TP24: Uncovering networks

Steffen Schaper
Supervisors: Dr N Jones, Dr E López, Dr M A Porter
Word count: 8708

Abstract

I investigate numerically the performance of random and guided strategies to discover the topology of a network. I find that simple guided methods can exploit correlations in complex networks to increase the number of discovered edges above the random methods. I then study estimates of the global and local network structure. Here I find that the more sophisticated techniques introduce serious biases which severely affect estimates of the local network topology. I show that these incorrect estimates depend even qualitatively on the structure of the network being discovered. I also briefly study the impact of erroneously found edges on network discovery. I argue why the large scale topology is generally easier to estimate than the local structure. Finally I give recommendations for practical applications of the discovery techniques proposed here.

1 Introduction

The interdisciplinary field of network science has at least partially been driven by the increasing availability of computational power that has made it possible to work with large networks created from real-world data. Prominent examples of such networks are the world-wide web (WWW - the network of websites joined by hyperlinks) or the internet (which is the network of physical connections between computers, usually studied at the level of autonomous systems), protein interaction networks (PINs) or social networks like friendship networks [1, 2, 3].

In many cases, these empirical networks are not complete; for example, the WWW consists of billions of websites [4] and grows and changes continually. For PINs the detection of interacting protein pairs is expensive and time-consuming. Therefore an important question to ask is whether the networks constructed from incomplete data describe the properties of the underlying real systems reliably. Surprisingly, although this question has been raised often (for a recent example, see the concluding section of [5]), it has been addressed infrequently.

Apart from the question about the reliability of incomplete data and potential biases, there is another aspect of practical importance: Is there an 'optimal' strategy to adopt when we want to uncover a network?

The choice of methods that can be applied to discover a network depends on the context. For example, the topology of the internet is usually mapped by sending signals from a source to a specified target and following the path of the signal. This approach has been studied thoroughly by Dall'Asta et al. [6]. For the WWW, the standard technique is to 'crawl' the web by following hyperlinks from page to page. A similar method called 'snowball' sampling has been studied in [7]. Another rather well-studied setting is discovery by random walks [8]. All these approaches have in common that the edges in the network are easily accessible. Here the focus is on the opposite setting where the edges have to be found individually. This setting is commonly encountered in PINs; a recent paper by Schwartz et al. [9] has addressed the process of discovery on such networks. They have found that it is possible to improve the finding of interactions through prediction; a similar result

in a more general context has been obtained by Clauset et al [10].

I extend these findings by considering simple discovery strategies and by measuring the topology of the so obtained networks. To achieve this, I study numerically the behaviour of five methods on various example networks. In section 2.1 I provide a short background on networks in general and on the networks studied here. Then I describe our discovery strategies in section 2.2. We shall start our study in section 3 with the basic question of how many nodes and edges each methods discovers. We find that the guided methods are better than random techniques in finding edges, but are slower to cover all nodes of a network. After that, we direct our attention to topological quantities first at the large scale of the entire network (section 4) and then at the local level of individual nodes (section 5). In general we find that whilst global structure is captured well by all our methods, local properties are often estimated incorrectly. Finally in section 6, we briefly address the problems that arise through errors in the discovery process. In agreement with [11] we find that the impact of erroneously adding edges to the network is more severe than the omission of edges. We end by summarizing our key results and giving some directions for extensions of our work.

2 Network models and discovery methods

2.1 Networks

A network consists of *nodes* (sometimes also called vertices in the literature) and *edges* (or links) connecting them. In general, there could be different types of nodes in a network (or graph) - a famous example in the literature is the network of actors with two types of nodes, actors and movies, and edges between a movie and the actors featuring in it. Edges could be directed or weighted, for example in a road network weights could be distance and there could be one-way streets. Here we shall rule out all such complications and consider only undirected unweighted networks with only one type of node.

We can represent a network in terms of its adjacency matrix M [1, 2, 3]. For a network of N nodes, M is an $N \times N$ matrix. If the nodes i and j are connected, we have $M_{ij} = 1$, and $M_{ij} = 0$ otherwise. For undirected networks, M is symmetric. We only consider networks with no self-connections, that is $M_{ii} = 0$.

One of the most fundamental concepts of the study of network topology is the number of edges attached a node, which is called its *degree* and commonly denoted by the letter k. A widely studied summary statistic of networks is their degree distribution $p(k)$ which gives the probability that a node chosen uniformly at random has degree k.

Here we shall not discuss our results concerning the degree distribution of discovered networks; although this is certainly an important issue, we cannot make justified claims in the limited space available to present our results.

A significant contribution to the study of networks has been made by Erdös and Rényi [12]. They introduced the ensemble of random (ER) networks $\mathcal{G}_{N,P}$ with N nodes where each edge is present independently of any other with probability P. The notion of ensembles is the same as in statistical mechanics: Whereas it is hardly possible to make predictions about a single realization of an ER network, we can infer the properties of an average over many networks[1]. As we shall see, the uncorrelated nature of the ER networks makes them special for discovery; this makes them useful to contrast against the other networks we study.

Most networks found in real systems have degree distributions quite different from the ER networks. A much-studied ensemble which extends the idea of the random graph to arbitrary degree distributions is known as the configuration model (CM) [13]. The ensemble is formed by all networks for a given degree distribution, appearing with equal probability. The realization of configuration networks is non-trivial [14, 15, 16]. I give a brief description of the technique I employed in Appendix A.1.

The above ensembles deal with networks as

[1] Technically, the analogy also requires a counterpart of the Boltzmann factor in statistical mechanics. In the case of these random graphs, the networks in the ensemble are weighted by $P^E(1-P)^{(K-E)}$ where E is the number of edges in the network and $K = N(N-1)/2$ is the total number of possible edges [1]

Network	N	E	N_1	N_c	$\langle d \rangle$	r	C
Caltech	769	16656	762	4	2.33	-0.066	0.409
Reed	962	18812	962	1	2.46	0.023	0.318
Haverford	1446	59589	1446	1	2.23	0.068	0.323
ER	769	16642(50)	769(0)	1(0)	2.02(1)	-0.005(6)	0.0564(4)
CM	769	16656(0)	769(0)	1(0)	2.24(1)	-0.065(8)	0.161(4)
BA	769	16778(40)	769(0)	1(0)	2.06(1)	-0.035(8)	0.099(2)
Power grid	4941	6594	4941	1	18.99	0.004	0.080

Table 1: Topological properties of the networks studied. In the case of the simulated networks, averages over 10 realizations are given and uncertainties are one standard deviation. The numbers of nodes and edges are N and E; N_1 is the size of the largest component and N_c is the total number of components (section 4.1); $\langle d \rangle$ and d_{max} are the average and maximum geodesic lengths (section 4.2); r and C are the assortativity (section 5.1) and clustering coefficient (section 5.2).

static objects and give no justification as to how a network came to have its structure. The ensemble proposed by Barabási and Albert (BA) [17] is a widely studied attempt to answer this question. They model the growth of network by preferential attachment: Starting from a small network, nodes are added one at a time. Each node comes with m edges. The probability that it connects to a node A in the network is proportional to its degree k_A. Results for the BA networks were generally similar to the other complex networks; they are therefore not presented in the main body of the text and can be found in Appendix B.1.

Simulated ensembles do not generally capture all the varieties of network structures that exist in real systems. We therefore also work with empirically constructed networks. Three of them are taken from the online social networking site *Facebook* [18]: They were constructed by completely sampling the 'friendships' of students from the US universities Caltech, Reed and Haverford. In order to show more clearly the differences between these real networks and the simulated ones, the parameters for the simulations were chosen in such a way that the number of nodes N and edges E is similar to the Caltech network. Finally we include a network from a completely different field, the Western Power Grid [19].

The properties of all these network are summarised in Table 1. Fig. 1 shows the cumulative degree distributions of the networks under study. The cumulative distribution is given

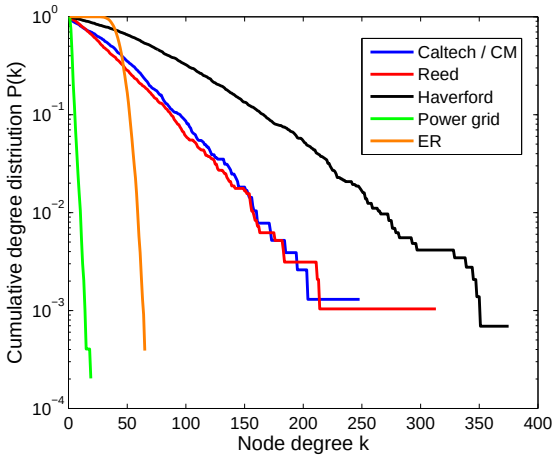

Figure 1: The cumulative degree distributions for the networks considered here on semi-logarithmic axes. Notice that only for the ER networks there are essentially no nodes with very small degrees.

by $P(k) = \sum_{\kappa=k}^{\infty} p(\kappa)$ and gives the probability that a randomly selected node has degree k or larger. Notice that only ER networks have essentially no nodes with degrees below 20 whereas on all the other networks, there is a significant fraction of nodes with very small degrees.

