INTRODUCTION TO NETWORKS

TILL HOFFMANN

1. Network structure

Networks are representations of interacting entities and are used to describe complex systems including social networks, protein interactions, interbank loans, food webs and many more. Understanding networks will enable you to follow a large literature of interdisciplinary research ranging from structural problems in the financial sector to the accumulation of toxins in certain animals. But what is a network?

Definition 1. A *network* G is a set of distinct entities N called *nodes* together with a set of *edges* E which encodes the connection between nodes. Two nodes i and j are said to be connected by an edge if $(i, j) \in E$.

Networks (or *graphs*) have been used in a wide range of disciplines which has lead to redundant terminology: nodes are often referred to as *vertices*. Edges are also called *links* or *connections*. Connected nodes are sometimes said to be *adjacent*. The number of nodes is usually denoted by $n = |N|$ and the number of edges is $m = |E|$. If the network is *complete*, i.e. all nodes are connected to one another, it has $m = \left(\begin{array}{c} n \\ 2 \end{array}\right) = n\,(n-1)\,/2$ edges. The fraction of existing edges is $p = \frac{2m}{n(n-1)}$. A network is said to be *sparse* if $p \ll 1$ and is called dense if $p \sim 1$.

1.1. **Types of networks.**

The most basic networks are *undirected* and *unweighted*, i.e. the only information about edges we have is whether they exist or not. A network is said to be *directed* if the elements of the edge set are ordered such that $(i, j) \in E$ implies that there is a connection from node j to node i but not necessarily from i to j. In a *weighted* network, each edge (i, j) is assigned a scalar value w_{ij}, e.g. an interaction rate. Examples of such networks are shown in Fig. 1.1. Unless noted otherwise, we will consider undirected, unweighted networks for the remainder of the course.

1.2. **Representation of networks.**

Once we have collected information about which entities interact with one another, we need to represent the network in memory before we can analyse it. We consider two commonly used representations:

Definition 2. The *adjacency matrix* A representing an undirected, unweighted network is a binary symmetric matrix with elements

$$A_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \\ 0 & \text{otherwise} \end{cases}.$$

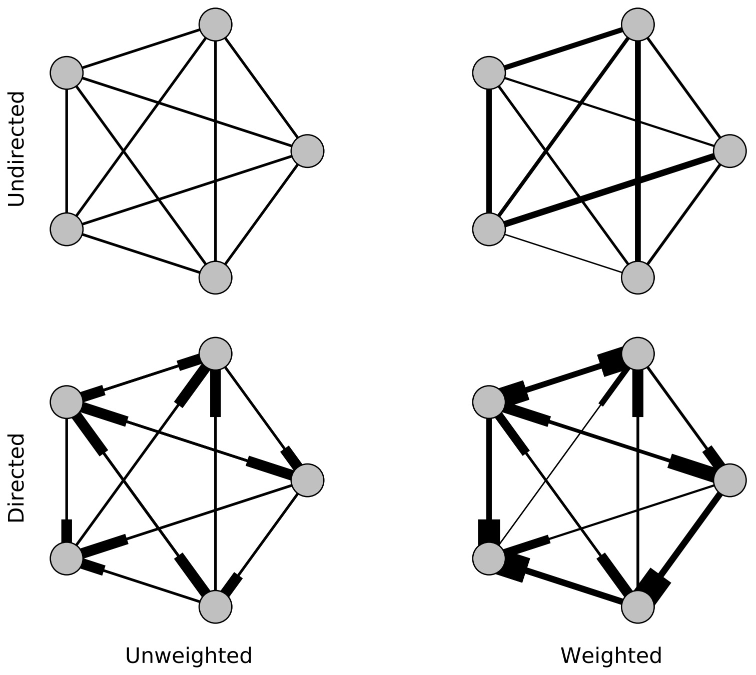

FIGURE 1.1. Reciprocated friendships such as Facebook friendships are an example of undirected, unweighted networks (top left). Internet traffic between different servers can be represented as an undirected, weighted network (top right). Twitter users do not need to follow their followers such that the Twitter "listening" network is a directed, unweighted network (bottom left). Interbank loans are an example of directed, weighted networks (bottom right) because each loan has a direction (money flow) and weight (loan amount).

