Weighted networks

9

In this lecture we consider networks in which links differ from each other. The links are made individual by ascribing a positive number—a *weight*—to each of them. These *weighted networks* enable us to quantitatively represent processes and flows in real-world networks—from various transportation and information nets to social ones.

9.1 The strength of weak ties

In 1973 American sociologist Mark Granovetter published his landmark paper 'The strength of weak ties' [100]. In this remarkable work whose influence has spread far beyond sociology, Granovetter pioneered a quite new understanding of social networks. According to Granovetter, individuals in social networks are connected by 'ties'—weighted links. The 'tie strength' (actually, the link weight in a one-partite weighted network) is defined to be proportional to the frequency (intensity) of social interaction between two individuals [100, 101]. According to strength, we can distinguish 'strong' and 'weak' social ties. In the view of Granovetter, 'our acquaintances (weak ties) are less likely to be socially involved with one another than are our close friends (strong ties). Thus the set of people made up of any individual and his or her acquaintances comprises a low-density network ... whereas the set consisting of the same individual and his or her close friends will be densely knit'. As a result, a social network looks as shown in Fig. 9.1: dense communities of strongly tied close friends are connected together by weak acquaintance ties. In this scheme, 'The weak tie between Ego and his acquaintance ... becomes not merely a trivial acquaintance tie but rather a crucial bridge between the two densely knit clumps of close friends. ... These clumps would not, in fact, be connected to one another at all were it not for the existence of weak ties...'

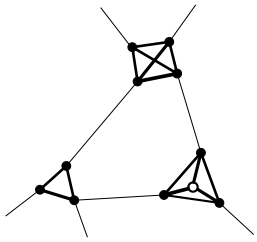

Fig. 9.1 Organization of social networks according to Granovetter. The width of each link represents the strength of the corresponding social tie. Weak ties connect together dense communities of strongly tied individuals. An individual in the centre of the right community (an open dot) has no weak ties and receives any 'external' information only after his close friends.

This organization of social networks has an important consequence also indicated by Granovetter. Ego receives information from his or her close friends through strong ties, while information from other, outer parts of a social network reaches Ego through weak ties. So a lack of weak ties would significantly delay the receipt of information coming from the outer social world. Indeed, in this case, Ego can hear all news only after his or her close friends. This puts Ego in a disadvantaged position, for example, in the labour market (which was a point of interest for Granovetter). Contrastingly, individuals with many weak ties have an apparent advantage in job hunting.

Granovetter assumed this structure of social networks[1] and tested its direct consequence by surveying workers to find who did tell them about their current job. In most cases, the information came from 'not a friend, an acquaintance'. Interestingly, later sociological studies found a number of exceptions from this picture [101]. Recently (2007), researchers found a new possibility of verifying Granovetter's ideas more thoroughly [144, 145]. Large log files stored by cellular network operators contain traces of all mobile phone calls made within these networks. From these records, in particular, one can get the total duration of calls between each pair of customers during a given period. This number characterizes the intensity of social interaction. So the duration of calls can naturally be interpreted as the strength of a given social tie, or, in other words, as the weight of a link in a social network.[2] In this way, these authors constructed a large weighted undirected network of 4.6×10^6 nodes and 7.0×10^6 links. A great majority of nodes in this network (about 85% of the total number) belonged to a giant connected component. The questions were: how does the weight of a link relate to its position in this network? What is the relation between the weight of a link and information flow though this link?

Let us first discuss the former question. To answer this and so to confirm or reject the picture shown in Fig. 9.1 one has to do two things: (i) analyse the structure of the close neighbourhood of a link of a given weight and (ii) reveal the role of links of given weight in the global organization of this network. The closest environment of a tie connecting two individuals i and j is essentially characterized by the relative number of their close friends which are friends of each other. This relative overlap can be characterized by the following expression:

$$O_{ij} = \frac{t_{ij}}{q_i - 1 + q_j - 1 - t_{ij}}\,, \tag{9.1}$$

where t_{ij} is the number of common friends of nodes i and j, which is also the number of triangles attached to the link; q_i and q_j are the degrees of nodes i and j. Instead of the relative overlap, the link-clustering coefficient

$$C_{ij} = \frac{t_{ij}}{\min(q_i, q_j) - 1} \tag{9.2}$$

can be used (compare with the definition of the clustering coefficient). Here $\min(q_i, q_j) - 1$ is the maximum possible number of common friends for the given q_i and q_j. Figure 9.2 explains these two closely related characteristics. The authors of papers [144, 145] measured the average relative overlap $\overline{O}(w)$ for a link of weight w as a function of this weight. The result was a dependence shown schematically in Fig. 9.3. The monotonously increasing $\overline{O}(w)$ would clearly support the picture of Fig. 9.1—the stronger the tie, the stronger the overlap of friends between two individuals. However, the curve in Fig. 9.3 is non-monotonous. Does this argue against the weak ties hypothesis? It does not. The point is that only a small fraction (about 5%) of links have weights in the range where the curve $\overline{O}(w)$ declines. The remaining links have lower weights,

[1] This assumption is often called the *strength–of–weak–ties hypothesis* or simply the *weak ties hypothesis*.

[2] Phone calls have directions, and so, in principle, the log files allow us to construct a directed network. In the studies of Onnela *et al.* [144,145], the directions of phone calls were ignored for the sake of simplicity.

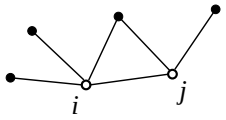

Fig. 9.2 The relative overlap of the common friends of nodes i and j in this configuration equals $O_{ij} = 1/(4 - 1 + 3 - 1 - 1) = 1/4$. This number differs from the link-clustering coefficient $C_{ij} = 1/(3 - 1) = 1/2$.

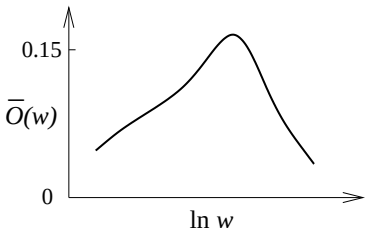

Fig. 9.3 Schematic plot of the dependence of the average overlap $\overline{O}(w)$ for a link of weight w as a function of the logarithm of this weight found in [145]. Only 5% of all links contribute to the declining part of this curve.

being in the region of monotonously increasing dependence. Thus, despite the non-monotonous form, this curve supports Granovetter's hypothesis.

What about the place of weak ties in the global structure of this network? More precisely, how can one verify the hypothesis that the weak ties interconnect dense communities? For this, Onnela and his coauthors used a very standard method of network research. They ranked all of the links in this network according to their weights w and studied how the giant component size changes if links are successively removed based on this ranking: first, starting from high w and, second, starting from low w. Figure 9.4 shows schematically the two resulting curves. The reader can see that the network is disintegrated more rapidly when the weak ties are first broken. This shows that the scheme in Fig. 9.1 is indeed valid. In the same way, instead of S one could study the variations of other network characteristics, global or local. For example, consider the changes in average clustering $\overline{C}$ with f for these two kinds of random damaging—the breaking of links starting from weak and strong ties. This quantity, $\overline{C}$, is, of course, not a global characteristic. It rather characterizes the 'local cliquishness' of a network. It turns out that the removal of links starting from the strong ones diminishes $\overline{C}$ more rapidly than if we delete links starting from the weak ties [145]. This is again consistent with the assumption that strong ties are within dense communities while weak ties connect these communities.

Finally, we approach the problem of the distribution of information flows over the network. Like Granovetter, we are interested in the the information flow coming to a node from remote sources in the network. Through which sort of ties—strong or weak—do larger 'outsider information' flows pass? Fortunately, it is possible to answer the question by only analysing the structure of this weighted network. Let us accept a minimal model for the information transfer in a network. Namely, assume that each node produces the flow of news at equal rate, and that this information is sent to all other nodes through the shortest paths. Then the information flow through a link is proportional to the number of shortest paths between all pairs of nodes, which pass through this link.[3] In the phone call network, this number turned out to be, on average, high for weak ties and low for strong ties.[4] So it is the weak ties that ensure access to remote information in these networks.

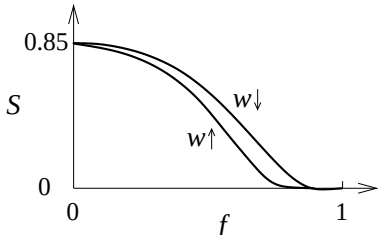

Fig. 9.4 Schematic plot of the variation of giant component size S with a fraction of removed links f. The curves labelled by $w\uparrow$ and $w\downarrow$ were obtained by the successive removal of links based on their weights, starting from low and high weights, respectively. Onnela *et al.* observed dependences of this sort in the phone call weighted network [144, 145]. Note that $S(0) < 1$ since even in the undamaged network, a fraction of nodes are in small clusters.

[3] Since the number of remote nodes is large, one can neglect the contribution from paths between close nodes.

[4] Instead of the numbers of shortest paths, Onnela and coauthors studied the betweenness centralities of links. These quantities are closely related—the larger the number of shortest paths, the larger the betweenness centrality.

