statistics Netherlands.

[2)] The author is grateful to Sander Scholtus for reviewing a mature draft of the present paper and for suggesting several improvements.

[3)] For brevity we drop the suffixes G and Π, as they are superfluous in the context in which an ensemble is discussed. Here we want to emphasize the dependence of an ensemble on these objects.

‘Generate’ usually means that elements of the ensemble are constructed, step-by-step. To proceed with the next step a random procedure is involved, so that the final result is an object that can be viewed as an element that is randomly drawn from the ensemble.

Ideally each element in the ensemble should have the same probability to be drawn into the sample. This is not always straightforward, partiular in cases when the elements ‘drawn’ into the sample are actually constructed ‘from the ground up’. In this way it is possible that an element can be constructed in several ways which yields a higher probability of being drawn/generated. Such elements have a higher probability to be ‘drawn’ than, for instance, elements which can be generated in a single way.

The inspiration for the concept of ensembles came from statistical mechanics and statistical thermodynamics; also the name was borrowed. In these areas of physics one is dealing with systems, such as gases in a container, which consist of many particles (molecules). Describing such a system at the level of individual particles is impossible: there are too many of them and the dynamics (particles colliding with each other and with the walls of the container) are far too complicated. And luckily, for understanding the behavior of a gas at the macro-level, it is not necessary to know the details of the movements of individual particles in the gas. Concepts at the macro-level, such as temperature, pressure, average energy, volume of the container, etc., suffice to describe the behavior of the gas, is such a way that is useful for practical applications. The subject of thermodynamics studies problems related to heat (generation, conduction, dissipation, etc.) of many-particle systems such as gases, fluids, metals, cristals and other solid objects. However, it is also of interest to understand the behavior of such systems at a more detailed level. Not as detailed as that of the individual particles, but at a higher level of aggregation. For instance by considering particles with the same kinetic energy, which corresponds to particles with the same speed.

The main text concentrates on (deterministics) samples, but we consider briefly statistical ensembles (in Appendix D) to illuminate the origin of this concept in physics a bit more and also to point at a possible generalization of the ensemble concept (of networks) as used in the present paper.

The paper deals with networks, which are general directed graphs (digraphs) but which also includes graphs as an important subclass. It is sometimes profitable to single them out, as they are often easier to handle than (general) digraphs. The fact that each arc (a, b) in a graph also contains its counter-arc (b, a) often simplifies things considerably.

The paper suggests several algorithms, but in a rather informal way, with the expectation that most readers can fill in the necessary details, if needed. Due to this hand waving approach, certain issues, such as convergence of the suggested algorithms, have not been checked—let alone be proved—so that it is not guaranteed. It then is up to the reader to come up with a variant that does converge, at least for the cases at hand. As an excuse for this informal approach is that the focus in this paper is on concepts rather than computation.

Algorithms are needed if one wants to draw elements from an ensemble. In fact this ‘drawing of elements’ is usually a construction of an element, using random steps. The question whether all elements of the ensemble have an equal probability to be drawn into the sample is not always an easy one to answer. In the present paper we are satisfied with describing how one can construct random elements from ensembles, without being bothered by the problem just described, for which we do not have a solution (at best an idea how to find one). As it is not clear that the algorithms we suggest stand the scruteny in the light of the problem just mentioned. That is another reason not to spend a lot of time analyzing them and developing them in great detail.

The paper is organized as follows. In Section 2 we define the problem that the present paper addresses. It is about a network G and some of its properties P, which are used to define a set of networks $\mathcal{E}_{G,P}$ associated with G and P, called an ensemble. This may be seen as a generalization of the original network G. This is useful in case one wants to consider the given network as a representative of a class of networks. The original network G and the properties P are its generator.[4] By randomly drawing networks from this ensemble, one can study the distribution of certain characteristics of these networks. In most cases this drawing of elements from an ensemble is actually replaced by constructing them. In the next several sections examples of different types of ensembles are considered. In Section 3 we consider ensembles of graphs with a given number of connectivity components, as well as a given number of nodes and edges. An algorithm is sketched to compute random elements from this ensemble. In Section 4 the approach of the previous section is extended to digraphs. The aim is to define an ensemble of digraphs with a given number of strongly connected components. A strongly connected component is a set of nodes with the property that from any node one can reach all the other nodes in this component. Also it has maximum size: it is not possible to add another node from the digraph and get a bigger set of nodes with the reachability property just mentioned. The strongly connected components of a digraph induce a partition of its nodes, as is the case of connectivity components in graphs. In Section 5 we consider the neighborhoods of nodes in a graph. This is an obvious way of defining ensembles: consider networks that have, in a certain way, the same neighborhood structure as the original network. In our case only the number of neighbors of each node is given (that is, its degree). Also the numbers of neighbors of neighbors is given. So we are dealing with neighbors at distance 1 (direct neighbors) or distance 2 (neighbors of neighbors that are not also direct neighbors). The approach to these problems is algebraic. Section 6 considers ensembles of connected graphs with a given diameter, which is the largest distance of two nodes in a graph, where distance is defined as the length of the shortest path between two nodes. Each edge on a path counts for 1, that is has length 1. Section 7 considers ensembles of trees and ditrees with certain imposed constraints. Such constraints may concern the maximum degree of nodes (in trees). This section is very much about generating trees or ditrees from the various ensembles defined. One of them stems from Albert and Barabási and generates trees with the scale free property: it is likely to have a few big hubs (nodes with large degrees) and many nodes with smaller neighborhoods. A basic procedure is given to generate trees in an ensemble. It is very easy to modify this algorithm so that it can produce trees or ditrees from other, but similar ensembles. It is also very easy to extend it to ensembles of certain graphs or digraphs (which may have loops, which trees or ditrees do not possess). Section 8 considers node clustering, the process in which nodes of the network are combined in a cluster which is viewed as a node in a new, aggregated network. The arcs or edges in this new network are derived from those in the original network. Node clustering implies an equivalence relation on the set of networks, as the clustering of different networks may result in the same network with clustered nodes. Section 9 concludes the paper by discussing the main results as well as work left for future researches. This section is followed by a list of references, which concludes the main text. Four appendices conclude the paper: two with examples of ensembles, one on an interesting and useful operator for numbers and matrices and the final one about statistical ensembles in physics.

[4] Typically, only G is mentioned as the generator of $\mathcal{E}_{G,P}$ and the properties P are forgotten.

2 Properties of networks

In this section we consider some characteristics of networks that can be used to generate ensembles of networks from the original network. The characteristics considered here are relatively easy to generate networks that satisfy them. They are used as an introduction to more complex characteristics that will be considered later in the paper.

The idea is to start with a particular network and identify the characteristics that are interesting. These then are used to define an ensemble (set) of networks that are similar to the original network in certain ways. Such an ensemble can be used in simulation studies, to sample from. Each sampled network can then be used to compute certain characteristics. The sampling procedure provides an insight into the average behavior of networks like the original one.

It should be stressed that it is not always necessary to use a given network as an example and base an ensemble on a certain property (or properties) of that network. One can also base an ensemble on certain desirable properties. Think of networks with a small world characteristic: relatively small neighborhoods of acquaintances for most individuals and large overlap of the neighborhoods of individuals who are neighbors. One can then vary the size distributions of the neighborhoods, as well as the distribution of the degree over overlap of the neighborhoods of acquaintances. This example is discussed below as item 13.

When discussing the characteristics it is sometimes useful to distinguish between graphs and digraphs, whereas in other cases such a distinction is not necessary. A graph is in fact a special kind of digraph, with the property that if (a, b) is an arc so is (b, a), its counter-arc. Usually this property makes things easier for graphs than for general digraphs. The relevance of this distinction comes to light when generating a network from an ensemble.

Examples of ensembles and their characteristics are the following:

1. Networks with a given number of nodes and edges / arcs.
2. Graphs with a given number of connectivity components.
3. The number of strong connectivity components.
4. Graphs with a given diameter.
5. Graphs with a given distribution of degrees of the nodes.
6. Digraphs with a given distribution of indegree and outdegree combinations of the nodes.
7. Graphs with a given number of cliques. [5)]
8. Graphs with a given number of elementary cycles.
9. Graphs with a given cycle structure.[6)]
10. Trees with a given number of nodes (and hence edges). This is a special case of item 5.
11. Trees with a given number of nodes and distribution of degrees of the nodes. This is a special case of item 6.
12. k-regular graphs with a given number of nodes and degree k of each node.