The adjacency matrix of a directed network does not need to be symmetric. If the network under consideration is weighted, the entries of the adjacency matrix are the edge weights $A_{ij} = w_{ij}$.

The toy network shown in Fig. 1.2 is represented by the adjacency matrix

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Definition 3. The *adjacency list* L representing an undirected, unweighted network is a list of unordered pairs of nodes (i, j) such that $(i, j) \in E$, i.e. a list representation of the edge set. If the network under consideration is directed, the pairs of nodes in the adjacency list are ordered. The adjacency list of a weighted network is a list of tuples (i, j, w_{ij}), where w_{ij} is the weight associated with the edge.

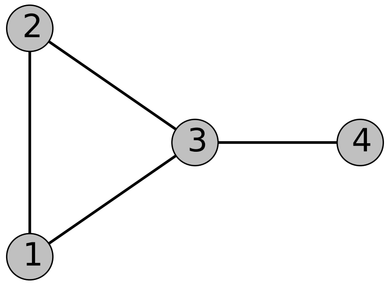

FIGURE 1.2. An unweighted, undirected toy network.

The adjacency list representing the toy network is

$$L = \begin{pmatrix} 1 & 2 \\ 1 & 3 \\ 2 & 3 \\ 3 & 4 \end{pmatrix}.$$

Adjacency matrices are useful for theoretical investigations of networks and proofs as we shall later see. Checking if a particular edge (i, j) exists is a constant-time operation because we can simply check the entry A_{ij}. However, they are often wasteful representations because they require n^2 values to be stored in memory. In contrast, adjacency lists only require $2m$ values to be stored in memory but looking up if an edge exists is a more expensive operation. As a rule of thumb, adjacency lists are more suitable for computational tasks unless the network is *dense*, i.e. a large proportion of all possible edges exists.

1.3. **Neighbourhoods and degree.**

Networks can be studied at different scales from individual nodes and their immediate neighbourhood to the network as a whole. In the following, we will start with microscopic structure and work our way up the scale ladder to investigate large scale properties of networks. We are only going to consider network statistics in this section but will consider models of networks in the Sec. 2. First, let us define some more terminology.

Definition 4. The *neighbourhood* N_j of a node j is the set of all nodes to which j is connected such that

$$N_j = \{i \in N : (i, j) \in E\}.$$

The size of the neighbourhood $k_j = |N_j|$ is called the *degree* of node j. If the network is weighted, we can also define the *strength* of a node as the sum of the weights of all of its edges

$$s_j = \sum_{i \in N_j} w_{ij}.$$

The degree distribution $P(k)$, i.e. the probability that a node chosen uniformly at random has degree k, is a central object of study in network analysis. It is the analogue of an income distribution except income is measured in number of connections. In a regular graph, i.e. a graph in which all nodes have the same number of edges, the degree distribution is a discrete δ-function. If the degree distribution is heavy-tailed, a small fraction of nodes has a large fraction of all connections, i.e. there is an "elite". Real-world networks often have a heavy-tailed degree distribution and a lot of effort has been devoted to constructing models that can generate such degree distributions. In fact, one way to assess the importance of a node in a network is to consider its *degree centrality*–just another term for degree. Degree centrality is motivated by the idea that nodes with more connections are more important.

1.4. **Clustering coefficient.** Are your friends friends? Many of them probably are because they share the same social circle. In fact, most social networks exhibit a large degree of *clustering*, i.e. the probability that two nodes i and j are connected is much higher if they have a common connection k than if they do not.

Definition 5. The *clustering coefficient* c_j of a node j is the fraction of existing edges in its neighbourhood N_j, i.e.

$$c_j = \frac{2\left|\{(i,k) \in E : (i,j) \in E \wedge (k,j) \in E\}\right|}{k_j\,(k_j - 1)}. \tag{1.1}$$

The larger the *average clustering coefficient* $\bar{c} = \frac{1}{n}\sum_{j \in N} c_j$, the more likely neighbourhoods are to be densely connected.

