Discussion Paper

Sampling restricted access networks

2018 | 02

Leon Willenborg

Content

1. Introduction 4

2. Networks and network types 6
2.1 Digraphs 6
2.2 Graphs 7
2.3 Partition networks 7
2.4 FAN: Full access network 9
2.5 RAN: Restricted access network 9
2.6 Massive network 11
2.7 Static network 12
2.8 Dynamic network 12
2.9 Example: Internet 12

3. Some network characteristics 13
3.1 Degree related characteristics 13
3.2 Path related characteristics 15
3.3 Spectrum of AA' 21
3.4 Node rank 21

4. Sampling RANs 24
4.1 Webpages or domains? 25
4.2 Methods of access 25
4.3 Sample size 25
4.4 Search strategies 26

5. Analysis of the sampling results 29

6. Simulating RANs 30

7. Discussion 33

References 33

Appendix A. Some network terminology 36

Summary

The paper discusses some topics concerning the sampling of restricted access networks (RANs). These are networks that are usually too big to hold all structural information neatly accessible, say in the form of node and arc lists. Often they are very dynamic as well, so that it is impossible to obtain instant snapshots: they change while they are being explored. The internet is a key example of such networks, or social networks like Facebook and LinkedIn that are part of it. Structural information about such networks can only be obtained by sampling them. As there is no list of nodes and arcs available, such information has to be collected, for instance by random searches of the network. Such searches can be carried out in many ways. It is interesting to apply them to the same RAN and compare the information collected by each method. For each node it is relatively easy to determine the outgoing arcs. But it is difficult to determine the arcs pointing to it. This requires information about the entire network, or a great part of it. An interesting question concerns the size of the sample in order to be able to say something about the structure of the network with sufficient precision. The paper is only a first exploration of the area. It is partly based on existing literature and partly on ideas from the author. It may very well be that these ideas have been described in the literature before, however unbeknownst to the present author.

Keywords

Sampling, network analysis, data collection, network sampling, network search, random access network, massive network, dynamic network, network characteristics, page rank, network simulation.

1. Introduction

A lot of situations in real life can be modelled by networks. [1] Situations where there are 'individuals' represented by nodes (or points) and the relations these individuals have are represented by links. These links may be undirected (in which case they are called edges) or directed in which case they are called arcs. Edges can be used to model symmetric relations and arcs asymmetric ones.

Examples of symmetric relationships between individuals are 'being family members'. One might think that 'being friends' is also symmetric but it often isn't. Person A may consider person B as a friend, but person B may not consider person A as a friend. Empirical studies among school children have shown that this property actually is true quite often. It seems to be related to social status. Children (and perhaps people in general) tend to favour persons with whom they are acquainted as their friends if they have (or are perceived to have) a higher social status. Being merely acquaintances for the one may be considered being friends by the counterpart.

The nodes need not necessarily be concrete objects like persons. For a given network one can consider its links as nodes in a newly to be defined network, and consider two nodes connected by a link (in the new network) if the links in the original network have at least one node (of the original network) in common. For another network defined for a given one, its cycles can be the nodes, and two cycle are connected by a link if the cycles have an edge in common.

Network sampling is necessary when a network is too big to study it in its entirety. Too big in terms of collecting the data one is interested in (it simply takes too long to collect them, which is particularly a problem if the network is constantly changing). Or too big, because the amount of data exceeds the computer memory available.

Social media (Facebook, LinkedIn, Twitter, etc.) are examples of such massive networks. The nodes in such a network are the personal pages and the (directed) links are the hyperlinks from such pages to other pages, which are those of 'friends'. Besides being large these networks are dynamic: the nodes constantly change as personal pages are added (new members), disappear (from individuals tired of being active in the network) as do the links (new 'friends' are included and some old ones are 'unfriended').

Networks are studied in various disciplines for a long time, like sociology, management science, computer science, traffic studies, telecommunications to mention but a few. Internet is a relatively new kid on the block, and is an example of a network of huge proportions. But it is not the only massive network in existence. Think, for instance, of the financial world with transactions likely transferring money between

[1] The author is grateful to Edwin de Jonge for stimulating discussions on the topic of this paper and to Sander Scholtus for reviewing it. The review led to several improvements in the text.

bank accounts. These networks are so huge and dynamic that it is impossible to know them at any point of time. One can study portions of them, while in the meantime other parts have been already changed. But one should describe them at a level that is more stable.

Which examples can be given of RANs in which official statistics could be interested? One such example is in the area of business statistics. Traditionally such statistics view companies as separate entities, not as actors in a (world-wide) network, trading services and goods in exchange for payments. A major problem here is the availability of relevant information. But even if it would be available, it would be fragmented and probably only available locally (per country). So in the best case one only could view part of this network, and the problem is what one can learn from the structure of the observed part about the structure of the entire network.

Another example of an application in official statistics is the LGBT-community.[2] This at the moment is a topic of interest among sociologists. Who belongs to this group? Who is acquainted with, married to, etc. whom? Are these relationships predominantly national or international? These or only a few of the questions that can be asked about this group. The LGBT-community can be viewed as a RAN. One can only hope that on the basis of samples one gets an idea of the size and structure of this group of people.

Network science typically is focused on big networks.[3] The properties it is typically interested in are specific for this relatively new field: how to characterize the networks that one is dealing with in terms of a few striking characteristics. Things like the distribution of in-degrees (or page ranks), out-degrees, diameters, hubs (important nodes to which a lot of links point), etc. In short, its aim is to characterize networks statistically. And to analyse them for their peculiarities. Simulating networks on the basis of a few characteristics is a good method to find out whether all relevant characteristics (for a particular application) have been identified, or whether some crucial ones are missing.

The organization of the rest of the document is as follows. In Section 2 we discuss some network types that are frequently used in network science. In Section 3 we discuss the network characteristics that we shall focus on in the present paper: out-degree and in-degree (for digraphs), degree (for graphs), degree asymmetry (the relative number of arcs that do not have a counter arc), distance and the derived concept of diameter (for graphs and digraphs). Section 4 deals with sampling RANs. Various issues that play a role here are discussed, such as deciding what are the nodes and links in a RAN? In case of the WWW one can choose, for instance separate pages to be nodes, or domains. The former is very volatile as webpages appear and disappear constantly. Domains are usually (much) bigger and less volatile, as they consist of a (potentially varying) set of (also potentially varying) webpages. The links

[2] LGBT = Lesbian, Gay, Bisexual and Transgender. Bart Bakker suggested this example to the author.
[3] But 'big' is relative. It is not so much related to the absolute size, as to the effort it takes to collect information about the network, in particular the links.

in both case are defined by hyperlinks pointing from one webpage to another. Other subject discussed here are about the sampling of RANs: sample and search strategies, sample size. Samples are used to collect information from a RAN from which structural information (about the network structure) about this RAN can be computed or estimated. This is considered in Section 5. The question also is: what information should actually be extracted from a RAN in order to 'understand' it sufficiently. Of course, strictly speaking this question cannot be answered unless the exact goals of the application are known. But certain structural characteristics of a network such as average outdegree, average indegree, etc. are basic and likely to be part of any characterization of the structure of a network. Such information can be used to simulate RANs with given characteristics in order to mimic the original version. Simulating RANs is considered in Section 6. The final section, Section 7, contains a discussion on the main results of the paper. The paper is completed with a reference list and an appendix, Appendix A, containing a glossary with network terminology that is used in the present paper.

2. Networks and network types

In the present section we discuss different types of networks that one may encounter in network science. The various concepts are not ‘orthogonal’. On the contrary often we find networks with a mix of the properties we discuss. We use the word network as a generic term that includes both the concepts of ‘digraph’ and of ‘graph’. In a sense (that will be made clear below), a graph is a special case of a digraph. The remaining concepts are special cases of graphs or digraphs that are worth considering in the context of network sampling.

2.1 Digraphs

A digraph – or directed graph – is a network with oriented edges, or arcs. If a and b are nodes and there is an arc from a to be we write this as (a,b), the ordered pair of the nodes a and b. It is clear that (a,b) is different from (b,a). We say that (a,b) and (b,a) are arcs with different orientations. And also that (a,b) is the counter-arc of (b,a) and vice versa. It is possible that a digraph possesses an arc (a,b) but not its counter-arc (b,a), for certain of its nodes a and b.

Plots of digraphs

When plotting a digraph the nodes are denoted typically as dots (or squares, triangles, etc.) and the arcs as ‘arrows’. So in case we have an arc (a,b) there is a line segment or curve connecting a and b with an arrow-head pointing in the direction of b. In case (a,b) and (b,a) are arcs it may be handy to plot this as an edge in a graph (see Section

2.2, namely by a line segment or a curve without an arrow-head.[4] The advantage of using this convention is that the picture of a digraph becomes less cluttered.

1-Neighbourhoods

For a node a in a digraph G=(V,E) its out-neighbourhood NO_a is the set of nodes in G that are linked to a where a is the starting point. That is: $NO_{a,1} = \{b \in V | (a,b) \in E\}$. The subscript '1' indicates that the neighbouring points are just one step away from a. We can extend this neighbourhood concept using paths (see the next subsection) and consider nodes that are further away from a. Another kind of neighbourhood that can be defined in a digraph which is more or less dual to the one just introduced, is the in-neighbourhood NI. For a node $a \in V$ its in-neighbourhood is the set $NI_{a,1} = \{b \in V | (b,a) \in E\}$, where, again, '1' denotes that the neighbouring points are just one step away.

2.2 Graphs

A graph is a network where the edges are represented by unordered pairs of nodes. The adjacency matrix of a graph is symmetric.

A graph can be viewed as a special kind of digraph, namely one in which every arc has an arc in the opposite direction. So if (a,b) is an arc in such a digraph, the counter-arc (b,a) is as well.

We can compare a digraph with its underlying graph. This can be viewed as a completion (in terms of arcs) of the given digraph: for each arc we make sure that it also contains its opposite version. We can measure the distance between the two in terms of arcs without its counterpart. The more such cases exist, the bigger the distance. The bigger the distance the more the digraph is unbalanced.