[5)] A clique in a graph is a maximal sub-graph in which every pair of nodes is an edge.

[6)] For instance by starting with a graph with cycles (not a tree), determine the cycle matrix. Then consider its rows sums (which count the total number of edges on each elementary cycle) and its column sums (which count the number of elementary cycles that contain a given edge). So instead of providing the cycle matrix one only provides its two marginals.

13. Networks with neighborhoods with a controlled overlap (NCOs). If i, j are nodes and N_i, N_j are their respective neighborhoods $|N_i \cap N_j|/|N_i \cup N_j|$ should be big enough, say at least $\alpha > 0$.
14. Networks which have a certain cluster network in common.

A cluster network is a network that is obtained from a given network by clustering the nodes of the original network. The clusters form nodes in the cluster network. Two clusters A and B are connected by an edge $\{A, B\}$ if there is a node $a \in A$ and $b \in B$, in the original network with $\{a, b\}$ and edge. Likewise for arcs: (A, B) is an arc at the cluster level provided (a, b) is an arc in the original network. In case one finds that for clusters A and B both (A, B) is an arc and $\{A, B\}$ is an edge, then one concludes that $\{A, B\}$ is an edge, as this is the more inclusive structure.

To illustrate item 5, there is an example in Appendix B featuring a sample of graphs in an ensemble $\mathcal{E}_T$ generated by some graph T. These graphs possess the same number of nodes and edges, with the same degree distribution as T).

When an ensemble is defined in terms of a property, or a set of properties, it is not necessarily clear that the ensemble is non-empty, unless a particular network is chosen some of whose properties are used to define the ensemble. So checking that an ensemble is not empty is a first task that one has to perform, in particular if no network is used to ‘generate’ the ensemble.

A second question would be to count (or estimate) the size of the ensemble. This is a combinatorial enumeration problem, which can be quite tough to answer. It is perhaps well-advised to skip this problem and turn to the third one, which is of great practical value: how to generate a sample of elements from the ensemble, ideally in such a way that each element has equal probability to be drawn into the sample. Sometimes the drawing process may in fact be a construction process of a specimen graph from the ensemble.

It is therefore of interest to also define a sampling scheme such that the networks in the ensemble have the same probability to be drawn into the sample. Because we started with an existing graph this ensemble is therefore not empty, since at least the original graph is an element of it. Hopefully, there are many more networks in the ensemble.

The applications of ensembles are to compute certain average properties of its members, or distributions of properties, which are too complex to compute analytically. We start with considering ensembles of graphs and then move on to ensembles of digraphs.

In the following sections several ensembles of networks are discussed, which are based on some of the characteristics mentioned in Section 2. Of particular importance is how ensembles are made operational, that is, how to (randomly) generate members of the ensemble so they can be used in computations, in particular simulations. Ideally this should be done in such a way that each member has equal probability to be drawn in a sample. But to achieve this is not always easy.

Since we generate an ensemble by generalizing the properties of a specific network we know from the outset that the ensemble is not an empty set. We will not address the problem of actually counting the size of an ensemble, as this kind of problem is usually not so easy. It requires dedicated work which is beyond the present paper.

3 Connectivity components

We are dealing with graphs. For a graph a connectivity component is defined as a maximum set of connected nodes, which means that any pair of them can be connected by a path within the (sub)graph.

Suppose the number of connectivity components of the original graph is k. We are interested in an ensemble with this number of components and the same number of nodes and edges as the orginal graph, say n and m, respectively.

How can this ensemble be generated? We sketch a possible approach, with the details left out. First divide the n nodes into k nonempty clusters. Then make sure that the nodes in each cluster are connected, using the minimum number of edges. If a cluster has ℓ nodes, $\ell - 1$ edges are needed. They yield a tree. If some of the m edges are still left, randomly select any of the clusters that are not saturated with edges (for instance, the clusters with a subgraph with 2 nodes) and for each edge randomly select a feasible cluster. Repeat this until all the remaining edges have been allocated.

4 Strongly connected components

In this case we are dealing with digraphs. Suppose that we start with a particular digraph, with n nodes, m arcs and k strongly connected components. A strongly connected component is a maximal subgraph such that any pair a, b of nodes are mutually reachable. 'Maximal' means that no node can be added while preserving the reachability property. A digraph can be partitioned into strongly connected components.

With a digraph G one can associate a new digraph H, as follows. The vertices in H are the strong components of G. The arcs in H present arcs in G that point from a node in one strongly connected component to a node in another strongly connected component. There may be several of these arcs in G, all represented by a single arc in H. In H all nodes are strongly connected components. Hence nodes are the only strongly connected components in H.

How to generate similar digraphs? The idea to do so is very similar to that given in Subsection 3. First we divide the nodes into k (disjoint) clusters. For each cluster with at least two nodes, we orden the nodes in some way. Each node, except the last one, is connected with its direct successor; the last node is connected to the first node. This (directed) cycle structure ensures that each pair of its nodes are connected (in both directions), so that this cluster is a strong component. Proceeding in this way with every cluster with at least two nodes, we arrive at the situation where we have k (intermediate) disjoint clusters, and if there are still arcs left, we are adding them one by one. In a cluster of n_g nodes connected by a (directed) cycle (using n_g arcs, we still have room for $\binom{n_g}{2} - n_g$ more arcs. Also arcs connecting two clusters can be used as 'extra arcs'. One has to be careful. If there are two clusters A and B, one has to point the connecting arcs always in the same direction, to prevent violation of the maximum principle for each node.[7] If A has $|A|$ nodes and B has $|B|$ nodes, there are $|A| \cdot |B|$ ways to connect them, always from A to B (or vice versa, whichever the chosen direction is).

It is clear that the procedure described above yields digraphs with the prescribed number of nodes, arcs and strongly connected components. But does the procedure do so giving them all equal probability? This is not clear, and is open to investigation. If it is not, then the question is, which procedure could be used to generate the graphs from.[8]

The problem considered in the present subsection should be contrasted to that in Subsection 3. The latter is much easier to handle.

5 Neighborhoods and degrees

For graphs the degree of a node is the number of other nodes it is connected to via an edge. Otherwise it is the row (or column) sum of the adjacency matrix. Another view is that the degree of a node is the size of the neighborhood, in terms of direct neighbors. Of course, one can extend this idea and define neighborhoods in a wider sense, not only including direct neighbors but also neighbors of neighbors, etc. For graphs ‘being a neighbor’ is a symmetric property: if a is a neighbor of b then b is also a neighbor of a. In digraphs this need not be the case.

A possibility to define an ensemble is by specifying the number of nodes of the graphs as well as a degree distribution for the nodes.

5.1 k-regular graphs

A simple case of an ensemble based on a degree distribution is by specifying the number of nodes of the graphs in the ensemble, say n, as well as the degree of each node, say k. In that case we obtain the ensemble of k-regular graphs, consisting of n nodes. In Appendix A examples of 3-regular graphs with 10 nodes, as well as with 12 nodes, are shown. The famous Petersen graph is a member of the first ensemble. □

5.2 Given distribution of degrees

We now consider an example of an ensemble that is generated by the graph which is depicted in Figure 5.1. It is a graph with the following properties. It has

- 15 nodes, of which
  - 4 nodes of degree 1.
  - 8 nodes of degree 2.

[7] If the connecting arcs are in both directions then they join A and B to create a bigger strongly connected component, which is to be avoided.

[8] For a quick introduction to strongly connected components and several useful related algorithms see the Wikipedia.en lemma on this topic (`https://en.wikipedia.org/wiki/Strongly_connected_component`).

  - 1 node of degree 3.
  - 1 node of degree 4.
  - 1 node of degree 5.
- $16 = (4 \cdot 1 + 8 \cdot 2 + 1 \cdot 3 + 1 \cdot 4 + 1 \cdot 5)/2$ edges.