2.2 Discovery methods

Imagine a searcher who is given a list of the nodes in a network and the task to find the edges among them. We will assume that the searcher remembers which node pairs he has already in-

vestigated and does not repeat a test on the same pair. In some applications he could employ specialized tests; for example, for the discovery of interacting proteins, the technique of pooling can be employed where one protein is first tested against a large number of others and only if any interaction is observed, one-to-one tests are performed [9]. Such approaches are not necessarily possible in all applications, so we exclude them here.

A particularly important parameter is the network coverage fraction t. We define this to be the fraction of node pairs that the searcher has tested relative to all $N(N-1)/2$ pairs (we focus on undirected networks and hence we take all discovered edges as undirected). We can view t as a time coordinate; the longer the discovery goes on, the larger the discovered part of the network. Alternatively we could imagine that the searcher has limited resources and can only partially discover the network. In that sense, we associate a cost with each test of a pair of nodes. I have taken data starting at $t = 0.1$ up to $t = 0.95$ in intervals of 0.05.

Mathematically, discovery strategies can be described as follows. To reach the given coverage t, the searcher performs $tN(N-1)/2$ *steps*. At the beginning of a step, the searcher is standing on a node A. He then chooses another node B according to some strategy S. We can view S as a probability distribution so that the probability of selecting B is

$$P(B|A) = S(A, B, G) \tag{1}$$

Here G is the adjacency matrix of the partially discovered network. After selecting B, the searcher determines if A and B are connected by some experiment. Finally he moves on to another node C which can be any node in the searcher's list, including A and B.

We can directly translate the requirements of forbidding self-checks and double-checks into this framework. We require

$$S(A, A, G) = 0 \text{ for all } A, G \tag{2}$$

$$S(A, B, G) = S(B, A, G) = 0 \text{ if } A \text{ and } B \text{ have been tested} \tag{3}$$

Equipped with this formalism, we can define the strategies which we will investigate in the following sections. The above requirements are understood to take precedence over the different strategies we define below.

The simplest approach is the *uniform* method. Here the searcher selects a pair of nodes uniformly at random from the entire network for each step. We can define the strategy in terms of n_A, the number of nodes which have not been tested against A as

$$S_u(A, B, G) = \frac{1}{n_A} \tag{4}$$

The next node C is also chosen uniformly at random from the entire network.

It is natural to ask if we can 'do better' than this. To answer this question, I introduce three methods in which the selection of node pairs is not completely random. Therefore we shall refer to these as the 'guided' strategies.

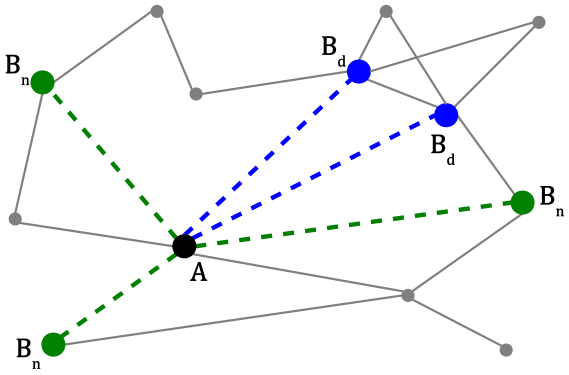

Figure 2: Illustration of the node selection for the degree and neighbour methods. The searcher is standing on the black node A. The candidate nodes (marked B_n) for the neighbour method are in green; for the degree method, the most likely nodes B_d are coloured blue.

The *degree* method selects B with a preference to high degrees:

$$S_d(A, B, G) \propto k_B \tag{5}$$

where k_B is the degree of B, calculated by summing all the elements of G in the row corresponding to the node B. This selection strategy is illustrated in Fig. 2. If the searcher finds an edge, he moves on to B; otherwise he remains at A. I have also included a small probability $p = 0.01$

that both the selection of B and the transition to C is the same as in the uniform method. This has the effect of preventing the searcher from being 'trapped' on a node in some pathological cases; also it ensures that nodes with degree zero can be selected.

The *neighbour* strategy chooses B from the nodes that have an edge to one of the neighbours of A (for an illustration, see Fig. 2). If A has m such nodes from which to choose B, the strategy is given by

$$S_n(A,B,G) = \begin{cases} m^{-1} \text{ if } B \text{ is a } 2^{\text{nd}} \text{ neighb. of } A \\ 0 \text{ otherwise} \end{cases} \quad (6)$$

So the searcher chooses B uniformly at random among the second neighbours of A. If $m = 0$, and generally with probability $p = 0.01$, B is selected according to the uniform strategy. If the searcher finds an edge, he moves on to B; otherwise he makes a uniform move with probability p or remains at A, just as in the degree method.

There are plenty of other strategies to study. Here I shall only present one important idea, namely to combine the two above methods into the *hybrid* strategy. Here the searcher alternates between the degree and neighbour methods. For simplicity, I only consider the case where the number of steps per method is fixed from the outset. Of course, there are many other possibilities where the more 'successful' strategy is given more steps; I have obtained some preliminary results for such adaptive approaches but since they are quite similar to the simple splitting, I shall not discuss them any further. To be precise, each strategy is applied 50 times in total (that is, up to $t = 0.95$), and the number of steps per application is constant.

It is clear that the guided methods cannot work right from the start. If no edges have been found, there are no nodes with non-zero degrees and also no second neighbours. To circumvent this problem, I start these methods by performing a uniform discovery with coverage $t = 2.5\%$.

In all the above methods, the searcher works with all the nodes in his list; if he only works up to a limited coverage, he has to accept that he cannot test the connection between all the node pairs. I have also devised a complementary approach which is called the *subnet* method. Here the searcher concentrates on a smaller number of nodes

$$N_s = \frac{1}{2} + \sqrt{\frac{1}{4} + tN(N-1)} \approx N\sqrt{t} \quad (7)$$

which are chosen uniformly at random from the entire network. Then he tests all node pairs in that list for edges. Eqn. (7) is derived in Appendix A.2.

An important difficulty in dealing with partially discovered networks is to handle the undiscovered part. In contrast to the actual network, each pair of nodes can exist in three different configurations: connected, unconnected, untested. The last term indicates simply that the searcher has not yet investigated the respective pair of nodes. For all the results shown here, the simplest method in dealing with the undiscovered network has been adopted: All untested pairs of nodes are treated as unconnected.

I have implemented and extensively tested all the code that has been used to obtain the data that is presented in the following sections. The most severe limitations were those of memory and computational complexity and particularly the trade-off between memory and processor usage. Most importantly, networks were represented in terms of their adjacency matrices. This means that memory usage increases quadratically with N; whilst the Caltech-sized networks with $N = 769$ occupy about 600kB of memory, the power grid network has $N = 4941$ nodes and requires 24MB of storage. Since all the networks to which the discovery strategies were applied are quite sparse (meaning that the average degree $\langle k \rangle = 2E/N \ll N$), the memory usage can be reduced by working with sparse matrices which only store the non-zero entries. The complication here is that for the discovered network, the information to keep in the adjacency matrix is more than binary: We have to discriminate not only between 'edge found' and 'no edge found' as usual but also between 'not yet tested' (see above). As we go up to 95% coverage, working with full matrices is necessary.

The problem of memory usage has limited the size of networks to work with. Equally important is the time taken to perform the actual computations. Since the number of node pairs to check is $N(N-1)/2$, the duration also increases quadratically with N. In fact, the guided strate-

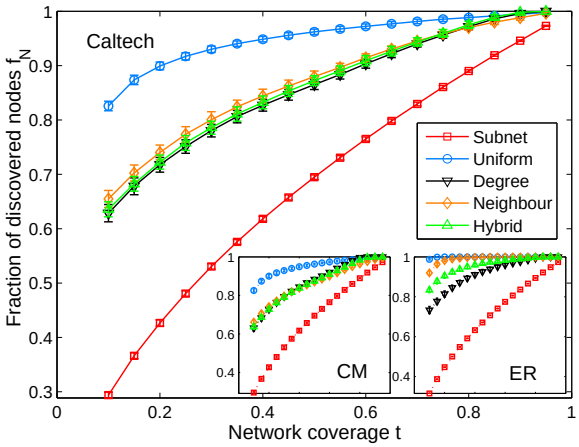

Figure 3: The fraction of non-isolated nodes f_N against network coverage t for the Caltech, CM and ER networks. Error bars are one standard deviation.

gies are even more time consuming because the time taken to select a node for testing also increases with the network size. For all the simulations, I ran each strategy 50 times on each network; for the simulated network, these runs were split over 10 realizations. As an order of magnitude, the duration for running each strategy once (up to $t = 0.95$) on the Caltech network is about 10 minutes on a standard desktop computer.

