

Networks: An Introduction Research Notes

W. J. Cunningham

February 28, 2014

Chapter 6

Mathematics of Networks

6.1 Networks and Their Representation

Terminology:

- Network/Graph
- Vertex/Node/Site/Actor
- Edge/Link/Bond/Tie

The number of vertices is denoted by n and the number of edges by m .

6.2 The Adjacency Matrix

If we denote an edge between vertices i and j by (i, j) then the complete network can be specified by giving the value of n and a list of all the edges. Such a specification is called an *edge list*. A better representation of a network is the *adjacency matrix*. The diagonal elements are all zero and it is symmetric for simple undirected graphs. The elements are defined such that

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge between vertices } i \text{ and } j, \\ 0 & \text{otherwise.} \end{cases} \quad (6.1)$$

6.3 Weighted Networks

In some situations it is useful to represent edges as having a strength, weight, or value. Such *weighted networks* can be represented by giving the elements of the adjacency matrix values equal to the weights of the corresponding connections. Vertices can also have weights.

6.4 Directed Networks

A *directed network* is a network in which each edge has a direction, pointing *from* one vertex *to* another. Such edges are called *directed edges*. The adjacency matrix of a directed network has the matrix elements

$$A_{ij} = \begin{cases} 1 & \text{if there is an edge from } j \text{ to } i, \\ 0 & \text{otherwise.} \end{cases} \quad (6.2)$$

Note that the direction indicates the edge runs *from* the second index *to* the first.

We can think of undirected networks as directed networks in which each undirected edge has been replaced with two directed ones running in opposite directions between the same pair of vertices. The adjacency matrix for such a network is then symmetric and exactly the same as for the original undirected network.

6.4.1 Cocitation and Bibliographic Coupling

A directed network may be transformed into an undirected one in one of the following ways:

1. Ignore the edge directions entirely
2. Use *cocitation* (do this one!)

The *cocitation* of two vertices i and j in a directed network is the number of vertices that have outgoing edges pointing to both i and j :

$$C_{ij} = \sum_{k=1}^n A_{ik} A_{jk} = \sum_{k=1}^n A_{ik} A_{kj}^T \quad (6.3)$$

We can also define the (symmetric) *cocitation matrix* \mathbf{C} to be the $n \times n$ matrix with elements C_{ij} :

$$\mathbf{C} = \mathbf{A} \mathbf{A}^T \quad (6.4)$$

This allows for us to build a *cocitation network*. This network conventionally has no self-edges. We can make it a weighted network by identifying vertex pairs cited by more common neighbors as having a stronger connection than those cited by fewer. This matrix plays a similar role to the adjacency matrix for this network.

The *bibliographic coupling* of two vertices in a directed network is the number of other vertices to which both point. The bibliographic coupling of i and j is given by

$$B_{ij} = \sum_{k=1}^n A_{ki} A_{kj} = \sum_{k=1}^n A_{ik}^T A_{kj}, \quad (6.5)$$

and we define the (symmetric) *bibliographic coupling matrix* \mathbf{B} to be the $n \times n$ matrix with elements B_{ij} so that

$$\mathbf{B} = \mathbf{A}^T \mathbf{A} \quad (6.6)$$

In general, bibliographic coupling is a more uniform indicator of similarity between papers than cocitation. It also does not change over time, so it is a static variable.

6.4.2 Acyclic Directed Networks

A *cycle* in a directed network is a closed loop of edges with the arrows on each of the edges pointing the same way around the loop. Networks which have no cycles are called *acyclic* networks, and ones with cycles are called *cyclic*. **Causal networks are necessarily acyclic.**

In acyclic networks there must be at least one vertex somewhere on the network that has ingoing edges only and no outgoing ones. A simple algorithm for determining whether a network is acyclic is:

1. Find a vertex with no outgoing edges.
2. If no such vertex exists, the network is *cyclic*. Otherwise, if such a vertex does exist, remove it and all its ingoing edges from the network.
3. If all vertices have been removed, the network is *acyclic*. Otherwise go back to step 1.

In the case that the network is acyclic, the adjacency matrix is an upper triangular matrix. For every acyclic directed network there exists at least one labeling of the vertices such that the adjacency matrix will be strictly upper triangular. Further, all of its eigenvalues will be zero, so it will be a *nilpotent matrix*.

6.5 Hypergraphs

These networks have links which join more than two vertices at a time (probably not useful for causal networks).

6.6 Bipartite Networks

These networks have two kinds of vertices (probably not useful for causal networks).

6.7 Trees

A *tree* is a connected, undirected network which contains no closed loops. If all parts of a network are trees, the complete network is called a *forest*. Topologically, a tree has no particular root, but in some applications there are other reasons for designating a root. In trees, there is exactly one path between any pair of nodes, and a tree of n vertices always has $n - 1$ edges.

6.8 Planar Networks

A *planar network* is a network that can be drawn on a plane without having any edges cross. Any network that contains a subset or expansion of vertices (or subgraph) in the form of K_5 or UG is non-planar. The converse is true by *Kuratowski's Theorem*. Real-world networks are rarely precisely planar. No widely accepted metric for degree of planarity has emerged.