We shall take these properties as the defining ones of the ensemble that we are interested in in the present section. The graph in Figure 5.1 is also connected, but we do not consider this property of relevance for our ensemble. Topologically this graph is defined by how the subgraphs consisting of 3, 4 and 5 ‘spikes’ are mutually connected.

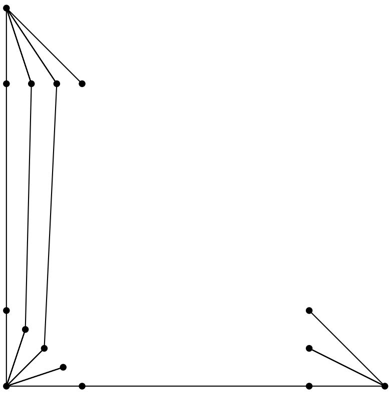

Figure 5.1 The initial graph.

5.3 Neighbors in graphs

We now want to describe the problem to find the ensemble consisting of graphs with a given distribution of degrees (=number of neighbors) in an algebraic setting.

In case G is a graph the situation can be represented as in Table 5.1, where the b’s are the unknown entries of an adjacency matrix B corresponding to a graph with the same immediate neighborhood structure (distance 1) as the original graph with adjacency matrix A. The δ’s are the degrees of each of the nodes of G. They are functions of A, namely row sums or column sums. The graph we are looking for has the same number of nodes and edges or arcs as G. . Furthermore, the graph should be loop-free, as is G, which implies that $b_{ii} = 0$, for $i = 1, \ldots, n$. As the networks we are looking for are graphs then $B' = B$, i.e. $b_{ij} = b_{ji}$, for $i, j = 1, \ldots, n$.

b_{11}	…	b_{1n}	δ_1
⋮	⋱	⋮	⋮
b_{n1}	…	b_{nn}	δ_n
δ_1	…	δ_n	

Table 5.1 Tableau for a graph, with degrees (columns at the right and row at the bottom).

The δ values in Table 5.1 are obtained from the original matrix, as follows. If we let A denote the adjacency matrix of the original graph $G = (V, E)$. Of course, $A = A'$. We then have

$$\delta = (\delta_1, \ldots, \delta_n)' \triangleq A\iota, \tag{1}$$

where ι is the all 1 vector of length $n = |V|$. In fact δ is a vector with the degree d_i of node i as its i-th entry. We may assume that $d_i > 0$ for all i. If this would not be the case for some i_0, delete row and column i_0. Node i_0 is isolated and should be discarded from our computations.

We start with a weight matrix $W = (w_{ij})$ and require that $W > 0$, that is, each $w_{ij} > 0$ and with $W = W'$. The idea is now to distribute the marginals, i.c. δ for the rows as well as the columns, as we are dealing with graphs—so that indegree equals outdegree for each node—over the interior of the matrix using Iterative Proportional Fitting (IPF); see e.g. [1]. W is the matrix that regulates the mass distribution of the marginals over the 'interior' cells of the matrix. That W should be a strictly positive matrix is discussed below; it is for a technical reason, which, fortunately, is not a restriction in practice.

The question now is how to choose W. There is no definitive best answer it seems. We can only make some remarks and observations to guide one to a sensible choice. We can consider an approach from information theory. In general this leads to considering the marginals as independent distributions (when properly normalized). We then obtain for interior cell (i, j) the probability:

$$p_{ij} = p_{i\cdot}p_{\cdot j} \tag{2}$$

But this is under the assumption of independence of the column and row vectors. However, in practice this is not always the case. For instance when some entries are structurally 0. Or they are 0 for another reason: for instance, the diagonal cells are 0 because we do not want loops in the graphs that are in the ensemble. Now we can again apply information theory, in the form of entropy maximization under constraints (in this case using the fact that some interior cells are 0). The nonzero cells are given the same value as we have no particular information to distinguish one from the other. Of course, if there is other prior information about the interior cells, we can use that as well. In practice, we have to replace 0 values in the interior by very small positive values. This is only a trick to be able to apply Sinkhorn's theorem, as discussed below.

We can also use our knowledge about the initial graph, that is the generator of the ensemble. We could produce a weight graph W that uses this information (in the form of its adjacency matrix) in combination with the matrix expressing almost total ignorance (except for 0 cells). The weight matrix could be chosen as a convex combination of an ignorance matrix N and the adjacency matrix A of the initial graph

$$W = \lambda_N N + \lambda_A A, \tag{3}$$

where $\lambda_N, \lambda_A \geq 0$ and $\lambda_N + \lambda_A = 1$. The weights λ_N, λ_A can be used to express a stronger or weaker bias for the initial digraph.

Let the resulting matrix from the IPF application be denoted by F. Now, in a second IPF step we use F as the interior matrix and the all-ones vector ι as marginals. In this way we compute a doubly stochastic matrix D.[9]

[9] A doubly stochastic matrix is a nonnegative matrix in which each row and column sum equals 1.

In both IPF steps we tacitly assumed that IPF converges to a solution. In both cases convergence is guaranteed by Sinkhorn's theorem (see [10], [11], [5] and [4]), which reads:

Sinkhorn's theorem For a matrix $X > 0$ there are diagonal matrices $\Delta_1, \Delta_2 > 0$ such that $\Delta_1 X \Delta_2$ is a doubly stochastic matrix. The matrices Δ_1, Δ_2 are unique, except for a proportionality factor $q > 0$ so that $q \cdot \Delta_1$ and $(\frac{1}{q} \cdot \Delta_2)$ yield the same stochastic matrix.

Remark As noted before, all entries of W have to be strictly positive in order to be able to apply Sinkhorn's theorem. This condition is often too severe in practice. Even if some some entries are 0 IPF may still converge and yield a (unique) solution. However if the number of 0's increase one may well find weight matrices with some entries equal to 0 that do not yield a solution. The problem is that it is not easy to find necessary conditions for the weight matrices to yield a solution. But the whole matter is somewhat academic, as one can replace the 0s by small values $\epsilon > 0$ and adjust the nonnegative values accordingly. □

The next step is to use the doubly stochastic matrix D, to randomly generate edges, using a sampling procedure without replacement. The idea is to generate an edge (a 1 at some entry (i, j)). Then the values of the marginal totals are updated: δ_i and δ_j are both decreased by 1, as we are dealing with graphs and not merely digraphs. If one of these new values is 0, the corresponding rows and columns are deleted, also from the probability matrix for the interior cells. This is then also updated, so that a new doubly stochastic matrix is obtained. We are then in a similar position as before and we can draw the next edge, provided the remaining marginal distribution vector has not grouped to 0. We then repeat this procedure until no more edges can be drawn. One finishes with a randomly generated adjacency matrix of an element of the ensemble.

5.4 Neighbors of neighbors

We now want to extend the scope of the neighborhoods of each point in the network. Instead of looking at nodes at distance 1 from each node as in the previous section, we now look at sets of nodes at distance 2 from each node in a network. So these are second degree neighbors, or neighbors of neighbors.

We start with a G with adjacency matrix A. This matrix in fact contains information of the neighbors of each node i: row i provides its outgoing arcs and column i its ingoing arcs. Now we can compute the adjacency matrix A_2 of $G_2 = (V, E_2)$ which provides information of the nodes at distance 2 as follows:

$$A_2 \triangleq [A^2], \tag{4}$$

where $[\cdot]$ is the 01- operator (see Appendix C). It operates on nonnegative numbers and matrices and simply indicates if such a number is zero or positive, and for matrices it indicates this for each entry.

Example The 01-operator $[\cdot]$ can be illustrated by the following examples, for scalars: $[0] = 0$, $[5] = 1$, and for matrices:

$$[\begin{pmatrix} 5.2 & 3 \\ 0 & 7.6 \end{pmatrix}] = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}. \tag{5}$$

□

The entry at location (i, j) in A^2 in (4) equals the numbers of paths of length 2 from node i to node j, whereas the entry at (i, j) in $[A^2]$ indicates whether there is a path of length 2 from i to j (if the entry is 1) or not (if the entry is 0). So $[A^2]$ is the adjacency matrix of the digraph G^2 derived from G by considering the nodes at distance 2 in G from each node: i in G: (i, j) is an arc in G^2 if and only if there is a node k in G such that (i, k) and (k, j) are arcs in G.