9.2 World-wide airport network

Ramified transportation systems are most naturally treated in terms of weighted networks. The weight of a link in these networks is normally defined as traffic through this link. Here we consider a very representative network of airports, where nodes are airports and links are direct flight connections. In sharp contrast to basically two-dimensional railway networks, this network is a small world. Alain Barrat and coauthors analysed the world-wide airport network including 3880 major airports

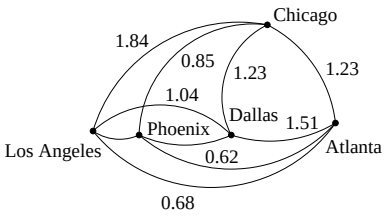

Fig. 9.5 A small part of the world-wide airport network analysed by Barrat and coauthors [19]. The weights of links are the numbers of available seats (million/year) on these direct connections.

and 18 810 direct flight connections [19]. This sparse network is actually small (less than 10^4), but this size was sufficient for a conclusive analysis. The weights of links were the numbers of available seats on given connections for 2002. Figure 9.5 shows a small piece of this network. In terms of its degree distribution, this network is scale-free. The exponent γ of the degree distribution is close to 2. Furthermore, the distribution of link weights was also found to be heavy-tailed, although noticeably different from a power law. In addition to the degrees, nodes in the weighted networks are characterized by their *strengths*. The strength s of a node is the total weight of its connections: $s_i = \sum_j w_{ij}$. Similarly to weights, the strength distribution in the world-wide airport network is heavy-tailed, but the network size is not sufficient to determine whether this distribution is scale-free or not.

As is clear from the definition, any weighted network can be treated as unweighted if we ignore the weights of its links. Then the first questions to be asked are: what is the relation between a weighted network and the underlining unweighted one? How do their statistical characteristics correlate with each other? In particular, how do the degrees of nodes and their strengths relate to each other? This question being especially relevant for scale-free networks, where node degrees and strengths fluctuate strongly. Barrat and coauthors measured the average strength of a node of degree q, $\overline{s}(q)$, in a wide range of degrees in several real weighted networks. They found that this dependence is close to a power law:

$$\overline{s}(q) \propto q^{\theta} . \tag{9.3}$$

Exponent $\theta \approx 1.5$ for the world-wide airport network. In other real-world weighted networks, exponent θ takes different values, typically, in the range from 1 to 2. That is, the hubs—nodes with many connections—usually have a high strength. In some other scale-free networks the dependence $\overline{s}(q)$ is proportional.[5] This is the case, for example, for the one-partite weighted networks of scientific collaborations [19]. In these networks, link weights indicate the intensity of cooperation within the pairs of researchers. In the weighted networks of coauthorships, the weight of a link between two researchers is proportional to the number of their joint papers. Thus a power-law $\overline{s}(q)$ curve is typical in real-world scale-free networks. What is the origin of this dependence?

[5] In this case, one can assume that $\overline{s}(q) = \langle w \rangle q$, where $\langle w \rangle$ is the average weight of a link in the network.

9.3 Modelling weighted networks

'To explain the nature of some observed phenomenon' actually means to find a reasonably realistic model demonstrating this effect. In our case, this model must provide a network in which all of the three distributions—the degree distribution, the weight distribution, and the strength distribution—have a power-law form. In addition, $\overline{s}(q)$ must be a power law or, at least, proportional to q. Barrat and coauthors proposed the first model of this kind for a growing network, exploiting a self-organization mechanism [18].

The rules for the evolution of this network are as follows.

(i) At each time step, add a node to a network. By a link of some weight, say $w_0 = 1$, attach this node to a preferentially chosen node, say node i. The probability to select a node for the attachment is taken to be proportional to the strength of this node.

(ii) Increase the strength of the node i by a constant value δ:

$$s_i \to s_i + \delta\,.$$

Do this by distributing this addition among the q_i links of this node proportionally to their weights. So the weights of these links increase:

$$w_{ij} \to w_{ij}(1 + \delta/s_i)\,.$$

This simple model is a good starting point. The reader can see that if $\delta = 0$, then the network is actually unweighted, and the model is reduced to the Barabási–Albert one. The non-zero parameter δ produces a deviation of the strength of a node from its degree and makes the evolution of weights nontrivial. For transportation, in particular, for airport networks, evolution rules (i) and (ii) are quite reasonable. Indeed, (i) a new flight connection is preferentially to a hub with higher traffic; (ii) the additional traffic due to this new connection certainly increases traffic in other flight connections from this hub. The resulting growing network has scale-free distributions of degree, weight, and strength with the same value of distribution exponent. This exponent, γ, is determined by the only parameter of the problem, that is δ.[6] As for the curve $\overline{s}(q)$ for this model network, the dependence was found to be proportional.[7] Thus this model is too simplistic to explain the power law $\overline{s}(q) \propto q^{1.5}$ observed in the real world-wide airport network.

Ginestra Bianconi found how to reproduce the law $\overline{s}(q) \propto q^{\theta}$ with exponent θ greater than 1 [27]. She proposed combining two different preferential attachment processes: (i) attachment to a node selected with probability proportional to its degree, exactly as in the Barabási–Albert model and (ii) practically the same strength–preferential attachment, as in the model described above, followed by the redistribution of weights. These two different kinds of attachment occur with complementary probabilities, p and $1 - p$. For these evolution rules, in some range of the parameters p and δ, the resulting exponent θ turns out to be greater than 1. On the other hand, there is still a wide region of p and δ in which θ is exactly 1. Thus the two classes of scale-free weighted networks can be indicated: with exponent $\theta > 1$, like the world-wide airport network, and with $\theta = 1$, like the weighted network of scientific collaborations.

For the sake of brevity, we have touched upon only local quantities here—weight and strength—and missed other characteristics. Remarkably, some characteristics of unweighted networks can easily be generalized to the case of weighted networks, and others in principle cannot. For example, in addition to the usual path length, consider the sum of the weights of the links in a path—the weight of the path. Then the notion of *an optimal path* between two nodes is naturally introduced. This

[6] The expression for the exponent is

$$\gamma = 3 - \frac{2\delta}{1 + 2\delta}\,.$$

So increasing δ results in more skewed distributions of degree, weight, and strength than for the Barabási–Albert model.

[7] Note that the model may be even simplified. Use, for example, the following rules: at each time step, increase the weight of a preferentially chosen link by some constant number and attach a new node to one of the ends of this link. This leads to the same results.

is the path with the optimal weight (minimal or maximal, depending on the problem under consideration). Optimal paths in real-world weighted networks and their models have been extensively studied. On the other hand, there is still no satisfactory analogy of the clustering coefficient in weighted networks. Note that the particular interest in weighted networks emerged rather recently, and far fewer have been studied than unweighted ones.

Motifs, cliques, communities

10

In this lecture we discuss subgraphs in complex networks. First we consider 'building blocks' of networks—*motifs*—subgraphs which are present as many copies in a network. We have already considered a simple motif—the triangle—but what is the place and role of other motifs? We also discuss ways to detect relatively weakly interconnected modules in a network—*communities*.

10.1 Cliques in networks

The k-clique is a fully connected subgraph of k nodes.[1] In particular, the triangles, which we discussed in the context of clustering, are 3-cliques. Recall that in sparse infinite classical random graphs, the number of triangles is finite, $\mathcal{N}_3 = \langle q \rangle^3/6$, where $\langle q \rangle$ is the mean degree of a node. The higher k-cliques must occur even more rarely than triangles, and therefore we can expect that the k-cliques with k greater than 3 are almost surely absent in these networks. Mathematicians have proved that this is indeed the case for sparse classical random graphs, and for them, the maximal observable clique size is 3. When are the higher cliques significant? The obvious answer is: when a network is highly clustered. This, in particular, takes place in dense classical random graphs. For us, however, widespread sparse networks are far more interesting. We have explained that sparse uncorrelated networks may have high clustering if their degree distributions decrease sufficiently slowly. Keeping this in mind, Ginestra Bianconi and Matteo Marsili inspected sparse uncorrelated networks with a divergent second moment $\langle q^2 \rangle$ and indeed found numerous (k>3)-cliques [33].

Given that at least in some networks, k-cliques are in abundance, one can ask: how are they connected with each other? Or, what is the organization of a network in terms of its k-cliques? Hungarian researchers Derényi, Palla, and Vicsek introduced the following definition: two k-cliques are adjacent (or, one may say, 'linked' or 'connected') if they share $k-1$ nodes [69]. Figure 10.1 explains this definition in the case of 3-cliques, that is, triangles. One can replace a network by the complete set of its k-cliques connected in accordance with this definition. Derényi and coauthors focused on the architecture of this k-clique network obtained from a dense classical random graph.

[1] For simplicity, in this lecture we consider only undirected networks.

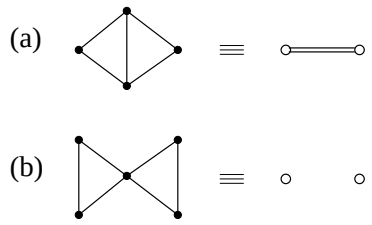

Fig. 10.1 These two 3-cliques (a) are adjacent and these two (b) are not. The open nodes and double link denote 3-cliques and a connection between them, respectively.