High-degree nodes often have small clustering coefficients because their neighbourhoods are large and only a small fraction of edges exist. For example, a social butterfly has a large number of acquaintances who may not know each other. A scatter of the degrees and clustering coefficients of members of Zachary's karate club is shown in Fig. 1.3. The network consists of 34 members of a karate club at a US university.

1.5. **Shortest path and diameter.** We have considered nodes and their immediate neighbourhoods in the previous sections. Now let us investigate properties of the network as a whole.

Definition 6. The *shortest path* between a *root node* r and a *target node* t is the shortest sequence $\ell_{rt} = \{r, \ldots, t\}$ of connected nodes that starts with r and ends with t. The *geodesic distance* between r and t is $d_{rt} = |l_{rt}| - 1$. If the graph is weighted, the shortest path is the sequence that minimises the total weight of edges between r and t. If the graph is directed, the geodesic distance is not necessarily symmetric under exchange of root and target nodes.

Finding shortest paths in networks may seem like a daunting task because the number of different paths grows combinatorially with the number of edges. Fortunately, efficient algorithms exist that can find shortest paths with time-complexity $\mathcal{O}\left(n^2\right)$ [3]. Being able to find short paths efficiently is extremely important for routing problems such as finding your way home using GoogleMaps.

Definition 7. The *diameter* d of a network is the largest shortest path between any pair of nodes defined by

$$d = \max_{r,t \in N} d_{rt}.$$

The diameter of a network gives an intuitive notion of how "big" the network is. Surprisingly, the diameter is often very small compared with the number of nodes. This phenomenon has entered popular culture as the concept of six degrees of

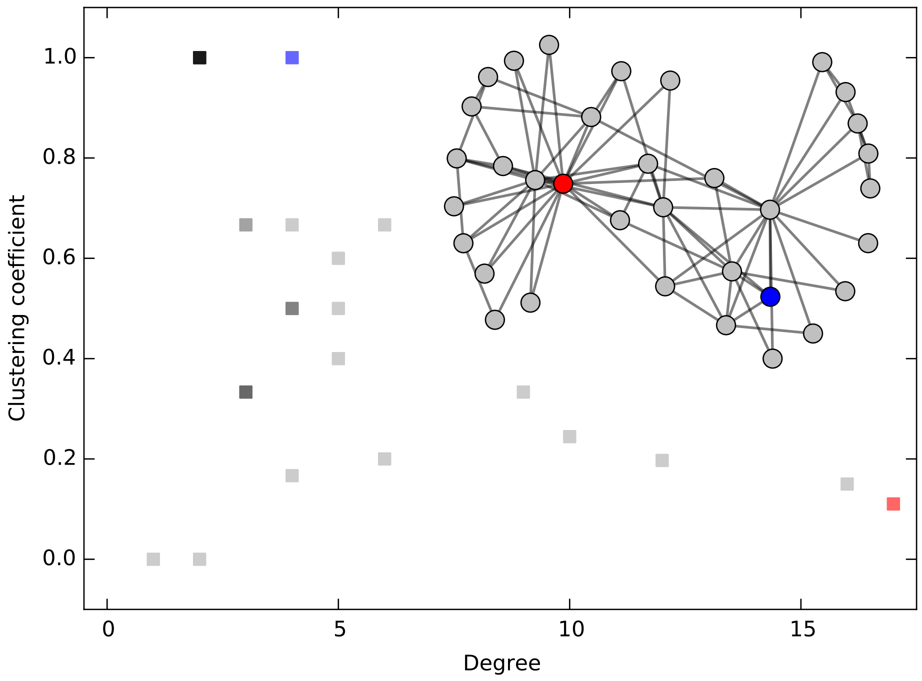

FIGURE 1.3. Scatter plot of the degrees and clustering coefficients of members of Zachary's karate club. The node with the highest degree shown in red has a large number of connections who do not socialise with one another leading to a low clustering coefficient. All friends of the lower-degree node shown in blue are friends with each other such that it has a clustering coefficient of one. The network is shown as an inset.

separation between any pair of individuals. Networks with small diameter are called *small-world* networks and are usually defined by having the property $d \propto \log n$.