So we are now facing the inverse problem: given A_2 we look for adjacency matrices A such that (4) holds. Unfortunately we have no solution to offer here for this problem.[10)]

Remark The operator $[\cdot]$ can also be used to define ensembles with more distant neighbors as well. Let G be a digraph with adjacency matrix A and let $k > 2$ be an integer. Consider the ensemble $\mathcal{E}_k$ consisting of all $(0, 1)$-matrices B of the same order as A such that $A_k \triangleq [A^k] = [B^k]$. The matrices B are to be interpreted as adjacency matrices of digraphs. The challenge, again, is to find these matrices B. □

6 Diameter

The diameter is defined for graphs. Let G be a graph. The diameter of G is the length of the longest path connecting two points in G.

The goal is to define an ensemble of connected graphs with a given diameter d. A simple example of a graph with diameter d is the linear graph with $d + 1$ nodes (and hence d edges).

When starting to generate graphs with a given diameter d one can start with this linear graph and then add edges and nodes where this is possible and choosing randomly from possible positions at each step.

A more difficult task is to generate connected graphs with a given diameter d as well as a given number of nodes n. In fact these parameters cannot be chosen independently of each other. For a given diameter d, the possible number of nodes n should be in a particular range (depending on d) to be compatible.

[10)] This does not necessarily mean that it is a hard problem, but rather that we did not have enough time to study this problem more thoroughly. We leave it for the interested reader to tackle.

7 Trees

The ensemble we are interested in consists of trees of a given number of nodes n (and hence edges $n-1$). The problem we consider is how to generate such graphs. We provide a simple method that builds a tree in the ensemble iteratively.

So an algorithm to generate random graphs with a given number of nodes, could be as follows. Start with a node. In the next step, add a new node and connect it to the previously generated node. Repeat the generating of a node, and the decision to connect it to a previously chosen node. Choose such a node at random. Stop this iteration if the tree obtained has n nodes.

This construction can be modified to generate different ensembles as well. We mention a few of them. For instance, instead of randomly choosing a node from those previously obtained with equal probability, we draw a node proportional to its current degree. In this case we produce trees that favor hubs, that is, nodes with large degrees (see also [2], a copy of which is also included in [6], pp. 349–352).

Another construction uses an upper-bound for the degree of any node in the tree. If a previously generated node has reached this bound at some step, it cannot be a candidate in a next step to connect it to a new node.

In another modification we obtain an algorithm that generates random digraphs. If we look at the base construction, we modify it as follows. With each new node ν we pick a previously generated node μ and randomly select either (μ, ν) as a new arc, or (ν, μ). So not only the connecting node is randomly chosen, but also the direction of the arc.

With this modification of the base algorithm it is easy to carry on and generate another ensemble of digraphs, in which the indegrees and/our outdegrees are limited by predetermined upper-bounds.

By another construction, we first randomly choose the direction of a new arc. If its head is the new node, we connect its tail to a randomly chosen previously generated node. If its tail is the new node, we connect its head to a previously chosen node proportional to the indegrees of the possible candidate nodes. In this way we simulate the attractiveness of certain nodes, symbolizing, say, (popular) websites or (favorite) persons.

Remark Above we describe some simple algorithms to generate various ensembles of trees and ditrees. But with a slight modification of the drawing procedure we can generate all kinds of ensembles of graphs or digraphs that contain loops.

We first define a base procedure for generating general graphs, with or without loops (i.e. trees). We start with the first node and generate a second one, and connect it to the first node (with an edge). Then we generate a third node, and we select one of the first nodes to connect it to (with an edge). Now we have three nodes, not all of which are connected directly. We choose a pair of nodes that can be connected. We throw a coin to decide to connect them or to add a new node and connect it at random with one of the previous nodes. In this way we proceed: at any step we throw a coin to decide to connect two existing nodes not already connected, or to add a new node and connect it randomly to a previous node. We repeat this until the graph we are constructing step-by-step has the desired number of nodes.

It is easy to modify this procedure to limit the degrees of all the nodes, to introduce arcs instead of edges, to work with hubs, etc. We shall not spell out the details, leaving it to the interested reader to provide them. □

8 Node clustering

Node clustering (NC) is a process to compute a network from a given one, as follows. The process is the same for graphs and digraphs, so there is no reason to discuss the method separately for these structures.

Starting from the original network, the idea is to cluster randomly chosen nodes into clusters which are the nodes in the cluster network. Let A and B be such clusters. There is an arc from A to B, denoted as (A, B), if (and only if) there are nodes $a \in A$ and $b \in B$ such that (a, b) is an arc in the original graph. If there is also an arc from B to A, we have an edge $\{A, B\}$.

We may also have to consider loops in the cluster network, as (different) nodes in the same cluster may be connected by arcs or edges in the original network. So if $a, b \in A$ where a, b and A are as above, and $a \neq b$, there may be an arc (a, b) in the original network, which would imply a loop (A, A) in the cluster network.

We may use generalized graphs for this. They allow parallel arcs between nodes as well as loops. The clustering operation amounts to a contraction of the adjacency matrix of the original graph. This is done by first partitioning this matrix according to the clusters, followed by adding the cell values within each part (corresponding to a cluster).

Let the adjacency matrix A be as in (6).

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{n1} & \cdots & a_{nn} \end{pmatrix} \tag{6}$$

If we assume that the set of nodes is partitioned into p parts, the adjacency matrix A can be partitioned correspondingly. We assume that nodes in the same partition are next to each other. This may require relabeling of the nodes. The resulting partitioned matrix, Q, is represented in (7).

$$Q = \left(\begin{array}{c|c|c} Q_{11} & \cdots & Q_{1p} \\ \hline \vdots & \ddots & \vdots \\ \hline Q_{p1} & \cdots & Q_{pp} \end{array}\right), \tag{7}$$

where each partition matrix Q_{ij} is a submatrix of A. From Q we determine a new (generalized) adjacency matrix $\mathcal{B}$ in (8). It is generalized because it may describe a generalized digraph, that includes (multiple) loops and parallel and anti-parallel arcs. The number p of nodes of such a structure will be smaller than the number n of nodes in the original digraph.

$$\mathcal{B} = \begin{pmatrix} \mathfrak{b}_{11} & \cdots & \mathfrak{b}_{1p} \\ \vdots & \ddots & \vdots \\ \mathfrak{b}_{p1} & \cdots & \mathfrak{b}_{pp} \end{pmatrix} \tag{8}$$

Example The digraph in this example is depicted in Figure 8.1. It consists of 12 nodes and 14 arcs. The underlying graph consists of two components: one consisting of the nodes numbered $9, 10, 12$ and the other of the remaining nodes $1, 2, 3, 4, 5, 6, 7, 8, 11$.

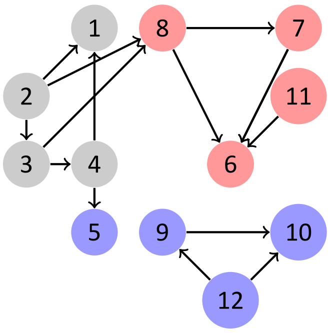

Figure 8.1 A directed digraph with some clusters of nodes indicated.

The adjacency matrix of the digraph in Figure 8.1 is presented in Table 8.1.

Node	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	0	0	0	0	0	0	0	0
2	1	0	1	0	0	0	0	1	0	0	0	0
3	0	0	0	1	0	0	0	1	0	0	0	0
4	1	0	0	0	1	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	1	0	0	0	0	0	0
8	0	0	0	0	0	1	1	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	1	0	0
10	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	1	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	1	1	0	0

Table 8.1 The adjacency matrix of the digraph in Figure 8.1

We now define three clusters of nodes:

- cluster I: nodes 1, 2, 3, 4.
- cluster II: nodes 6, 7, 8, 11.
- cluster III: nodes 5, 9, 10, 12.

