

Spectral methods for analyzing and visualizing networks: an introduction

Andrew J. Seary and William .D. Richards
School of Communication
Simon Fraser University
Burnaby BC Canada V5A-1S6
seary@sfu.ca richards@sfu.ca

Abstract

Network analysis begins with data that describes the set of relationships among the members of a system. The goal of analysis is to obtain from the low-level relational data a higher-level description of the structure of the system which identifies various kinds of patterns in the set of relationships. These patterns will be based on the way individuals are related to other individuals in the network. Some approaches to network analysis look for clusters of individuals who are tightly connected to one another; some look for sets of individuals who have similar patterns of relations to the rest of the network. Other methods don't "look for" anything in particular — instead, they construct a continuous multidimensional representation of the network in which the coordinates of the individuals can be further analyzed to obtain a variety of kinds of information about them and their relation to the rest of the network.

One approach to this is to choose a set of axes in the multidimensional space occupied by the network and *rotate* them so that the first axis points in the direction of the greatest variability in the data; the second axis, orthogonal to the first, points in the direction of greatest remaining variability, and so on. This set of axes is a coordinate system that can be used to describe the relative positions of the set of points in the data. Most of the variability in the locations of points will be accounted for by the first few dimensions of this coordinate system. The coordinates of the points along each axis will be an *eigenvector*, and the length of the projection will be an *eigenvalue*. The set of all eigenvalues is the *spectrum* of the network.

Spectral methods (eigendecomposition) have been a part of graph theory for over a century. Network researchers have used spectral methods either implicitly or explicitly since the late 1960's, when computers became generally accessible in most universities. The eigenvalues of a network are intimately connected to important topological features such as maximum distance across the network (diameter), presence of cohesive clusters, long paths and bottlenecks, and how random the network is. The associated eigenvectors can be used as a natural coordinate system for graph visualization; they also provide methods for discovering clusters and other local features. When combined with other, easily obtained network statistics (e.g., node degree), they can be used to describe a variety of network properties, such as degree of robustness (i.e., tolerance to removal of selected nodes or links), and other structural properties, and the relationship of these properties to node or link attributes in large, complex, multivariate networks.

We introduce three types of spectral analysis for graphs and describe some of their mathematical properties. We discuss the strengths and weaknesses of each type and show how they can be used to understand network structure. These discussions are accompanied by interactive graphical displays of small ($n=50$) and moderately large ($n=5000$) networks. Throughout, we give special attention to sparse matrix methods which allow rapid, efficient storage and analysis of large networks. We briefly describe algorithms and analytic strategies that allow spectral analysis and identification of clusters in very large networks ($n>1,000,000$).

Introduction

A standard method in statistics for handling multivariate data is to find the directions of maximum variability, usually of variance-covariance or correlation matrices. These directions are called Principal Coordinates or *eigenvectors*, while the relative importance of each direction is represented by numbers called *eigenvalues*. (Jolliffe, 1986) Finding this coordinate system may be accomplished by a series of *rotations* (although this is not the most efficient method) that end up pointing along the direction of maximum variability, with the second largest maximum variability at right angles, and so on. As a result, the data matrix is reduced to a diagonal matrix, with diagonal entries corresponding to the *importance (eigenvalue)* of each direction (*eigenvector*). The collection of *all eigenvalues* is called the *spectrum*. One goal is to reduce the problem so that only the most important dimensions (those with the largest eigenvalues) contain most of the variability. Implicit in these methods (variance-covariance or correlation) is that some kind of “expected” or “background” signal has been subtracted: in the case of variances, these would be the means of each variable in the original data matrix. To find these *eigenvectors* and *eigenvalues* we need to solve the *eigenvalue equation*:

$$\mathbf{E} \mathbf{e} = \epsilon \mathbf{e}$$

(we will derive this equation below) which states that along the direction represented by vector \mathbf{e} , multiplication by data matrix \mathbf{E} does not change the direction, but only the length (where ϵ may be any number, including 0).¹ The related pair (ϵ, \mathbf{e}) is called an *eigenpair* of matrix \mathbf{E} .

A *network* or *graph* $G(V, E)$ is a set of *nodes* V (points, vertices) connected by a set of *links* E (lines, edges). We will consider networks that are *binary* (edges have logical value 1 if an edge exists, 0 if not), *symmetric* (an edge from node i to j implies an edge from node j to i), *connected* (there is a set of edges connecting any two nodes, consequently only one *component*), and *without self-loops* (no edges between i and i). We may represent such a network as the *adjacency matrix* $\mathbf{A} = \mathbf{A}(G)$ with:

- 1 in row i , column j if i is connected to j ,
- 0 otherwise.

We will not directly discuss *weighted* networks, where the entries for an edge may be a number other than 1, although most of the results that follow generalize to such networks. For many “real world” networks, \mathbf{A} consists mostly of 0's: it is *sparse*. We will discuss efficient ways of storing and manipulating \mathbf{A} using *sparse methods*.

Associated with \mathbf{A} is the *degree distribution* \mathbf{D} , a diagonal matrix with row-sums of \mathbf{A} along the diagonal, and 0's elsewhere. \mathbf{D} describes how many connections each node has. We call the number of nodes, m , the *order* of G and it is equal to the number of rows or columns of \mathbf{A} . We represent the number of edges by $|E|$. We will also introduce two other matrices related to \mathbf{A} :

- the *Laplacian*: $\mathbf{L} = \mathbf{D} - \mathbf{A}$
- the *Normal*: $\mathbf{N} = \mathbf{D}^{-1} \mathbf{A}$

and will discuss the properties of the spectrum and associated eigenvectors of \mathbf{A} , \mathbf{L} , and \mathbf{N} .

¹ We have introduced some notation which will be followed throughout:

- matrices are represented by bold capitals: \mathbf{D}
- (column-)vectors are represented by bold lower case: \mathbf{e}
- inner products of vectors are represented as $\mathbf{e}^T \mathbf{e} = n$ (a scalar) where \mathbf{e}^T is the transpose of \mathbf{e} .
- outer products of vectors are represented as $\mathbf{e} \mathbf{e}^T = \mathbf{M}$ (a matrix)
- eigenvalues are represented by greek letters, usually with some relationship to the latin letters representing a matrix and an eigenvector. E.g., (α_i, \mathbf{a}_i) is an eigenpair of adjacency matrix \mathbf{A} .

Distances and diameter:

One important property of a network is the set of *distances* between any pair of nodes i and j ; that is, the least number of links between any pairs i and j . One way of calculating this is to take *powers* of the matrix \mathbf{A} as follows:

1st power $\mathbf{A} = \mathbf{A}$ by definition gives a matrix of all pairs of nodes linked to each other.

2nd power $= \mathbf{A}\mathbf{A}$ has a non-zero in row i column j if j is two steps away from i . Since i is 2 steps away from itself, the diagonal i,i entry counts the number of these 2-steps.

3rd power $= \mathbf{A}\mathbf{A}\mathbf{A}$ has a non-zero entry in row i column j if j is 3 steps away from i .

Eventually, some power of \mathbf{A} , say \mathbf{A}^N , will consist of entirely non-zero entries, meaning every node has been reached from every other node. We call N the *diameter* of the graph: the longest possible path between any pair of nodes.

This is a very inefficient way of calculating the diameter of a graph for two reasons:

- 1) calculating each power of \mathbf{A} requires m^3 calculations
- 2) as more nodes are reached, the powers of \mathbf{A} become less sparse until eventually no 0's remain: the amount of storage required approaches m^2 .