1.6. **Betweenness centrality.** Shortest paths between pairs of nodes also give us a handle on how important nodes are for making a network well-connected. Intuitively, nodes that are visited often by following shortest paths between nodes are more important than nodes that are rarely visited.

Definition 8. The *betweenness centrality* g_i of a node i is proportional to the number of shortest paths that pass through it. We define

$$g_i = \sum_{r,t \in N \setminus i} \begin{cases} 1 & \text{if } i \in \ell_{rt} \\ 0 & \text{otherwise} \end{cases}.$$

For example, the Panama canal has high betweenness centrality in the global shipping network and Houston airport has high betweenness centrality in the global air traffic network.

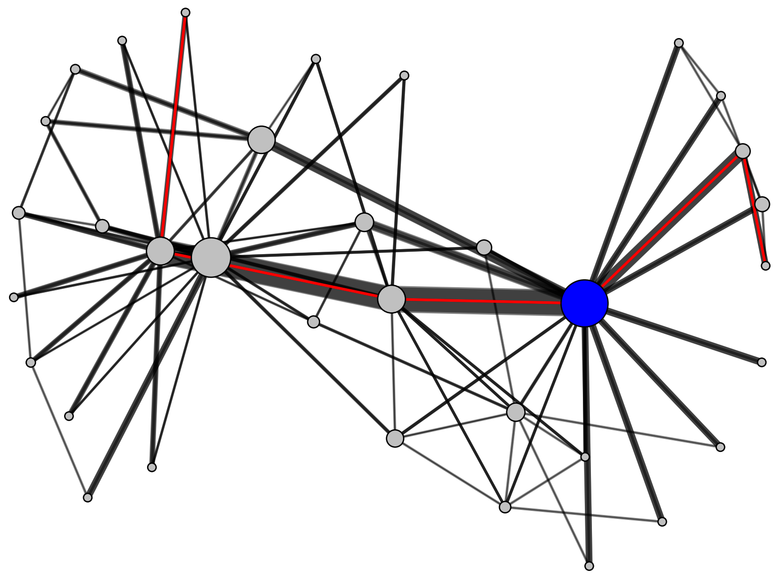

FIGURE 1.4. Betweenness centrality in the Zachary karate club. The size of each node is proportional to its betweenness centrality. The node with the largest betweenness centrality is shown in blue. The width of each edge is proportional to the number of shortest paths that pass through it. The red line shows one of the longest shortest paths in the network.

1.7. **Eigenvector centrality.** An alternative way to measure the importance of nodes is eigenvector centrality. The name will become obvious shortly. A flaw of degree centrality is that not all nodes are created equal. For example, you may have the same number of friends as president Obama but his friends are likely more important than your friends (no offence intended).

Definition 9. Let x_i be the eigenvector centrality of node i satisfying

$$x_i = \lambda \sum_j A_{ij} x_j,$$

where λ is a proportionality constant and A is the adjacency matrix, i.e. x is an eigenvector of A.

Informally, eigenvector centrality is a measure that assigns importance to nodes proportional to the importance of their neighbours. Eigenvector centrality would attribute more importance to president Obama than one of us.

1.8. **Community structure.** Finally, we briefly consider community structure in networks, i.e. groups of nodes that are more densely connected with each other than with nodes in other groups. For example, communities in protein interaction networks often correspond to functional groups and communities of coauthors resemble their disciplinary associations. Identifying communities in networks is often necessary to understand their function: networks can consist of a large number of nodes with even more edges which we can only make sense of after they have been reduced to a more manageable size. The combinatorial optimisation necessary to identify the best communities is however too advanced for this brief introduction.

1.9. **Recommended reading.** See [**?**] for a general introduction to networks. See [1, 6] for more concise reviews. See [5] for a review of community detection.

2. Models of Networks

Statistics are useful to get an intuition for the properties of networks but we need models for networks to perform proper inference and prediction.