Let p be the probability that two nodes in the original graph are linked.[2] Assume that pN (the mean degree of a node) diverges with N, so that the the original network is sufficiently dense, and, of course, practically all of its N nodes are in a giant connected component. It turns out that the total number of k-cliques in this graph is approximately $N^k p^{k(k-1)/2}/k!$, which is a very large number in comparison with N. The degree distribution of the resulting k-clique network is Poissonian, as in the original graph. The mean degree of a node in the k-clique network, $\langle q\rangle_k$, was found to be approximately Nkp^{k-1}, which is much less than the number of nodes in this net as N tends to infinity. So the derivative network is sparse, in contrast to the original one. Derényi and coauthors showed that this sparse derivative network has a giant connected component if the probability p exceeds some critical value $p_c(k)$,

$$p_c(k)N \sim N^{(k-2)/(k-1)}\,, \tag{10.1}$$

as N approaches infinity.[3] The emergence of this giant k-clique component is called *clique percolation* [69]. So we have the three scales for a mean node degree $\langle q\rangle$ in classical random graphs: (i) $\langle q\rangle \sim 1$, a giant connected component emerges in this range; (ii) $\langle q\rangle \sim N^{(k-2)/(k-1)}$, a giant k-clique connected component emerges; and (iii) $\langle q\rangle \sim N$, where the network becomes fully connected. If $k = 2$, then scales (i) and (ii) coincide. On the other hand, if k is large, scale (ii) approaches the regime of an extremely dense network.

One may say that clique percolation is about the emergence of a connected system of strongly overlapping communities of a given size in a network. We should acknowledge, however, that the constraint of fixed size and overlap looks somewhat artificial. What if communities have different sizes and overlaps? We will touch upon this problem in one of the next sections.

[2] Here we use the $G_{N,p}$ version of a classical random graph.

[3] The k-clique network resembles a classical random graph, and a giant connected component emerges when $\langle q\rangle_k$ is about 1. So, $Np_c^{k-1}(k) \sim 1$, which results in relation (10.1). Note that $p_c(k)$ depends on N.

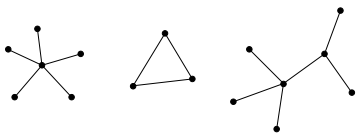

Fig. 10.2 Various stars, a triangle, and the pairs of neighbouring stars were used for the description of networks in the previous lectures.

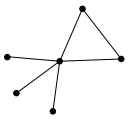

Fig. 10.3 A simple graph, which has three 1-point stars, two 2-point stars, one 5-point star, and one triangle. To present this list is actually the same as to give the degree distribution and clustering.

10.2 Statistics of motifs

We started these lectures by defining a network as a set of nodes connected by links. Then we explained that the properties of a network are to a large extent determined by its local features, namely, by its degree distribution. This can be treated as the distribution of q-pointed stars in a network. We approached an even better understanding of a network by studying the statistics of triangles (3-loops) and correlations between degrees of nearest neighbours—actually, the distribution of the neighbouring star pairs. These were the simplest patterns which we used extensively in describing the network structure, see Fig. 10.2. For example, in terms of stars and triangles, the graph shown in Fig. 10.3 is described as follows: the graph contains three 1-point stars, two 2-point stars, one 5-point star, and one triangle, and so on. In essence, this was our standard approach to networks in the previous lectures. Figure 10.4 explains another way of characterizing a network: index and enumerate network motifs starting from the smallest and simplest. In particular, in

this figure, we show the 3-node motifs (connected 3-node subgraphs). In its ultimate form, this description implies the statistics of all isomorphic subgraphs in a given network, starting from the smallest and simplest subgraphs.[4] Usually, however, only rather small motifs are discussed, that are present in many copies in a network, which are its elementary building blocks. There are only two different 3-node motifs in undirected networks, see Fig. 10.4. In directed networks, their number is markedly higher. If reciprocal links are allowed, as in Fig. 4.5 for the WWW, then this number is thirteen [126]. Figure 10.5 demonstrates some of these thirteen motifs. Draw the rest of the 3-node motifs.
In their renowned article, Milo and coauthors focused on these specific motifs in a number of real networks, including, in particular, the *E. coli* transcription network, a domain in the WWW, electronic sequential logic circuits, and food webs [126]. In addition, for each of these networks they constructed a simplified model, namely the uniformly random network of equal size, having the same sequence of in- and out-degrees. In Lecture 8 we described a randomization procedure allowing one to construct a network model of this kind, see Fig. 8.4. Milo and coauthors compared the occurrence frequencies of the motifs in all of these directed networks. It turned out that the occurrence frequencies of a given motif are very different in different directed networks and their randomized models. Moreover, the observed number of motif copies in a real network typically exceeds that for its randomized counterpart.

In principle, this difference is not so surprising and was expected. We have already described a strong difference in clustering in various undirected networks. So it is reasonable to characterize networks using a set of occurrence frequencies for motifs. One tempting idea stimulated the numerous studies of motifs in networks. The hope was that each specific motif is responsible for some function of a network.[5] If this be true, then the statistics of motifs would essentially describe and even determine the function of a network. This would enable researchers to predict details of the function of a network based on the distribution of motifs. Unfortunately, this program was never realized. Moreover, we believe that the idea is basically flawed. Here we present only one of the counter-arguments. Consider, for example, a protein interaction network, in which nodes are different proteins and links are pair-wise interactions between them. A motif in this network is a specific pattern of interactions between proteins. Distinct copies of the same motif contain different proteins. So, the different copies may be responsible for quite different functions in a network. Then, it is indeed impossible to strictly relate motifs to specific network functions.

The number of different n-node motifs rapidly grows with n. It is easy to check that there are already hundreds of different 6-node motifs each of which may be present in a network in many copies. These numbers provide sufficient statistics for meaningful analysis. Using this statistics, Baskerville, Grassberger, and Paczuski investigated the frequency of occurrence of different motifs of a fixed size in a network [21]. They measured the numbers of motif copies of sizes in the range from 8 to

[4] See Lecture 1 for the definition of graph isomorphism. Note that here only 'connected subgraphs' are considered, that is, motifs cannot have separate parts.

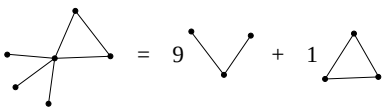

Fig. 10.4 The same graph as in Fig. 10.3 can be described as a combination of two distinct 3-node motifs. This graph also contains six 2-node motifs (two nodes and a link between them). Find the number of 4-node motifs in this graph.

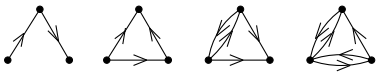

Fig. 10.5 A few of the thirteen 3-node motifs for directed networks, similar to that shown in Fig. 4.5. For the full list, see [126].

[5] The reader will find a strict formulation of this concept on the home page of Uri Alon (http://www.weizmann.ac.il/mcb/UriAlon) who is one of the authors of [126]. In application to networks of transcriptional interactions, he stressed: 'We find that much of the network is composed of repeated appearances of three highly significant motifs. Each network motif has a specific function in determining gene expression, such as generating temporal expression programs and governing the responses to fluctuating external signals. The motifs also allow an easily interpretable view of the entire known transcriptional network of the organism.'

10 in a set of real-world networks (in the *E. coli* and yeast transcriptional regulatory networks and others). For each size, they ranked all motifs according to their amount in the network and plotted the number of copies of a motif versus the rank of this motif.[6] As is very usual in network research, the implicit hope was that this distribution would be heavy-tailed or even scale-free. Interestingly, the first publication of Baskerville and Paczuski on the statistics of motifs indicated that this was the case, but finally a rapidly decreasing distribution was reliably reported for all of the studied networks. Nonetheless, no single motif was found to be prevalent in any of these networks. Instead, sets of different motifs are particularly abundant, say, dozens or hundreds of motifs, depending on motif size n. Due to overlapping, the number of motif copies may be astronomic even in relatively small and sparse networks. For example, in the *E. coli* network investigated in that work, there were only 230 nodes and 695 links. Nonetheless, the most frequent of the 6-node motifs was present in about 10^8 copies.

One can go even further and fix not only size n of a motif but also the number of its links m [176]. Remarkably, the number of the n, m-subgraphs in a network is determined by its degree-dependent clustering $\overline{C}(q)$ and degree distribution, and so this number can easily be estimated. Even without calculations, however, it is clear that in a sparse network, densely connected motifs of a given size are less frequent than motifs with a small number of links. In that sense, cliques are the least frequent motifs in networks.

[6] Researchers often represent statistical data in this way. This representation is sometimes called *Zipf's plot*. Based on the ranking, Zipf's plot provides a monotonously decreasing dependence with few noticeable fluctuations. This weakness of fluctuations allows better fitting and analysis of the data. Very similarly, instead of a degree distribution $P(q)$, empirical researchers usually plot a so-called *cumulative distribution* $P_{\rm cum}(q) = \sum_{q' \geq q} P(q)$, which has fewer visible fluctuations than $P(q)$. You can equivalently do the following. In the spirit of Zipf, rank the nodes of a network according to their degrees and plot the rank r of a node versus its degree (or, if you want, degree versus rank). Clearly, $r(q)/N = P_{\rm cum}(q)$, and so we have full equivalence.