If we continue taking powers of \mathbf{A} , an interesting thing happens: all the columns become multiples of each other. Taking higher powers of \mathbf{A} corresponds to taking longer “walks” along the edges and we can interpret this as a “loss of memory” about where we started from (Lovasz, 1995). We will see why this happens soon, as well as other examples of this phenomenon.

We can approach this problem another way by the properties of the *spectral decomposition* of \mathbf{A} (Parlett, 1980). Let α_i be the eigenvalues of \mathbf{A} and \mathbf{a}_i the corresponding eigenvectors, with $\alpha_0 \geq \alpha_1 \geq \alpha_2 \dots \geq \alpha_{m-1}$ and $\|\mathbf{a}_i\| = 1$ (the eigenvectors are *normalized* to length 1). Then the spectral decomposition of \mathbf{A} is:

$$(1) \mathbf{A} = \sum_i (\alpha_i) \mathbf{a}_i \mathbf{a}_i^T \quad \text{where } \mathbf{a}_i \mathbf{a}_i^T \text{ is an } m \times m \text{ matrix defining a 1-dimension subspace and} \\ (\mathbf{a}_i \mathbf{a}_i^T)^N = \mathbf{a}_i \mathbf{a}_i^T \text{ if } i=j; \mathbf{a}_i \mathbf{a}_j^T = \mathbf{0} \text{ if } i \neq j$$

therefore $\mathbf{A}^N = \sum_i (\alpha_i)^N \mathbf{a}_i \mathbf{a}_i^T$ for any power N , and this allows an easy way of calculating powers of \mathbf{A} , assuming we have already calculated all the eigenpairs (α_i, \mathbf{a}_i) .

Another important property of the spectral decomposition is the *approximation* property. If we take the first k of the eigenpairs (α_i, \mathbf{a}_i) , then $\mathbf{A}_k = \sum_{i=0}^k \alpha_i \mathbf{a}_i \mathbf{a}_i^T$ is the *best least-squares approximation* to \mathbf{A} , meaning that we have captured most of the variability of \mathbf{A} in the important eigenpairs. For example, we can *estimate* an *upper bound* for the diameter using the second-largest eigenvalue α_1 (Chung, 1989):

$$\text{Diam}(G) \leq \lceil \ln(m-1)/\ln(k/\alpha_1) \rceil$$

Unfortunately, this bound applies only to k -regular networks (all degrees = $k = \alpha_0$). We will get better bounds for general networks using different spectra. Nevertheless, this bound does show one relationship between the spectrum and an important property like diameter. In particular, when k/α_1 is *large* (there is a large *gap* between the first two eigenvalues), the upper bound on diameter is *small*, so all distances are short.

The Power Method and Sparse methods:

Using (1) and eigenpair (α_0, \mathbf{a}_0) we can see why taking large enough powers of \mathbf{A} results in columns that are multiples of one another – in fact, multiples of eigenvector \mathbf{a}_0 . This is the basis of the *Power method* (Hotelling, 1933) for finding eigenpairs. We have mentioned that taking powers of matrix \mathbf{A} is not efficient, so we introduce a representation and methods that are far more efficient.

A very simple way of storing and manipulating a sparse matrix \mathbf{A} is to use a *link list* representation, which stores only the non-zero entries of \mathbf{A} as a list of pairs i,j for each link in \mathbf{A} . We could then calculate the diameter of \mathbf{A} by starting at $i=1$ and following each link until we have reached every node, repeat for $i=2$, and save the maximum number of steps. This requires about $m|E|$ operations (Aho, *et al.*, 1987) and a very moderate amount storage equal to $2|E|$. We can now use (1) to devise a very efficient version of the Power Method for finding the largest eigenpair:

Starting with some random vector \mathbf{p} normalized to length 1:

Repeat $\mathbf{p}' \leftarrow \mathbf{A}\mathbf{p}$, $\mathbf{q} \leftarrow \mathbf{p}$, $\mathbf{p} \leftarrow \mathbf{p}'$ until \mathbf{p} is no longer changing in direction.

Then the largest eigenpair of \mathbf{A} is $(\mathbf{p}/\mathbf{q}, \mathbf{p})$. There are some bookkeeping details: $\mathbf{A}\mathbf{p}$ uses the link list representation, and the entries of \mathbf{p}' must be adjusted in size after each multiplication (for details see Richards & Seary, 2000), but the method will always work for any matrix *without repeated eigenvalues*, which is generally the case for social networks.

If we want more eigenpairs, we can iterate with

$$\mathbf{p}' \leftarrow \mathbf{M}\mathbf{p} - \alpha_0 \mathbf{a}_0 \mathbf{a}_0^T$$

to get the second, and with

$$\mathbf{p}' \leftarrow \mathbf{M}\mathbf{p} - \alpha_0 \mathbf{a}_0 \mathbf{a}_0^T - \alpha_1 \mathbf{a}_1 \mathbf{a}_1^T$$

to get the third, and so on, without destroying sparsity. However, we must store the (α_i, \mathbf{a}_i) eigenpairs somewhere; the procedure is subject to loss of precision on a computer; and the iterations may converge slowly if α_i/α_{i-1} is close to 1. There are better methods, such as Lanczos iteration (Parlett *et al.*, 1982) which converge very rapidly and do not have problems with loss of precision.

Some network invariants:

Some properties of \mathbf{A} remain unchanged (*invariant*) under the series of orthogonal rotations that diagonalize \mathbf{A} (eigendecomposition). We will relate these to some *network invariants* of \mathbf{A} .

The eigenvalues of any symmetric matrix \mathbf{M} are the roots of the *characteristic polynomial*:

$$x^m + c_1 x^{m-1} + c_2 x^{m-2} + c_3 x^{m-3} \dots + c_{m-1}$$

Therefore, $c_1 = \alpha_0 + \alpha_1 + \dots + \alpha_{m-1}$ (sum over all eigenvalues) of \mathbf{A} ;

$$c_2 = \alpha_0 \alpha_1 + \alpha_0 \alpha_2 \dots + \alpha_0 \alpha_{m-1} \dots + \alpha_{m-3} \alpha_{m-1} + \alpha_{m-2} \alpha_{m-1}$$
 (sum over all pairs);

$$c_3 = \alpha_0 \alpha_1 \alpha_2 + \alpha_0 \alpha_1 \alpha_3 + \dots + \alpha_{m-3} \alpha_{m-2} \alpha_{m-1}$$
 (sum over all triples)

The *trace* of a matrix is the sum of the entries on the diagonal, and this is invariant under orthogonal rotations. Since \mathbf{A} has trace of 0 (no self-loops), $c_1 = 0$. The sum of product pairs is equal to minus the number of edges so that $c_2 = -|E|$. Most important is c_3 which is twice the number of triangles in G . Higher coefficients are related to cycles of length 4, 5,... although they also contain contributions for shorter cycles (Biggs, 1993). It appears that the eigenvalues of \mathbf{A} encode information about the cycles of a network as well as its diameter. We will see related results for the other two spectra.

A *bipartite* network is one that can be partitioned so that the nodes in one part have connections only to nodes in the other part, and vice-versa. Such a network cannot have odd cycles (of any length) and hence no triangles. This means *all* the odd coefficients c_{2k-1} must be 0. It can also be shown (Biggs, 1993) that, in bipartite networks, the eigenvalues occur in pairs with opposite signs, so that $\alpha_0 = -\alpha_{m-1}$ and so on. Bipartite networks can be used to represent *two-mode networks* (Wasserman & Faust, 1994), for example the network relating people and the events they attend.