An edge in a graph can be viewed as an undirected link between two nodes. Or, viewed as a digraph it can be represented as two arcs – an arc and its counter-arc. The difference is representing an edge between two points a and b as a set, $\{a, b\}$ of the node a and b, or as two ordered pairs, i.e. (a,b) and (b,a), corresponding to arcs.

Each digraph yields a graph – the underlying graph - when forgetting about the orientation of the arcs. If (a,b) is an arc, the corresponding edge in the underlying graph is the set {a,b}. The counter-arc (b,a) would give rise to the same edge {a,b}.

2.3 Partition networks

Partition networks may surface in case we have a graph, and there is an equivalence relation on the set of nodes. For instance, all web pages on the Internet on the same

[4] More properly would be to use two arrow heads, one pointing in the direction of a and the other in the direction of b, but this is superfluous. No arrowheads means two arrowheads.

domain are considered equivalent in some application. Then we get a new digraph where the nodes are domains instead of webpages, and arcs are between domains instead of between web pages. (See Section 2.9 for more on this.)

Let G=(V,E) be a network. Suppose that $\mathcal{V} = \{V_1, \dots, V_p\}$ is a partition of V. That is, $\bigcup_{i=1}^{p} V_i = V$, $V_i \cap V_j = \emptyset$ if $i \neq j$, $V_i \neq \emptyset$ for all $i = 1, \dots, p$. This implies a new network, $G^p = (\mathcal{V}, \mathcal{E})$, so with the parts of $\mathcal{V}$ as nodes and

- If G is a graph: $\{V_i, V_j\} \in \mathcal{E}$ if and only if there is a $v_i \in V_i$ and a $v_j \in V_j$ with $\{v_i, v_j\} \in E$, and
- If G is a digraph: $(V_i, V_j) \in \mathcal{E}$ if and only if there is a $v_i \in V_i$ and a $v_j \in V_j$ with $(v_i, v_j) \in E$.

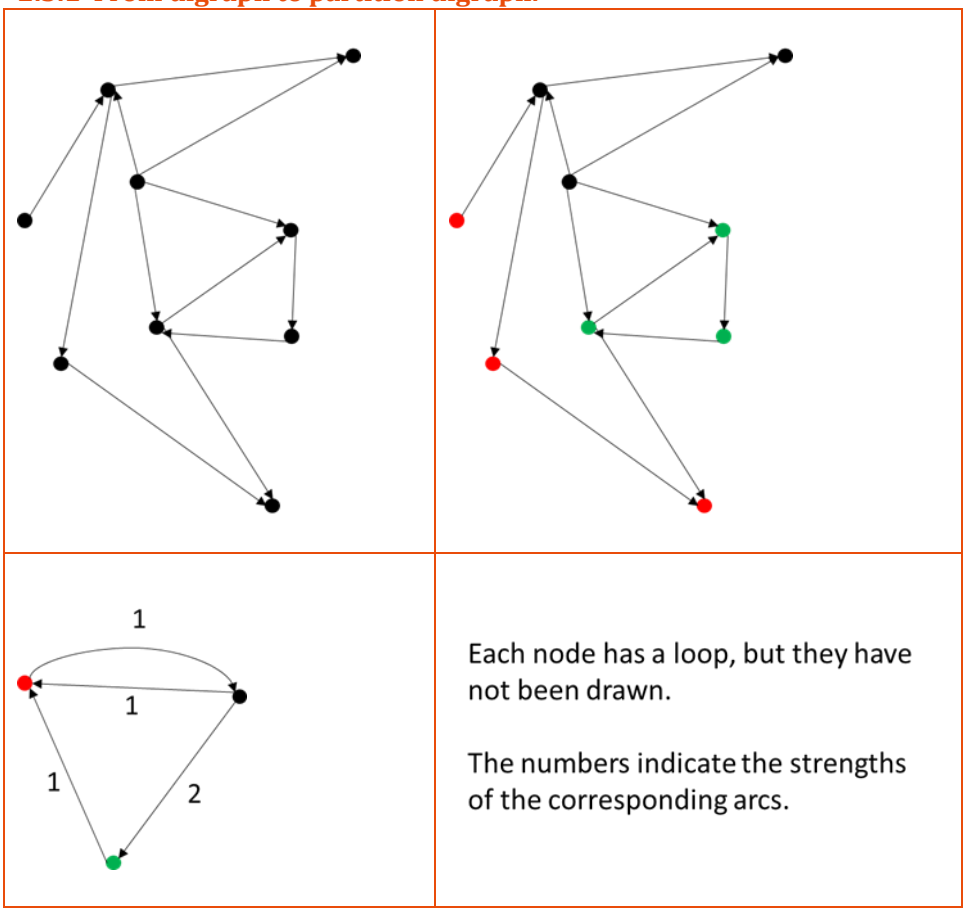
One can associate the strength of an edge or arc in its partition counterpart, which is the number of edges or arcs that are represented by the same edge or arc in the partition graph (or digraph).

To illustrate the concept of a partition graph we present the following example.

Example

In Figure 2.3.1 a digraph is plotted in the upper left corner. In the upper right corner colours are used to indicate the nodes that are in the same parts of a partitioning of the node set into three pieces. In the plot in the lower left corner the corresponding partition digraph is represented. To avoid a cluttered picture, the loops have been discarded.

2.3.1 From digraph to partition digraph.



2.4 FAN: Full access network

This type of network is what one usually encounters outside network analysis. These networks are entirely know: the nodes, the edges (in case of a graph) or arcs (in case of a digraph), for instance in the form of an adjacency matrix, or edge / arc list, etc.

2.5 RAN: Restricted access network

It may be that the network in its entirety is not available, but can only be crawled, hopping from one node to neighbouring ones. In this case it is easy to determine the nodes pointed to by a link from a node visited. But it is a major task to determine the arcs / links that point to a given node. So determining the outdegree of a node is easy, whereas determining its indegree ("page rank" in the www context) is difficult, in the sense that more work has to be invested to collect the necessary information. In fact, It may even be impossible: if the network is too big, or is partly hidden, or changes rapidly. Determining the outdegree of a node requires only local knowledge, whereas determining the indegree requires global knowledge, that is, of the entire network.

Remark

For some networks it may be the case that they are only partly accessible because some of its structure is not accessible due to privacy concerns. This is also a form of being invisible, but it should be distinguished from the concept at the focus in the present section. For our purposes we are inclined in the present paper to treat nodes that are inaccessible due to privacy concerns not as part of the network we are interested in. They are comparable to persons who never participate in surveys, as a matter of principle, that is, hard core non-respondents. But what to do with such nodes, in different applications, depends on the kind of problem one is interested in. ■[5]

For a partially explored RAN we have three types of nodes: explored ones (e), observed ones (o) and the unobserved ones (u). Only for the explored nodes we know all the nodes referred to (at the moment of observation). For the observed nodes we do not know at all to which other nodes they refer. In general, the adjacency matrix for a partially explored RAN (PERAN) has the following form in terms of block matrices:

$$A = \begin{pmatrix} A_{ee} & A_{eo} & A_{eu} \\ A_{oe} & A_{oo} & A_{ou} \\ A_{ue} & A_{uo} & A_{uu} \end{pmatrix}. \tag{2.1}$$

Each block matrix $A_{\xi\lambda}$contains information about arcs connecting node of type ξ (that is, e, o, u) with nodes of type λ (that is, e, o, u). A_{ee} and A_{eo} are the only sub-matrices from A that we know. About A_{eu} we know that it is a zero matrix, and we know its number of rows, but not its number of columns. So we may put $A_{eu} = 0$. All the remaining sub-matrices of A are unknown.

This implies that there is a difference between usefulness of AA' and $A'A$. In the former case there is one useful sub-matrix (namely the one in the upper-left corner) , whereas in the latter case no entry is known.

Instead of concentrating on the adjacency matrix A in (2.1) we can, just as well concentrate on the known part

$$\breve{A} = (A_{ee} \quad A_{eo}). \tag{2.2}$$

If the network is sufficiently explored $\breve{A}$ has a lot in common with A in terms of structural characteristics, or approximations thereof. Instead of $\breve{A}$ we may consider

$$\breve{A}\breve{A}' = A_{ee}A'_{ee} + A_{eo}A'_{eo}, \tag{2.3}$$

which is a square symmetric positive semi-definite matrix, which only has real, nonnegative eigenvalues.

[5] The symbol '■' is used to indicate the end of an example.

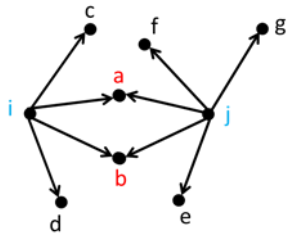
To illustrate (2.3) we consider the following example.

Example

In Figure 2.5.1 two explored nodes are shown, namely nodes i and j. The other nodes, i.e. nodes a, b, c, d, e, f, g , are only observed. So for these nodes we do not know to which other nodes they point.

The nodes and links that not have been observed have not been plotted, as they are not known. But we should not forget that the picture is incomplete. We only do not know which nodes and links are missing.

2.5.1 Nodes with outgoing links.



We can specify the known part of the adjacency matrix:

$$\check{A} = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \end{pmatrix}, \tag{2.4}$$

where the columns correspond to the nodes $i, j, a, b, c, d, e, f, g$, respectively and the rows with the nodes i, j, respectively. The left most 2×2 submatrix corresponds to A_{ee} and the remainder to A_{eo}.

Then

$$\check{A}\check{A}' = \begin{pmatrix} 4 & 2 \\ 2 & 5 \end{pmatrix}. \tag{2.5}$$

This matrix tells us that there is one node with outdegree 4 (node i) and one node with outdegree 5 (node j), and that these nodes point to 2 common nodes (nodes a, b). ■

The aim of the sampling of RANs is to obtain $\check{A}$'s and $\check{A}\check{A}'$'s that have properties that resemble that of A and AA', respectively, and concern the global structure of the network.