3 Node and edge discovery

We begin our comparison of the discovery methods described above by asking how many nodes and edges they find for a given network coverage. Whereas counting edges is straightforward, a node is only counted as 'discovered' if it has at least one edge attached to it. This procedure of removing isolated nodes from the network is common in the literature [1]; this is justified because such nodes have no impact whatsoever on the structure of the network.

3.1 Finding nodes

From Fig. 3 we see that the uniform method is the fastest to find at least one edge for each node; unsurprisingly, the subnet strategy is much slower. This of course is an artifact of its construction. More interesting is the fact that the guided methods perform nearly equally on the complex networks. It is clear that they should be slower than the uniform method because they only look for edges between nodes that have already at least degree 1. On the random graphs, the neighbour method is more successful.

Generally, we observe that finding the first edge for each node is easier on the random graphs. This is a direct consequence of the fact that in the ER ensemble the number of nodes with very small degree is much smaller than on the other networks (cf. Fig. 1).

3.2 Edge discovery

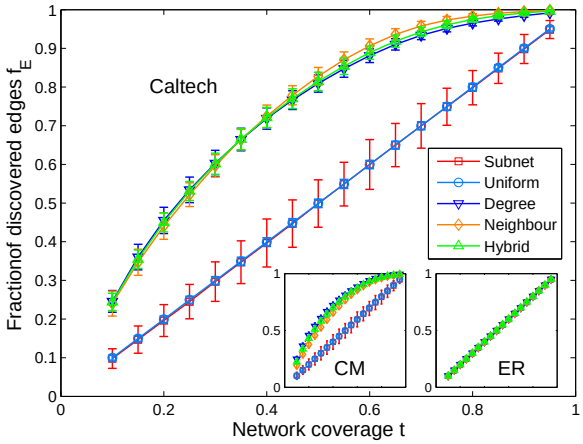

Figure 4: The fraction of discovered edges f_E against network coverage t for the Caltech, CM and ER networks. Error bars are one standard deviation.

When it comes to discovering edges, we see that the guided methods clearly beat the random techniques (see Fig. 4). However, this does not apply to the ER networks where all methods perform equally. Combining these two results, we are lead to the important conclusion that the guided methods can exploit correlations (if existing) to find more edges than a uniform search. Taking into account also the results for the Reed and Haverford networks (Figures B.4 and B.5 in Appendix B.2), the neighbour method seems to perform slightly better than the degree method.

From our study of the power grid (see Fig.

B.10 in Appendix B.3) we learn that there is a caveat to these results: On that network, the mean degree $\langle k \rangle = \frac{2E}{N} \approx 3$ is an order of magnitude smaller than for the above networks (which have $\langle k \rangle \approx 43$). This extreme sparseness complicates the task for the guided methods: Many nodes have only very small degrees and hence it is hard to make use of any correlations. Indeed we find that the degree method is the worst method to find edges; the neighbour method is slightly better than the random strategies.

We note at this stage a common feature that we will find also in the following sections: Except for the ER networks, the error bars for the subnet method are generally larger than those of the uniform strategy. This is a consequence of inhomogeneity of the underlying networks. Concerning the number of found edges, this is expressed in the degree distribution (cf. Fig. 1): There are quite a few nodes with very high degrees. If they are included in the subnetwork, the number of discovered edges will be much larger than if they are missing. So the variance on the number of edges is rather high. In contrast we see that the uniform strategy is not sensitive to the underlying degree distribution.

We now address some questions that follow from this result: Naively, we should expect that finding more edges improves our estimate of the network topology (in particular as these edges are spread among a smaller number of nodes). We investigate if this intuition is correct by studying the structure of the discovered networks first at large scales and then at the level of individual nodes and edges.

4 Large scale structure

Here we focus on the global topology of the networks discovered by our strategies. To this end, we first study the *components* of the networks. A component is formed by all the nodes that can be reached from each other only by following edges [1]. Then we look at the *geodesic* distances (shortest path lengths) between nodes in the largest connected component, measured by the number of edges to cross in order to get from one node to the other. Both of these properties are generally important for the function of a network, in particular for transport processes.

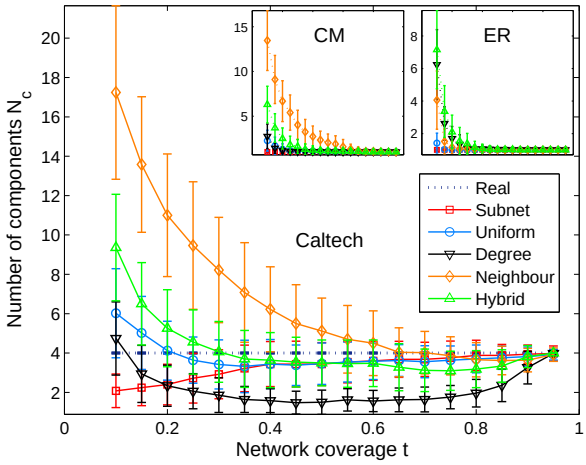

Figure 5: The number of components N_c against network coverage t for the Caltech, CM and ER networks. Error bars are one standard deviation.

4.1 Component structure

In general we can deduce from Fig. 5 that the guided methods take longer to establish just one large component. The fact that the neighbour method is slowest can be explained from its design: Since the searcher tests for edges between nodes with a common neighbour, they are already parts of the same component. Thus edges joining two components are only found when the searcher performs a uniform test.

The presence of 3 tiny components (with 3, 2 and 2 nodes) makes the interpretation of the Caltech graph more difficult: However we can conclude that it is generally hard to find small components. By analysing the fraction of nodes in the largest component (see Fig. B.11 in Appendix B.4), we can establish that the degree method has particular difficulties with the small patches.

4.2 Geodesic lengths

When analysing path lengths, we need to bear in mind that they generally depend on component sizes. This at least partially accounts for the large average path length found by the uniform method, as seen in Fig. 6. We also see that the networks discovered by the neighbour strategy in general tend to have longer path lengths than those coming from the degree and hybrid

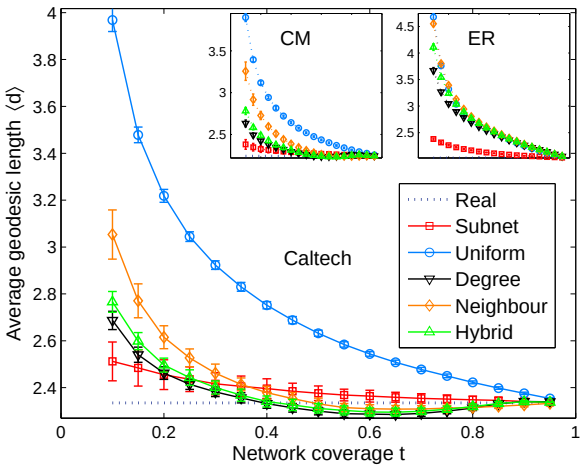

Figure 6: The average geodesic length $\langle d \rangle$ against network coverage t for the Caltech, CM and ER networks. Error bars are one standard deviation.

method. A possible explanation is illustrated by Fig. 2: An edge found by the neighbour method can only reduce a path length by one; the degree method instead is able to find more significant shortcuts.

In summary, we see that the global topology is captured rather well by all the methods we consider. In all cases, the hybrid method mediates between the degree and neighbour strategies. We now turn to the questions concerning the local structure of the discovered networks.

5 Microscopic correlations

Above we have shown that we can exploit correlations in complex networks in the search for edges. We now ask a reverse question: Given we have found disproportionally many edges, how does this affect our estimates of local correlations? To answer this, we consider two different measures of correlations: First we look at the degrees of a node's neighbours, and after that we turn to the edges connecting these neighbours. Both of these questions are 'of higher order' than the mere existence of edges but are still local in the sense that they only consider the direct neighbourhood of each node.