2.1. **Erdős–Rényi model.** The prototypical model of a network is the Erdős–Rényi (ER) model [4]. The ER model assumes that the probability that an edge between nodes i and j exists is a constant p, i.e.

$$P\left(A_{ij}|p\right)=\begin{cases}p & \text{if } A_{ij}=1\\ 1-p & \text{if } A_{ij}=0\end{cases}.$$

The entries of the adjacency matrix are independent, identically-distributed (i.i.d.) Bernoulli random variables with success probability p.

Recalling the introduction to Bayesian statistics from last week, what is the posterior on p?

The degree of a node i is

$$k_i=\sum_j A_{ij}$$

and is a binomial random variable with $n-1$ trials and success probability p. The degree distribution is sharply peaked about the mean-value $p\left(n-1\right)$. In the limit $n\to\infty$, the distribution approaches a δ-function and all nodes have the same degree $k=np$.

Using this property it is easy to calculate the clustering coefficient: the number of possible connections between neighbours of a node is $\frac{1}{2}k\left(k-1\right)$ and the average number of existing connections is $\frac{1}{2}pk\left(k-1\right)$. Plugging into Eq. (1.1) the clustering coefficient is

$$c=p$$

and must vanish in the limit $n\to\infty$ if nodes have finite degree.

We note without proof that the ER model generates small-world networks with diameter $d=\mathcal{O}\left(\log n\right)$ provided that p is sufficiently high for the network to be connected.

The ER model is theoretically appealing because of its simplicity but has a number of shortcomings. Firstly, real-world networks often have degree distributions with much heavier tails. Secondly, the independence assumption of edges leads to low clustering coefficients which are not observed in real networks. For example, the probability that two of your friends are friends with one another is much higher than the probability that two random people are friends with one another.

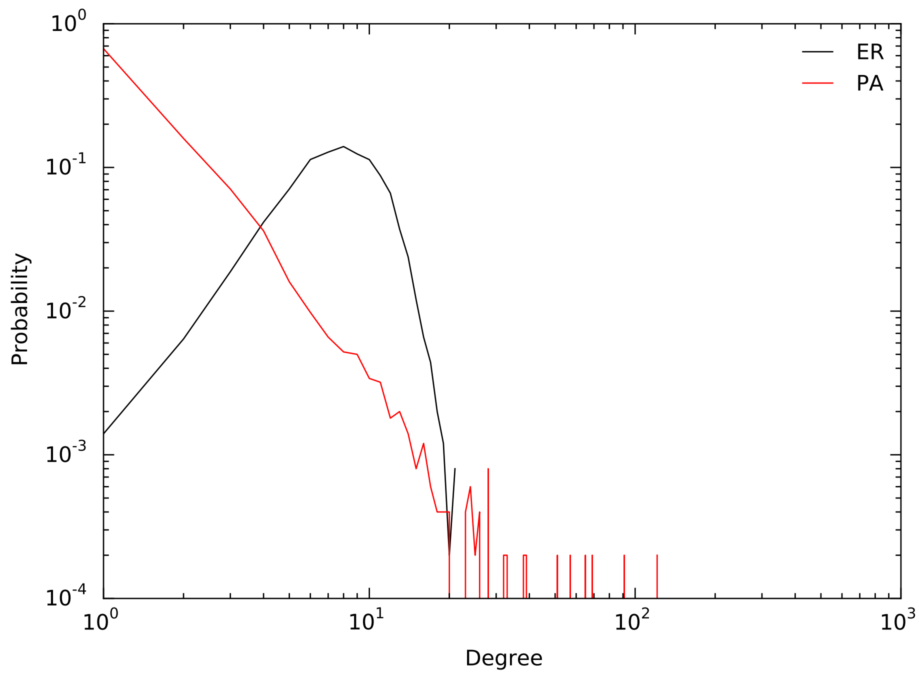

FIGURE 2.1. The black (red) line shows the degree sequence of a realisation of an ER network with $p = \frac{\log n}{n}$ (a PA network with $m = 1$). Both networks have $n = 5000$ nodes.