10.3 Modularity

We have demonstrated that nodes in a typical social network fall into a set of well-distinguished dense modules, see Fig. 9.1. In contrast to network motifs, these modules have no joint nodes: overlapping is absent by definition. Importantly, there are many links within modules and relatively few links between nodes in different modules. Note that, in principle, a given network may contain modules of very different sizes. Various situations are possible. In one network, modules of, say, 3 and 300 nodes can be found, and another network is of only two large modules. Consequently a modular organization is usually about a large-scale clustering of nodes in a network. A pronounced modular structure, in other words, high modularity, is typical not only for social networks but also for many others. So, how can we measure the modularity of a network? This important question is addressed frequently in network science, and it will be instructive to discuss this problem in detail.

Note that, in principle, details of the modular organization of a given network are not known *ab initio*. This modular structure—a set of modules in which nodes naturally fall—should be found. This difficult task can be carried out in two steps. Firstly, assume some particular division of the network into modules and define a measure of the 'quality' of this division. We will explain below what is a high-quality partition. It is this

quality measure—a characteristic number—that is called *the modularity* of a given partition of a network. The second, most difficult step is to compare all possible partitions of the network and find the best of them, with optimal modularity. The optimal partition gives the actual set of modules in the network. The modularity for this partition is the modularity of a network. Unfortunately, the second step—optimization—cannot be done explicitly. The point is that the number of possible divisions grows rapidly as a factorial of the network size. This number is huge even in small networks. This makes the optimization problem computationally very hard, requiring some approximations and efficient numerical algorithms. The main content of hundreds of published papers on network communities consists of a frantic search for the best approximations and algorithms of this kind.

Let us focus on the modularity of a particular partition of a network, leaving the discussion of step 2 for the next section. Our immediate task is to define modularity Q for a given division of a network into a set of modules. Assume that a network of L links is divided into n modules, $r = 1, 2, \ldots, n$, with l_r links in module r. The definition should preferably provide the maximum modularity if all modules are separated from each other, and the minimum, say $Q = 0$, in a homogeneous situation, in which modules are imperceptible. Mark Newman and Michelle Girvan found a way to meet these requirements [139]. They proposed to base the definition on the comparison of two networks: a given network and its zero modularity counterpart—the uniformly random network with the same sequence of degrees as the original one. A quantity for comparison was the ratio of the total number of links within modules and the number L of links in the network. In other words, this is the probability that a link is within one of the modules. This gives the definition:

$$Q = \frac{1}{L}\sum_{r=1}^{n} l_r|_{\text{given net}} - \frac{1}{L}\sum_{r=1}^{n} l_r|_{\text{uniform counterpart}}\,. \tag{10.2}$$

In the uniformly random counterpart, the ratio $\sum_r l_r/L$ can easily be calculated since we know the degrees of all nodes. It is actually enough to know the total degrees q_r of the nodes within individual modules, where $r = 1, 2, \ldots, n$ and $\sum_r q_r = 2L$. We must find the probability $p_r = (l_r/L)|_{\text{uniform}}$ that a randomly chosen link in the uniformly random network is in module r. Similarly to the configuration model, the probability that a given end of a link is in module r is proportional to q_r. Consequently, the probability p_r is proportional to q_r^2.[7] Then $p_r = (q_r/2L)^2$. Indeed, if, for example, the entire network consists of a single module, we have the probability $(2L/2L)^2 = 1$. Then the definition of modularity takes the form:

[7] Note that in the uniformly random counterpart, multiple and self-connections are allowed.

$$Q = \sum_{r=1}^{n}\Big[\frac{l_r}{L} - \Big(\frac{q_r}{2L}\Big)^2\Big]\,. \tag{10.3}$$

Clearly Q is smaller than 1. If a network has strong community structure, that is, well-distinguished modules, then for this partition we have

Q close to 1. Another division of the same network gives a smaller Q. In a uniformly random network, $Q = 0$. However, this definition allows even negative Q in some situations. In particular, this can occur if multiple and self-connections are absent in a given network. For example, let a network without 1-loops be divided into N modules, each of a single node. These 'modules' clearly have no internal links, and the first term in the definition equals zero. Contrastingly, the uniformly random counterpart contains self-connections, which guarantees a positive second term. Together, these terms give negative modularity.

Instead of expression (10.3), researchers often use an equivalent one, in terms of the adjacency matrix a_{ij} and the node degrees q_i,

$$Q = \frac{1}{2L} \sum_{r=1}^{n} \sum_{i,j \in r} \left(a_{ij} - \frac{q_i q_j}{2L} \right). \tag{10.4}$$

Here the sum $\sum_{i,j \in r}$ is over all pairs of nodes in module r, and the difference $a_{ij} - q_i q_j / 2L$ is called a *modularity matrix*.[8]

Typically, positive modularity is observed in real-world networks, in the range about 0.3—0.8.[9] For demonstration purposes, here we compile only a short list of modularity values, which Mark Newman obtained for a few diverse networks of different sizes and architectures [135].

- For the karate club network of Zachary of 34 nodes, Q is 0.419.[10]
- For a metabolic network for the nematode *C. elegans* of 453 nodes, Q is 0.435.
- For a coauthorship network of scientists working on condensed matter physics of 27 519 nodes, Q is 0.723.

Importantly, it is hard to find a real-world network with low modularity. Even this short list shows that modular network architectures are widespread.

The useful modularity definition (10.2) has been implemented in hundreds if not thousands of studies. Nonetheless, there is some controversy. First of all, one may wonder: is it in principle possible to describe so complicated a feature as modularity by a single number, Q? The second point is: in the definition, an uncorrelated network with multi- and self-connections is set as a zero modularity counterpart of a given network. This choice is not that obvious if, say, the original network has no multiple connections. We noted that, strangely, when applying to networks without multi- and self-connections, the original definition may produce a negative modularity. Furthermore, for a linear chain, the definition counterintuitively gives non-zero modularity, despite the fact that chains can hardly be called modular objects.[11] In short, be careful when using this definition of modularity.

10.4 Detecting communities

There is no unique definition of a network community. In many works, the communities are natural, non-overlapping modules of networks,

[8] To understand this form of the definition, recall that $\sum_{i,j} a_{ij} = 2L$ (see Lecture 1.

[9] Here modularity is for the optimal partition of a network into modules.

[10] The small Zachary karate club graph [184] was a reference network in numerous studies of modularity and communities. This is a one-partite network of friendships between members of a karate club at one of the US universities. A social conflict in this group divided the network into two overlapping communities, which made community indexing non-trivial. Researchers often test new community detection algorithms on this network.

[11] There is a natural way to improve this definition. Namely, impose the structural constraints of an original network on its zero modularity counterpart. For example, let an original network have no multiple connections. Then its zero modularity counterpart must also have no multiple connections. This way, however, was still not realized.

which we have discussed in preceding sections. Other studies suppose that communities can overlap. In the third type of study, the communities are hierarchically embedded one into another—the nested structure of communities. Finding/indexing communities is a remarkably attractive research direction in network science. How can we explain such a keen interest? The main aims of these studies are to find natural groups of nodes in a network and connections between them, to uncover and understand the coarse structure of a network, and to indicate particularly 'influential' communities. Of course, we can rather easily achieve these aims if a network is small, say, of 10, or 20, or 30 nodes. Analyse this small network visually and indicate its dense, closely connected parts—communities. Certainly, this can be done only if these communities are sufficiently distinct from each other. If, however, a network is large, say of hundreds nodes or bigger, the 'manual' detection of communities is impossible and an automated-indexing solution is the only option. For large networks, solution is an extremely time-consuming task. In addition, communities are poorly defined and often hardly distinguishable, so it is very difficult to develop a universally efficient numerical algorithm for the problem [94].

For a brief survey of various techniques used for indexing communities, the reader should address article [94]. Here we outline a few key ideas. Let us start with the problem of detecting non-overlapping communities.[12] The distinguishing feature of these communities is a relatively high density of links. The idea is to trace the motion of a random walker on the network over a long period of time. A random walker spends more time in areas with a higher density of links, and so efficiently indicates communities. One can consider different random walk processes. For example, in one process, at each time step, a walker moves from a node to any of its q nearest neighbours with equal probability $1/q$. In another process, a walker moves from a node to any of its nearest neighbours with a probability p and remains at the node with the complementary probability $1 - pq$.[13] Note that these two random walk processes differ sharply from each other. We will explain the difference in detail in one of the following lectures. Here we stress that results (a particular picture of communities) obtained using these algorithms, depend strongly on an implemented random walk process. So the idea is to attain the clearest recognition of communities in a wide set of networks by choosing some particular random walk process. Often, this is possible.

Instead of tracing a random walk, researchers usually inspect the action of a mathematical operator, which generates this random walk. Technically, this is more convenient. In mathematical terms, the problem is about the eigenvalues and eigenvectors of this operator. Moreover, the general idea of analysing the spectral properties of some specific mathematical operator defined on a network leads this field [135].

These 'spectral' algorithms allow us to extract non-overlapping communities. In their renowned paper, Mark Newman and Michelle Girvan considered another, 'hierarchical' picture of network communities [139].

[12] In the preceding section, we explained that the maximum modularity Q corresponds with the actual set of network communities. In large networks, it is in principle impossible to explicitly optimize Q over all possible network partitions. So all of the numerous algorithms for network community discovery perform this optimization approximately or, alternatively, bypass the optimization problem completely by exploiting some heuristic arguments. For example, use the fact that the density of connections is higher within communities.