6.9 Degree

The *degree* of a vertex in a graph is the number of edges connected to it. For an undirected graph of n vertices the degree can be written in terms of the adjacency matrix as

$$k_i = \sum_{j=1}^n A_{ij} \quad (6.7)$$

The number of edges is equal to the sum of the degrees of all the vertices, so

$$2m = \sum_{i=1}^n k_i \quad (6.8)$$

The mean degree c of a vertex in an undirected graph is

$$c = \frac{1}{n} \sum_{i=1}^n k_i = \frac{2m}{n} \quad (6.9)$$

The maximum possible number of edges in a simple graph is $\binom{n}{2} = \frac{1}{2}n(n-1)$. The *connectance* or *density* ρ of the graph is the fraction of edges that are actually present.

$$\rho = \frac{m}{\binom{n}{2}} = \frac{2m}{n(n-1)} = \frac{c}{n-1} \approx \frac{c}{n} \quad (6.10)$$

A network for which the density ρ tends to a constant as $n \rightarrow \infty$ is said to be *dense*. A network for which $\rho \rightarrow 0$ as $n \rightarrow \infty$ is said to be *sparse*. In particular, a network is sparse if c tends to a constant as n becomes large. **Causal networks are sparse networks.**

In directed networks each vertex has two degrees. The *in-degree* is the number of ingoing edges connected to a vertex and the *out-degree* is the number of outgoing edges. If there is an edge from j to i , these quantities can be written

$$k_i^{\text{in}} = \sum_{j=1}^n A_{ij}, \quad (6.11)$$

$$k_j^{\text{out}} = \sum_{i=1}^n A_{ij}. \quad (6.12)$$

The mean in-degree c_{in} and the mean out-degree c_{out} of every directed network are equal:

$$c_{\text{in}} = \frac{1}{n} \sum_{i=1}^n k_i^{\text{in}} = \frac{1}{n} \sum_{j=1}^n k_j^{\text{out}} = c_{\text{out}} \quad (6.13)$$

We can denote both by the variable c so that

$$c = \frac{m}{n} \quad (6.14)$$

6.10 Paths

A *path* is any sequence of vertices such that every consecutive pair of vertices in the sequence is connected by an edge in the network. *Self-avoiding paths* are more important - Hamiltonian and geodesic paths are two special cases. The *length* of a path in a network is the number of edges traversed along the path. The number of paths of length r from vertex j to vertex i is given by

$$N_{ij}^{(r)} = [\mathbf{A}^r]_{ij} \quad (6.15)$$

and the number of loops of length r is given by

$$\begin{aligned} L_r &= \sum_{i=1}^n [\mathbf{A}^r]_{ii} \\ &= \text{Tr} \mathbf{A}^r \\ &= \sum_i \kappa_i^r \end{aligned} \quad (6.16)$$

where κ_i are the eigenvalues of the adjacency matrix. Note that for a directed graph, the adjacency matrix is asymmetric and therefore may have complex eigenvalues.

6.10.1 Geodesic Paths

A *geodesic path* is a path between two vertices such that no shorter path exists. These paths are necessarily self-avoiding, though they are not necessarily unique. The *diameter* of a graph is the length of the longest geodesic path between any pair of vertices in the network for which a path actually exists.

6.10.2 Eulerian and Hamiltonian Paths

An *Eulerian path* is a path which traverses each edge in a network exactly once. A *Hamiltonian path* is a path that visits each vertex only once.

6.11 Components

The disconnected subgroups of a network are called *components*. The adjacency matrix of a network with more than one component can be written in block diagonal form.

6.12 Independent Paths, Connectivity, and Cut Sets

The number of independent paths between given vertices gives a simple measure of how strongly the vertices are connected to one another. The number of independent paths cannot exceed the number of edges in the network. The number of independent paths between a pair of vertices is called the *connectivity* of the vertices. A *cut set* is a set of vertices whose removal will disconnect a specified pair of vertices. A *minimum cut set* is the smallest cut set that will disconnect a specified pair of vertices.

Menger's Theorem states that if there is not cut set of size less than n between a given pair of vertices, then there are at least n independent paths between the same vertices. It implies that the size of the minimum vertex cut set that disconnects a given pair of vertices in a network is equal to the vertex connectivity of the same vertices.

6.13 The Graph Laplacian

6.13.1 Diffusion

The diffusion equation is

$$\frac{d\psi_i}{dt} = C \sum_j A_{ij} (\psi_j - \psi_i) \quad (6.17)$$

We can now define the *graph Laplacian* as

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \quad (6.18)$$

where \mathbf{D} is the diagonal matrix of vertex degrees and \mathbf{A} is the adjacency matrix. The elements of this matrix are

$$L_{ij} = \begin{cases} k_i & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and there is an edge } (i, j) \\ 0 & \text{otherwise} \end{cases} \quad (6.19)$$

This becomes an eigenvector equation $\mathbf{L}\mathbf{v}_i = \lambda_i \mathbf{v}_i$, with the solution

$$a_i(t) = a_i(0)e^{-C\lambda_i t} \quad (6.20)$$

where C is the *diffusion constant*.

6.13.2 Eigenvalues of the Graph Laplacian