These results scratch the surface of the information contained in the spectrum of \mathbf{A} for k -regular graphs. For general graphs, we need to turn to other spectra.²

The Laplacian spectrum:

The Laplacian of a network was originally discovered by Kirchoff (1847). There are a number of definitions and derivations, perhaps the most revealing due to Hall (1970), who was interested in situating the nodes of any network so that total edge lengths are minimized.

He considers the problem of finding the *minimum* of the weighted sum

$$(2) \quad z = 1/2 \sum_{i,j} (x_i - x_j)^2 a_{ij}$$

where a_{ij} are the elements of the adjacency matrix \mathbf{A} . The sum is over all pairs of squared distances between nodes *which are connected*, and so the solution should result in nodes with large numbers of inter-connections being clustered together.

Equation (2) can be re-written as:

$$\begin{aligned} &= 1/2 \sum_{i,j} (x_i^2 - 2x_i x_j + x_j^2) a_{ij} &&= 1/2 \sum_i x_i^2 a_{ij} - 1/2 \sum_{i,j} 2x_i x_j a_{ij} + 1/2 \sum_j x_j^2 a_{ij} \\ &= \sum_i x_i^2 a_{ij} + \sum_i \sum_{i,j} x_i x_j a_{ij} &&= \mathbf{X}^T \mathbf{L} \mathbf{X} \end{aligned}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is the Laplacian. In addition to this, Hall supplies the condition that $\mathbf{X}^T \mathbf{X} = 1$, i.e., the distances are normalized. Using Lagrange multipliers (a standard method for solving problems with constraints), we have:

$$\mathbf{z} = \mathbf{X}^T \mathbf{L} \mathbf{X} - \lambda \mathbf{X}^T \mathbf{X}$$

and to minimize this expression, we take derivatives with respect to \mathbf{X} to give:

$$(3) \quad \mathbf{L} \mathbf{X} - \lambda \mathbf{X} = 0 \text{ or } \mathbf{L} \mathbf{X} = \lambda \mathbf{X}$$

which is the eigenvalue equation. It is not hard to show that $\lambda_0 = 0$ with $\mathbf{1}_0 = \mathbf{1}$, the constant (or *trivial*) eigenvector, and that $0 = \lambda_0 \leq \lambda_1 \leq \dots \leq \lambda_{m-1}$. For \mathbf{L} , the most “important” eigenvectors belong to the smallest eigenvalues (Pothen, *et al.*, 1990).

It turns out that the discrete network Laplacian shares many important properties with the well-known continuous Laplacian *operator* ∇^2 of mathematical physics. This has led to an explosion of research and results, mostly concerned with λ_1 (Bein, 1991).

The definition of \mathbf{L} shows that there is no loss of sparsity (except for the diagonal) and that the sparse methods mentioned earlier can be applied to find all or some of the eigenpairs. The requirement that we must find the *smallest* eigenpairs is easily overcome by subtracting a suitably large constant from the diagonal of $-\mathbf{L}$ (which subtracts that constant from the eigenvalues without changing the eigenvectors). This guarantees that the first eigenpairs returned by the Power Method or Lanczos iteration are associated with the smallest eigenvalues of \mathbf{L} .

² Similar results may be obtained from the *moments* of the eigenvalue distribution (Farkas. *et al.*, 2001; Gho, *et al.*, 2001)

Some of the coefficients of the characteristic polynomial of \mathbf{L} have an easy interpretation:

$$c_1 = \text{Trace}(\mathbf{L}) = 2|E| \text{ (i.e., twice the number of edges)}$$

$$c_{m-1} = 0 \text{ (since 0 is an eigenvalue)}$$

$$|m c_{m-1}| = \lambda_0 \lambda_1 \dots \lambda_{m-1} + \lambda_0 \lambda_2 \dots \lambda_{m-1} + \dots + \lambda_1 \lambda_2 \dots \lambda_{m-1} = \lambda_1 \lambda_2 \dots \lambda_{m-1} = \text{the number of spanning trees of } G \text{ (this is the Matrix-tree theorem of Kirchoff, 1847).}$$

In general, the eigenvalues of \mathbf{L} encode information about the *tree-structure* of G (Cvetkovic, *et al.*, 1995). The spectrum of \mathbf{L} contains a 0 for every connected component. There is no such direct way to find the number of components of a network from the spectrum of \mathbf{A} . There is also a bound on diameter related to λ_{m-1} and λ_1 for general graphs from Chung, *et al.*, (1994):

$$\text{Diam}(G) \leq \left\lceil \cosh^{-1}(m-1) / \cosh^{-1}((\lambda_{m-1} + \lambda_1) / (\lambda_{m-1} - \lambda_1)) \right\rceil$$

Intuitively, if λ_1 is close to 0, the graph is almost disconnected, while if $\lambda_1 \gg \lambda_0$ (an eigenvalue gap) the diameter is small.

The Normal spectrum:

We can repeat the same argument as Hall to derive the Normal spectrum, with the normalization constraint that $\mathbf{X}^T \mathbf{D} \mathbf{X} = 1$ (Seary & Richards, 1995) to give:

$$\mathbf{L} \mathbf{X} = \mu \mathbf{D} \mathbf{X}, \text{ or assuming that } \mathbf{D} \text{ can be inverted,}$$

$$\mathbf{D}^{-1} \mathbf{L} \mathbf{X} = \mathbf{D}^{-1} (\mathbf{D} - \mathbf{A}) \mathbf{X} = (\mathbf{I} - \mathbf{D}^{-1} \mathbf{A}) \mathbf{X} = \mu \mathbf{X}$$

where \mathbf{I} is an *identity* matrix of proper size. In fact, we usually take the defining equation to be

$$(4) \mathbf{D}^{-1} \mathbf{A} \mathbf{X} = \mathbf{N} \mathbf{X} = \nu \mathbf{X} \text{ with } \mathbf{D}^{-1} \mathbf{A} = \mathbf{N} \text{ and } \nu = 1 - \mu$$

since adding an identity matrix shifts the eigenvalues by 1 without changing the eigenvectors. Note that for connected networks \mathbf{D} not only has an inverse, it also has an inverse *square root* $\mathbf{D}^{-1/2}$.

The *Normal* matrix \mathbf{N} has a number of interesting properties:

- 1) It is a *generalized* Laplacian (with a different definition of orthonormality)
- 2) It therefore has a *trivial* eigenvector \mathbf{n}_0 with eigenvalue $\nu_0 = 1$
- 3) The spectrum of \mathbf{N} is bounded by $1 = \nu_0 \geq \nu_1 \dots \geq \nu_{m-1} \geq -1$
- 4) The rows of \mathbf{N} sum to 1 (it is a *stochastic* matrix)
- 5) The spectrum of \mathbf{N} contains a 1 for every connected component
- 6) The eigenvalue -1 only occurs if G is bipartite, in which case all eigenvalues occur in pairs.
- 7) \mathbf{N} has been rediscovered a number of times: generalized or combinatorial Laplacian (Dodziuk & Kendall, 1985; Chung, 1995); Q-spectrum (Cvetkovic, *et al.*, 1995).

The descriptive name Normal is suggested by points 2) - 5), although it is not standard terminology.