5.1 Assortativity

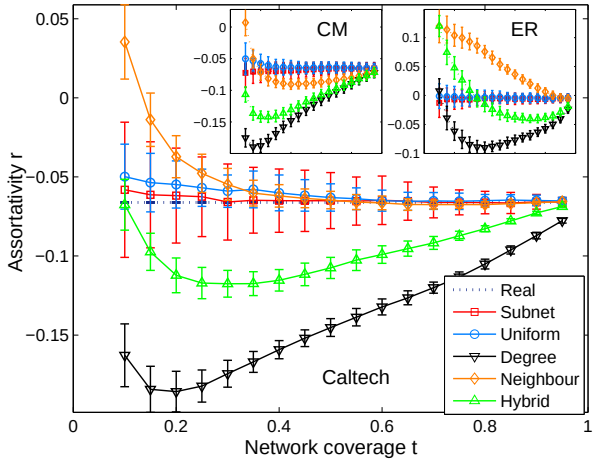

Figure 7: The assortativity r against network coverage t for the Caltech, CM and ER networks. Error bars are one standard deviation.

The assortativity r has been proposed as a measure of degree-degree correlations [20]. It is defined as an average over all edges:

$$r = \frac{E^{-1}\sum_n j_n k_n - [E^{-1}\sum_n (j_n + k_n)/2]^2}{E^{-1}\sum_n (j_n^2 + k_n^2)/2 - [E^{-1}\sum_n (j_n + k_n)/2]^2} \tag{8}$$

where E is the number of edges in the network, the sums go over all edges and j_n, k_n are the degrees of the nodes at the ends of the n^{th} edge. We note that when correlations are positive (that is, high-degree nodes prefer to be connected to each other) then $r > 0$ and the network is said to be assortatively mixed. In the opposite case when high-degree nodes are mostly connected to low-degree nodes, a network has $r < 0$ and displays disassortative mixing. r is normalized such that $|r| \leq 1$.

We can see from Fig. 7 that the guided methods generally do not provide reliable estimates of the assortativity. The degree method generally finds disassortative mixing; this is even the case for networks which are actually assortative such as the Reed and Haverford networks (see Figs. B.6 and B.7 Appendix B). The performance of the neighbour method is less consistent across the different networks, and the hybrid strategy generally mediates between the two. This makes it very hard (if not impossible) to extrapolate the

assortativity of the underlying network from the discovered ones. In contrast, both the subnet and the uniform method get the correct value of the assortativity for essentially any coverage, and all networks considered here.

5.2 Transitivity

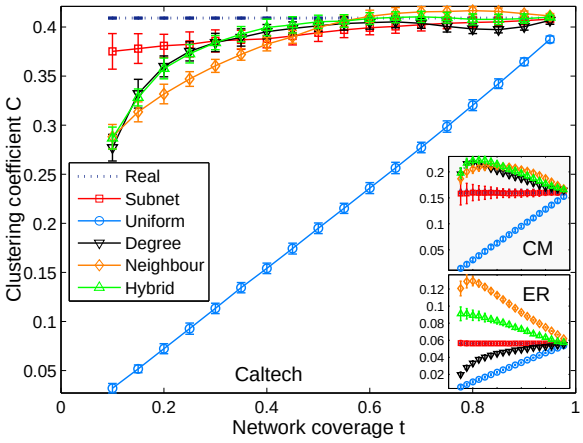

Figure 8: The average clustering coefficient C against network coverage t for the Caltech, CM and ER networks. Error bars are one standard deviation.

Transitivity refers to the phenomenon that nodes with a common neighbour tend themselves to be connected. This feature is particularly prominent in social networks (cf. Table 1): Often people with a common friend are also friends of each other. This effect can be quantified by the clustering coefficient [19] which for the node A is defined to be

$$C_A = \frac{\text{no. of edges between } A\text{'s neighbours}}{k_A(k_A-1)/2} \tag{9}$$

where k_A is the degree of the node and the denominator effectively is the maximum number of edges that could exist between the neighbours. If $k_A < 2$, C_A is conventionally set to zero. In simple terms, the clustering coefficient measures the probability that there is an edge between a pair of neighbours of the node A.

In Fig. 8 we show the clustering coefficient averaged over all non-isolated nodes in the network. The general picture is quite similar to the assortativity: The guided strategies discover subnetworks that have quite a different local structure than the real network. Except for the Caltech network, the subnet method provides very good estimates for every coverage. Its failings on the Caltech (and also Reed and Haverford, see Figs. B.8 and B.9 in Appendix B.2) network tell us important information: It appears that the triangles of mutually connected nodes are not spread evenly throughout the network. This can be understood as a special kind of inhomogeneity. In contrast, the distribution of triangles in the simulated networks seems to be homogeneous: There the subnet strategy is produces much better estimates of the clustering coefficient. We can view this distinction between the real and simulated networks as a shortcoming of the network models.

The most remarkable feature of Fig. 8 however is the result of the uniform method. On all networks the curves appear very nearly linear, although on closer inspection the linearity turns out not to be exact[2]. A similar finding has been reported in [7]. We can understand this behaviour by a simple rough argument: The probability that a given edge has been discovered by the uniform method is equal to the coverage fraction t. The probability of finding a closed triangle therefore scales with t^3 and the two edges required to establish a common neighbour is proportional to t^2. The clustering coefficient measures the ratio of triangles to all common neighbours, and hence is linear in t. In Fig. B.12 in Appendix B we show that dividing out a factor of t provides a reasonable yet not quite correct estimate of the actual clustering coefficient, as we should expect.

It is important to note that there is no consistent trend in the guided strategies: On the CM networks, all methods overestimate the clustering coefficient. On the Caltech network, the degree method underestimates C while the neighbour strategy changes to an overestimate at about $t = 0.6$. As with the assortativity, this dependence of the qualitative behaviour on the underlying network makes it very difficult to ex-

[2] An exactly linear dependence can also be excluded by a simple theoretical argument: At $t = 0$, clearly $C = 0$. However it requires at least 3 tests before the first triangle can be established; hence at $t = 2/(N(N-1))$, we always have $C = 0$ and hence the exact relation cannot be $C \propto t$.

trapolate a result from a partially discovered network to the entire underlying graph.

In summary we have found that the estimates of the local network structure are estimated well only by the random methods. The guided methods introduce severe biases. What is more, we have seen that the behaviour is distinct for each method: Depending on the method, different edges are discovered. Before summarizing our results, we briefly discuss the influence of random errors in the determination of edges.

6 Errors in the discovery

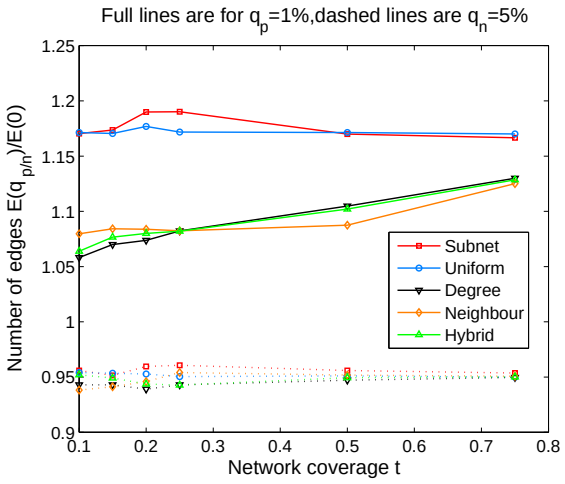

Figure 9: The number of found edges when random errors occur in the discovery, normalized by the number of edges found without errors. Solid lines are for $q_n = 0, q_p = 0.01$ and dotted lines show the case $q_p = 0, q_n = 0.05$. The underlying network is Caltech.

So far, we have assumed that the edges of a network can be discovered without any errors. In many circumstances, this assumption is definitely not valid and the consequences of erroneous discoveries must be addressed.

In principle, we have to differentiate between false-positive edges (i.e. those which are 'found' although not actually present) and false-negatives (i.e. erroneously omitted edges). We allow for such errors to occur randomly with probabilities q_p and q_n respectively. It is important to stress that these parameters are *not* the error probabilities for a single experiment (which can be as high as $q_p \approx 50\%$ in protein interaction measurements [9]). Rather these paramaters give the probabilities of making an erroneoues assertion after repeated experiments.

For the sparse networks under study here, most of the tests that the searcher performs are between nodes that are unconnected in the real network. Hence we should expect that the number of erroneous test outcomes is much larger if $q_p = q, q_n = 0$ than if $q_n = q, q_p = 0$. In Fig. 9 we confirm that this is indeed the case: With $q_n = 5\%$, we observe that only 95% of all edges are found, as expected. In contrast, with $q_p = 1\%$ the error rate is generally larger. The guided strategies are initially produce less false-positives. This is a direct consequence of the greater number of discovered edges at low t (cf. Fig. 4).