2.6 Massive network

There can be issues with networks when analysing them. First of all they can be massive, too big to process off-line. Massiveness of a network typically implies that the network has restricted access. It may force one to apply network sampling in order to get an impression of the characteristics of the entire network. Also, because of its

size, the entire network may change while one is sampling it. One can only hope that these changes are only minor and do not affect the properties one is interested in too much.

2.7 Static network

In a static network the structure of the network (in terms of the nodes and edges / arcs) does not change over time. In a sense this is an ideal situation. In practice networks studied by network science typically change over time. In order to study dynamic networks sometimes an option is to take snapshots of the network at different points in time and study the differences of the various snapshots. They give an idea about how the network changes. But in case the network is big or hidden (or both) this is not an option.

2.8 Dynamic network

A third issue with many networks is that they are dynamic. Their structure changes constantly. New nodes and/or links are added or deleted continuously. In order to get an idea about the changes that take place in the network, that is, its dynamic character, one on can take snapshots at regular time intervals, or, if the network is too big, one can sample it regularly. But taking snapshots takes time during which the network can change. See Section 4 for remarks on this aspect.

For a dynamic full access network (DFAN), it would be an option to consider snapshots at different points in time. At each point in time one obtains a FAN. By comparing the different FANs one could focus on finding differences: added nodes, deleted nodes, added arcs, deleted arcs. Or one could simple consider various characteristics for each of these networks, diameter, connectivity aspects, (in and out) degrees, etc., and compare these results for the different snapshots.

A dynamic restricted access network (DRAN) is the kind of network that we would be interested in, given the focus of the present paper. But such a network cannot be treated it in the same way as a DFAN, since no snapshots can be taken. What we can do instead to study its dynamics is to retrace paths though the network made earlier and study what has changed: nodes deleted, arcs removed or added. If a node is missing on a path it is not a problem to jump to the next node, as this node is known from an earlier path.

2.9 Example: Internet

The internet is a massive, dynamic RAN, with which most of us are familiar, in the sense that we use it a lot. But it works so well that there is no need to delve into its architecture. But in case we want to sample it (or part of it) this is a necessity.

One question one has to answer when sampling the internet is which objects to take as nodes: webpages or domains. At first sight it seems natural to take webpages as nodes and the link they contain as the outgoing links. This is fine, but one should realize that at this level things are a bit hazy: webpages appear and disappear constantly. If we want to have a structure that is more stable then we should look for other entities as nodes. Domains spring to mind. If we view a domain as a collection of webpages at a given moment, we in fact are deriving a new network from one at the page level. Domain A has a link to domain B if there is a webpage W_A in A with a link to a webpage W_B in B. If fact there can be many such pages. A measure expressing how well domain A is connected to domain B is by the number of webpages in domain A with a link to webpages in B. There may also be links from webpages in domain A to other webpages in domain A. It is interesting to know the number of such links to the same domain. It is measure for internal cohesion of a domain. Likewise it is interesting to compare the number of links from domain A to domain B, with the number from domain B to domain A. The domain with the higher number of links pointed to it has a higher status of the two domains, so to speak. Instead of the indegree of a domain, we can also look at its page (or node) rank. See Section 3.4.

3. Some network characteristics

In network analysis one is interested in specific characteristics of networks. We briefly consider some of these in the present section.

The structure of a network is completely specified by its adjacency matrix or by its incidence matrix. These matrices are usually sparse for the kind of networks typically studied in network analysis.

3.1 Degree related characteristics

The concept of degree needs to be treated in case of digraphs and graphs separately, as there are essential differences.

Outdegree and indegree in digraphs

Given a digraph G=(V,E) with $|V| = n$ and $|E| = m$, we can consider a node $p \in V$. Given this p we have the sets $E_p^{out} = \{(p,q) \in E | q \in V\}$ and $E_p^{in} = \{(q,p) \in E | q \in V\}$ of arcs in G pointing away from p (E_p^{out}) and to p (E_p^{in}). The sizes of these sets, $\left|E_p^{out}\right|$ and $\left|E_p^{in}\right|$, are called the outdegree and indegree of p, respectively. If we look at the adjacency matrix A_G of G, the outdegree of p is the sum of elements in the p-th row, and the indegree of p the sum of the elements of the p-th column. So $A_G \iota_n$ and $\iota_n' A_G$ are vectors with outdegrees and indegrees for the nodes of G. We have that the total number of arcs $\iota_n' A_G \iota_n$ equals $\iota_n'(A_G \iota_n)$ which can be viewed as the sum of the

outdegrees over the nodes. But the total number of arcs $\iota_n' A_G \iota_n$ also equals $(\iota_n' A_G)\iota_n$ which can be interpreted as the sum of the indegrees over the nodes.

Degree in graphs

As graphs are balanced digraphs, in the sense that each arc has a counter-arc, the results for outdegrees and indegrees of the previous section coincide. Instead the concept degree is used. For a graph (V,E), where E consists of sets of nodes (with 1 or 2 elements), the neighbours of p in G is the set $N_p = \{\{p,q\} \in E | q \in V\}$. The size of this set, i.e. $|N_p|$, is the degree of p. The sum of the degrees over all nodes is twice the numbers of edges in G, that is, $\iota_n' A_G \iota_n = 2|E|$, since $A_G = A_G'$. Viewed as a di-graph, the number of arcs in G is twice the number of edges, as each edge is represented by an arc and its counter-arc. From this result it also follows that the number of nodes with an odd degree is even.

Degree asymmetry in digraphs

The adjacency matrix of a graph is symmetric, but that of a digraph is not necessarily. It is quite interesting to compare the deviation of the adjacency matrix of a digraph with that of its underlying graph. This is in fact a question about how many arcs in a digraph that do not have a counter-arc, that is an arc in the opposite direction, so with the opposite orientation. It can also be seen as the asymmetry between indegrees and outdegrees in a digraph. If we view a graph as a digraph where each arc has its (unique) counter-arc, it is perfectly symmetric in this respect.

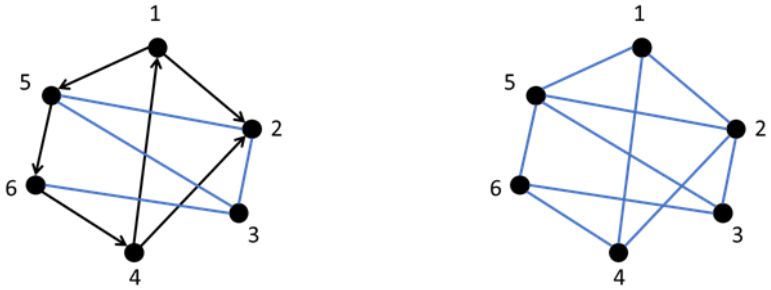
A highly asymmetric digraph has many arcs (a, b) for which (b, a) is not an arc. So it makes sense to define a concept that quantifies the deviation of a digraph from its underlying graph. We call that concept asymmetry of the digraph. It is related to the symmetry of the corresponding adjacency matrix when it is compared with is transpose.

But before we consider the definition, we look at an example. It is the one that is also presented in Section 3.4.

Example

We consider the digraph pictured on the left in Figure 3.1.1. Some line segments have been drawn. They symbolize a pair of arcs going in one direction and the reverse direction. Using such line segments has the advantage of making the picture look more clear. But also they mimic what is customary for graphs, where line segments are used to depict edges.

3.1.1 A digraph (left) and its underlying graph (right).



From Figure 3.1.1 it is immediately clear how asymmetric the digraph is: there are 6 arcs without counter-arcs.

We now consider the computation of the degree asymmetry of a digraph, using the digraph G in Figure 3.1.1 as an example. Let A be the adjacency matrix of G. From this we can compute the adjacency matrix of the underlying graph of G. We obtain this from adding A and A', using binary addition).[6] We denote this by $A \oplus A'$.

$$\mathrm{A} = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}, \qquad \mathrm{A}' = \begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{pmatrix}, \tag{3.1}$$

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

The degree of asymmetry can now be expressed as $\iota'(A \oplus A' - A)\iota = \iota'(A \oplus A' - A')\iota = \iota' A \oplus A'\,\iota - \iota' A\iota = \iota' A \oplus A'\,\iota - \iota' A'\iota$, where we used that $\iota' A\iota = \iota' A'\iota$. ■

3.2 Path related characteristics

In this section we discuss some properties that are related to path connectedness. It is important to make a distinction between digraphs and graphs. In the digraph situation we have an asymmetric relationship and a symmetric relationship in the graph context.

Paths and cycles

For digraphs the important notion of a path can be defined. In a sense it is an extension of an arc. An arc is a path of one step. A path in a digraph from nodes a to b is a (finite) sequence of arcs that, if traversed, allows one to go from a to b. If there is one intermediate node, c, then (a,c), (c,b) are the arcs 'traversed'. More generally, if the

[6] Binary addition is defined by the following rules: $0 + 0 = 0,\ 0 + 1 = 1 + 0 = 1 + 1 = 1$.

nodes we pass on our way from a ('starting point') to b ('end point') are (in this order): $s_1, \dots, s_k$, with $k > 3$, $s_1 = a$ and $s_k = b$, then $(s_1, s_2), (s_2, s_3), \dots, (s_{k-1}, s_k)$ is the sequence of arcs in the digraph. The path length is the number of arcs traversed, which is $k - 1$ in this case. A cycle is a path with the same starting point and end point. A loop is a cycle with one node.

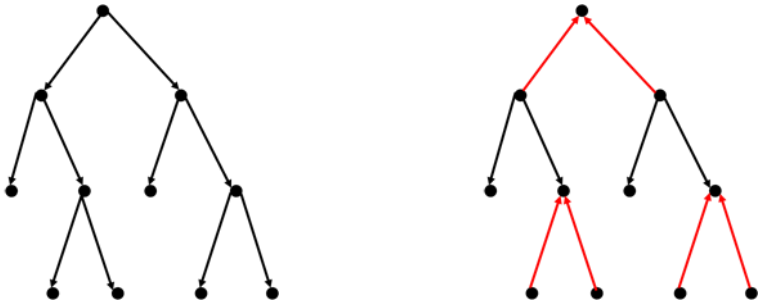
Trees and ditrees

A tree is a connected graph without cycles.[7] A tree is also a graph that has the property that for any pair of nodes there is a unique path joining them. What about such properties in the realm of digraphs? How does a tree generalize to a ditree?