It is easy to see that there is no loss of sparsity in the definition of \mathbf{N} . Each 1 in row i is simply replaced by $1/\mathbf{D}_{ii}$ and the 0's are unchanged, but \mathbf{N} is no longer symmetric. However, the matrix $\mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$ is *similar* to \mathbf{N} (has the same eigenvalues) and we can apply the sparse methods described above to solve for the eigenpairs (ν_i, \mathbf{e}_i) and then calculate $\mathbf{D}^{-1/2} \mathbf{e}_i = \mathbf{n}_i$ to get the corresponding eigenvectors without losing precision or sparsity.

For \mathbf{N} , the coefficients of the characteristic equation are harder to interpret except in special cases, but the eigenvalues encode information about *both* the cycle *and* tree structure of G

(Cvetkovic, *et al.*, 1995). Some examples:

$$c_1 = \text{Trace}(\mathbf{N}) = 0$$

$$c_3 = c_{2k-1} = 0 \text{ (no triangles or other odd cycles) if } G \text{ is bipartite}$$

$$\prod_i \text{deg}(i) / \sum_i \text{deg}(i) \sum_{i=1}^n (1-v_i) = \text{number of spanning trees of } G$$

In the last example we see how details of the degree distribution are *also* encoded in the spectrum. Fan Chung uses this to derive two remarkable bounds (see Chung, 1995 for details):

$$\text{Diam}(G) \leq \max \left[\frac{\cosh^{-1} \sqrt{\frac{\text{vol } \bar{X} \text{ vol } \bar{Y}}{\text{vol } X \text{ vol } Y}}}{\cosh^{-1} \frac{(v_{n-1} + v_1)}{(v_{n-1} - v_1)}} \right] \quad \min_{i \neq j} \text{dist}(X_i, X_j) = \max_{i \neq j} \left[\frac{\ln \sqrt{\frac{\text{vol } \bar{X}_i \text{ vol } \bar{X}_j}{\text{vol } X_i \text{ vol } X_j}}}{\ln \frac{(v_{n-1} + v_k)}{(v_{n-1} - v_k)}} \right]$$

where $\text{vol } X$ is the total number of edges in a subset of nodes $X \subset V$ and \bar{X} is $V-X$. Chung's first bound applies to *any* graph (regular or not) and is much tighter than the previous bound (for the Laplacian). Intuitively, if v_1 is close to 1, the network has a long path or is almost disconnected, and if $v_1 \ll 1$, the diameter is small.

Chung's second bound describes the distance between subsets for any number k of subsets, based on the k^{th} eigenvalue. The result suggests that we can use the eigenvalues to estimate how many subsets we should look for in a network without forcing distances that are too short (and hence too many subsets).

Interpreting the Spectra:

Many important properties of the spectrum of $\mathbf{A}(G)$ where G is k -regular are true for $\mathbf{L}(G)$ and $\mathbf{N}(G)$, even when G is not regular. Another way of looking at this is that these properties of \mathbf{A} are true because the spectrum of \mathbf{A} is simply related to those of \mathbf{L} and \mathbf{N} for regular graphs: ($\alpha_i = k - \lambda_i = v_i/k$ for k -regular graphs, with the corresponding eigenvectors being identical). In other words, both \mathbf{L} and \mathbf{N} are more natural function of graphs. This point of view is shared by the authors of recent papers on the Laplacian (Grone, *et al.*, 1990, 1994). Mohar (1991) presents a collection of important results relating to the spectrum and eigenvectors of \mathbf{L} . Chung (1995) has written several papers and a book about \mathbf{N} .

We return to the goal expressed in the opening paragraph. We would like to find the most important global features of a network, after accounting for what could be considered "expected" for a random network with the same number of nodes and edges. The biggest problem with interpreting the spectrum of \mathbf{A} is the lack of an "expected" eigenvector (again, except for k -regular graphs). There is a lot of literature on the so-called "main eigenvectors" of \mathbf{A} : those which have a *projection* on the "all-ones" vector (e.g., Harary & Schwenk, 1979), but the results remain hard to interpret (Cvetkovic and Rowlinson, 1990). Both \mathbf{L} and \mathbf{N} have an "expected" all-ones eigenvector for which the interpretation is clear (though different in each case).

To interpret \mathbf{L} , we turn to physical analogy and the relation to ∇^2 as discussed by Friedman (1993). He considers a graph G as a discrete *manifold* (surface) subject to "free" boundary conditions.³ For illustration, consider ∇^2 as the spatial part of the wave equation (Fisher, 1966,

³ no external constraints need to be satisfied

Chavel, 1984). Think of a fishing net subject to no forces. It just lies there at 0 energy with nothing happening. As we subject it to regular oscillations, the net vibrates with the most highly-connected regions moving together. Friedman shows how the Hilbert Nodal theorem (Courant & Hilbert, 1965) can be applied to a discrete network, which generalizes Fiedler's result (described below): the k^{th} eigenvector divides the network into no more than $k+1$ *disconnected* components.⁴

To interpret \mathbf{N} , we have a number of choices:

- 1) \mathbf{N} is the Laplacian for a network of nodes, each *weighted* by its degree
- 2) \mathbf{N} is the transition matrix of a Markov Chain for a simple random walk on the nodes
- 3) \mathbf{N}^2 is similar to the χ^2 matrix, thus treating \mathbf{A} as a *contingency table*

The first leads to a physical analogy similar to that for \mathbf{L} , so we consider 2) and 3):

The Normal spectrum and Random walks:

Specifically, we consider nearest-neighbour random walks on a network (Lawler & Sokal, 1988). Define the probability-transition matrix for such a walk as

$$\mathbf{N} = \mathbf{D}^{-1} \mathbf{A}$$

Then the probability of moving from vertex i to any vertex adjacent to i is uniform. \mathbf{N} is a row-stochastic matrix, and the random walk is a Markov chain. In this case $\mathbf{1}$ (the trivial all-ones eigenvector) is related to the *stationary state* of the Markov Chain: the probability is $1 = v_0$ that such a probability distribution is eventually reached.⁵ The vector $\mathbf{p}_0 = \mathbf{1}^T \mathbf{N} = \mathbf{N}^T \mathbf{1}$ is the stationary state, and it is *proportional* to the degree distribution. The second eigenpair (v_1, \mathbf{n}_1) has become important in the analysis of *rapidly mixing* Markov chains – those that reach the stationary state quickly (Sinclair, 1995). From the previous discussion it should not be surprising that these are associated with $v_1 \ll 1$ (a large eigenvalue gap), which means that the walk quickly “forgets” where it started.⁶ Moreover, when v_1 is close to 1, there must be parts of the network that are hard to reach in a random walk, implying long paths or a nearly disconnected network.

Normal spectrum and χ^2 :

The χ^2 matrix is defined in terms of the row and column *marginals* (sums). A typical element is $(\text{Observed}_{ij} - \text{Expected}_{ij})^2 / \text{Expected}_{ij}$ which is not sparse. For a sparse network \mathbf{A} , consider χ which has a typical element $(\text{Observed}_{ij} - \text{Expected}_{ij}) / \sqrt{\text{Expected}_{ij}}$ where

$$\text{Expected}_{ij} = \frac{\text{deg}(i) \text{deg}(j)}{\sum \text{deg}(i)}$$

⁴ This interpretation of the eigenvectors may be even more useful when considering ∇^2 as the spatial part of the *Diffusion* equation (for example, when considering diffusion of innovation or disease).