Results for the other topological measures are shown in Figs. B.13-B.16 in Appendix B.6: For $q_p = 0, q_n = 0.05$ we see that the estimates are relatively stable. However for $q_n = 0, q_p = 0.01$ we observe large deviations from the error-free values. We only show the data for the Caltech network; on the other networks, we have obtained similar results.

It would however be premature to claim that false-positives should be avoided at the cost of increasing the rate of false-negatives: We have to bear in mind that when the underlying network is very sparse (like the power grid), too many false-negatives can have dramatic consequences, in particular on the discovered component structure.

7 Conclusions and Outlook

We have studied the performance of different discovery strategies on a selection of both simulated and empirical networks. In particular, we compared the topologies of the discovered network on both global and local scales over the complete range of network coverage. Our most fundamental result is that correlations can be exploited (provided they exist) by simple guided methods in order to increase the number of found edges significantly above the results of a completely random strategy. Even though our guided strategies are quite different, their performance in terms of edge findings

Method	N	E	N_c	$\langle d \rangle$	r	C
Subnet	534(4)	8307(341)	3(2)	2.38(4)	$-0.065(18)$	0.394(9)
Uniform	**740**(4)	8314(61)	3(2)	2.63(2)	$-0.063(9)$	$0.390(12)^{\dagger}$
Degree	665(7)	13477(124)	2(1)	2.30(2)	$-0.145(6)$	0.401(6)
Neighbour	676(8)	**13789**(127)	5(2)	**2.33**(3)	$\mathbf{-0.065}(6)$	0.399(7)
Hybrid	671(7)	13566(132)	**4**(2)	2.31(3)	$-0.108(7)$	**0.405**(7)
Real network	769	16656	4	2.33	-0.066	0.409

Table 2: Overview of the discovery results of the Caltech network at coverage $t = 0.5$. Uncertainties are one standard deviation. The estimates that are closest to the actual values of the real Caltech network are printed in bold face. For the number of components N_c, the mean geodesic length $\langle d \rangle$, the assortativity r and the clustering coefficient C, results are in bold face if they are significantly different from the actual values. †: For the uniform method, the calculated clustering coefficient has been divided by t to give a better result.

are surprisingly similar, given the network to discover is not too sparse.

By the investigation of topological properties we have shown that the success of the guided methods comes at a cost: Due to their inherent bias towards a particular kind of edges, they discover a network that is not representative of the underlying network. This contrasts with the behaviour of the uniform strategy which in general gives results that are reliable across different networks and which appear connected to the network coverage in simple ways. For the guided methods, we have found that while the component structure and path lengths are captured rather well, the estimates of microscopic correlations (in terms of assortativity and clustering) are generally far from the correct values. Furthermore we have seen that even the qualitative behaviour of the guided strategies depends on the underlying network. This makes is very difficult if not impossible to extrapolate from a partially discovered network to the full underlying graph.

The relative stability of the large scale measures compared to the microscopic properties can be understood by the idea of 'redundant' edges: The component structure often does not rely on a single edge. This result is prominent in the study of resilience of networks against the removal of edges [1, 2]. Similarly, the path from one node to another can in many cases follow different routes, and the exclusion of a single edge will not have a large impact on the average path length. Locally however every edge is important: The assortativity is an average over the degrees of the nodes at the ends of each edge, and hence depends not only on the number of edges that are found, but also on the nodes to which they are attached. Clearly the clustering coefficient is also very sensitive to the existence of each individual edge: To establish a common neighbour requires two edges, a closed triangle needs three. As we have seen, the biased selection of edges has severe consequences for these local measures.

In drawing together all our results, we would like to answer the question with which we have set out: Can we recommend any single method for network discovery? To answer this question, we compare the findings of the different strategies. As an example, Table 2 shows the results on the Caltech network at coverage $t = 0.5$. Most importantly, we see that no single strategy is the best for all the properties considered here. So the first part of the answer to the above question is no: There is no 'best' strategy that produces correct estimates for every network property. However, we can say that if one is only interested in certain special quantities, some methods are better than others: If the total number of discovered edges is most important, it is advisable to adopt one of the guided strategies. In contrast, the uniform method generally outperforms the guided methods in estimating microscopic correlations such as assortativity and clustering.

We can sum up our results in the following advice: If the primary objective in the discovery

of a network is to find as many edges as possible for limited time or resources, then a guided strategy should be employed. However if one is interested in the detailed network structure, then the uniform method is generally preferable.

The theory of network discovery is still in its infancy, and the present study opens many further questions. Apart from conceiving other, more sophisticated discovery strategies it is certainly necessary to widen the range of applicability by testing our methods on further networks, both simulated and empirical.

A possible extension of our methods could include the introduction of a cost not only for testing a pair of nodes, but also for making transitions from one node to another. In many applications, this parameter is of interest: For PINs, moving to another node at least requires the experimenter to obtain another chemical; in the context of social networks, changing the node often means to interview another participant in a study. In that sense, moves in the network are penalized; due to time constraints, we could not address this aspect here.

From an applicational standpoint, it would be very useful to look for 'optimized' strategies. In view of our results, such methods will presumably be restricted to a subset network properties, and possibly also to a limited class of networks. In conjunction with such results, another question would need to be addressed: Given a partially discovered network, how can we assign it to a certain class? Without such results, optimal strategies for different networks would be of little use.

It is also worthwhile to investigate the sizes of error bars further. In most cases a network is only discovered once. So at the end of partial discovery, one is only left with a single result for each measurement, and an important task is to estimate the uncertainty of the obtained value, or equivalently to determine confidence intervals. This problem has been addressed for example by Salganik [21]; it would certainly interesting to compare his techniques to create confidence intervals to the error bars that we have found from our repeated discoveries.

Finally, a very important problem is to deal with the undiscovered part of the network at the end of a partial discovery. In this report, we have chosen the simplest solution which is to assume that no edges are present in the undiscovered part. This is a clear oversimplification and results in a number of false-negatives. We have obtained some preliminary results for another solution: We have placed further edges randomly between edges, with the probability for a node to receive more edges given by the ratio of its degree to the tests performed on that node. Yet, this simple approach has not proved very fruitful. Work on this problem would be particularly useful, and would complement the search for an optimal strategy. Ultimately we aim for an understanding of the network as a whole so the ability to extrapolate from partial information would be highly valuable.

Acknowledgements

I thank my supervisors Nick Jones, Eduardo López and Mason Porter for many interesting conversations and useful insights and comments.

I am grateful to Peter Mucha who provided me with access to the Inventor Cluster at the University of North Carolina at Chapel Hill. The cluster is supported by the National Science Foundation through award DMS-0645369 and by start-up funds provided by the Institute for Advanced Materials, Nanoscience and Technology and the Department of Mathematics at the UNC. All simulation results were obtained on the cluster.

The data on the Facebook networks was kindly provided by Adam d'Angelo and Facebook, and the power grid network was taken from Mark Newman's website. The original data was compiled and published by Duncan Watts. The basis for the code used to convert that network into MATLAB was provided by Thomas Richardson.

References

[1] M.E.J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.

[2] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74(1):47–97, 2002.

[3] S.N. Dorogovtsev and J.F.F. Mendes. *Evolution of Networks: From Biological Nets to the Internet and WWW*. Oxford University Press, 2002.

[4] www.worldwidewebsize.com.

[5] M.E.J. Newman. The physics of networks. *Physics Today*, 11:33–38, 2008.

[6] L. Dall'Asta, I. Alvarez-Hamelina, A. Barrata, A. Vazquez, and A. Vespignani. Exploring networks with traceroute-like probes: Theory and simulations. *Theo. Comp. Sci.*, 355:6–24, 2006.

[7] S.H. Lee, P.J. Kim, and H. Jeong. Statistical properties of sampled networks. *Phys. Rev. E*, 73:016102, 2006.

[8] L. da F. Costa and G. Travieso. Exploring complex networks through random walks. *Phys. Rev. E*, 75:016102, 2007.

[9] A.S. Schwartz, J. Yu, K.R. Gardenour, R.L. Finley Jr., and T. Ideker. Cost-effective strategies for completing the interactome. *Nature Methods*, 6(1):55–61, 2009.

[10] A. Clauset, C. Moore, and M.E.J. Newman. Hierarchical structure and the prediction of missing links in networks. *Nature*, 453(7191):98–101, 2008.

[11] P. R. Villas Boas, F. A. Rodrigues, G. Travieso, and L. da F. Costa. Sensitivity of complex networks measurements. arXiv:0804.1104, 2008.

[12] P. Erdös and A. Rényi. On random graphs. *Pub. Math.*, 6:290–297, 1959.

[13] M. Molloy and B. Reed. A critical point for random graphs with given degree sequence. *Random Struct. Algorithm*, 6:161, 1995.