2.2. **Preferential attachment model.** The preferential attachment (PA) model mitigates one of the shortcomings of the ER model [2]. Under the assumptions of the PA model, networks are generated in a two-step process: (1) the network is initialised with m fully-connected nodes at time $t = 0$, and (2) at each time step $t > 0$ a new node is added to the network and is connected to m existing nodes selected with probability proportional to their degree. The edges are obviously no longer independent.

Can you come up with an efficient $\mathcal{O}(1)$ algorithm to generate such a network?

Consider the average degree $k_i(t)$ of a node that was added to the network at time i as a function of time t. By definition

$$k_i(i) = m.$$

The average number of additional connections the node gains at each step $t > i$ is

$$\frac{k_i(t)}{2t + m - 1}$$

because each new node makes m connections and the probability for the node to receive a connection is $\frac{k_i(t)}{2mt+m(m-1)}$, where the denominator ensures the correct normalisation. The term $2mt$ accounts for all connections added in step (2) and the term $m(m-1)$ accounts for the initial connections. In the limit of large time t we may neglect the constant term and the degree of node i satisfies the differential

Why is a differential treatment appropriate even though the process is discrete-time?

equation

$$\frac{dk_i(t)}{dt} = \frac{k_i(t)}{2t}$$

$$\text{with solution } k_i(t) = m\sqrt{\frac{t}{i}}.$$

Consider the nodes that have degree smaller than some threshold θ, i.e.

$$m\sqrt{\frac{t}{i}} < \theta$$

$$\therefore i > \frac{m^2 t}{\theta^2}.$$

The fraction of nodes with degree smaller than θ, i.e. the cumulative distribution function, is

$$P(k < \theta) = 1 - \left(\frac{m}{\theta}\right)^2.$$

After differentiating with respect to θ we obtain the degree distribution

$$P(k) \propto k^{-3}$$

which is heavy tailed as observed in real-world networks.

Unfortunately, the clustering coefficient still decays to zero in the limit $n \to \infty$. The PA model also generates small-world networks. A comparison of the degree distributions of ER and PA networks is shown in Fig. 2.1.

2.3. **Recommended reading.** See [4] for ER graphs and [2] for a preferential attachment model.

3. Dynamics on networks

So far we have only considered static networks that do not change in time. However, ecological networks such as food webs are rarely static. In the following we will consider dynamics *on* networks, i.e. dynamics that occur on a network with fixed connections. We will not touch on dynamics *of* networks, i.e. changing connections.

3.1. **Discrete-time random walks.** Discrete-time random walks (DTRWs) are a simple example of dynamics on networks. Consider an agent residing on node $x(t) = j$ at time t. In the next time step, the agent jumps to one of the neighbours of node j at random such that $x(t+1) \in N_j$. The probability that the agent resides on node i at time $t+1$ given that it resided on node j at time t is

$$P(x(t+1) = i | x(t) = j) = T_{ij},$$

where $T_{ij} = A_{ij}/k_j$ is the *transition matrix* which is normalised such that

$$\sum_i T_{ij} = 1 \tag{3.1}$$

for all j . DTRWs are Markov processes because the future evolution of the system only depends on the current location of the agent.

How would you simulate such a process?

Let $p_j(t) = P(x(t) = j)$ be the probability that the agent resides on node j at time t. Suppose the agent starts on node l at time $t = 0$ such that $p_j(0) = \delta_{jl}$, where

$$\delta_{jl} \equiv \begin{cases} 1 & \text{if } j = l \\ 0 & \text{if } j \neq l \end{cases}$$

is the Kronecker delta. The probability that the agent resides on node i at time $t = 1$ is

$$p_i(1) = [Tp(0)]_i .$$

More generally, the probability that the agent resides on node i at time t given initial conditions $p(0)$ is

$$p_i(t) = \left[T^t p(0)\right]_i .$$

Evaluating powers of matrices is hard work but we can simplify the problem by diagonalising the transition matrix

$$T = U^{-1}DU,$$

where U is a matrix whose columns correspond to the right eigenvectors of T, U^{-1} is the inverse of U and its rows are the left eigenvectors of T, and D is a diagonal matrix containing the eigenvalues of T. Then

$$T^t = U^{-1}D^tU$$

where we have made use of the fact that UU^{-1} is the identity matrix. Taking powers of the diagonal matrix D is trivial.