⁵ A problem can arise with bipartite graphs: \mathbf{p}_0 does not exist since the chain oscillates between the two sets of vertices (period = 2). Probabilists deal with this by a simple trick: divide \mathbf{N} by 2 and add a self-loop of probability 1/2 to every vertex: $\mathbf{N}' = \mathbf{I}/2 + \mathbf{N}/2$. The eigenvalues of \mathbf{N}' are then: $1 = v'_0 \leq v'_1 \leq \dots \leq v'_{m-1} \leq 0$ so that $v' = (1 + v)/2$ and again the eigenvectors are not affected. In effect, this suppresses *all* negative eigenvalues. However, negative eigenvalues are useful for directed and bipartite networks.

⁶ As we saw, \mathbf{A}^K shows a similar phenomenon, but there is no simple relation to \mathbf{D} , due to “leakage” from all the “main eigenvectors” (Cvetkovic, *et al.*, 1988)

We can write χ as: $\frac{O}{\sqrt{E}} - \sqrt{E}$ so that non-zero elements of \mathbf{A} become $A_{ij}/\sqrt{E_{ij}}$ while the 0 terms are unaffected, maintaining sparsity. The second term corresponds to the trivial eigenvector which can be dealt with separately. In matrix notation $\chi = \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$ which has eigenpairs $(v_i, \mathbf{D}^{1/2} \mathbf{n}_i)$. Thus we have

$$(5) \quad \chi^2 = \sum_{j=1} v_j^2 / \sum_i a_{ij} \quad (\text{omitting the } v_0=1 \text{ expected term for } \mathbf{n}_i = \mathbf{1})$$

This equation shows how much each dimension contributes to χ^2 which is a measure of *dependence* between rows and columns. In this interpretation, if v_1 is small ($v_1 \ll v_0 = 1$), then χ^2 is also small: there is no relation between rows and columns of \mathbf{A} , and so there is no “signal” above the expected “background”. If v_1 is close to 1, then χ^2 will be large and there is a relation between rows and columns of \mathbf{A} , with the first eigenvector pointing in the direction of the maximum variability in χ^2 . If v_2, v_3, \dots, v_k are also large, we need $k+1$ eigenvectors to describe the patterns in the χ^2 matrix. With (5) we can tell how many eigenvectors we need to explain most of the χ^2 of the network.⁷

Compositions

The *Kronecker product* of two binary matrices \mathbf{A}_1 and \mathbf{A}_2 makes a *copy* of \mathbf{A}_1 for each 1 in \mathbf{A}_2 . It is well-known that for two matrices \mathbf{A}_1 and \mathbf{A}_2 of order m_1 and m_2 the eigenpairs of the Kronecker product $\mathbf{A}_1 \otimes \mathbf{A}_2$ *behave well* (West, 1996):

If \mathbf{A}_1 has eigenpairs (α_i, \mathbf{a}_i) and \mathbf{A}_2 has eigenpairs (β_j, \mathbf{b}_j) , then

$$\mathbf{A}_1 \otimes \mathbf{A}_2 \text{ has eigenpairs } (\{\alpha_i \times \beta_j\}, \{\mathbf{a}_i \otimes \mathbf{b}_j\})$$

It is also well-known that \mathbf{A}_1 and \mathbf{A}_2 behave well under *Cartesian sum*:

$$\mathbf{A}_1 \oplus \mathbf{A}_2 = \mathbf{A}_1 \otimes \mathbf{I}_2 + \mathbf{A}_2 \otimes \mathbf{I}_1 \quad (\text{where } \mathbf{I}_1 \text{ and } \mathbf{I}_2 \text{ are identity matrices of appropriate size})$$

has eigenpairs $(\{\alpha_i + \beta_j\}, \{\mathbf{a}_i \otimes \mathbf{b}_j\})$.

The Laplacian \mathbf{L} also behaves well under Cartesian sum. For $\mathbf{L}_1 \oplus \mathbf{L}_2$ the eigenpairs are $(\{\lambda_i + \kappa_j\}, \{\mathbf{l}_i \otimes \mathbf{k}_j\})$

This fact is used by Pothen, *et al.*, (1990) to study the Laplacian of grids, which are Cartesian sums of paths. Further, the eigenvalues of \mathbf{L}_1 and \mathbf{L}_2 always contain a $\lambda_0 = 0$ with corresponding constant eigenvector, so that the corresponding eigenpairs of $\mathbf{L}_1 \oplus \mathbf{L}_2$ are $(\lambda_i + 0, \mathbf{l}_i \otimes \mathbf{1})$. The term $\mathbf{l}_i \otimes \mathbf{1}$ means that the components of \mathbf{l}_i are *replicated* m_2 times. Since the Cartesian sum of two paths is a grid, this produces a perfectly rectangular representation (Fig. 1b). The Laplacian is therefore a useful tool in problems involving regular grids (or hypergrids). However, \mathbf{N} does *not* behave well under Cartesian sum (Fig. 1c).

The Laplacian does *not* behave well under Kronecker product. However, the Normal spectrum *does* (Chow, 1997), so that $\mathbf{N}_1 \otimes \mathbf{N}_2$ has eigenpairs

$$(\{v_i \times \mu_j\}, \{\mathbf{n}_i \otimes \mathbf{m}_j\})$$

Further, the eigenvalues of \mathbf{N}_1 and \mathbf{N}_2 always contain a $v_0 = 1$ with corresponding constant eigenvector, so that the corresponding eigenpairs of $\mathbf{N}_1 \otimes \mathbf{N}_2$ are $(v_i \times 1, \mathbf{n}_i \otimes \mathbf{1})$. The term $\mathbf{n}_i \otimes \mathbf{1}$ means that the components of \mathbf{n}_i are *replicated* m_2 times. Because all the coordinates are the same within each copy, this produces *clustering* of the components of $\mathbf{N}_1 \otimes \mathbf{N}_2$ for these eigenvectors.

⁷ While PCA results tell how much of the variance each dimension accounts for, the Normal eigenvalues tell how much of the network’s chi-squared each eigenvector (dimension) accounts for.

It appears that the behaviour under Kronecker product explains why both the Adjacency and Normal eigenvectors are good at detecting both on- and off-diagonal *blocks* (clusters of edges).

Visualization

The Laplacian can \mathbf{L} provide good visual representations of graphs which are Cartesian products (such as grids and hypercubes); while \mathbf{N} can provide good visual representations of graphs which are Kronecker products (such as graphs consisting of blocks). The reasons for this are suggested above and have mostly to do with the behaviour of eigenpairs which are sums and products with 0 and 1, respectively. For graphs that are not k -regular, eigenpairs of \mathbf{A} do not provide such good representations since, in general, there is no constant (expected, trivial) eigenvector to combine with.