[13] Here $pq_{\max} \leq 1$, where $q_{\max}$ is the maximum degree in a network.

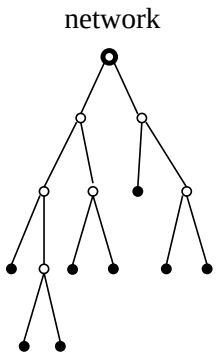

Fig. 10.6 The hierarchy of communities in a network according to Newman and Girvan [139]. The nodes are the communities, the root is the network itself, and the dead ends (filled dots) are the nodes of the network, that is, the smallest communities. Each link connects a community with a directly embedded community. In this example, the network is of 8 nodes. The two largest communities are of 5 and 3 nodes.

In this picture, communities are successively nested one into another, so that a network consists of two complementary communities, each of these communities consists of two communities, and so on. This splitting is always possible if a community has two or more nodes. This hierarchy of network communities has athe structure of a tree, shown in Fig. 10.6. Each node in this tree is a community, and the root is the original network. The dead ends of this tree (excluding the root) are the smallest possible communities, which cannot be further split. Clearly, these dead ends are the nodes of a given network. So the number of dead ends in this dendrogram (excluding the root) coincides with the number of nodes in the network.

The algorithm of Newman and Girvan uses ideas which we have already discussed in the context of the 'strength of the weak ties' concept. Recall that according to Granovetter the connections between communities in a network (interlinks) carry larger information flows than links within communities (intralinks). The information flow through a link may be roughly estimated as the number of shortest paths between pairs of nodes, that pass through this link. So, the numbers of shortest paths passing through interlinks is greater than those passing through intralinks. One can believe that this is the case, not only in social networks, but generally. Newman and Girvan used this feature directly. They proposed to uncover communities by progressively removing the links that carry the highest numbers of shortest paths. Instead of these numbers, they used the link betweenness centrality, which we discussed in Lecture 5. In more strict terms, the algorithm of Newman and Girvan is as follows [139].[14]

(i) Compute betweenness centrality for all links in a network.

(ii) Remove the link with the highest betweenness centrality.

(iii) Recalculate betweenness centrality for the remaining links.

(iv) Repeat from step (ii) until no links remain.

This algorithm progressively divides a network into a set of separated clusters. Note step (iii), which is the recalculation of the betweenness centrality of all remaining links after each removal.

Why do we have to perform these time-consuming repetitive recalculations instead of consecutively removing links with the highest betweenness centralities in the original network? The authors explained the necessity of the 'recalculation step' using a simple example. Suppose that two of the communities in a network are connected by two links, one of which has the highest betweenness centrality in the network, the second having a very low betweenness centrality. Removing these two links uncovers the obvious division into these two communities. It is clear that the link with the highest betweenness must be removed immediately. After removal, recalculation will give a high betweenness centrality for the second link. So this link will also be quickly removed, and the natural division will be readily uncovered. On the other hand, without the recalculation, the second, low betweenness centrality link

[14] We assume that a network is one-partite and undirected.

would be deleted much later, after many other links. This would hamper the discovery of these communities or even make it impossible.

As the algorithm runs, the number of links decreases from L, which is the total number of links in the original network, to 0, and, in parallel, the number of extracted communities increases from 1 (the network itself) to N (the total number of nodes). The result is a dendrogram presented in Fig. 10.7, where vertical lines show communities. Note that this is a more informative representation of the same network community structure than in Fig. 10.6. For each number of removed links, this dendrogram presents a natural partition of a given network (see, for example, the intersection of the dendrogram and the dashed line in Fig. 10.7). So the algorithm provides a set of less than L partitions into communities, say, n partitions. This number is far smaller than the total number of all possible partitions for a given network, which is a good point. Thanks to the smallness of this set, we can quickly compute a modularity Q for each of these n specific partitions and select the 'best' partition having the highest modularity. For this, use the formulae from the preceding section. Thus, the Newman–Girvan algorithm finds the optimal division of the network into non-overlapping communities, but also uncovers the hierarchical organization of network partitions—the hierarchy of nested communities, see Figs. 10.6 and 10.7. We stress that the Newman–Girvan algorithm is approximate. It is based on heuristic arguments. Indeed, the key assumption about the high betweenness centralities of the interlinks is only a hypothesis. At best, this property can only be valid statistically, with less than 100 per cent probability. As a result, we have the chance to miss the optimal partition. Nonetheless, despite the strong assumption behind this famous algorithm, it uncovers the community structures of various networks with reasonable success. The algorithm correctly finds the modular structure in clear situations and approximately reproduces the results of more sophisticated algorithms in more difficult cases.[15]

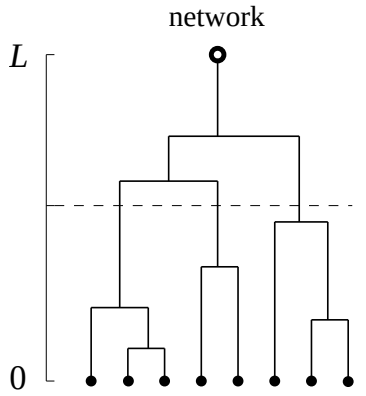

Fig. 10.7 Another representation of the community structure of the same network as in Fig. 10.6. This dendrogram is a direct result of the application of the Newman–Girvan algorithm [139]. The vertical axis shows a current number of links in the network during the execution of the algorithm. For example, at the level indicated by the dashed line, the algorithm shows three communities.

The idea that each node in a network enters into only one community, can easily be disputed. Actually, this is a very simplifying assumption. For example, in developed social networks, an individual is inevitably involved in many social, cultural, and professional activities, which results in simultaneous participation in a number of communities. This overlapping of communities severely hampers the uncovering of the community structure of a network. A few attempts have been made to fulfil this difficult task. Let us consider one of them.

In 2005, Palla, Derényi, Farkas, and Vicsek proposed defining communities using their concept of k-clique percolation [147]. We have explained this notion. Let the number k be given. According to Palla and coauthors, a community is a connected k-clique component. In other words, it is a subgraph consisting of all the k-cliques that are mutually reachable through $(k-1)$-node overlaps. It is clear that two communities of this kind can overlap, but the overlapping is not by more than $k-2$ nodes. Figure 10.8 explains the idea of Palla and coauthors. Unfortunately, community structure strongly depends on k. So the problem is

[15] There are situations, where the algorithm fails: e.g. three equal cliques interconnected by three links.

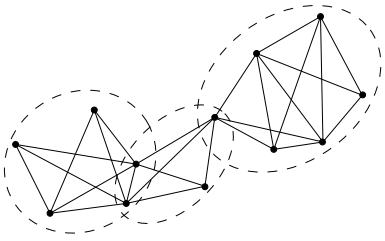

Fig. 10.8 Overlapping k-clique communities according to Palla and coauthors [147]. In this example, $k=3$. The networks consists of three communities: of two, one, and three 3-cliques.

how to choose this number. Both large and small k lead to an apparently incorrect community structure. Indeed, if k is too large, the resulting communities will be separate. On the other hand, if k is too small, the communities will be unrealistically large. If, for example, $k=2$, then these communities are simply connected components, and so a network of a single connected component is one community. Therefore, the researchers chose some intermediate value ($k = 4$, 5, or 6, depending on the network), which gave a sufficiently rich community structure with numerous overlaps by 1—30 nodes. They investigated this community structure in a number of highly clustered biological and social networks. Remarkably, the size distribution of these communities was found to have a power-law large-degree part, with exponent in the range 2.0—2.6. This power-law distribution was observed even in non-scale-free networks. Furthermore, one can naturally introduce the 'degree' of a community. This is the number of its overlaps with other communities, that is, the number of nearest neighbours. The observed degree distribution for the communities is also scale-free.

The work of Palla and coauthors indicates the difficulty of this rapidly progressing research field. It is not easy to arrive at a uniform community picture since different researchers can understand and define communities very differently.

10.5 Hierarchical architectures

Using tree or, equivalently, dendrogram schemes similar to those in Figs. 10.6 and 10.7 allows one to describe a hierarchy in the set of nodes in a network. In particular, in the tree in Fig. 10.6, the hierarchical position of a node is naturally characterized by the distance of this node from the root. Importantly, the forms of tree or dendrogram in these figures reflect the global organization of a network. The difference between these trees for different networks may be spectacular.[16] Figure 10.9 shows two contrasting examples of dendrograms for two different networks. Apparently, the right dendrogram indicates the presence of well-distinguished communities in the network and, one may say, its strictly 'hierarchical architecture'. In other words, this net contains a clear hierarchy of communities, unlike the network producing the left dendrogram. Hierarchical architectures of this kind have been reported in a number of real-world networks, for example, in networks of metabolic reactions [155, 154].

Deterministic graphs in Fig. 7.6 also have a clear hierarchical design. For these deterministic graphs, one can easily find degree-dependent clustering $\overline{C}(q)$. This quantity is inversely proportional to degree: $\overline{C}(q) \propto 1/q$. It is tempting to guess that a decreasing dependence $\overline{C}(q)$ is a generic feature of hierarchically organized networks. We believe that this is not correct. Many networks with degree–degree correlations show this dependence even without being hierarchical.