[14] M. Catanzaro, M. Boguña, and R. Pastor-Satorras. Generation of uncorrelated random scale-free networks. *Phys. Rev. E*, 71:027103, 2005.

[15] R. Milo, N. Kashtan, S. Itzkovitz, M.E.J. Newman, and U. Alon. On the uniform generation of random graphs with prescribed degree sequences. arXiv:cond-mat/0312028v2, 2004.

[16] F. Viger and M. Latapy. *Efficient and simple generation of random simple connected graphs with prescribed degree sequence,* Lectures notes on computer science, pages 440–449. Springer, 2005.

[17] A.-L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 1999.

[18] A.L. Traud, E.D. Kelsic, P.J. Mucha, and M.A. Porter. Community structure in online collegiate networks. arXiv:0809.0690, 2008.

[19] D. J. Watts and S. H. Strogaty. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

[20] M.E.J. Newman. Assortative mixing in networks. *Phys. Rev. Lett.*, 89(20):208701, 2002.

[21] M.J. Salganik. Variance estimation, design effects, and sample size calculations for respondent-driven sampling. *J. Urb. Health*, 83(7):i98–i112, 2006.

Appendices

A Some details on implementation

In general, I have tried to keep all code as modular and reusable as possible while at the same time keeping the computational complexity low. For example, choosing nodes with selection probability proportional to their degree is realized in the way proposed by Newman [1]: Instead of computing the list of degrees each time a node is selected, I maintain a list in which the index n of a node appears k_n times (k_n is the degree of the node n as usual). Then the probability that an element drawn uniformly at random from the list is n is proportional to k_n.

A.1 Realizing arbitrary degree distributions

In order to construct an uncorrelated network with an arbitrary degree distribution, I employed the algorithm described in [14, 15, 16].

I first build deterministically a network that has the desired degree distribution. This is achieved by first assigning the degree to each node according to the given distribution. This can be pictured as having k_n 'half-edges' sticking out of the node n. Then I take the node with the lowest number of free half-edges and connect it to the node with the highest number of free half-edges that is not connected to the first node. Repeating this procedure, I create a network without multiple edges between any pair of nodes and with the desired degree distribution. This of course will only work if the total number of half-edges is even. If this is not the case, the degree distribution cannot be realized.

In the ensemble of networks with arbitrary degree distribution, all possible realizations occur with equal probability. The above procedure does not reproduce this ensemble; the only reason to follow it is that it works without any potential deadlocks (except in a few pathological cases which are not necessary to consider here). In the second stage, I shuffle the edges in the network in the following way: Two nodes A and C are drawn uniformly at random from the network, and for each of them one of their neighbours is selected (again, uniformly at random). Let the selected neighbour of A be the node B, and that of C be D. Then the edges are rewired so that A and C, and B and D, are connected (see Fig. A.1), unless this introduces a self-edge connecting a node to itself, or a multi-edge, that is an edge between nodes that are already neighbours. This type of rewiring leaves the degree distribution untouched while at the same time randomizing the nodes at the ends of the edges. After repeating these swaps sufficiently often (I perform $50E$ such steps, where E is the total number of edges in the network), the network is essentially completely randomized.

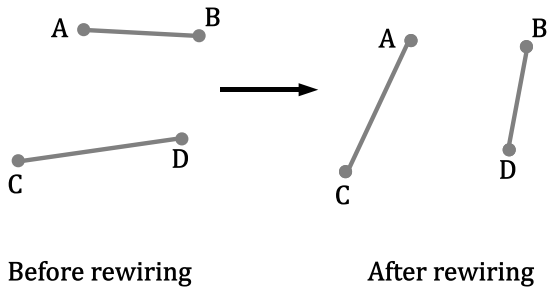

Figure A.1: Illustration of the rewiring process that randomizes the configuration model networks.

A.2 The number of nodes to check in the subnet method

In order to take snapshots of the network at any given coverage fraction t, I have implemented the discovery strategies in such a way that they take as input parameters (among others) a partially discovered network and the number of steps to perform before the next snapshot. This is straightforward for all methods except the subnet method. Here the number of steps cannot be any integer: Instead, it must be such that it is possible to test a number of completely new nodes among each other, but also with the nodes that have been included in the subnet previously.

Let T be the number of steps to perform be-

fore the next snapshot, and let N_o be the number of nodes that are already included in the subnet. Then the number of nodes to add to the subnet N_a is determined from

$$T = N_a(N_a - 1)/2 + N_a \cdot N_o \qquad \text{(A.1)}$$

The first term on the RHS of Eq. (A.1) represents the tests between the added nodes, and the second term accounts for the tests between added and old nodes. Solving the quadratic for N_a is straightforward:

$$\begin{aligned} 0 &= N_a^2 + 2N_a(N_o - \frac{1}{2}) - 2T \\ \Rightarrow N_a &= \frac{1}{2} - N_o + \sqrt{\left(N_o - \frac{1}{2}\right)^2 + 2T} \end{aligned} \qquad \text{(A.2)}$$

Clearly we get an integer solution for N_a only for certain values of T. In general, these values do not coincide with the coverage steps of 0.05 that I used for the data collection. I have dealt with this problem in the following way: Given the number of steps to perform between two snapshots, I add as many nodes to the subnet as possible without using more steps than allowed. The remaining steps are carried over and added to the steps for the next snapshot.

We can also use Eq. (A.2) to derive Eq. (7): The number of steps to perform is $T = tN(N-1)/2$, and the number of nodes initially in the subnet is $N_o = 0$. Thus we obtain

$$N_a = \frac{1}{2} + \sqrt{\frac{1}{4} + tN(N-1)} \qquad \text{(A.3)}$$

as before.

B Supplementary results

B.1 The BA networks

The results for the discovery of the BA networks are quite similar to the Caltech and CM networks. In Fig. B.1 we see that the guided methods are again better at finding edges. However it appears that here the degree methods is slightly better than the other two. Fig. B.2 shows that the guided strategies tend to find a disassortative mixing which is not present in the actual networks. Finally we observe from Fig. B.3 that the clustering coefficient is overestimated by the guided strategies. As in the other cases, the subnet and uniform methods produce useful estimates of the assortativity and clustering coefficient.

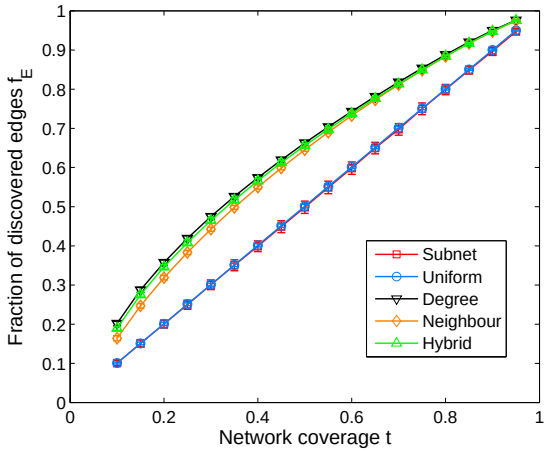

Figure B.1: The fraction of discovered edges f_E against network coverage t for the BA network. Error bars are one standard deviation.

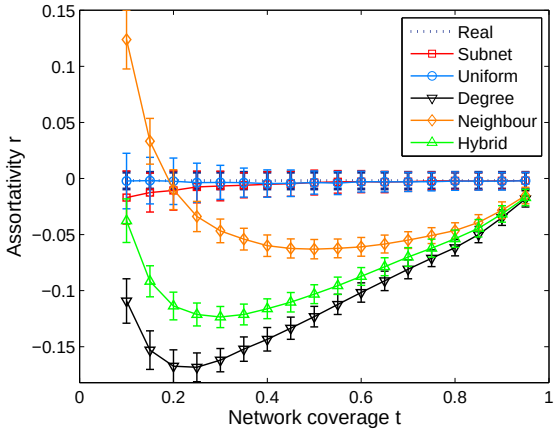

Figure B.2: The assortativity r against network coverage t for the BA network. Error bars are one standard deviation.

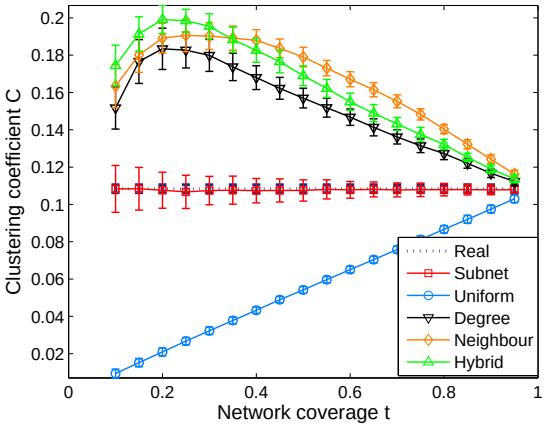

Figure B.3: The clustering coefficient C against network coverage t for the BA network. Error bars are one standard deviation.