Another way of describing these results is to consider the relationship between the eigenvector components for a node and those it is connected to (Seary & Richards, 1999). It is evident from the definition of eigendecomposition that (where “ $u \sim v$ ” means “ u is connected to v ”)

$$(6) \quad \mathbf{a}_i(u) = \sum_{u \sim v} \mathbf{a}_i(v) / \alpha_i \quad \text{for eigenpair } i \text{ of } \mathbf{A}$$

$$(7) \quad \mathbf{l}_i(u) = \sum_{u \sim v} \mathbf{l}_i(v) / (\lambda_i - \text{deg}(u)) \quad \text{for eigenpair } i \text{ of } \mathbf{L}$$

$$(8) \quad \mathbf{n}_i(u) = \sum_{u \sim v} \mathbf{n}_i(v) / (v_i \times \text{deg}(u)) \quad \text{for eigenpair } i \text{ of } \mathbf{N}$$

Note that \mathbf{A} has no control for node degree. Consider the effect for "important" eigenpairs: ($|\alpha| \approx k \gg 1$, $\lambda \approx 0$ and $|v| \approx 1$) when $\text{deg}(u)$ is small, $\mathbf{a}(u)$ will be folded toward the origin, while $\mathbf{l}(u)$ and $\mathbf{n}(u)$ will sit further *away* from the origin than its neighbours. This effect makes it difficult to interpret visual representations based on \mathbf{A} , except for k -regular graphs where all three spectra are essentially the same (Fig. 1)

The equation for \mathbf{n}_i shows that for v_i near 1, each node is approximately *at the centroid* of those it is connected to. The exact difference from the centroid for node u of eigenvector \mathbf{n}_i is:

$$\mathbf{n}_i(u) - \sum_{u \sim v} \mathbf{n}_i(v) / \text{deg}(u) = (1 - v_i) \mathbf{n}_i$$

For important eigenvalues v_i near 1, this produces very good visualization properties. In addition, the eigenvector representation may be combined with derived properties such as *betweenness* (Freeman, 1979) to produce very helpful displays of large networks (Brandes *et al.*, 2001)

Interpreting the eigenvectors

1. Partitions

Powers (1988) and others have shown how eigenvectors of \mathbf{A} can be used to find partitions of highly connected subsets (clusters) of nodes, but these methods are not as general or as clear as those derived from \mathbf{L} or \mathbf{N} .

The first non-trivial eigenvector \mathbf{l}_1 of \mathbf{L} is the subject of extensive literature (Lubotzky, 1994; Alon & Millman, 1985). Fiedler (1975) first suggested that the eigenvector \mathbf{l}_1 associated with the *second-smallest* eigenvalue λ_1 could be used to solve the *min-cut* problem: separate the network into two approximately equal sets of nodes with the fewest number of connections between them, based on the *signs* of the components of \mathbf{l}_1 .⁸ In fact, more recent derivations of \mathbf{L} use the min-cut property as a starting point (Walshaw, *et al.*, 1995) and the results are used to partition compute-intensive problems into sub-processes with minimal inter-process communication (Pothen, *et al.*, 1990). This technique is called *Recursive Spectral Bisection* (Simon, 1991). Other researchers have used \mathbf{l}_2 , \mathbf{l}_3

⁸ Hagen also uses deviations from median

and higher eigenvectors to produce multi-way partitions of networks (Hendrickson & Leland, 1995).

The graph bisection problem (Mohar & Poljak.,1991) is to find two nearly equal-sized subsets $V_1, V_2 \subset V$ such that $\text{cut}(V_1, V_2) = \sum_{ij} a_{ij}$ is minimized, where $i \in V_1, j \in V_2$. (i.e. nodes in V_1 and V_2 have few connections to each other).

This problem is known to be NP-hard (Garey & Johnson, 1979), but a good approximation is given by the *signs* of \mathbf{I}_1 (Walshaw & Berzins,1995). This gives two sets of nodes of roughly the same size, but has no control for the number of *edges* in each part, and so any clustering of nodes is a *side-effect* of the partition.

However, we can add an additional constraint that the number of edges in each part also be roughly equal by *weighting* the node sets by their total degrees. This is exactly what a partition based on \mathbf{n}_1 from \mathbf{N} gives us, since it \mathbf{n}_1 points in the direction of *maximum variability* in χ^2 (Greenacre, 1984).⁹ Similarly, further partitions based on $\mathbf{n}_2, \mathbf{n}_3, \dots$ will also produce sets of nodes with a large number of edges in common (as long as v_2, v_3, \dots make significant contributions to χ^2). Partitions based on *positive* eigenvalues will produce blocks *on the diagonal* of \mathbf{A} of edges associated with each set of nodes, while those based on *negative* eigenvalues produce nearly bipartite *off-diagonal* blocks (which occur in pairs if the network is symmetric) (Seary & Richards,1995).

2. Clustering

Ideally, the important eigenvectors should be at least *bimodal* to induce clustering based on sign-partitions, and often they are *multi-modal* (Hagen,1992), suggesting that standard clustering methods can be used on the coordinates of these vectors. Equations (7) and (8) show that \mathbf{L} and \mathbf{N} place nodes approximately at the centroids of their neighbours. For \mathbf{N} , the distances are actually measured in χ^2 space, meaning that nodes with very similar *patterns* of connections will be close together (Benzecri, 1992). This clustering happens with either positive or negative eigenvalues (on- or off-diagonal). The latter are important in *nearly bipartite* networks with few triangles (Fig. 3).

3. Problems

Farkas, *et al.*, (2001) and Goh, *et al.*, (2001) report that the important eigenvectors of \mathbf{A} are very *localized* on nodes of high degree, and suggest that this effect may be used to distinguish certain types of networks. This effect does not occur for \mathbf{L} or \mathbf{N} (fig. 2), since each include some control for degree, and so far no similar results for distinguishing network types have been reported for these spectra. The biggest problem for \mathbf{L} and \mathbf{N} is their sensitivity to “long paths”, especially to pendant trees attached to the main body of the network (Seary & Richards, 2000). For \mathbf{N} , these may be interpreted as nodes that are hard to reach (distant) in a random walk. For long paths *internal* to a network, this effect is actually an advantage, since these cycles are detected as “locally bipartite” and emphasized in important eigenvectors. Nodes on such paths can have a large effect on global properties such as diameter (Fig. 3-4).

Two-mode networks

Two-mode networks mix two different kinds of nodes and connections. A simple example is an affiliation network such as people and the events they attend. We could be interested in finding sets of people with events in common (or, equivalently, sets of events attended by the same people): this is an example of *co-clustering*. Affiliation networks can be represented by bipartite graphs for which \mathbf{A} and \mathbf{N}

⁹ see Dhillon (2001) for a formal derivation and proof

are most suited, since they have symmetric spectra for these (the eigenvalues occur in pairs with opposite sign). Because of this we don't need the entire bipartite matrix: we can work with the rectangular representation, and infer the missing parts of the eigendecomposition. If we assume m_1 people and m_2 events, the resulting eigenvectors consist of m_1 components for people followed by m_2 components for events. The resulting blocks will be strictly off-diagonal and once again the eigenvectors of \mathbf{N} provide a superior solution by maximizing χ^2 . In fact, this solution is identical to that provided by Correspondence Analysis, a statistical technique for finding patterns in 2-mode data (Benzecri, 1992).

Partial Iteration

For large networks, it is not necessary or desirable to calculate the entire eigendecomposition. For *very* large networks, it may not be possible in terms of time and space to calculate even a *few* eigenpairs. Nevertheless, it is possible to get at a large amount of the global and local network structure by partially iterating using the Power Method. A few iterations of $\mathbf{N} = \mathbf{D}^{-1} \mathbf{A}$, with each iteration placing nodes at the means of their neighbours, will produce a *mixture* of the most important eigenvectors. Consider the spectral representation

$$\mathbf{N}^K = \sum_i (v_i)^K \mathbf{n}_i \mathbf{D} \mathbf{n}_i^T$$

We know that $(\mathbf{n}_i \mathbf{D} \mathbf{n}_i^T)^K = \mathbf{n}_i \mathbf{D} \mathbf{n}_i^T$ for all K , so the contributions of eigenvectors with small v_i quickly drop out as these $(v_i)^K$ approach 0. This means that \mathbf{N}^K is dominated by the dimensions with v_i near 1. We start with a random vector, and quickly (6-10 iterations) produce such a mixture. Moody (2001) describes a procedure in which this process is repeated a number of times, each producing a slightly different mixture of the important eigenvectors (Fig 5). The results are then passed to a standard cluster analysis routine (such as k-means, Ward's method) to find any clusters of nodes.