In Figure 8.1 the nodes belonging to the same cluster have been colored in the same color: gray for cluster I, red for cluster II and blue for cluster III. Table 8.2 shows the results of aggregating the information from the adjacency matrix per cluster. Note that there are arcs between clusters and also within clusters. The resulting matrix in Table 8.2 can be viewed as a generalized adjacency matrix, corresponding to a generalized digraph, which has multiple loops per node (being a cluster) and parallel arcs in some cases between different nodes (clusters).

So we cannot present them in a partitioned matrix as in (7), unless we would relabel them, which we do not want to do.

Node	I	II	III
I	4	2	1
II	0	4	0
III	0	0	3

Table 8.2 The adjacency matrix of the clustered digraph in Figure 8.2

Of course, Table 8.2 is not a standard adjacency matrix, but one that belongs to a generalized network, in which there are parallel arcs and edges and loops. If we discard loops and represent parallel arcs and edges by a single arc or edge representing them all, we obtain an ordinary adjacency matrix. See Table 8.3. This digraph is represented in Figure 8.2.

Node	I	II	III
I	0	1	1
II	0	0	0
III	0	0	0

Table 8.3 The adjacency matrix of the digraph derived from the cluster digraph with adjacency matrix in Table 8.2

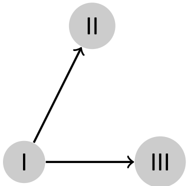

Figure 8.2 Digraph showing the connection of the clusters of nodes. Not shown are parallel arcs and loops in this picture of what is a generalized digraph.

So the digraph pictured in Figure 8.2 is the digraph representing the network after node clustering. It discards loops and parallel edges and arcs. So any network that after node clustering results in this digraph is part of the ensemble. An extra requirement would be that the number of nodes and arcs should be the same as that of the original network that implied the ensemble. Looking at Figure 8.2 we can immediate deduce some properties of the networks that transform into this network: there is at least one arc (a, b) with a in cluster node I and b in cluster node II. The same holds for cluster node I and cluster node III instead of cluster node I and cluster node II, respectively. Also, no arcs from cluster node II to cluster node I, no arcs from cluster node III to cluster node I. No edges between cluster node II and cluster node III, and hence also no arcs connecting these cluster nodes. There is no restriction to the numbers of edges or arcs from nodes in cluster node I. The same holds for cluster nodes II and III.

This example should illustrate the idea of node clustering and how it generates an ensemble from a given network and a specific node clustering applied to it.

Remark Node clustering in networks reminds us of the description of many-particle systems in physics (more particularly in statistical thermodynamics) such as gases. Three levels of description can be considered: at the extremes we have the molecular level on the one hand and the macro-level at the other. This describes a gas by parameters such as temperature, pressure, volume (of the container of the gas), etc. For practical purposes the macro-level is most suited. At this level, it is totally irrelevant to posses information about position and speed of the individual gas molecules at any moment in time.[11] But apart from this, a complete description at the molecular level of the gas is impossible. There are far too many particles, too many interactions (collisions), and small errors of measurement in position and speed result in large uncertainties concerning these parameters after some time. For this reason a third level is chosen, which is intermediate to the molecular level and the macro-level. We call this level the micro-level. At the micro-level we have various energy levels. Each level involves a sizeable number of molecules. Yet compared to the macro-level it is fairly detailed, although not als detailed as the molecular level. But the micro-level is sufficiently coarse to be of any use.

It is important to understand how the macro-level is linked to the micro-level. This is the realm of both statistical thermodynamics and statistical mechanics. The micro-level chosen as the starting point of the description is not the individual particle in the gas, but energy levels at the micro-level, say the kinetic energy. One considers the distribution of speeds of the particles, which are numbers, not vectors. Under certain assumptions one can derive the speed distribution of the gas molecules, at the micro-level, namely the Maxwell-Boltzmann distribution.

The important point we want to make is that the description used in physics for a gas is that it involves several levels of detail: a micro-level and a macro-level. Also a probability distribution on the microlevel (of energy) is used to link it to a known (i.e. measurable) average energy at the macro-level of the gas, which is the realm of thermodynamics.

In the present section we can view the initial graphs as elements at the microlevel (like the energy levels defining the micro-levels of a gas) and the induced networks by the node clustering operation as macro-level objects. With one macro-level object several (or even many) micro-level objects are (or may be) associated.

The approach in physics extends to the one taken in the present section, in the following sense. Each state at the macro-level corresponds to a set of states at the micro-level together with a probability distribution defined on the micro-states. This probability p_i for micro-level state i is used to weight the micro-level quantities defining them, such as the energy E_i, such that the weighted energy $\sum_i p_i E_i$ equals the average energy $\bar{E}$ measured at the macro-level. Such ensembles, consisting of micro-states with probabilities defined on them, are called statistical ensembles in physics. It is convenient to borrow this name for the generalization of our approach to ensembles in the present paper. A bit more information on statistical ensembles in physics can be found in Appendix D, which also contains references to the physics literature in case one is interested in an extensive coverage of this topic.

It should be noted that an energy concept for networks can be defined (see [12]) that may be useful in the context of statistical ensembles of networks. □

9 Discussion

The aim of this paper is to introduce the concept of ensembles of networks, either graphs or digraphs, and illustrate this concept with a few examples. The goal of working with ensembles is

[11] The molecules in a gas move in other ways as well: they are spinning around, they vibrate and they collide with each other and with the walls of the container in which they are trapped.

to be able to perform computations, i.c. simulations, on networks with certain characteristics they have in common with an initial network. The ensemble thus created can be seen as a kind of neighborhood of networks resembling the initial one. The ensemble can be used to generate similar networks to the initial one and compute certain things with them. By using a sample of these networks one may gain an idea how the properties one is interested in fluctuate. The underlying idea for taking this approach may also be that one considers the initial network as a random realization of a network, that could just as well be different, similarly as one may view a population as a realization of some superpopulation. This empirical approach may be a lot easier and less cumbersome than using an analytical, non-experimental approach.

A particular problem is to generate elements from an ensemble. Ideally one would like to have a sampling scheme, or a construction method, that generates specimen of the ensemble in such a way that each member of the ensemble has equal probability of being drawn or constructed. This, however, is not always easy. One may find a sampling or construction scheme, but it is not always clear that it produces all elements of the ensemble with equal probability.

A suggestion[12)] for a possible approach to solve this problem is to look at the Metropolis-Hastings algorithm. See e.g. [7], ch. 6. The idea of this method is to use an ergodic Markov chain with a Markov matrix P such that the distribution v (a row vector) one is interested in (in our case, a uniform distribution, that is $v \sim (1, \cdots, 1)$) is an invariant measure of P, that is $vP = v$ holds. If the Markov chain is 'run' for some time, we have $v_k \to v$ if $k \to \infty$, where v_k is the frequency vector of visited states until time k. The challenge is to adopt this idea to the construction process of ensemble elements in the sample. In quite some cases elements from an ensemble are not drawn 'ready made', but they are constructed, where the construction involves random choices. Using the idea behind the Metropolis-Hastings method, this would imply a change of construction method as suggested in some examples: not to construct each element from scratch, but to modify the current element in a particular way to obtain another element in the ensemble considered. How this should be done is not clear to the present author at the time of writing. It is left as an interesting topic for future research.

To illustrate the intricacy of the concept of ensemble, we considered several examples in varying degrees of detail. These examples also show that there is a variety of techniques to draw or construct random elements that belong to the ensemble. It is very likely that the examples considered do not exhaust the techniques that can be used to draw a sample, or to construct random elements from the ensemble. It is likeley that if more ensembles of networks are studied, more methods to generate random elements will be developed. In particular, it is not only of interest to generate random elements from an ensemble, but also also to do this in such a way that all elements of the ensemble have equal probability of being 'drawn into the sample'. Or, more generally, in case of statistical ensembles, that elements are drawn with a predefined probability.

Of special interest may be the generation of specimen from intersections of ensembles. We have not systematically investigated these objects. In some examples we showed how extra restrictions could be introduced. But these were easy cases. In general this may be a lot harder.

So it is clear that the interesting subject of ensembles of networks leaves enough challenges to be explored in the future.

[12)] Made by Sander Scholtus when he reviewed the current paper.