B.2 The Reed and Haverford networks

The Reed and Haverford networks show qualitatively similar discovery behaviour as the Caltech network: The neighbour method is best at finding edges (Figs. B.4 and B.5). Although the networks are assortatively mixed, the degree method discovers a dissassortatively mixed network even for medium coverage. The neighbour method captures the assortativitiy rather well (see Figs. B.6 and B.7. In Figs. B.8 and B.9 we again observe the sensitivity of the estimated clustering coefficient on the detailed network structure: The estimates are initially below the real value for all guided methods. At later stages, the hybrid method produces an overestimate. On the Reed network, the same applies to the neighbour method whilst the degree method keeps underestimating the actual value. On the Haverford network, this behaviour is reversed. The estimates of the subnet and uniform strategies are generally close to the actual values, with the usual exception of the clustering coefficient's estimate by the uniform method.

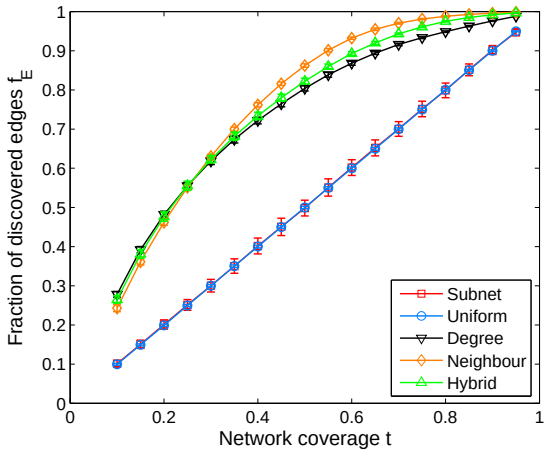

Figure B.4: The fraction of discovered edges f_E against network coverage t for the Reed network. Error bars are one standard deviation.

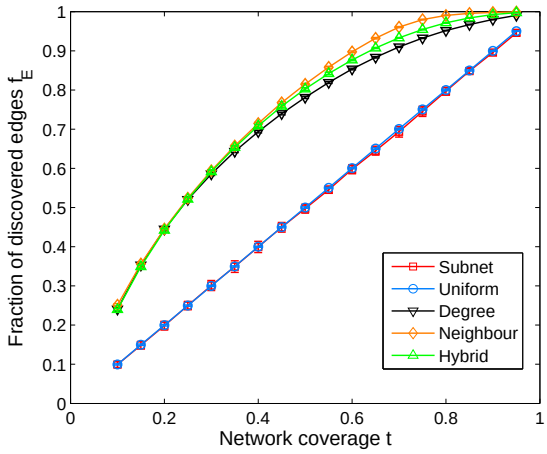

Figure B.5: The fraction of discovered edges f_E against network coverage t for the Haverford network. Error bars are one standard deviation.

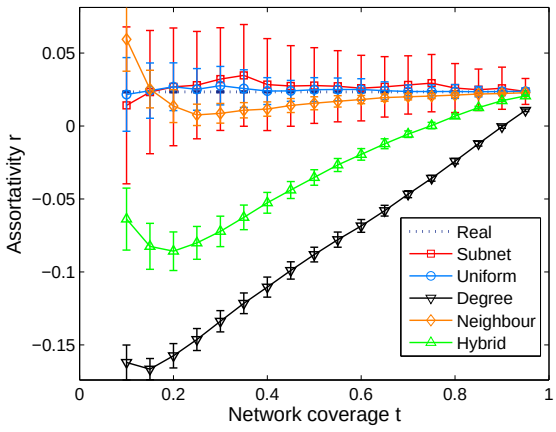

Figure B.6: The assortativity r against network coverage t for the Reed network. Error bars are one standard deviation.

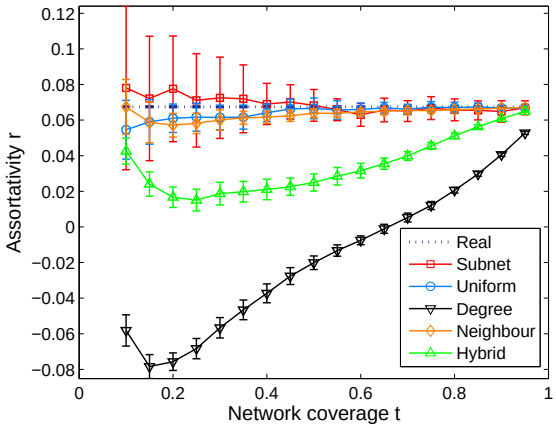

Figure B.7: The assortativity r against network coverage t for the Haverford network. Error bars are one standard deviation.

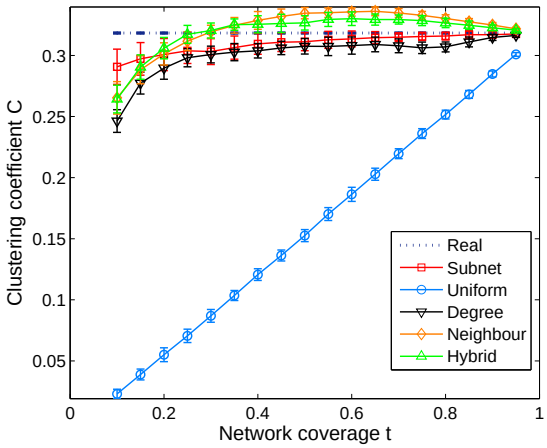

Figure B.8: The clustering coefficient C against network coverage t for the Reed network. Error bars are one standard deviation.

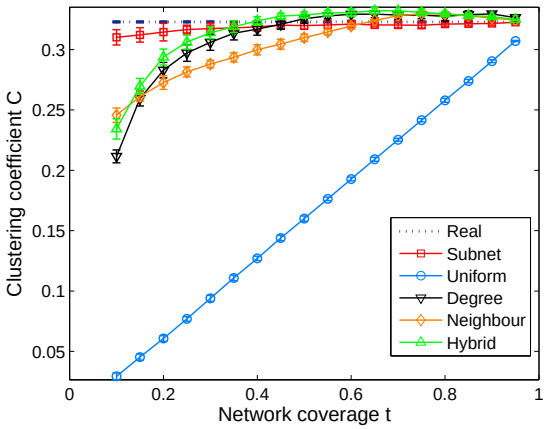

Figure B.9: The clustering coefficient C against network coverage t for the Haverford network. Error bars are one standard deviation.

B.3 Edge discovery on the power grid

The power grid is different from the other networks considered here in that the average degree is only about 3. This extreme sparseness has important consequences for the discovery process: In Fig. B.10 we can see that the degree method is worst at finding edges above a coverage of about $t \approx 0.4$. The neighbour method still outperforms the uniform and subnet method but even here the difference is much smaller than on the other networks (except the ER ones where there is no difference at all).

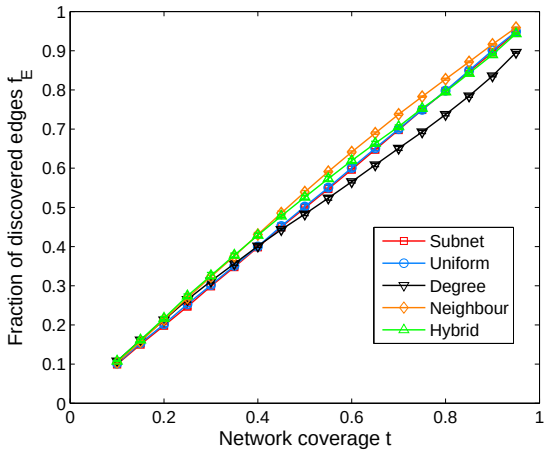

Figure B.10: The fraction of discovered edges f_E against network coverage t for the power grid network. Error bars are one standard deviation.

B.4 Component discovery on the Caltech network

The Caltech network features three very small components. This makes the interpretation of the number of discovered components (Fig. 5) more complicated. From Fig. B.11 we can deduce that the degree method has particular difficulty in finding the small components: From $t = 0.3$ until $t = 0.7$ the degree method only finds two components, and one of them is the giant component of the real network. Only around $t = 0.8$ the other small components are established. In contrast, the neighbour method finds the small components earlier; on the other hand, this strategy takes longer to find the edges that link all the nodes of the largest component together.