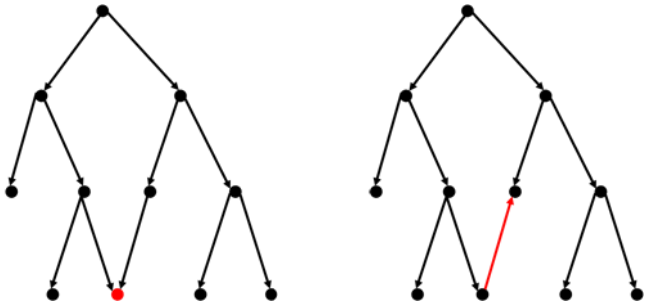
The first intuition is perhaps that of a tree, but with arcs instead of edges. If we look at the top two digraphs of Figure 3.2.1 they would probably qualify as ditrees, because the underlying graph is a tree, so is connected and has no cycles. But for neither of these two digraphs holds that any pair of points can be joined by a path. For a digraph that is not a graph that is not possible. In fact, for the digraph at the top right no path involves more than 2 nodes.

We could demand that for a ditree there is at most one path connecting two points. That property is satisfied by the two digraphs on the top in Figure 3.2.1. In the example at the bottom left of Figure 3.2.1, this property does not hold. There are two paths from the top node to the red node. But it is true for the digraph at the bottom right, which differs from the one to the left of it by the direction of one arc (in red). Of course, in both cases the underlying graph is not a tree. So these examples show that different types of digraphs can be considered generalizations of trees. Which of them to call a ditree is a matter of taste.

3.2.1 Ditrees and lookalikes.



[7] A graph without cycles is a forest, which can be viewed as a disjoint union of trees. Disjoint in the sense that no points or edges are shared.



Reachability and connectedness are properties for which paths are needed to compute them. These concepts are discussed in the next two subsections.

Reachability in a digraph

If there is a path from node a to node b in a digraph G=(V,E), we denote this by $a \rightsquigarrow b$. We say that b can be reached from a. Although G is not mentioned in the notation it is the context within which the concept of reachability is understood. We have, for each node a in G, that $a \rightsquigarrow a$ (identity). And also, if a, b and c are nodes in G and $a \rightsquigarrow b$, $b \rightsquigarrow c$ then $a \rightsquigarrow c$ (transitivity).

To summarize, we have the following properties for reachability:

Reflexivity: $a \rightsquigarrow a$, for each node a,
Transitivity: $a \rightsquigarrow b$, $b \rightsquigarrow c$ then $a \rightsquigarrow c$, for nodes a, b, c.

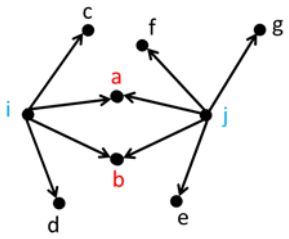
The following property ('symmetry') may are may not hold for pairs of nodes a, b in a digraph. In a graph however it is true for any pair of nodes:

Symmetry: $a \rightsquigarrow b$ implies $b \rightsquigarrow a$, for nodes a and b.

Reachability is illustrated in Figure 3.2.2.

For a node a in G we can define the transitive closure of a in G: $T_a = \{b \in V | a \rightsquigarrow b\}$. We call the node a the source of T_a, which is referred to as the reach of a in G. T_a consist of all the nodes in G that can be reached from a. Obviously $a \in T_a$ for all nodes a in G. It is a maximal set in the sense that for a $c \in V$, such that $a \rightsquigarrow c$, it holds that $c \in T_a$. If $a, b \in V$ and $a \rightsquigarrow b$ then $T_a \supseteq T_b$.

3.2.2 A digraph with two reaches. Nodes i and j are sources. Nodes a and b are contact points.



For a graph G=(V,E) the set $\{T_a | a \in V\}$ is a cover of V, that is $\bigcup_{a \in V} T_a = V$. It is an interesting question to find minimum subcovers of V, that is, sets $C \subseteq V$ such that $\bigcup_{a \in C} T_a = V$ and $|C|$, the size of C, is smallest.

A digraph is totally reachable, if for any pair of nodes a, b both $a \rightsquigarrow b$ and $b \rightsquigarrow a$ holds. The underlying graph of a totally reachable graph is connected. The other way round is not necessarily true.[8] In a totally connected digraph that is not a graph, the path from a to b may be different from the path from b to a.

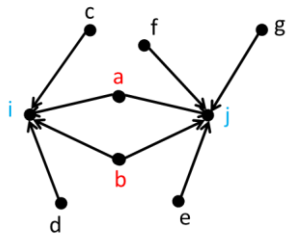
We can also consider reachability from a different perspective, and look at sets of points or arcs, that, once removed, produce a disconnected digraph. Instead of removing these points we can just mark them. The remaining points in the digraph may be subdivided into two groups. If a is a node in one group en b in the other, and if $a \rightsquigarrow b$ holds, then at least one of the marked points has to be on a path. So each of these points plays a role similar to that of a border crossing point: any path from a to b should contain at least one of these points. Put more picturesquely: one cannot go from one area to the other without crossing a border crossing point. Generally such border crossing points are called cut-points and together they form a cut set. There is an interesting relation between cut points and flows on digraphs. The maximum flow through such a digraph is equal to the flow through a cut-set with minimum capacity (hence for short: max flow = min cut).

Reachability in a digraph does not require knowledge of the entire digraph, because one only has to deal with outgoing arcs. So in RANs they can be computed, provided, of course, the reaches are not too big. The dual concept of attraction, which is considered in the next section, is much more elusive in a RAN, and requires knowledge of the entire digraph.

Attraction in a digraph

We can define a concept dual to reachability by reversing the arcs. We then obtain the concept of (what we have coined) 'attraction'. For a node a in G=(V,A) we define $T^a = \{b \in V | b \rightsquigarrow a\}$, that we shall call the basin of attraction[9] for the sink a.

3.2.3 Two basins of attraction with two sinks(i and j) and two points at the watershed (a and b).



[8] Take the digraph with two node a, b and only arc (a,b).
[9] Inspired by hydrology: areas that feed rivers with water.

A sink has only ingoing arcs, provided the basis of attraction has more than one node. Contact points only have outgoing arcs. Intermediate points (neither sinks nor contact points) have ingoing and outgoing arcs. They are not represented in Figure 3.2.3. But if arcs like (d,i) in fact represent a path from d to i, an intermediate point would be any point on the path, except d and i. In Figure 3.2.3 nodes a and b are part of two basins of attraction, and are therefore on the watershed of the two basins.

Although a basin of attraction is a dual concept of a reach, the reversing of the arcs has big consequences for a RAN. Whereas a reach is easy to compute (to a reasonable depth), similar computations from a basin of attraction (even at depth 1) is a formidable task in a massive network as it requires to explore this first. So the innocuous 'reversal of arcs' has major computational consequences for a large RAN.

Connectivity in a graph

For a graph one can define an equivalence relation on the set of nodes, as follows: two nodes a, b are related if there is a path in the graph connecting the two, denoted $a \sim b$. It is easy to verify that holds:

Reflexivity: $a \sim a$, for each node a,
Symmetry: $a \sim b$ implies $b \sim a$, for nodes a and b,
Transitivity: $a \sim b$ and $b \sim c$ implies $a \sim c$, for nodes a, b, c.

When applying this equivalence relation to a graph, the effect is that the set of nodes are partitioned into one or more nonempty sets. Each set is called a connectivity component of the graph. If a graph has only one connectivity component it is called connected. If it has more than one it is called disconnected, or simply not connected.

We can also look at connectivity in another way, namely in terms of separation. Instead of looking for paths connecting nodes in a graph, it is possible to consider removing points or edges and see if the resulting graph remains connected. If not, a major property has changed as a result of the transformation. It is natural to look at the minimum number of edges or points to be removed from a connected graph to produce a disconnected one. Such points form a minimum cut-set.

Measure for connectivity

The definition of connectivity is section 3.2 is rather strict. When a graph has two connectivity components, adding a single edge between the two component produces a connected graph. But there is only this single link. We want a more gradual concept of connectivity at the sub graph level. The number of links between two sub graphs should count. The more links there are between two sub graphs in a graph, the more connected they are (in the parent graph). Suppose that energy is associated with unbonding two nodes, i.e. removing an edge. The total unbonding energy is supposed to be equal to the sum of the unbonding energies associated with the individual connecting edges.

We can apply similar ideas to a graph and ask for the minimal number of links that have to be removed so that the resulting graph is disconnected. This minimum number is a measure for the connectedness of the graph.

Instead of looking at edges, we can look at points in a graph. For instance sets of points that when deleted from a graph (including all edges connected that are connected to them) produce a disconnected graph. Such sets are called cut sets. Particularly interesting are minimum cut sets, that is cut sets of minimum size.

Path length

For a path in a digraph or a graph we can define its path length: the number of arcs traversed. In that case all arcs have the same length. If this is not the case, we can use a nonnegative number that represents the distance between two nodes a, b connected by an arc $d_{a,b}$ and another one for the opposite arc: $d_{b,a}$.

In case we are dealing with a graph, we assume that if there is an edge in the graph connecting a and b, the distance between a and b along the path π connecting equals $d_{\{a,b\}} = d_{a,b} = d_{b,a}$.

Distance

Using path length, we can define a pseudo-distance or pseudo-metric d on a digraph G=(V,E), as follows. If a and b are nodes in G the pseudo-distance from a to b is the minimum length of all paths $\pi_{a,b}$ from a to b in G. If there is no such path, we say this pseudo-distance is infinite, ∞. Let $d(.,.)$ denote this pseudo-distance for G=(V,E) then we have $d(a,b) \geq 0$ for all $a, b \in V$. Also, it holds that $d(a,b) = 0$ implies $a = b$. Furthermore, $d(a,a) = 0$ for each node a in G, and $d(a,b) \leq d(a,c) + d(c,b)$ (triangle inequality). Because it is not guaranteed that $d(a,b) = d(b,a)$ for all $a, b \in V$, we are dealing with a pseudo-metric rather than a metric.