[16] Aside from the Newman–Girvan algorithm, there are a number of ways to build tree schemes of the node hierarchy for networks. The resulting trees and dendrograms also depend on the algorithm used.

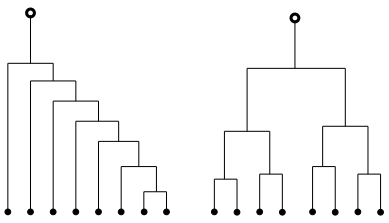

Fig. 10.9 Two contrasting examples of dendrograms for two networks. The right dendrogram indicates the hierarchical organization of a network.

Navigation and search

11

The starting point in the science of networks—Euler's solution of the Königsberg bridge problem—addressed walks on a graph. In comparison to lattices, complex networks provide remarkably rich environments for walks and related processes. In this lecture we consider the specifics of walks, navigation, and search processes in networks of various architectures and geometries.

11.1 Random walks on networks

For the sake of brevity, we touch upon only one version of a random walk. At each time step, a walker moves randomly from node to node along the links of a graph. In a so-called *simple random walk*, walker moves from a node to any of the nearest neighbours of this node with equal probability (the drunkard's walk). We can start a walk from a given node in a network and keep track of how it evolves: namely, how the probability of finding the walker at various nodes varies, how and when the walker returns to the starting node or moves off to infinity, and so on. Since a walker spends more time in dense parts of a network, we can use this process to explore the heterogeneous structure of the network. We mentioned that random walk based algorithms are applied to the detection of communities. Numerous chaotic 'one-particle' processes taking place on networks can be treated as random walks.[1] Recall, for example, Internet traffic. Optimally, packets in the Internet traffic are routed along the shortest path between a source and destination. In reality, however, the routing of packets is often so chaotic that it rather resembles random walks.

[1] Physicists use the term 'one-particle processes' for processes in which a single particle participates or particles move independently.

Suppose that a graph (or lattice) of size N is regular, that is all nodes have equal degrees. Then, at infinite time, we will find the walker at any node with equal probability, $1/N$, irrespective of the starting point of the walk. Heterogeneity changes this result. Let a network have no separate parts. Then it is easy to show that the probability of finding a walker at a node of degree q approaches the following final value:

$$p_{\text{fin}}(q) = \frac{q}{Q}\,, \tag{11.1}$$

where $Q = \sum_i q_i$ is the total degree of nodes in a network. In a random network, $p_{\text{fin}}(q) = q/(N\langle q\rangle)$. Figure 11.1 explains this formula. Note that this result does not depend on the architecture of the network. Relation (11.1) is correct only for undirected networks. The structure

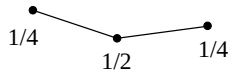

Fig. 11.1 The numbers indicate the final probabilities of finding a random walker at the nodes of this graph. In the final stationary state, for each link, the probabilities of moving in the opposite directions between the end nodes must be equal: $1 \cdot 1/4 = 1/2 \cdot 1/2$.

of connections in a directed network is more rich, as we observed for the WWW. There may be an area or areas connected to the rest of a network only by incoming links. These areas (components) are deadly traps for a random walker. If a network is finite, then the final probability $p_{\rm fin}$ is non-zero only in these traps. Consequently, random walks, enable us to detect specific components in directed networks.

The simple relation (11.1) leads to another important result for random walks. The problem is: how soon will a walker return to the origin of his walk? In 1947, Mark Kac derived his famous formula which directly relates the average first-return time $\langle\tau\rangle$ to the origin of a walk to the probability $p_{\rm fin}$ of finding a walker at this point at infinite time: $\langle\tau\rangle = 1/p_{\rm fin}$. From the Kac formula, we readily obtain the value of the average first-return time to starting node i of degree q_i [141]:

$$\langle\tau_i\rangle = \frac{\langle q\rangle}{q_i}\,N\,. \tag{11.2}$$

This time diverges in an infinite network. Note, however, that for hubs, the mean return time may take a rather moderate value even for extremely large networks. Let us estimate, for example, $\langle\tau_{\rm h}\rangle$ for a hub in a scale-free recursively growing network with a degree distribution $P(q) \sim q^{-\gamma}$. Let exponent $\gamma = 2.1$ and $N = 10^{11}$. We have shown that the largest number of connections of a node in this network is of the order of $N^{1/(\gamma-1)}$. Then for this node we have the average return time of the order of $N^{(\gamma-2)/(\gamma-1)} = 10$. Even if a network is of astronomical size, $N = 10^{33}$, we obtain a rather moderate number $\langle\tau_{\rm h}\rangle \sim 1000$.

We can naturally generalize the average first-return time $\langle\tau_i\rangle$ to $\langle\tau_{ij}\rangle$. This is the average first-passage time of a walk which starts from node i and passes node j. This quantity characterizes separation between nodes i and j in a network from the point of view of a drunken sailor who started his walk from node i. In respect of stochastic processes of this kind, $\langle\tau_{ij}\rangle$ is a more adequate characteristic of a network than a shortest-path distance. For example, for random traffic, $\langle\tau_{ij}\rangle$ is more relevant. Note that, in heterogeneous networks, $\langle\tau_{ij}\rangle \neq \langle\tau_{ji}\rangle$ [141]. This asymmetry allows us to find which one of two nodes is more rapidly approachable by random walking.

Let us focus on infinite networks. The next natural question about a random walk is: what is the probability that a walker will return to the starting point at all? Is there a chance of escaping to infinity without returning to the origin? Apparently, the answer depends on the number of ways of escaping to infinity, that is on the dimensionality of a medium for walking. It is easier to escape in a highly-dimensional lattice than, say, in a one-dimensional chain. According to Pólya's theorem (1921), there are two possible situations for walks on a D-dimensional medium.[2]

(i) If $D \leq 2$, then a walk certainly returns to the starting point, that is the walk is *recurrent*.

(ii) If $D > 2$, a walk has a finite probability of escaping to infinity without returning to the starting point, that is, the walk is *transient*.

[2] George (György) Pólya (1887–1985) was a famous Hungarian mathematician.

Therefore random walks on small worlds (infinite-dimensional networks) are transient.[3]

Physicists traditionally study a more detailed characteristic of random walks, namely, an autocorrelation function $p_0(t)$, which is the probability of finding a random walker at the starting node after t steps. We have already mentioned that for random walks on infinite D-dimensional lattices and fractals, $p_0(t) \sim t^{-D/2}$. Consequently for small worlds, we should observe a slower decay of the autocorrelation function than any power law. In particular, for classical random graphs, $p_0(t) \sim \exp(-Ct^{1/3})$, where C depends on $\langle q \rangle$ [157].[4] Unfortunately, in contrast to lattices, this specific decay can be observed only in unrealistically large networks. Their diameter, $\ln N$, must be very much greater than 1. It is easy to show, however, that in finite networks, the decay of the autocorrelation function is exponential $p_{0i}(t) \sim e^{-t/\tau_i}$, where the relaxation time τ_i, in principle, depends on a node. Somewhat counterintuitively, it turns out that, in contrast to the average first return time, eqn (11.2), this dependence is rather weak. Even in scale-free networks, where degrees vary by a few orders of magnitude, τ_i of all nodes were found to be surprisingly close to each other [141].

[3] This feature—transience—is closely related to another classical result for random walks. If $D > 2$, then after t steps, a random walker visits, on average, n_v different nodes, where n_v is a finite fraction of t. In particular, this is the case for small worlds.

[4] One can show that the decay of the autocorrelation function is determined by low-degree nodes, so this law is also valid for other uncorrelated networks, including scale-free. For this, a network must have nodes with less than three connections, otherwise, the decay is exponential.

11.2 Biased random walks

Let us return to the routing of packets in the Internet. Because of a number of reasons, which we will touch upon in the next lecture, the traffic is usually far from optimal. So the path of a packet from source to destination may be very far from the shortest path. In this regime, a packet moves at random. The key point is that the destination node attracts a packet, so that the random walk is biased, and the bias is towards the target. Importantly, the parameters of this walk depend strongly on the magnitude of the bias. It is interesting to understand how much bias can change the character of a random walk. The presence of bias means that the probability of a jump from a node in the direction of the destination exceeds the probability of a jump from this node in the opposite direction, see Fig. 11.2. One can show that, for biased random walks on networks, the relevant bias is exponential, that is the ratio of the probabilities is fixed:

$$\frac{p(i;\ell \to \ell-1)}{p(i;\ell \to \ell)} = \sqrt{\lambda} = \frac{p(i;\ell \to \ell)}{p(i;\ell \to \ell+1)}\,, \qquad (11.3)$$

where $\sqrt{\lambda}$ is some number greater than 1. The square root is introduced for the sake of convenience, since $p(i;\ell \to \ell-1)/p(i;\ell \to \ell+1) = \lambda$.