References

[1] M. Bacharach (1970). *Biproportional Matrices and Input-Output Change*. Cambridge University Press.

[2] A.-L. Barabási & R. Albert (1999). Emergence of Scaling in Random Networks, *Science*, **286**, 509–512.

[3] K. Huang (1963). *Statistical Mechanics*. Wiley.

[4] M. Idel & M. Wolf (2015). Sinkhorn Normal Form for Unitary Matrices. *Linear Algebra and Its Applications*, **471**, 76–84.

[5] A. Marshall & I. Olkin (1967). Scaling of Matrices to Achieve Specified Row and Column Sums. *Numerische Mathematik*, **12 (1)**, 83–90.

[6] M. Newman, A.-L. Barabási & D. Watts (2006). *The Structure and Dynamics of Networks*, Princeton University Press.

[7] C. Robert & G. Casella (1999). *Monte Carlo Statistical Methods*. Springer.

[8] E. Schrödinger (1962). *Statistical Thermodynamics*. Cambridge University Press.

[9] F. Schwabl (2002). *Statistical Mechanics*. Springer.

[10] R. Sinkhorn (1964). A Relationship between Arbitrary Positive Matrices and Doubly Stochastic Matrices. *Ann. Math. Statist.*, **35**, 876–879.

[11] R. Sinkhorn & P. Knopp (1967). Concerning Nonnegative Matrices and Doubly Stochastic Matrices”. *Pacific J. Math.*, **21**, 343–348.

[12] L. Willenborg (2022). Elastic Energy of Networks. Discussion Paper. Statistics Netherlands, The Hague.

Appendix

A Examples of 3-regular graphs

In this appendix some special networks have been included. Examples of regular graphs are shown, one set consists of graphs with 10 nodes each of degree 3, the second set consists of graphs with 12 nodes each of degree 3. There was no attempt to be complete, only to give some examples of each class of graphs. The pictures of these graphs can be found in Figures A.1 and A.2.

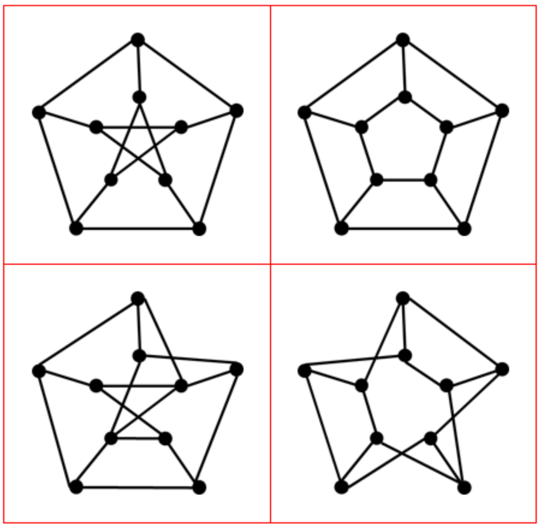

Figure A.1 Examples of 3-regular graphs with 10 nodes. Top left: the famous Petersen graph.

The third graph from the top in the right-most column of Figure A.2 is like a Möbius band: A non-orientable surface. The orientable counterpart of this graph is the one on the top left of Figure A.2.

It may be instructive to point out the relationship between these two graphs. The 'inner pentagon' of the variant of the Petersen graph (the one at the top right of Figure A.1) can be taken as a separate graph, for the moment. If we take the graph that describes the two-step graph, we get the pentagram at the inside of the Petersen graph. Note that 5 nodes at the outside are similarly connected in both graphs. These graphs differ only in the connections of the 5 inner nodes.[13)]

[13)] This description is based of the pictures used for both graphs in Figure A.1, which are representations of these graphs. For the abstract graphs there are no inner and outer nodes.

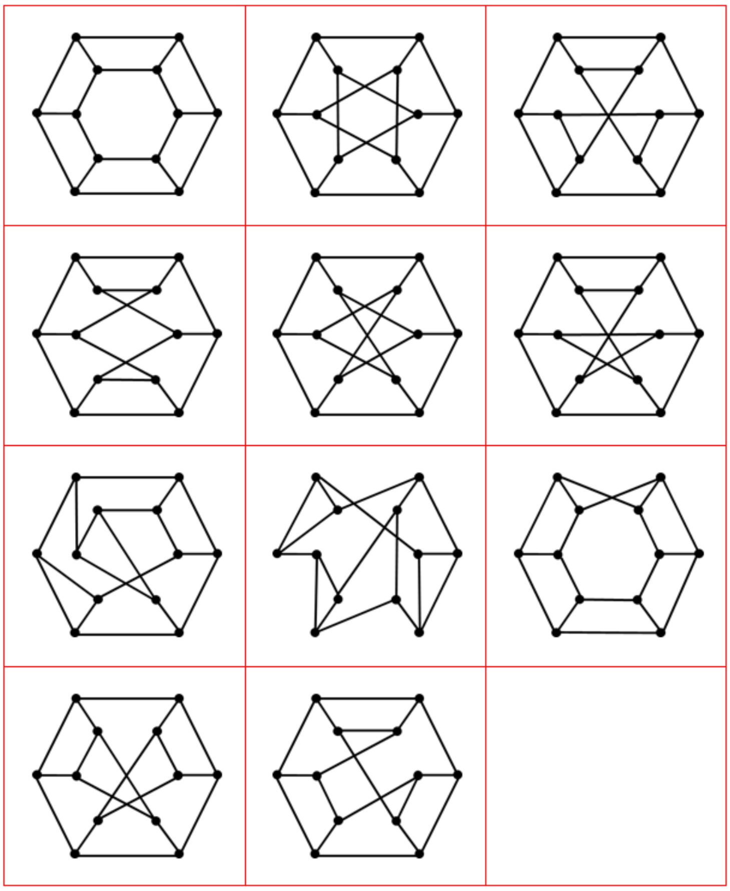

Figure A.2 Examples of regular graphs with 12 nodes and of degree 3. The Petersen graph is among them (middle, top row).

B Examples of graphs with a given degree distribution

In the present appendix we consider an example of an ensemble of graphs using the degrees of the nodes of a given graph as well as the number of its connectivity components. The graphs may look a bit odd, perhaps, but in this way it was easy to generate easily some examples from this ensemble.

We use the graph in Figure 5.1 as the basis for the ensemble of graphs that we consider in the present appendix.

Below we present some variants of Figure 5.1, with the same characteristics as listed above. The aim is to show the (topological) variety of graphs that satisfy the requirements specified above.

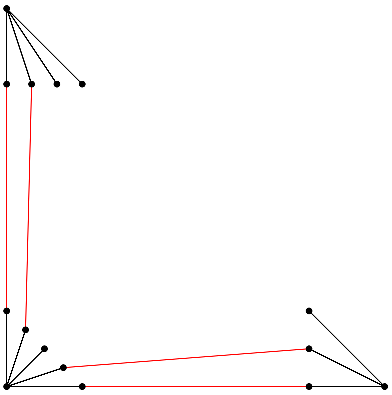

Figure B.1 Variant 1. One connectivity component.

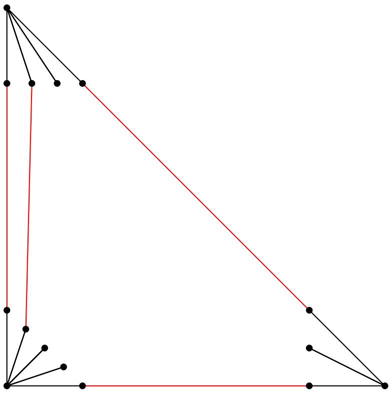

Figure B.2 Variant 2. One connectivity component.

B.1 Alternative specimen of the ensemble

The graph in Figure 5.1 was generated by starting with nodes with degrees 3, 4 and 5. Four edges were use to connect 8 nodes adjacent to the nodes with degree 3, 4 or 5. The same principle will be applied in generating variants below. It should be noted that this does not generate all elements in the ensemble. In this construction the nodes with degrees 3, 4 or 5 are only adjacent to nodes with degree 1 or 2. But there are other options. An example is shown in Figure B.12.