If the graph we consider is weighted (with nonnegative weights associated with each edge of the graph) we obtain a distance between two points in the graph by taking the sum of the weights of a path connecting these points, and looking for the minimum value of these sums.

Diameter

The distance between two nodes in an unweighted graph is the length of the shortest path connecting them. The length of such a patch is the minimum number of steps to go from one point to the other (or vice versa), where an edge corresponds to one step. This length is the distance between the two points. The diameter of a graph is the maximum distance of any two points in the graph. If the graph is connected, the diameter is finite. If it is disconnected, the diameter is formally infinite (∞).

In case the graph is weighted, then with each edge a nonnegative weight is associated. For such a weighted graph we can also define a distance between points, namely as the minimum sum of the weights of any path connecting these points. The weight of a path is the sum of the weights of the edges on the path. The diameter is likewise

defined in terms of weighted paths, instead of the case were all edges had weight equal to 1.

Transitive closure and transitive reduction

For a digraph G=(V,E) we can define the transitive closure G^*. This is a digraph with the same node set V as G does. The set of arcs of G, E, is a subset of the arcs of G^*. If a and b are nodes in G and $a \rightsquigarrow b$ then (a,b) is an arc in G^*, and vice versa.

The transitive reduction of G is a digraph G^r which has the same node set V as G does. Its arc set is a subset of the set of arcs of G, i.e. E. The transitive closure of G^r is the same as that of G, that is, G^*. Finally the arc set of G^r is minimal, in the sense that removal of one of its arcs results in a digraph whose transitive closure is not equal to G^*. Each digraph has a unique transitive reduction.

3.3 Spectrum of AA'

For each node visited during a search, we assume that the complete arc list is identified, at least at the time of observation. That is, the list of all arcs starting at that node. For the observed part of the network we have $\check{A}\check{A}'$ as a proxy to AA', with

$$\check{A}\check{A}' = A_{ee}A'_{ee} + A_{eo}A'_{eo}. \tag{3.2}$$

Here A_{ee} and A_{eo} are the only submatrices from A that we know, and they are the adjacency matrix for the explored nodes and the observed nodes in the network (see Section 2.5). The matrix $\check{A}\check{A}'$ is symmetric and nonnegative. Hence it has real, nonnegative eigenvalues. The question is: what we can learn from its spectrum (= the set of eigenvalues), in particular from its most important, that is, largest, eigenvalues? And what does this tell us about the RAN that we have only partially observed?

3.4 Node rank

In Brin and Page (1998) the concept of page rank is introduced. This is used by Google to rank the pages of the web according to their popularity, in terms of referencing. Derived from this we study here the idea of node ranking.[10] It seems the most direct way to do this is by way of an example.

The idea is that in a digraph G=(V,E) all the nodes carry a rank, which for each node is a value describing the popularity of the node. Popularity is defined by being referenced or being referred to. The more it is referred to, the more popular it is. But the ranks of the nodes that refer to a node also matter: a reference from a node with a high node rank counts for more than one from a node with a lower node rank. If a

[10] They use a subjective factor in their definition of page rank as well, that we discard in our example. It can be dealt with separately.

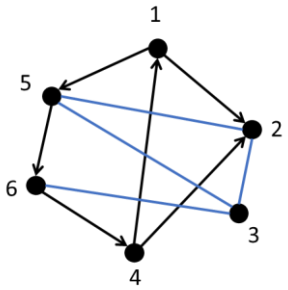
node has page rank r and n outgoing arcs, each arc carries a weight r/n. If we look at a node, we can calculate its node rank by adding all the weights carried by the arcs referring to it. So this description puts some constraints on the values a node rank can take for all the nodes in G. Of course, it does not define it uniquely: when we multiply a solution by a constant factor, we find another solution. We obtain a unique solution if we set a node rank for one of the nodes.

The following example shows that the node rank can be found as a solution to a set of linear equations. In fact, it can be shown that there is a natural link to the theory of Markov.

Example

In Figure 3.4.1 we have pictured the digraph that we consider in the present example. Arcs have been drawn (black). In case there are arcs between nodes in both directions we have drawn a line segment (without direction, and in blue). This is to make the picture look less cluttered. But as in a graph a line segment stands for two arcs, with opposite orientations.

3.4.1 Plot of the digraph in the example.



Consider the digraph G in Figure 3.4.1, which has the following adjacency matrix:

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{pmatrix}. \tag{3.3}$$

From A we compute M which is obtained from A by dividing the elements in each row by the corresponding row sum, which is the outdegree of the corresponding node:

$$M = \begin{pmatrix} 0 & 1/2 & 0 & 0 & 1/2 & 0 \\ 0 & 0 & 1/2 & 0 & 1/2 & 0 \\ 0 & 1/3 & 0 & 0 & 1/3 & 1/3 \\ 1/2 & 1/2 & 0 & 0 & 0 & 0 \\ 0 & 1/3 & 1/3 & 0 & 0 & 1/3 \\ 0 & 0 & 1/2 & 1/2 & 0 & 0 \end{pmatrix}. \tag{3.4}$$

Note that M is a stochastic (or Markov) matrix: its entries are nonnegative and for each row they add to 1. Note also that this property does not hold for the columns of M, so that M is not doubly stochastic.[11]

Let the node ranks be represented by a vector:

$$\rho = (\rho_1, \dots, \rho_6)'. \tag{3.5}$$

Next we compute the matrix R whose elements are the weights associated with the arcs of G. For a given node the weights associated with its outgoing arcs are equal to the node rank of this node divided by its outdegree. Then we have:

$$R = \begin{pmatrix} 0 & \rho_1/2 & 0 & 0 & \rho_1/2 & 0 \\ 0 & 0 & \rho_2/2 & 0 & \rho_2/2 & 0 \\ 0 & \rho_3/3 & 0 & 0 & \rho_3/3 & \rho_3/3 \\ \rho_4/2 & \rho_4/2 & 0 & 0 & 0 & 0 \\ 0 & \rho_5/3 & \rho_5/3 & 0 & 0 & \rho_5/3 \\ 0 & 0 & \rho_6/2 & \rho_6/2 & 0 & 0 \end{pmatrix}. \tag{3.6}$$

If we add these weights for all incoming arcs for each node they should be the same as the node rank of the arc they are pointing at. So we get the following vector equation:

$$\begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \\ \rho_5 \\ \rho_6 \end{pmatrix} = \begin{pmatrix} \rho_4/2 \\ \rho_1/2 + \rho_3/3 + \rho_4/2 + \rho_5/3 \\ \rho_2/2 + \rho_5/3 + \rho_6/2 \\ \rho_6/2 \\ \rho_1/2 + \rho_2/2 + \rho_3/3 \\ \rho_3/3 + \rho_5/3 \end{pmatrix}. \tag{3.7}$$

The right-hand side of (3.7) contains the column sums of the matrix R. We can rewrite this as:

$$\begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \\ \rho_5 \\ \rho_6 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1/2 & 0 & 0 \\ 1/2 & 0 & 1/3 & 1/2 & 1/3 & 0 \\ 0 & 1/2 & 0 & 0 & 1/3 & 1/2 \\ 0 & 0 & 0 & 0 & 0 & 1/2 \\ 1/2 & 1/2 & 1/3 & 0 & 0 & 0 \\ 0 & 0 & 1/3 & 0 & 1/3 & 0 \end{pmatrix} \begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \\ \rho_5 \\ \rho_6 \end{pmatrix}. \tag{3.8}$$

We can recognize the matrix in (3.8) as the transpose of M in (3.4). More succinctly we can write (3.8) as

$$\rho = M'\rho, \tag{3.9}$$

[11] In the particular case of the example. In general it may be doubly stochastic, by chance, but this is unlikely.

so that ρ is an eigenvector of M' for eigenvalue 1. In Markov chain theory ρ is called an invariant measure of M, with $\rho > 0$ and $\sum_{i=1}^{6} \rho_i = 1$. In general, a Markov chain that is irreducible, with only ergodic states, has a unique invariant measure ρ (cf. Feller, 1968, p. 393).

Solving (3.9) yields:

$$\begin{pmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \\ \rho_5 \\ \rho_6 \end{pmatrix} = \begin{pmatrix} 0.04081683 \\ 0.22448980 \\ 0.26785714 \\ 0.08163265 \\ 0.22193878 \\ 0.16326531 \end{pmatrix}. \tag{3.10}$$

■

Whatever the meaning of these rather technical conditions at the end of the example, the main message is that a unique invariant measure – and hence a node rank – does not always exist, but only under certain conditions. It is to be doubted if they apply to the Internet digraph.[12] But this is a fuzzy creature, constantly changing, impossible even to capture in a snapshot. This is a topic discussed in the next section.

4. Sampling RANs

Sampling a RAN is a quite different affair from sampling a full access network. There is no list of nodes or edges / arcs available. Access to a RAN is always through an entry point, and the access is only to nodes in the same connectivity component as the entry point.

When sampling a massive RAN, one should realize that it is impossible to access it in a split second. In fact, if the method presented in the present section is applied, nodes are accessed sequentially, and each visit to a node and processing information about the node takes time. In the meantime the network may change, depending on the dynamics of the network how much is changed. The point is that the measurements we intend to do on the network take time, and during this time a dynamic network will change. So instantaneous snapshots are impossible. One can only hope that the observation times are relatively short so that no dramatic changes have occurred in the meantime. On the other hand, it is unlikely that the global structure of the network dramatically changes in a split second. So capturing all micro-changes over the entire network elude us (at least at the time of writing). On the other hand, they do not define the big picture of a massive network. It is like a gas, in which it is impossible to measure the movement of every molecule it contains. But for the gas

[12] In Brin & Page (1998) these technical details are not discussed. This leaves the interesting question how the page rank is actually computed in practice.

considered as a macroscopic entity, such detail is not necessary, and not interesting. A few thermodynamic quantities, like temperature and pressure, suffice to characterize a gas. For massive networks we would like to find similar characteristics.