Here we discuss only one characteristic of the biased random walks on a network, namely, the mean return time, also averaged over all nodes, $\langle \tau \rangle$. Specifically, the dependence of $\langle \tau \rangle$ on the network size is important. It turns out that the dependence $\langle \tau \rangle(N)$ changes sharply at a certain, critical value of the bias parameter, λ_c, [167, 23]. This critical value exactly coincides with the mean branching in the network, $\lambda_c = \overline{b}$. There are three contrasting regimes:

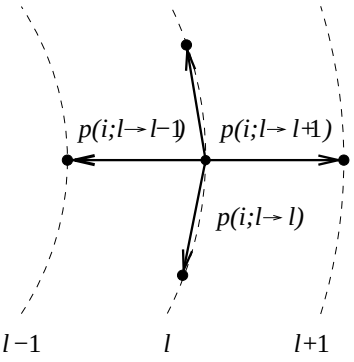

Fig. 11.2 The nodes in a network are additionally labelled according to their shortest-path distance ℓ from a destination (node-attractor). In a biased random walk, the probability of a jump from a node depends on the direction of this jump. Namely, $p(i;\ell \to \ell-1) > p(i;\ell \to \ell) > p(i;\ell \to \ell+1)$.

(i) If $1 \le \lambda < \lambda_c$, then $\langle \tau \rangle \sim N^{\ln(\overline{b}/\lambda)/\ln \overline{b}}$.

(ii) If $\lambda = \lambda_c$, then $\langle \tau \rangle \sim \ln N$.

(iii) If $\lambda > \lambda_c$, then $\langle \tau \rangle$ approaches a finite value at large N.

The regime below λ_c is delocalization. In this regime, in an infinite network, a walker escapes the bias. That is, the walks are transient. At the critical value λ_c, a *localization transition* occurs, and above this point, a walker is trapped by the bias. In this regime, the walks are recurrent, that is most walks approach the destination in a finite time. The localization transition disappears if $\overline{b} \sim \langle q^2 \rangle$ diverges, that is, hubs hamper localization. One should stress that routing in the Internet is much more complicated than this idealistic biased random walk model, which prevents meaningful comparison. In reality, packets travel not quite independently of each other, queueing at network routers.

11.3 Kleinberg's problem

Recall Milgram's small-world experiment, in which he found the famous six degrees of separation between individuals. We described the experiment in detail in Lecture 1. Note that these six degrees are not the length of the shortest path between two persons in the network of acquaintances but rather a rough estimate from above. Indeed, what was the essence of Milgram's idea if we ignore less important details? The participants in the experiment were asked to forward a letter to those of their acquaintances who were closer to the target person. The target's address was known, and the 'closer', in this idealization, simply means 'geographically closer'. Each participant knew the addresses of his or her acquaintances, and so it was easy to select a proper recipient. In this search process (searching for the shortest route to the target), all participants used very reduced, local information, namely the coordinates of their acquaintances, and, of course, the target's address. The participants had no idea about the full structure of their network.

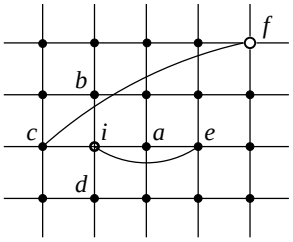

Fig. 11.3 How the greedy algorithm works. Let the goal be to reach the target (node f), starting from node i, in the shortest time. The greedy algorithm results in a four-step path passing through node a, while the shortest path through node c is of only two steps.

Remarkably, Milgram's algorithm is the standard one in computer science. Computer scientists call algorithms of this kind *decentralized search algorithms*. Specifically, this is the simplest process—*the greedy search algorithm*: repeatedly pass to the neighbour node, geographically closest to the target. Importantly, the nodes use only local information. We can roughly model the network of acquaintances in Milgram's experiment by a two-dimensional lattice with shortcuts—a small-world network. Figure 11.3 explains the greedy algorithm for this network. Each step is to the nearest neighbour, geographically closest to the target. So, with each step, we are getting geographically closer to the destination. This restriction—strong bias towards the target—simplifies the algorithm and makes it ultimately efficient and quick. On the other hand, 'geographically closer' does not mean 'closer in the network'. So moving in this manner, we easily miss the shortest path to the target even in very simple situations like that in the figure. We conclude that

Milgram's six degrees of separation are simply the delivery time of the greedy algorithm for his specific problem.

How quickly can we find a target in a network by using the greedy algorithm (greedy routing)? Or, equivalently, how easily can we navigate through a network? The answer depends strongly on the architecture of the network and on its size. A relevant characteristic quantity for a network is an average delivery time of the algorithm, $\overline{\tau}$. Here the average is over all pairs of nodes–starters and nodes–targets in the network. Let the substrate of the network be a D-dimensional lattice of $N = L \times L \times \ldots \times L$ nodes. Then $\overline{\tau}$ depends on L and D. The size dependence $\overline{\tau}(L)$ is the main issue of interest. For a lattice without shortcuts, clearly, $\overline{\tau} \sim L$, and it takes lot of time to reach the target. Apparently, added shortcuts diminish delivery times, but how much?

In 2000 computer scientist and applied mathematician Jon Kleinberg considered this problem for a generalization of Watts–Strogatz small-world networks [110–112]. Originally, his network was based on a two-dimensional lattice substrate, but here we assume it to be D-dimensional. Each node of the lattice has a shortcut to a node at the Euclidean distance ℓ drawn from a power-law probability distribution, $p(\ell) \propto \ell^{-\alpha}$.[5] If exponent α equals zero, then shortcuts connect uniformly randomly chosen nodes, and so we arrive at one version of a small-world network. If α is large (short-range shortcuts), then the network, in effect, approaches a D-dimensional lattice, and $\overline{\tau} \sim L$. Long-range shortcuts surely decrease the delivery time, but how much? In particular, uniformly distributed shortcuts ($\alpha = 0$) give a small chance of getting closer to the target and so they they do not substantially improve navigation compared to a pure lattice. Kleinberg studied the size dependence $\overline{\tau}(L)$ in the entire range of exponent α values, from zero to infinity, and found dramatically different dependences. Figure 11.4 shows the resulting average delivery time versus exponent α for a network of a given size. The main finding was that the delivery time of the greedy algorithm has a deep minimum at $\alpha = D$. Kleinberg proved that at this unique point, the size-dependence of the delivery time is very slow—polylogarithmic, $\overline{\tau} \sim \ln^2 L$, while the delivery time increases as a power of L, $\overline{\tau} \sim L^x$, at all other values of α. Exponent x approaches zero as $\alpha \to D$. This sharp difference allowed Kleinberg to introduce the notion of navigability. In terms of Kleinberg, a network is *navigable* if the greedy algorithm provides rapid navigation, that is if $\overline{\tau}(L)$ increases slower than any power law of L.

Kleinberg's construction provides a navigable network at one point, $\alpha = D$. Note that even at this point, the delivery time much exceeds the average shortest-path length, $\ln^2 L \gg \ln L$, $L \gg 1$. However, for realistic network sizes, the logarithm is not large, and the difference is not impressive. So at this, and only at this, point the greedy algorithm does a good job. Although we do not show how these results were obtained, it is worthwhile explaining a few key issues. Remarkably, this system is a small world, that is infinite-dimensional, in a wider range of α than one could expect. This takes place for $0 \le \alpha \le 2D$. In this

[5] In principle, the number of shortcuts per node can be taken to be equal to an arbitrary finite number, without loss of generality.

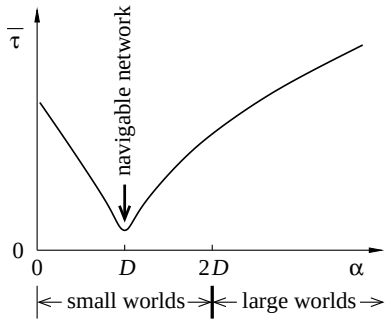

Fig. 11.4 Schematic plot of the average delivery time provided by the greedy algorithm versus exponent α according to Kleinberg [110]. The size of the network is fixed. When the dimensionality D of the underlying lattice is below or equal to $2D$, the network is a small world.

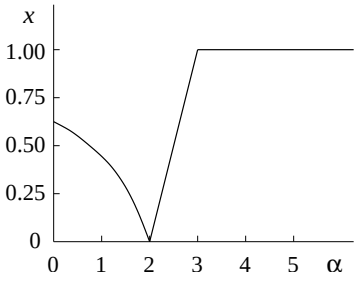

Fig. 11.5 Delivery time exponent x ($\overline{\tau} \sim L^x$) versus exponent α for the two-dimensional Kleinberg model according to Carmi *et al.* [51] and Cartozo and De Los Rios [52]. The network is navigable at $\alpha = 2$.

region, except when $\alpha = D$, $\overline{\tau}$ is dramatically greater than the average shortest-path length. Compare the analytical expressions for asymptotic dependencies $\overline{\tau}(L)$ at various α [51, 52]:

$$\overline{\tau}(L) \sim \begin{cases} L^{(D-\alpha)/(D+1-\alpha)} & 0 \le \alpha < D\,, \\ \ln^2 L & \alpha = D\,, \\ L^{\alpha-D} & D < \alpha < D+1\,, \\ L & \alpha > D+1\,. \end{cases} \tag{11.4}$$

Figure 11.5 illustrates these formulae showing the dependence $x(\alpha)$ of the exponent in the power law $\overline{\tau} \sim L^x$.