Further analysis

The method of partial iteration of \mathbf{N} has been used for years in the program NEGOPY (Richards and Rice, 1981; Richards, 1995), as the first step in a more complex analysis. A key concept in NEGOPY is that of *liaisons*. These are nodes which do not have most of their connections with members of a cohesive cluster of nodes, but rather act as connections *between* clusters (Fig 3-4). Often it is the liaisons that provide the connections that hold the whole network together. Finding the liaisons requires detailed knowledge about the members of (potential) clusters and *their* connections, and is not an immediate result of a partition based on eigenvectors or clustering methods. Nevertheless, eigendecomposition methods – full or partial – are an excellent strategy to begin such analysis.

Future prospects

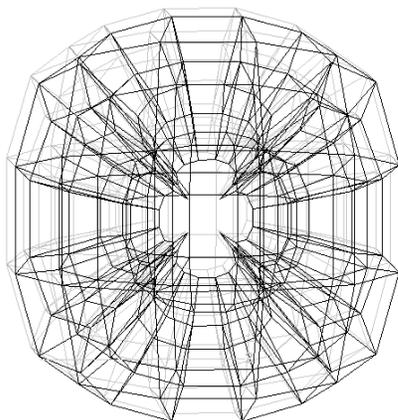
More work needs to be done on the categorization of networks based on important eigenpairs of \mathbf{L} and \mathbf{N} . Recent reports (Koren, *et al.*, 2002; Walshaw, 2000) suggest we might not need to resort to partial methods after all; we can find important eigenpairs exactly for enormous networks ($m > 10^5$) using “small” amounts of time and memory by first *reducing* the network in some way by sampling, solving the reduced eigenproblem, then interpolating back up with a very good “first guess” for the Power Method. Preliminary tests show that this should work equally well for Lanczos iteration.

References

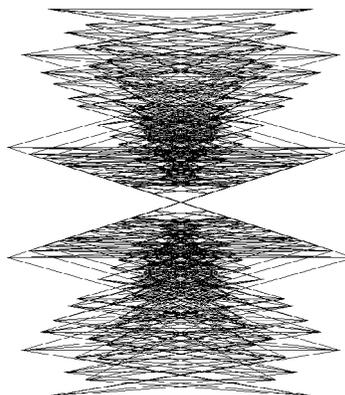
- Aho, A., Hopcroft, J., & Ullman, J. (1997). *Data structures and algorithms* Addison-Wesley.
- Alon, N. and Millman, V. (1985). λ_1 , Isoperimetric Inequalities for Graphs, and Superconcentrators. *J. Comb. Theory B.* 38: 73-88.
- Barnard S. and Simon, H. (1994). Fast Implementation of Recursive Spectral Bisection for Partitioning Unstructured Problems. *Concurrency: Practice and Experience* 6(2): 101-117.
- Benzecri, J-P. (1992). *Correspondence Analysis Handbook*, Marcel Dekker Inc.
- Bien, F. (1989). Constructions of Telephone Networks by Group Representations. *Notices of the Am. Math. Soc.* 36: 5-22.
- Biggs, N. (1993). *Algebraic Graph Theory*. Cambridge University Press.
- Brandes, U., Cornelsen, S. (2001). Visual Ranking of Link Structures, *Lecture notes in Computer Science*, 2125, Springer-Verlag.
- Chavel, I. (1984). *Eigenvalues in Riemannian Geometry*, Academic Press.
- Chow, T.Y. (1997). The Q-spectrum and spanning trees of tensor products of bipartite graphs", *Proc. Am. Math. Soc.*, 125(11): 3155-3161.
- Courant, R. and Hilbert, D. (1966). *Methods of Mathematical Physics*. Interscience Publishers.
- Cvetkovic, D., Doob, M., and Sachs, H. (1995). *Spectra of Graphs*. Academic Press.
- Cvetkovic, D., Rowlinson, P. (1990). The largest eigenvalue of a graph: a survey. *Linear and Multilinear Algebra*. 28: 3-33.
- Cvetkovic, D., Doob, M. Guttman, I, Torgasev, A. (1988). *Recent results in the theory of graph spectra*, North Holland.
- Chung, F.K.R.(1988). Diameters and Eigenvalues, *J. Am. Math. Soc.* 2(2): 187-196.
- Chung, F.K.R, Faber, V.& Manteuffel, T.A. (1994). An upper bound in the diameter of a graph from eigenvalues associated with its Laplacian, *SIAM J. Disc. Math.* 7(3): 443-457.
- Chung, F.K.R. (1995). *Spectral Graph Theory*, CBMS Lecture Notes, AMS Publication.
- Dhillon, I.S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning, UT CS Technical Report #TR 2001-05.
- Diaconis, P. and Stroock, D. (1991). Geometric Bounds for Eigenvalues of Markov Chains, *Ann. Appl. Prob.* 1: 36-61.
- Dodziuk, J. and Kendall, W. S. (1985). Combinatorial Laplacians and Isoperimetric Inequality. In K. D. Ellworthy (ed.). *From Local Times to Global Geometry*, Pitman Research Notes in Mathematics Series 150: 68-74
- Farkas, I., Derenyi, I., Barabasi, A-L., Viscek, T. (2001). Spectra of "Real-world" graphs: beyond the semi-circle law, cond-mat/100235
- Fiedler, M. (1973). Algebraic Connectivity of Graphs, *Czech. Math. J.* 23: 298-305.
- Fisher, M. (1966). On hearing the shape of a drum, *J. Combin. Theory*, 1: 105-125.
- Freeman, L. (1979). Centrality in social networks: Conceptual clarification, *Social Networks*, 1: 215-239.
- Freidman, J. (1993). Some Geometrical Aspects of Graphs and their Eigenfunctions. *Duke Mathematical Journal*, 69: 487-525.
- Garey, M., Johnson, D. (1979). *Computers and Intractability*. W. H. Freeman.
- Goh, K-I., Kahng, B. Kim, D. (2001). Spectra and eigenvectors of scale-free networks, cond-mat/0103337.
- Greenacre, M., (1984). *Theory and Application of Correspondence Analysis*. Academic Press.
- Grone, R., Merris, R. and Sunder, V. (1990). The Laplacian Spectrum of a Graph. *SIAM J. Matrix Anal. App.* 11(2): 218-238.
- Grone, R. and Merris, R. (1994). The Laplacian Spectrum of a Graph II. *SIAM J. Discrete Math.* 7(2): 221-229.
- Hagen, L. (1992). New Spectral Methods for Ratio Cut Partitioning and Clustering, *IEEE Trans. CAD*, 11(9): 1074-1085.
- Hall, K. (1970). R-dimensional Quadratic Placement Algorithm, *Management Sci.* 17: 219-229.