When sampling a RAN we assume that at any node we can identify all links to other nodes it contains when it is observed. Of course, we cannot guarantee that links will be deleted or added later, while the search process continues. So for each node visited we assume that we observe it completely. And in some cases the nodes referred to are similarly observed, in the sense that at the links it contains are copied. It is probably a good idea to timestamp nodes when visited. This information can be used to say something about the dynamics of a network. Information on nodes visited earlier (and still existing) can be compared with similar information obtained at a later visited. Differences give an idea of the changes of the network in the meantime.

4.1 Webpages or domains?

In case of the Internet it may be a bit confusing what a node is and what a link. A node is a web page, but it is identified by a link. A link on a webpage (or node) points to another webpage (or node). Besides, there is the problem of existence: a webpage may be created when accessed. Such pages with a fleeting existence should be excluded, for conceptual and for practical reasons. If included the web becomes overwhelmingly large, too large to cope with. A way to circumvent this problem is by considering to focus on domains and links between domains instead of on webpages with their hyperlinks.

4.2 Methods of access

First one has to find an entrance point. The visiting process starts from here. Below we describe several methods that can be used for traversing the network. The visiting methods only explore nodes in the same connectivity component as the entrance point. How to explore different connectivity components may be a difficult problem. One can try (in some way) to use different entry points (at random, say). But it is hard to decide if one is in the same or a different connectivity component.

4.3 Sample size

An important question is what the size of the sample of nodes should be so that we can make reliable inferences about the structure of the network. This question has obviously been considered in the literature on network sampling. See for instance Frank et al. (2012) and Leskovec & Faloutsos (2006).

4.4 Search strategies

The aim of these strategies to serve the purpose of the present paper is to collect useful information of the network being searched. As an ordinary sample should provide information on individuals in a population, a search through a (large) network should provide structural information about this network: about the distribution of degrees (outdegrees (easy) or indegrees (difficult)), the relation between indegrees and outdegrees, the hubs, the diameter, etc. Some of these strategies concentrate on finding the important parts of the digraph (nodes with a high indegree or with a high node rank). Others should give a picture of the less important points as well, if only to give a correct picture of the digraph. This information can be used to simulate networks with these characteristics, just to see if all relevant characteristics have been used.

Many network sampling techniques have been devised. We discuss only a few of them below. For more information see e.g. Ahmed et al. (2014), Al Hasan et al. (2014+), Chandrasekhar & Lewis (2016), Chierichetti et al. (2016), Das et al. (2008), Shou-De Lin et al. (2012+), which can all be found on the Internet.

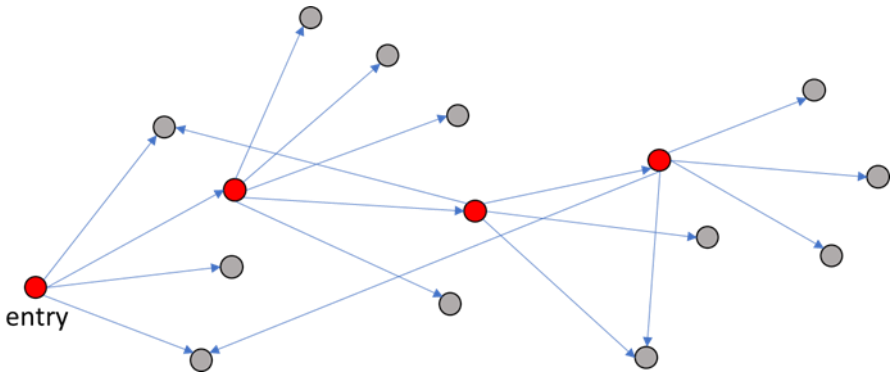
Finding an initial entry point

To start a search in a RAN we need one or several initial entry points. It is impossible to give a good procedure for this in a general case. But each time a search has been carried out the nodes on the search path can be used as initial entry points for new searches. But, obviously, in this way we only find entry points in the reach (or connectivity component) of the original starting point. So this method does not yield truly random initial entry points. For instance, if the RAN we are dealing with is the Internet we could use a search engine like Google to find entry points of the network. It could be fed with some key words randomly selected in a randomly chosen book. Among the entries generated there might be useful ones as initial entry points.

Random walk

This method, Method 0, is not for collecting information about the network. It is only used to penetrate quickly into the network, for instance to generate new entry points. In Figure 4.4.1, this method is illustrated by a picture of a small example. The red nodes are the nodes that are on the path that the method generates.

4.4.1 Random walk method applied.



The grey nodes are not kept. For each new red node the links from this node are collected, but only to randomly select a node to jump to. This selection can be done in one of several ways. The easiest one is to use with replacement sampling. In that case one may revisit a previously visited node. The advantage of this method is that the process will always continue. If one wants to avoid that one could use without probability sampling. Then it may be the case that one gets stuck: there are no unvisited nodes at the end of the links in the last node visited.

Note that the arcs in Figure 4.4.1 all start at a red node. Furthermore, the red nodes are the ones that are kept in a special list of visited nodes, which is updated each time a new red node has been determined.

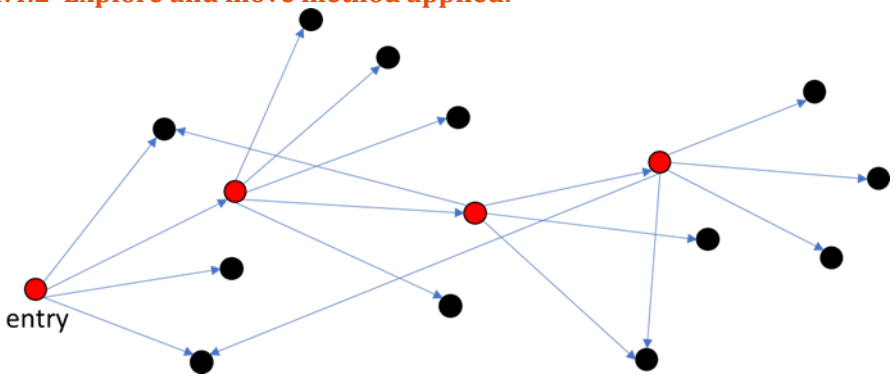
Explore and move

This method is intended to collect information about the nodes visited. These are the red nodes in Figure 4.4.2, which illustrates the method, which is as follows:

1. At a node, explore all the links emanating from there. Put them in a list, the explored nodes list (ENP).
2. Randomly select a link from the ones identified in step 1. Add this link to the selected links list (SLL). This list is used for checking.

In step 1 information on all nodes explored is kept. There is no checking if the node has been visited before. This will be done later. In step two it is possible to revisit a node. The probability that the same path is followed for a long time is probably negligible. And it makes the search process easier to implement and run. The process stops after a pre-specified number of steps, or before that, if it cannot be continued because it has come to a dead end (a node without an outgoing arc).

4.4.2 Explore and move method applied.



For this base method all sorts of variants are possible, in particular to cope with dead ends: the process is allowed to revisit an earlier visited node when it has come to a dead end. Such a previously visited node is chosen at random. Or, even without meeting a dead-end, it is decided by a separate random process to continue the process from a randomly chosen previously visited node.

Reach-search

We now want to consider a process where the randomness is basically only in the choice of the entry point to a RAN. Given such a point the reach of this point is explored systematically to some depth. This depth is chosen depending on the available computing resources.

The idea is to explore part of a reach of an initial entry point. As exploring the entire reach may be unfeasible in practice because it is too big, it may be possible to explore it to within a certain depth, that is, distance from the initial entry point. This maximum distance then is a parameter of the search method. It may actually be carried out by breadth-first search (BFS) (cf. Aho et al. (1983), Bang-Jensen & Gutin (2002), Gibbons (1985) or another book on algorithms, in particular (with sections) on graph algorithms).

BFS is a suitable method if the decision how deep a reach will be searched is taken during execution of the process. The methods explores a reach in layers. It starts with exploring the nodes directly linked to the entry point s. These are nodes in the first layer $N_1(s)$. Then the search proceeds to the points linked to the points in $N_2(s)$, consisting of the points at most two steps removed from s, or that are in $N_1(t)$ for $t \in N_1(s)$, etc. This procedure is repeated as long as one can. The more information can be gathered in this way the more is revealed of the RAN. This method, in a sense illuminates, the neighbourhood of an entry point up to a certain distance. We can liken such a neighbourhood with a ball of a certain diameter around the entry point s. One can take several random entry points and explore balls around them. It is clear that balls provide interesting information about the network, especially when they overlap.

Various variants of this method exist. For instance one can consider a randomized version, where a node is visited with a probability p. So if $p = 1$ we have the original method. Another option is to visit up to n neighbours, where n like p are parameters to be set by the analyst. Both methods are in fact variants of BFS, called forest fire and snowball sampling.

Taking care of degree bias

The methods discussed in the previous section favour nodes with a high indegree. For some purposes this is fine, as it gives a good impression of the most important parts of a digraph. But if we want to have a good impression of the entire network, including the (possibly many) nodes with a small indegree, these methods are not suitable, and others need to be devised for this purpose. The Metropolis-Hastings method is an example of such a method. We do not explain this algorithm here, as it is a bit involved. Instead we refer to Wikipedia (2017) or for a brief description Al Hasan et al. (2014). Roughly speaking one can say that the method is a version of a random walk method but one that also rejects sampled walks with a well-defined probability. The result is that the nodes in a graph have equal probability of being sampled.

Instead of modifying a random walk as in case of the Metropolis-Hastings case, another possibility is to use the data obtained by a sampling procedure that has ‘degree

bias', in the sense that nodes with a high indegree are more likely to be sampled. By using weighted estimation one can try to deal with this bias in the sample. This is quite common in ordinary sampling practice. One such method is proposed by Sagalnik & Heckathorn (2004), which is briefly discussed in Al Hasan et al. (2014).