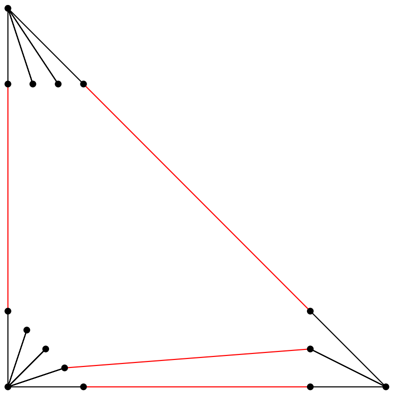

Figure B.3 Variant 3. One connectivity component.

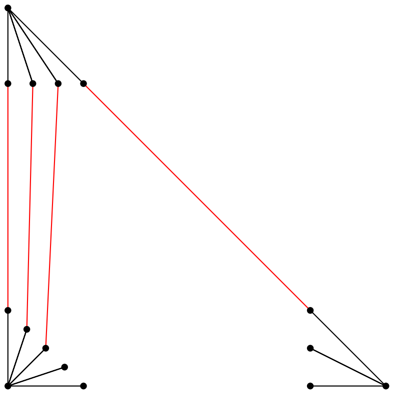

Figure B.4 Variant 4. One connectivity component.

A second specimen of a deviant type of graph (compared to the ones paraded above) is the one in Figure B.13

B.2 Sampling mechanism

Above we have shown a few specimens of the ensemble of graphs satisfying the requirements above. The interesting question remains how to devise a sampling mechanism that produces any graph satisfying these properties with positive probability. This problem is equivalent to finding an adjacency matrix $A = (a_{ij})$ in which these requirements have been translated into constraints of the rows and columns of this matrix. If we start with an example of such a graph (and hence adjacency matrix) we know there is at least one solution.

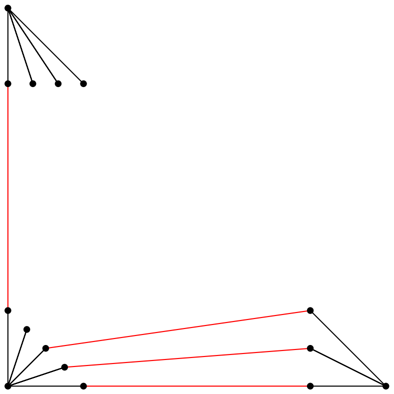

Figure B.5 Variant 5. One connectivity component.

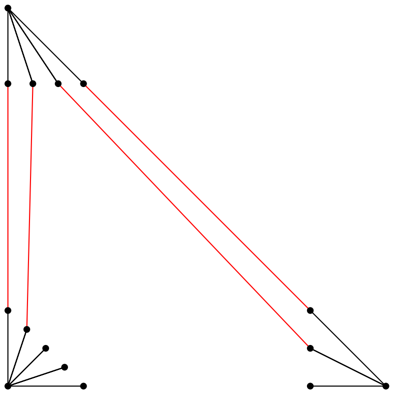

Figure B.6 Variant 6. One connectivity component.

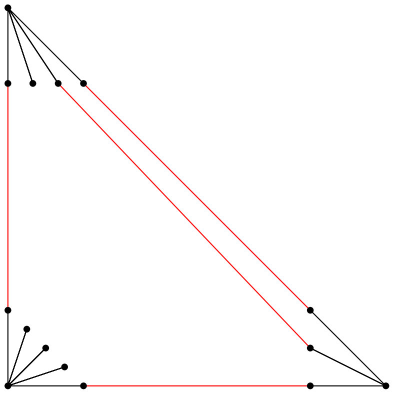

Figure B.7 Variant 7. One connectivity component.

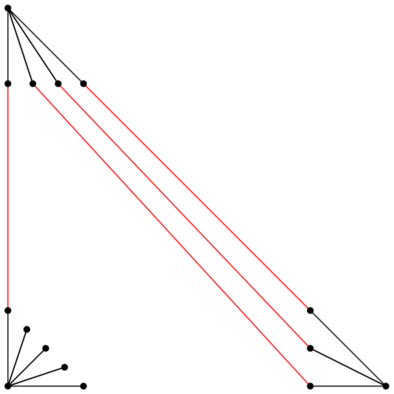

Figure B.8 Variant 8. One connectivity component.

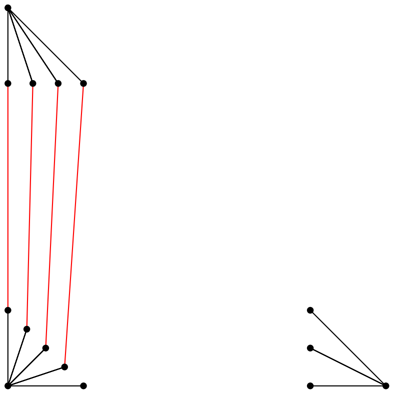

Figure B.9 Variant 9. Two connectivity components.

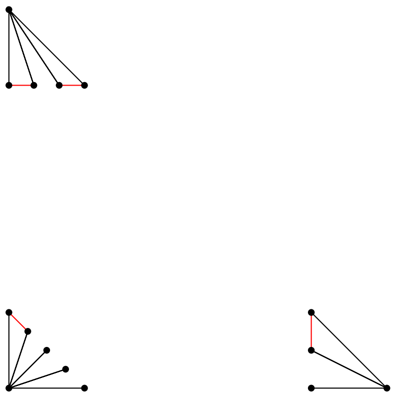

Figure B.10 Variant 10. Three connectivity components.

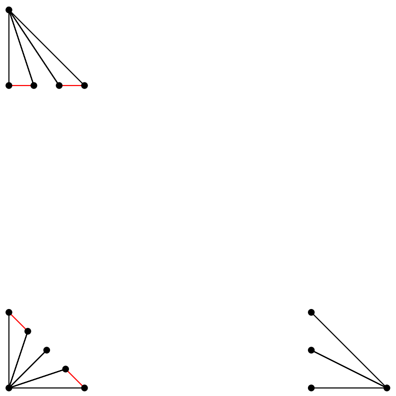

Figure B.11 Variant 11. Three connectivity components.

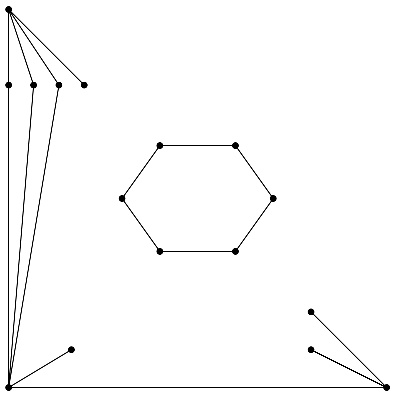

Figure B.12 A specimen of the ensemble of a different type. Two connectivity components.

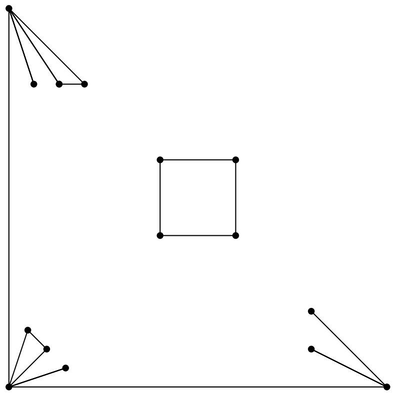

Figure B.13 A second deviant specimen of the ensemble. Two connectivity components.

C The 01-operator $[\cdot]$

In this appendix we define a simple but useful operator for numbers of matrices. It can be used to define certain ensembles, namely those defining neighbors of neighbors or neighbors of neighbors of neighbors, etc. See section 5.4.

The properties of $[\cdot]$ mentioned below are all trivial. Some of those for matrices are useful as they can be used to simplify computations.

C.1 $[\cdot]$ for numbers

We define the operator $[\cdot]$ for numbers $x \geq 0$ as follows:

$$[x] = \begin{cases} 0 & : \quad x = 0 \\ 1 & : \quad x > 0 \end{cases} \tag{C.1}$$

Some trivial properties of $[\cdot]$ for nonnegative numbers follow.