Computer scientists widely discussed Kleinberg's major result—the optimal navigation time $\overline{\tau} \sim \ln^2 L$ for greedy routing. Importantly, they could not reduce this delivery time by using more sophisticated algorithms. Suppose that we try to 'improve' the greedy algorithm by additionally accounting for the positions of the second nearest neighbours of a node or in an even larger neighbourhood, say, involving n nodes.[6] This will certainly reduce the path to the target. Its length will be closer to that of the shortest path. However, estimating the effective time for navigation, we must multiply the length of the path by n. It turns out that this product cannot be less than of the order of $\ln^2 L$. In that respect, $\overline{\tau}$ is a basic network characteristic.

[6] Note that this 'improvement' uses non-local information about the network.

11.4 Navigability

We missed one important point in Milgram's experiment. Many chains of acquaintances in the experiment were not completed, so that a fraction of the letters failed to reach the target. That fraction was so large that at first Milgram's attempt was unsuccessful. This is not the case for networks of the kind shown in Fig. 11.3, that is for regular lattices with added shortcuts. On these networks, all search chains generated by the greedy algorithm surely reach the target. We can easily modify the model network to reproduce broken Milgram's chains. For example, remove from the underlying lattice a fraction of nodes or links. This results in a finite probability that the greedy algorithm will lead a searcher (navigator) into a trap—nodes without neighbours geographically closer to the target. The greedy algorithm does not allow a searcher to return backwards for a few steps to try to avoid the trap. When the number of removed nodes or links is large, the probability of being get stuck in this way can be high. In this situation, the greedy search algorithm is, of course, utterly impractical, and efficient searching and navigation are impossible without global knowledge. It is natural to call a network *searchable* if a sufficiently large fraction of the search chains in this net are successful [180]. In more quantitative terms, in a searchable network, the probability of an arbitrary search chain reaching its target must exceed some given value. This threshold value determines the quality of searching.

Whether the model network is searchable or not depends on its size, on how much the underlining lattice is damaged, and on the distribution of shortcuts (exponent α). Here we will not discuss the form of the area of searchability and other details. Instead, let us look at efficient navigation on networks from a more broad perspective. Instead of damaging the underlying lattice, we can use any other network substrate embedded in Euclidean space. For example, this can be a so-called random geometric graph.[7] Furthermore, instead of the somewhat artificial division of a network into a substrate and shortcuts in this construction, we can simply assume that the nodes of a given network are located in Euclidean space. The Euclidean coordinates of the nodes play the role of hidden variables allowing greedy routing. Even more generally, we can place the nodes of a network in any space where the notion of distance between points is defined [160,35], and so where it is possible to introduce the coordinates of the nodes. In mathematics these spaces are called *metric spaces*. A standard example is Euclidean space. Another example is provided by a, say, binary tree, whose leaves play the role of the points of this discrete metric space, see Fig. 11.6. The distance between two points is defined as the degree of separation of these points from their closest common ancestor in the tree. In social networks, these hidden 'tree metric spaces' naturally represent hierarchies of successively embedded communities to which an individual belongs [180].

[7] Choose N points uniformly at random in a bounded region in D-dimensional Euclidean space and connect each two points at a distance less than some fixed number. The resulting network—*a random geometric graph*—has a rapidly decreasing degree distribution and D-dimensional geometry [67].

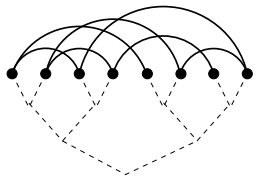

Fig. 11.6 This network (links are shown by arcs) is embedded in a binary tree metric space (dashed lines). In this discrete space, the distance between, for example, two left-most nodes equals 1, while the distance between the right-most and left-most nodes equals 3.

Marián Boguñá, Dmitri Krioukov, and kc claffy—scientists from Barcelona and San Diego—conjectured the presence of hidden metric spaces behind many real-world networks [35]. These underlining spaces are, in fact, fundamentally necessary. Individual nodes in such networks as cellular and many others, in principle, cannot have any global view of a network, and so without hidden metric spaces, nodes could not route signals, messages, and so on to intended targets. In a broad sense, it is the existence of these invisible metric substrates that makes a network navigable. In some networks, these underlining spaces are easily identified, for example, in airport networks. For other networks, these spaces are still unknown. Researchers hope to extract them based on the degree of similarity of nodes in networks, using their various intrinsic characteristics. Up to now this task has not been fulfilled.

The goal is particularly ambitious for Internet routing. Currently, routing protocols are actually based on a full knowledge of the Internet connections. Databases of individual routers—*routing tables*—store the routes to a large number of particular destinations in the network. This information is used by routers for forwarding messages by, roughly speaking, shortest paths.[8] The problem is that in the exponentially growing Internet, the total size of huge routing tables needed for this 'optimal' routing also grows exponentially with time. The urgent challenge is to find a way to cardinally reduce the routing tables to avoid rapidly approaching 'information processing overhead'. Greedy routing protocols, if realized, could resolve this serious problem.

[8] Interestingly, a typical Internet route follows a standard pattern. In the first half of the route, a message is progressively forwarded to more and more highly connected routers. In the second half, the message moves to fewer and fewer connected routers.

Efficient greedy routing assumes (i) sufficiently short routes to destinations and, necessarily, (ii) a high percentage of successful routes. Re-

fining Kleinberg's definition, one can call networks with efficient greedy routing navigable [35]. Recall that Kleinberg's network is navigable at a single point, exponent $\alpha = D$, which makes its navigability rather particular. In contrast to this, hidden-metric-space-based models of scale-free networks turn out to be navigable in a remarkably wide range of parameters [34]. This strengthens the case for the navigability of major real-world networks.

11.5 Google PageRank

Over half a million Google servers (in 2009) in a few dozen locations around the world do terrific work. They (i) permanently crawl the WWW, fetching web pages, (ii) store many copies of its open content, (iii) index this colossal array of text data, and (iv) process an astronomical number of search queries, returning results in a ranked (that is most informative) order.[9] Here we discuss an important part of Google's technology, namely the ranking of results—*the Google PageRank* [43].

[9] See a more detailed discussion 'How Google works' on http://www.googleguide.com/google_works.html.

A search query is a set of words and numbers. The result of a query is a ranked list of the addresses of pages containing this combination. The PageRank algorithm estimates the relative importance of a web page based on its popularity in the WWW. In this way PageRank ranks all pages in the WWW. After computation, this global ranking is used for the ranking of entries in the list of results of each search query. How does the Google PageRank work? The key idea is that the popularity of a web page is proportional to the number of times a crazy web-surfer visits this page when randomly surfing the WWW. So the problem is essentially reduced to a random walk on a directed network. In one respect, this random walk differs from the one on an undirected network that we discussed earlier. In directed networks, there may be clusters connected to the remaining part of a network only by incoming connections. If a walker moves only following directed links he will be trapped finally in one of these clusters. To avoid this, we have to allow our imaginary surfer to restart the process, say, from a random node. Let r_i be the final stationary probability of finding the surfer on node i at infinite time. Then the page with the highest r receives the top rank. It is easy to show that these probabilities satisfy the following set of equations:

$$r_i = \frac{p}{N} + (1-p) \sum_{j:j\to i} \frac{r_j}{q_{\mathrm{out},j}} \,. \tag{11.5}$$

Here N is the size of a network, $q_{\mathrm{out},j}$ is the out-degree of node j, the sum is over all incoming connections of node i, and finally p is the probability that, instead of moving to one of its nearest neighbours, the walker jumps to a node chosen uniformly at random. The probability $p \neq 0$, which is the only parameter of this algorithm, was claimed to be about 0.15. It is clear why only out-degrees are present in the sum on the right-hand side of this equation. Indeed, a walker escapes from a node along each of the outgoing links with equal probability.

Equations (11.5) are solved numerically by iteration. This is a simple task even for the huge WWW since these iterations rapidly converge.[10] The PageRank of a page depends on the entire structure of the network, but, clearly, the PageRank depends strongly on the number of connections of a node. Typically, the higher the number of incoming hyperlinks of a page, the higher r_i. For uncorrelated networks, it is very easy to obtain from eqn (11.5) the average value $\overline{r}$ for a node with a given number of links [93]:

$$\overline{r}(q_{\rm in}, q_{\rm out}) = \frac{p}{N} + \frac{1-p}{N} \frac{q_{\rm in}}{\langle q_{\rm in} \rangle} \,. \tag{11.6}$$

Note that this average value is entirely determined by the in-degree of a node. This simple linear dependence describes the measured average PageRanks in the WWW surprisingly well, despite strong correlations in this network. In the range of high in-degrees, fluctuations of r_i from page to page with the same in-degree turn out to be relatively small. So in this range, the average value $\overline{r}(q_{\rm in})$ approximates r_i extremely well. Equation (11.5) shows that to be highly ranked, a page must have incoming hyperlinks from pages with high PageRanks. Manipulating the connectivity of the nearest neighbours of a page, in principle, allows one to improve its PageRank. All the tricks, however, become virtually useless for pages with really high PageRanks.

[10] The solution of eqn (11.5) is actually the eigenvector of some matrix corresponding to its maximum eigenvalue. Finding these eigenvectors is among the simplest of numerical problems for matrices.