5. Analysis of the sampling results

Extracting results from the sampled RANs seems to be the most difficult part of the endeavour to understand networks. The samples that one uses may be biased, and give a distorted picture of the entire network, although they may provide a good picture of certain elements, i.c. the hubs. Al Hasan (2014+) , Shou-De Lin et al. (2012+) are good starting points to get an idea of the issues one has to deal with. They contain pointers to the literature as well, in case one wants to delve more deeply into a problem. Of course, a large part of the literature on network sampling is about analysis problems.

Without specific examples, discussing the analysis of the network sampling results is like 'dry swimming', so we abstain from it here. We just make some general remarks, and imagine that some random procedure has been used to extract information from a RAN. Take for instance the explore and move method, discussed in Section 4.4.2. The question is, what can be done with such data, without using any sophisticated analysis techniques? The difficulty resides in assigning the suitable weights to the nodes or edges observed. But even without this, one can still chart the results obtained.

First of all one can find out which nodes are in the sample. This may require some deduplication of the nodes explored or observed. The situation is summarized in Table 5.1.1, where an adjacency matrix on the basis of the explore and move procedure has been drafted. The set of nodes is divided into three groups: the explored nodes, the observed nodes and the remaining (unobserved) nodes of the RAN. See section 2.5.

5.1.1 Adjacency matrix after an explore and move search.

	Explored	Observed	Remainder
Explored	A_{ee}	A_{eo}	$\ominus$

The entries in Table 5.1.1 are three matrices, A_{ee} which indicates which explored nodes pont to other similar nodes, A_{eo} which indicates which explored nodes point to merely observed (but not explored node), and finally $\ominus$ which symbolizes a zero matrix. For the explored nodes we have total knowledge to which nodes they point. This information is in A_{ee} and A_{eo}. For the explored nodes we know that they do not point to any other node in the RAN. By adding the results of Figure 5.1.1 row-wise, we obtain the outdegrees of the explored nodes. The distribution of the outdegrees

is a first idea of the outdegrees for the entire RAN. But if the explored nodes are not a random sample of the nodes, this picture may be biased. One can also compute the column sums (only for the explored and observed node), and determine their distribution. They give a first glimpse of the distribution of the indegrees of the entire network. Again this may be biased, and ready for revision, using appropriate weights reflecting the sampling probability. Also, one gets a picture of the relation between indegrees and outdegrees per node, which is input for the correlation between both quantities in the RAN. For other sampling techniques the information on indegrees and outdegrees may be less precise.

Also, if the set of explored nodes is sizable compared to the total number of nodes in the RAN, the adjacency matrix A_{ee} represents a digraph that resembles the entire adjacency matrix A quite well. So one can study the properties of A_{ee} and get a picture of the RAN.

6. Simulating RANs

A network may be too big to study conveniently, in particular to run a computationally intensive algorithm. In such a case it would be convenient to use a scaled down version of this large network instead. But to be able to do this meaningfully certain characteristics of the big network should be mimicked by the scale model. A sample from the large network should provide the material to compute these characteristics from.

For simulation studies it is important that networks with given characteristics can be generated. Such a network may then be studied and it may turn out that it misses certain characteristics that can be observed in 'real' networks. One then may try to formulate the missing characteristics and then try to simulate a network that satisfies the extended set of characteristics.

It may perhaps be thought that random digraphs in practice are generated by starting with a set of nodes and for each pair of nodes i, j decide with probability p if there will be an arc (i,j) or not, with probability $q = 1 - p$. Such a model was proposed by Erdös and Renyi (1959). Although such random networks are interesting in their own right, they do not look like many of the networks one encounters in practice. The Erdös-Rényi-networks tend to be too homogeneous in terms of the distribution of indegrees and outdegrees. In practice one often encounters networks with a small number of hubs and a large number of nodes of low indegree.

To remedy this defect Barabási proposed another model to generates digraphs – in fact directed trees (ditrees) – with a few hubs and a plethora of nodes with small indegree. The model builds the ditree in steps. It starts with a single node. At a given stage it contains n nodes. Each of these nodes have been generated before and each

has an indegree. Now a new node is added to the tree obtained so far and one of the n existing nodes is selected with a probability proportional to its current indegree.

It is somewhat unsatisfactory that the Barabási model generates ditrees and not more general digraphs. However, it is easy to find all kinds of variants of this model that yield more general digraphs and not only ditrees. For instance, one variant is to connect a new node to each of the previously generated nodes i with a probability proportional to the current indegree of i

$$p_i = \begin{cases} 1 & \text{with probability} \quad \pi_{in,i}/\pi_{in}, \\ 0 & \text{with probability} \quad 1 - \pi_{in,i}/\pi_{in} \end{cases} \tag{6.1}$$

where $\pi_{in} = \sum_i \pi_{in,i}$, the sum over the indegrees of the nodes existing then. Also the direction of each new arc (pointing to or from the new node) is determined by flipping a coin. Then the new node is added to the list of existing nodes and the indegrees for each node is updated. The situation that model (6.1) yields a new node that is not connected to a previously generated node.

One can extend the model by using a probability $p_0 > 0$ that the new node is connected to no previously generated node. So after the process just described has been carried out (except the updating step) one decides to keep the newly generated arcs (with probability p_0) or to discard them all (with probability $1 - p_0$). In this way one can control the number of connectivity components (of the underlying graph) in the resulting digraph.

It is not so difficult to extend the Barabási-model so that one can control the distribution of indegrees and outdegrees per node. We shall not delve into this matter here. We only make the remark to indicate that the iterative way to generate a network with predetermined specifications is attractive and versatile. This is in contrast with the Erdös-Rényi-model, that produces a network in one go so to speak (for each pair of nodes one could simultaneously generate a random number, independently of each other, and use this to decide if they will be connected by an edge or not).

We finish this section by discussing a problem that may arise when simulating a scaled down version of a RAN. This concerns the generation of an adjacency matrix with given indegrees and outdegrees per node. In Figure 6.1.1 an example of such a situation (in miniature) is presented. A requirement is that the sum of the indegrees should be equal to the the sum of the outdegrees.

6.1.1 Unknown adjacency matrix with given indegrees and outdegrees per node.

	A	B	C	D	E	F	G	H		outdegree
A										6
B										4
C										3
D										3
E										2
F										5
G										1
H										0
indegree	4	2	2	1	6	5	3	1		24

In Figure 6.1.2 a solution to this problem is shown. To arrive at it by hand is a bit of a puzzle.

6.1.2 Adjacency matrix that satisfies the constraints of Figure 6.1.1.

	A	B	C	D	E	F	G	H		outdegree
A	0	1	1	0	1	1	1	1		6
B	1	0	0	0	1	1	1	0		4
C	1	0	0	0	1	1	0	0		3
D	1	0	0	0	1	1	0	0		3
E	0	0	0	0	0	1	1	0		2
F	1	1	1	1	1	0	0	0		5
G	0	0	0	0	1	0	0	0		1
H	0	0	0	0	0	0	0	0		0
indegree	4	2	2	1	6	5	3	1		24

To determine the general solutions for such problems, let each cell of the adjacency matrix be represented by a variable that is either 0 or 1. The problem implies that these variables should satisfy 16 linear constraints, which correspond to the row-wise and column-wise sums. We then can specify a linear function in the variable and try to solve that with the given constraints. This requires solving an integer programming problem, or to be more precise, a binary (Boolean) value program.

In general there will be many solutions. It would be nice if a sample of them could be generated so that it would be possible to study the different digraphs that result, to investigate how they differ in the aspects we are interested in.

7. Discussion

The present paper tried to present some ideas with respect to sampling RANs. Originally the plan was to include simulation results in the paper, so as to illustrate some ideas numerically. But this proved to be impossible. This is postponed until sufficient capacity is available for this kind of work. But the result is a somewhat unsatisfying paper, as a major part is lacking. What remains are some ideas that need to be explored, as well as an overview of some of the concepts that are important in the area of network science (at least in the understanding of the present author).

While writing the present paper, the author realized that it is a particularly interesting and challenging problem to identify those characteristics of RANs that are good input to calculate faithful scaled down versions of RANs that can be huge. This is typical for any statistical problem, not only networks: how to characterize the population one is dealing with on the basis of a limited amount of parameters? Of course, the answer can only be given in the light of the particular application one is interested in.

It is felt that books like Barabási (no date) and Newman et al. (2006) should be studied in order to be well-prepared for contributions in network sampling. They also are useful sources for other aspects of network science. Newman et al. (2006) is an anthology of important papers that have been published before in several journals, supplemented with comments and introductions by the editors. This book is now more than 10 years old, but still a useful source to get a good grasp of the area. The Barabási book is of more recent origin, but also less extensive and deep. A good quick introduction to the entire area of network science.

The references given are interesting for getting an understanding of network sampling. They represent just a sample of the literature on this subject. They were easy to find on the Internet. There is no claim that the references given are the best that are available. But they should be useful to get a better understanding of the problems involved in network sampling.

References

Ahmed, N.K., J. Neville & R. Kompella (2014). Network sampling: from static to streaming graphs, ACM Transactions on Knowledge Discovery, Vol. 8, No 2, Article 7. (paper available on the Internet)

Aho, A.V., J.E. Hopcroft & J.D. Ullman (1983). Data structures and algorithms, Addison-Wesley.

Al Hasan, M., N.K. Ahmed & J. Neville, (2014+). Methods and applications of network sampling, Power point presentation. (available on the Internet)

Bang-Jensen, J. & G. Gutin (2002). Digraphs – Theory, algorithms and applications, Springer.

Barabási, A.-L., Network Science (available on barabasi.com/networksciencebook/).