Because the transition matrix is column-normalised (cf. Eq. (3.1)) its largest eigenvalue λ_1 is one and all remaining eigenvalues $\lambda_2, \ldots, \lambda_n$ have modulus smaller than one (excluding pathological cases). Furthermore, the left eigenvector corresponding to the largest eigenvalue has constant entries. In the limit $t \to \infty$, there will only be one nonzero entry in D^t such that the only contribution to the steady-state solution $p^\infty = \lim_{t\to\infty} p(t)$ is from the eigenvector with the largest eigenvalue. In other words, the steady-state solution is the (appropriately normalised) eigenvector of the transition matrix with the largest eigenvalue.

Can you come up with two different pathological cases?

The slowest transient will decay as λ_2^t, where λ_2 is the second-largest eigenvector in modulus. If $|\lambda_2|$ is very close to one, the system will take a long time to reach equilibrium. The characteristic time-scale is $\tau = -\log|\lambda_2|$. Fig. 3.1 shows the air passenger traffic in Australia and the corresponding steady-state solution. We will study this network more closely in the practical.

3.2. **Recommended reading.** See [7] for a review of dynamical processes on networks.

References

[1] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Reviews of Modern Physics*, 74:47–97, 2002.
[2] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.
[3] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271, 1959.
[4] Paul Erdős and Alfréd Rényi. On the evolution of random graphs. *Publications of the Mathematical Institute of the Hungarian Academy of Sciences*, 5:17–61, 1960.
[5] Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3–5):75–174, 2010.

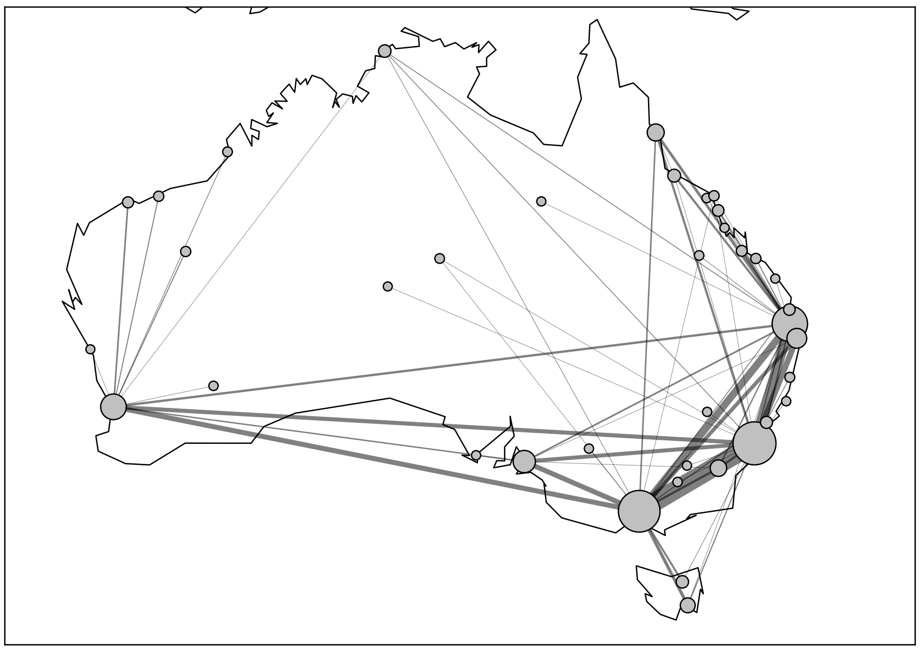

Figure 3.1. Air traffic network in Australia in 2013. The width of edges is proportional to the number of people travelling on the corresponding route. The size of each node corresponds to the steady-state probability to find a random walker at the corresponding airport.

[6] Mark Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[7] Mason A. Porter and James P. Gleeson. Dynamical systems on networks: A tutorial. *arXiv*, page 1403.7663, 2014.