For $x \geq 0$ we have

$$[[x]] = [x], \tag{C.2}$$

which means that $[\cdot]$ is idempotent.

We have the following behavior of $[\cdot]$ for products of numbers. If $x, y \geq 0$ then

$$[xy] = [x[y]] = [[x]y] = [x][y] = [x] \wedge [y], \tag{C.3}$$

where $\wedge$ is the minimum operator: $0 \wedge 0 = 0 \wedge 1 = 1 \wedge 0 = 0$ and $1 \wedge 1 = 1$. In fact, (C.3) shows that $[\cdot]$ is multiplicative.

For sums of numbers the behavior of $[\cdot]$ is as follows:

$$[x + y] = [[x] + [y]] = [x] \vee [y], \tag{C.4}$$

where $\vee$ is the maximum operator: $0 \vee 0 = 0$ and $0 \vee 1 = 1 \vee 0 = 1 \vee 1 = 1$. In fact, (C.4) shows that $[\cdot]$ is additive (sort of).

C.2 $[\cdot]$ for matrices

We can define $[\cdot]$ also for nonnegative matrices M, briefly characterized by $M \geq 0$, by applying the operator element-wise. So, if

$$M = \begin{pmatrix} M_{11} & \dots & M_{1n} \\ \vdots & \ddots & \vdots \\ M_{m1} & \dots & M_{mn} \end{pmatrix} \tag{C.5}$$

then

$$[M] = \begin{pmatrix} [M_{11}] & \dots & [M_{1n}] \\ \vdots & \ddots & \vdots \\ [M_{m1}] & \dots & [M_{mn}] \end{pmatrix}. \tag{C.6}$$

Note that $[\cdot]$ applied to matrices is order preserving: for a matrix $X \geq 0$, $[X]$ has the same order as X, that is, the same number of rows and columns.

We again list some trivial properties of this operator, this time for nonnegative matrices. In (C.5) the matrix elements M_{ij} are nonnegative numbers. But they could just as well be nonnegative matrices (of appropriate orders), so that M is in fact a block matrix.

For a matrix $X \geq 0$ we have

$$[[X]] = [X], \tag{C.7}$$

which means that $[\cdot]$ is idempotent for nonnegative matrices.

For matrices $X, Y \geq 0$ such that the sum $X + Y$ is defined, we have

$$[X + Y] = [X + [Y]] = [[X] + Y] = [[X] + [Y]], \tag{C.8}$$

which shows (a kind of) additivity to hold for $[\cdot]$ applied to nonnegative matrices.

For matrices $X, Y \geq 0$ such that the product XY is defined, we have

$$[XY] = [X[Y]] = [[X]Y] = [[X][Y]], \tag{C.9}$$

which shows (a kind of) multiplicity of $[\cdot]$. So if $X \geq 0$ is a square matrix (C.9) implies

$$[X^2] = [X[X]] = [[X]X] = [[X]^2], \tag{C.10}$$

from which it follows that for $k \in \mathbb{N}$

$$[X^k] = [X^{k-1}[X]] = [X[X]^{k-1}] = [[X]^k]. \tag{C.11}$$

For a matrix $X \geq 0$

$$[X'] = [X]', \tag{C.12}$$

where the prime denotes transposition, that is, reversing the roles of rows and columns in matrices.

Let $X \geq 0$ be a nonnegative matrix. If

$$[X] = X. \tag{C.13}$$

holds it implies that X is a $(0, 1)$-matrix, and vice versa. So (C.13) is a convenient way to characterize $(0, 1)$-matrices concisely.

D Statistical ensembles

D.1 Physics

The inspiration for looking at ensembles of networks came from a concepts in statistical mechanics, namely that of statistical ensembles,[14] We explain the ideas here briefly with a view on application to ensembles of networks, as dealt with in the present paper.

We remark that in statistical mechanics a notion of ensembles exists that could be useful to our area of ensembles of networks. It is the concept of statistical ensemble. It is related to the idea that a system like a gas in a container exists of many molecules, moving about frantically and colliding with each other and with the walls of the container. It is impossible to describe the position and velocity of each particle in a certain period of time. This is due to the fact that measurements are insufficiently precise: even small differences between the real and measured positions and velocities of particles would lead to significant discrepancies between observed and predicted positions and velocities in the future. Also the dynamics of the colliding molecules is far too complicated to model exactly. And most significantly, there are far too many particles in the gas. It is impossible to track them all.

For the understanding of the macro-properties of the gas (such as temperature, pressure, average energy, volume, density) it is not necessary to have an understanding of the gas at the micro-level. At the macro-level a description of the behavior of the gas in the container is by a few parameters such as temperature, pressure, volume of the container, etc.

However, it is desirable to link the description of the gas in the container at the micro-level and that of the macro-level. It is comparable to the situation in statistics where the microdata about some topic are used to summarize the data, using statistical quantities such as averages, (co)variances and regression models to show relationships between variables. Otherwise one may not be able to see the wood for the trees.

Statistical mechanics starts describing a many-particle system like a gas at the micro level and intends to explain the properties of the gas at the macro-level description provided by thermodynamics. The theory providing the link between the two extreme descriptions is often referred to as statistical thermodynamics.

A statistical ensemble is the collection of all the microstates which represent a macrostate, weighted by the frequency of their occurrence.

For instance, in a gas the microstates correspond to energy levels E_i, for $i \in S$, which represents the set of microstates. We may assume dat S is a finite set, containing $|S| = s$ elements. Let $p = (p_1, \dots, p_s)$ be a probability distribution, indicating the frequency of occurrence of each microstate i. Then (S, p) is a statistical ensemble.[15] The energy associated with it is given by

[14] There are other types of ensembles used in statistical mechanics, such as canonical (or Gibbs) ensembles. We only consider the concept of statistical ensemble here, as the others are not needed here and explaining them in the present paper would serve no purpose.

[15] Not that this is a detailed description, but stil one at some level of aggregation. In the most detailed case one is dealing with individual particles (e.g. molecules) which is beyond the possibility of description.

$$\bar{E} = \sum_{i \in S} p_i E_i. \tag{D.1}$$

$\bar{E}$ is the weighted average of energy values at the micro-level, so a parameter at the macrolevel of the system. Suppose that the energy (D.1) is given. Without any further information or guidance we cannot say anything about the probabilities p_i. However, using the maximum entropy principle, we can. We then have to consider the entropy

$$\mathcal{E}(p_1, \dots, p_s) = -\sum_{i \in S} p_i \log p_i, \tag{D.2}$$

and maximize this function under the constraint (D.1). This optimum exists and is the so-called Boltzmann distribution:

$$p_i = \frac{e^{-\beta E_i}}{\sum_{i \in S} e^{-\beta E_i}}. \tag{D.3}$$

So (D.3) shows that microstates with higher energy levels are less likely to be occupied than microstates with lower energy.

The parameter β in (D.3) is the Lagrange muliplier used to solve the constrained optimization problem that the maximum entropy approach amounts to. In a physical context this parameter has a specific meaning: it is proportional to the reciprocal of the temperature of the gas:

$$\beta = \frac{1}{kT}, \tag{D.4}$$

where k is Boltzmann's constant and T denotes the absolute temperature of the gas.[16)]

D.2 Network theory

We can apply the ideas used above to define a statistical ensemble for networks. As energy we use the concept of elastic energy for networks, as defined in [12]. We look at networks with a given number N of nodes. First we should make a list of elastic energies that can occur for such networks. Then we determine the temperature, or rather, the parameter β. Next we draw a set of elastic energies, using the Boltzmann distribution (D.3). The final task, which is the most difficult one, is to produce, for each elastic energy drawn, a network with precisely this elastic energy. This is left as a challenge for the interested reader.

[16)] As references for this theory see books on statistical mechanics or statistical thermodynamics, such as [3], [9] or the charming, concise [8].

Colophon

Publisher
Statistics Netherlands
Henri Faasdreef 312, 2492 JP The Hague
www.cbs.nl

Prepress
Statistics Netherlands, Grafimedia

Design
Edenspiekermann

Information
Telephone +31 88 570 70 70, fax +31 70 337 59 94
Via contact form: www.cbs.nl/information