Brin, S. & L. Page (1998). The anatomy of a large-scale hypertextual web search engine, Report, Computer Science Department, Stanford University, Stanford, Cal., USA. Presented at WWW 1998, April 14-18, 1998 in Brisbane, Australia. (paper available on the Internet)

Chandrasekhar, A.G. & R. Lewis (2016). Econometrics of sampled networks, Report, Department of Economics, Stanford University and NBER. (paper available on the Internet)

Chierichetti, F., A. Dasgupta, R. Kumar, S. Lattanzi & T. Sarlós (2016). On sampling nodes in a network. Paper presented at WWW 2016, April 11-15, 2016, Montréal, Québec, Canada. (paper available on the Internet)

Das, G., N. Koudas, M. Papagelis & S. Puttaswamy (2008). Efficient sampling of information in social networks, CIKM/SSM 2008, Nappa Valley, Cal., USA. (paper available on the Internet)

Erdös, P. & A. Rényi (1959). On random graphs, Publicationes Mathematicae, 6, pp. 290-297.

Feller, W. (1968). An introduction to probability theory and its applications, Vol. 1, Wiley.

Frank, D., Z. Huang & A. Chyan (2012). Sampling a large network: how small can my sample be? (paper available on the Internet)

Gibbons, A. (1985). Algorithmic graph theory, Cambridge University Press.

Leskovec, J. & C. Faloutsos (2006). Sampling from large graphs. ACM SIGKDD 2006.

Newman, M., A.-L. Barabási & D.J. Watts (2006). The structure and dynamics of networks, Princeton University Press.

Salganik, M.J. & D. D. Heckathorn (2004). Sampling and estimation in hidden populations using respondent-driven sampling, Sociological Methodology, Vol. 34. (2004), pp. 193-239.

Shou-De Lin, Mi-Yen Yeh & Cheng-Te Li (2012+). Sampling and summarization of social networks, Power point presentation (82 pages).

Wikipedia (2017). Article on the Metropolis-Hastings algorithm, https://en.wikipedia.org/wiki/Metropolis%E2%80%93Hastings_algorithm

Appendix A. Some network terminology

This appendix explains some of the graph theoretical terms and concepts used in the present paper. Some of it concerns standard concepts, some concepts are particular for the present paper. In the latter case, no attempt has been made to find out if a term already had been introduced by someone in the network literature for the same notion (that would be too much work). However, known this terminology was adopted.

Concept	Explanation
Acyclic graph	A graph without a cycle. A tree.
Adjacency matrix	0-1 matrix where element (i,j) indicated if there is an arc from i to j (if the value = 1) or not (if the value = 0). An adjacency matrix can be viewed as a representation of a (di)graph. In case of a graph it is symmetric. In case of a digraph it need not be symmetric.
Arc	An ordered pair of nodes (a,b). In a picture an arc (a,b) is denoted by an arrow pointing from a to b. The node a is called the start and b is called the finish of the arc (a,b).
Boundary point / node	In a reach this is a point with only ingoing arcs, that is a sink. In a drainage basin it is a point / node with only outgoing arcs, that is, a source.
Clique	A complete subgraph.
Complete graph	A graph where each pair of nodes is an edge.
Connected graph	A graph for which each pair of points can be connected by a path.
Contact point / node (in a digraph)	A point / node in a watershed.
Counter-arc	If (a,b) is an arc its counter-arc is (b,a), provided it is part of the digraph. Otherwise the arc has no counter-arc.
Cut set	The set of points in a network whose removal (including the arcs of edges incident to these points) results disconnecting the network (or component involved).
Cut-point	A cut-set of size 1. If a cut-point is deleted from a graph (that is, including all the edges attached to it) then the resulting graph is disconnected.
Cycle	A closed path. A path where all nodes are of degree 2. See also: path.
Cycle matrix	A (-1,0,1)-matrix where the rows correspond to the cycles in a graph. The columns correspond to the arcs

	of a graph or digraph. A 0 indicates that the corresponding arc is not on a cycle, a 1 that it is on a cycle, and a -1 that the opposite (or counter-) arc is.
Degree (of a node)	In a graph the number of edges attached to the node.
DFAN	Dynamic full access network, which is a FAN whose structure changes over time.
Digraph	Directed graph.
Directed edge	See: arc.
Directed graph	Consists of nodes and arcs. An arc is an ordered pair of nodes. If (a,b) is an arc node a is connected to node b, but not necessarily the other way round, unless (b,a) is also an arc.
Directed network	See: Directed graph.
Directed tree	A digraph where the underlying graph is a tree. But extra conditions may be desirable. See the discussion in Section 3.2 under the heading 'Trees and ditrees'.
Ditree	See: Directed tree.
Drainage basin (of a digraph)	The points in a digraph from which it is possible to reach to reach a particular point, a, the point of attraction of the drainage basin.
DRAN	Dynamic restricted access network. A RAN that is not static, where nodes and arcs / edges are added or removed over time, and where labels may be changed.
Dual concept	A concept defined on a digraph that emerges when the arcs are reversed. So a reach is and drainage basin are duals concepts.
Edge	In a graph an edge is an unordered pair of nodes {a,b}. In a picture an edge is represented by a line segment or arc without arrow heads (as there is no direction). Viewed in a digraph context an edge {a,b} is represen-ted by the arcs (a,b) and (b,a).
Edge list	A list structure containing all edges of a graph. A node list together with an edge list forms a representation of a graph.
Entry point / node	Node of a RAN where a search starts.
Ergodic state of a Markov chain	See Feller (1968, p. 389).
FAN	Full access network. A network where all the nodes and edges / arcs are known. A FAN allows random access to each node or arc.
Flow (in a digraph)	An abstraction of a transport system in which goods and/or vehicles are moved between locations connected by roads, waterways, etc.. It can be represented by a digraph. For instance the nodes are factories or warehouses. Arcs only connect factories and warehouses. With each arc a flow is associated indicating how much of a good is transported (say

	loaves of bread) from the factory to the warehouse in a given time period.
Graph	An example of a network. Or is special kind of digraph, where each arc (a,b) has its counter-arc (b,a).
Hub	A node in a digraph with a relatively large number of ingoing arcs.
Incidence matrix	A 0-1-matrix where the columns correspond to the arcs / edges and the rows to the nodes. The element (e,a) indicates if node a is incident with arc / edge e. If e= {a,b} is an edge nodes a, and b are incident with edge e. Similarly for arc e = (a,b).
Indegree	In a digraph the number of arcs ending at a given node and starting from other nodes in the digraph.
Irreducible Markov chain	See Feller (1968, pp.390 ff.).
Link	Either an edge in a graph or an arc in a digraph.
Loop	For a node v an edge of the type {v,v} and in a digraph an arc of the type (v,v). In a picture a loop is represented by a curve attached to a dot representing a node.
Markov chain	See Feller (1968, pp. 372 ff.).
Network	A generic term used to denote either graphs or digraphs, when it does not make sense to emphasize the (non)orientation of the edges.
Node	Point in a network.
Node rank	A system that ranks each nodes on the basis of the ranks of the nodes pointing to it.
Orientation	An edge {a,b}, were a and b are nodes, can be oriented in two ways, denoted by (a,b) and (b,a). Here (a,b) denotes the arc pointing from a to b. The second one is an arc pointing from b to a. Both are orientations of the original edge, and the orientations are opposite. An arc is represented by an arrow, pointing from the start to the end node.
Outdegree	In a digraph the number of arcs starting at a given node and pointing to other nodes in the digraph.
Page rank	A method devised by the founders of Google, to rank webpages on the basis of the page ranks of the webpages pointing to it. So a web page gets a higher rank when not only more webpages point to it, but more prestigious (in terms of page rank) web pages as well.
Passage point / node	A point / node with both ingoing and outgoing arcs.
Path	A path in a digraph G=(V,E) from $a \in V$ to $b \in V$ is a function $p: \{1, \dots, k\} \to V$, with $k \geq 2$ such that $p(1) = a$, $p(k) = b$ and $(p(i), p(i+1)) \in E$ for each $i = 1, \dots, k-1$. $k-1$ is the path length. The path p connects a to b. By definition, a node is also a path, of length 0. If a = b, the path is a cycle.
PERAN	Partially explored RAN. A RAN where some structural

	information is known as a result of a search.
Point	One of the ingredients of networks. The other ingredients are the arcs (in a digraph) or edges (in a graph).
RAN	Restricted access network. A network that only can be explored from choosing entry nodes, and jumping from node to node. A RAN does not allow random access to nodes or edges or arcs.
Reach	The nodes in a digraph that can be reached from a particular node s, the source of the reach.
SFAN	Static full access network. A FAN with a structure that does not change over time.
Sink (of a digraph)	A node with only ingoing arcs.
Sink of a drainage basin	The unique node in a drainage basin that only has ingoing arcs.
Source (of a digraph)	A node with only outgoing arcs.
Source (of a reach)	The unique node in a reach that only has outgoing arcs.
Spanning tree (of a graph)	A subgraph of a given graph with the same set of nodes as the graph and with a subset of the edges of the graph, which is a tree. The graph is obtained by adding extra edges (provided it is not a tree already).
SRAN	Static restricted access network. A RAN with a structure that does not change over time.
Structure (of a network)	The structure of a network is defined by the set of nodes and the set of edges or arcs. The structure changes if the set of nodes changes (new nodes are added or existing nodes are deleted) or the set of edges or arcs changes (new edges or arcs are added or existing ones are deleted).
Super sink (of a digraph)	A sink much bigger in size than the majority of sinks.
Tree	A connected graph without cycles.
Underlying graph of a digraph	The graph with the same nodes as the digraph and with {a,b} an edge if (a,b) or (b,a) is an arc in the digraph.
Watershed (in a digraph)	The set of nodes in a digraph belonging to two drainage basins.

Explanation of symbols

Empty cell	Figure not applicable
.	Figure is unknown, insufficiently reliable or confidential
*	Provisional figure
**	Revised provisional figure
2014–2015	2014 to 2015 inclusive
2014/2015	Average for 2014 to 2015 inclusive
2014/'15	Crop year, financial year, school year, etc., beginning in 2014 and ending in 2015
2012/'13–2014/'15	Crop year, financial year, etc., 2012/'13 to 2014/'15 inclusive

Due to rounding, some totals may not correspond to the sum of the separate figures.

Colofon

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

Prepress
Statistics Netherlands, Studio BCO

Design
Edenspiekermann

Information
Telephone +31 88 570 70 70, fax +31 70 337 59 94
Via contactform: www.cbsl.nl/information