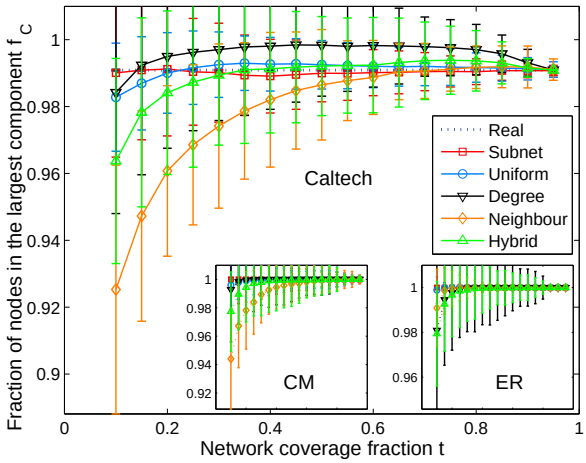

Figure B.11: The fraction of non-isolated nodes that are part of the largest component f_c against network coverage t for the Caltech, CM and ER networks. Error bars are one standard deviation.

B.5 Improving the estimate of the clustering coefficient

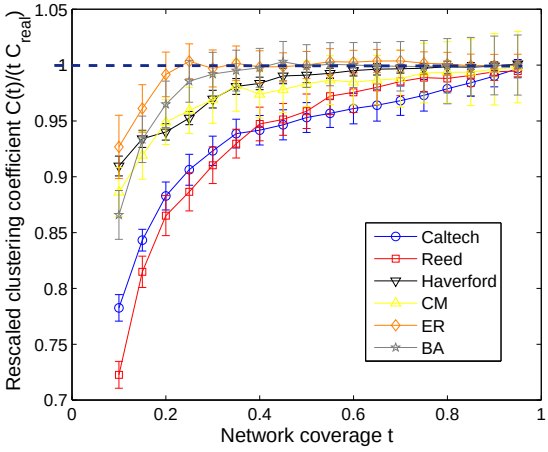

Figure B.12: The normalized clustering coefficient divided by t as found by the uniform method against network coverage t.

In Fig. 8 we have found that the clustering coefficient estimated by the uniform method appears to increase nearly linearly with the network coverage fraction t. Fig. B.12 shows the

estimated clustering coefficient divided by t and the actual value C_{real}. As already indicated in section 5.2, the curves are distinctly non-linear for small values of t. For higher coverage, dividing out the factor of t gets the estimated clustering coefficient closer than 90% to the actual value. The best result is obtained on the ER networks where the estimate is essentially correct for $t \geq 0.3$.

B.6 Discovery with errors

When we allow for errors to occur in the network discovery, we have seen in Fig. 9 the number of erroneously found edges is larger if only false-positives are present than if only false-negatives can occur. Here we study the influence of the errors on the topological properties of the Caltech network. Fig. B.13 shows that the existence of false-positive leads to the situation that the distinct components of the real network get connected by erroneously placed edges. On the other hand, false-negatives have a less severe impact. Only the neighbour method (and thus also the hybrid strategy) are strongly influenced.

From Fig. B.14 we learn that false-negatives have almost no impact on the average path length. Even the consequences of including false-positives is rather small.

The assortativity appears to change drastically (Fig. B.15): When looking at the diagram, we have to bear in mind that the absolute value of r is tiny, so even (absolutely) small deviations appear large on the relative scale. Furthermore the assortativity can be both positive and negative. It is remarkable how little the estimate of r changes with false-negatives; only for the neighbour method changes slightly for low coverage. The impact of false-positives is more dramatic: For the subnet, uniform and neighbour strategies, the assortativity changes sign for small t, and only the neighbour method recovers the original sign at higher t.

The picture is similar for the clustering coefficient. Fig. B.16 indicates that the estimate changes little under the inclusion of false-negatives; when false-positives are added, the variation is much larger.

Finally we note a surprising feature: With false-positives, the guided methods appear generally more stable in their estimates. Except for the number of components, the relative change in the estimates of the subnet and uniform strategies is much larger. However before we can make a justified claim here, further tests are certainly necessary.

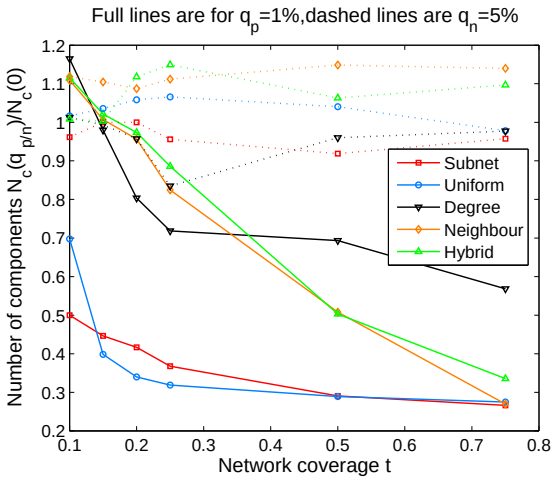

Figure B.13: The number of components when random errors occur in the discovery, normalized by the number of components found without errors. Solid lines are for $q_n = 0, q_p = 0.01$ and dotted lines show the case $q_p = 0, q_n = 0.05$. The underlying network is Caltech.

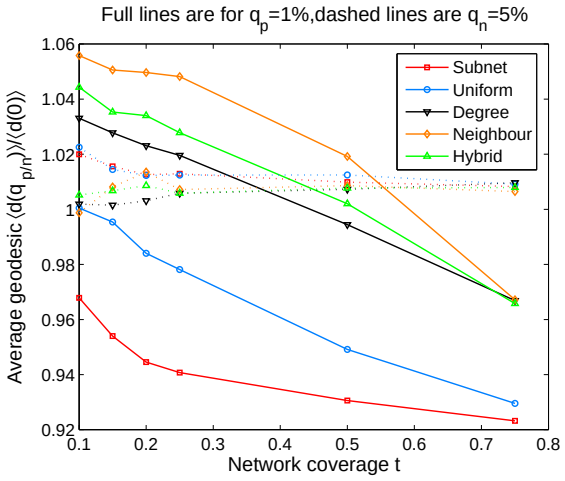

Figure B.14: The average geodesic length when random errors occur in the discovery, normalized by the average geodesic length without errors. Solid lines are for $q_n = 0, q_p = 0.01$ and dotted lines show the case $q_p = 0, q_n = 0.05$. The underlying network is Caltech.

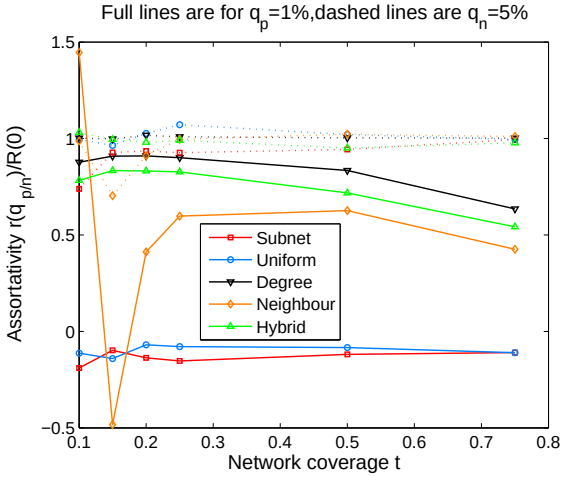

Figure B.15: Assortativity when random errors occur in the discovery, normalized by the assortativity without errors. Solid lines are for $q_n = 0, q_p = 0.01$ and dotted lines show the case $q_p = 0, q_n = 0.05$. The underlying network is Caltech.

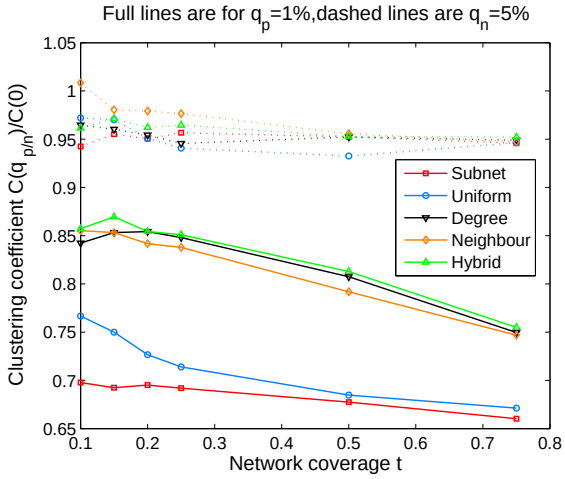

Figure B.16: The clustering coefficient when random errors occur in the discovery, normalized by the clustering coefficient without errors. Solid lines are for $q_n = 0, q_p = 0.01$ and dotted lines show the case $q_p = 0, q_n = 0.05$. The underlying network is Caltech.