- Harary, F. & Schwenk, A. (1979). The spectral approach to determining the number of walks in a graph, *Pacific J. Math.* 80: 443-449.
- Hendrickson, B. and Leland, R. (1995). An improved spectral graph partitioning algorithm for mapping parallel computations. *SIAM J. Comput. Sci.* 16(2): 452-469.
- Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components, *J. Educ. Psychol.* 24: 417-441, 498-520.
- Jolliffe, I.T.. (1986). *Principal Components Analysis*, Springer-Verlag, New York.
- Kirchoff, G. (1847). Über die Auflösung der Gleichungen.... *Ann. Phys.Chem*, 72: 497-508.
- Koren, Y. Carmel, L., Harel, D. (2002). ACE: a fast multiscale eigenvector computation for huge graphs, *IEEE Symposium On Information Visualization*.
- Lawler, G. F. and Sokal, A. D. (1988). Bounds on the L^2 Spectrum for Markov Chains and Markov Processes: a Generalization of Cheeger's Inequality, *Trans. Amer. Math. Soc.* 309: 557-580.
- Lovasz, L. (1995). Random Walks, Eigenvalues and Resistance, *Handbook of Combinatorics*, Elsevier, 1740-1745.
- Lubotzky, A. (1994). *Discrete Groups, Expanding Graphs, and Invariant Measures*, Birkhauser.
- Mohar, B. (1991). The Laplacian Spectrum of Graphs. In Alavi, Chartrand, Ollermann and Schwenk (eds.). *Graph Theory Combinatorics and Applications*, Wiley, 871-898.
- Mohar, B., Poljak, S. (1991). Eigenvalues and the Max-cut Problem, *Czech. Math. J.* 40: 343-352.
- Moody, J. (2001). Peer influence groups: identifying dense clusters in large networks, *Socia Networks*, 23: 261-283.
- Parlett, B. (1980). *The Symmetric Eigenvalue Problem*, Prentice-Hall.
- Parlett, B., Simon, H. and Stringer, L. (1982). On Estimating the Largest Eigenvalue with the Lanczos Algorithm. *Mathematics of Computation.* 38(157): 153-165.
- Pothen, A., Simon, H. and Liou, K-P., (1990). Partitoning Sparse Matrices with Eigenvalues of Graphs. *SIAM J. Matrix Anal. App.* 11(3): 430-452.
- Powers, D. (1988). Graph partitioning by eigenvectors. *Linear Algebra Appl.* 101: 121-133.
- Richards, W.D. & Seary, A.J. (2000). Eigen Analysis of Networks, *J. Social Structure*.
<http://www.heinz.cmu.edu/project/INSNA/joss/index1.html>
- Richards, W.D. and Rice, R. (1981). The NEGOPY Network Analysis Program, *Social Networks*, 3(3): 215-224.
- Richards, W.D.(1995). *NEGOPY 4.30 Manual and user's Guide*. School of Communication, SFU.
<http://www.sfu.ca/~richards/Pdf-ZipFiles/negman98.pdf>
- Seary, A.J. & Richards, W.D. (2000). Negative eigenvectors, long paths and p^* , Paper presented at INSNA XX, Vancouver BC. <http://www.sfu.ca/~richards/Pages/longpaths.pdf>
- Seary, A.J. and Richards, W.D. (1998). Some Spectral Facts. Presented at INSNA XIX, Charleston.
<http://www.sfu.ca/~richards/Pages/specfact.pdf>
- Seary, A.J. and Richards, W.D. (1995). Partitioning Networks by Eigenvectors. Presented to European Network Conference, London. Published in Everett, M.G. and Rennolls, K. (eds). (1996). *Proceedings of the International Conference on Social Networks*, Volume 1: Methodology. 47-58.
- Simon, H. (1991). Partitoning of Unstructured Problems for Parallel Processing. *Computing Systems in Eng.* 2(2/3): 135-148.
- Sinclair, A. (1993). *Algorithms for Random Generation and Counting*, Birkhauser
- Walshaw, C. (2000). A multilevel algorithm for force-directed graph drawing, *Proc. Graph Drawing 2000*, Lecture Notes in Computer Science, 1984: 171-182.
- Walshaw, C. and Berzins, M. (1995). Dynamic Load-balancing for PDE Solvers on Adaptive Unstructured Meshes, *Concurrency: Practice and Experience.* 7(1): 17-28.
- Wasserman, S. & Faust, K. (1994). *Social Network Analysis*, Cambridge
- West, D. B. (1996). *Introduction to graph theory*, Prentice Hall, 1996.

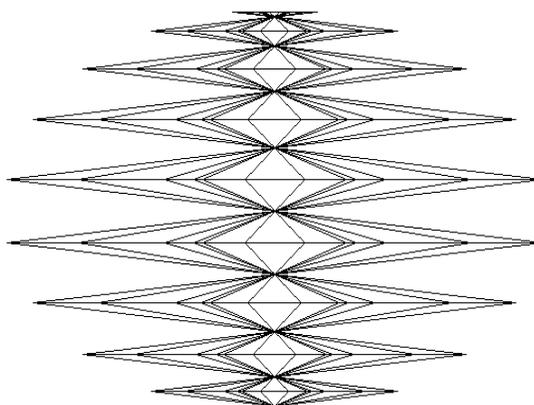
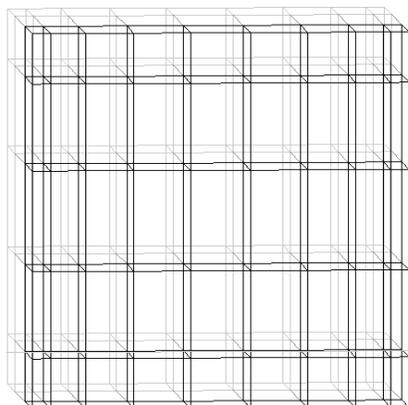
Cartesian sum



Kronecker product

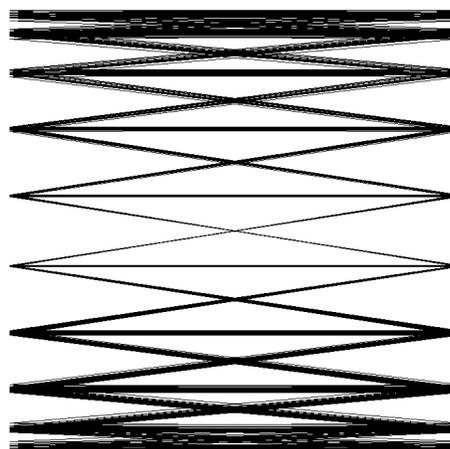
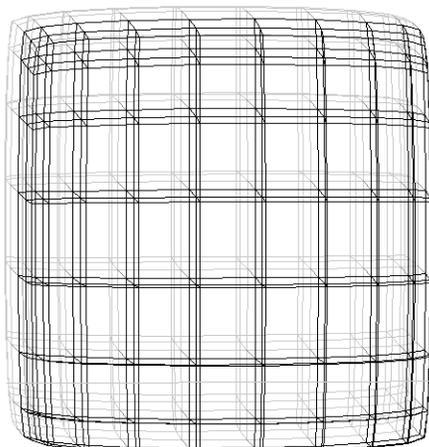


a) Using the eigenvectors of \mathbf{A} . With no “trivial” vector to combine with, the displays are distorted.



b) Eigenvectors of \mathbf{L} *behave well* under Cartesian sum. Coordinates of a 1-D path are replicated as straight lines.

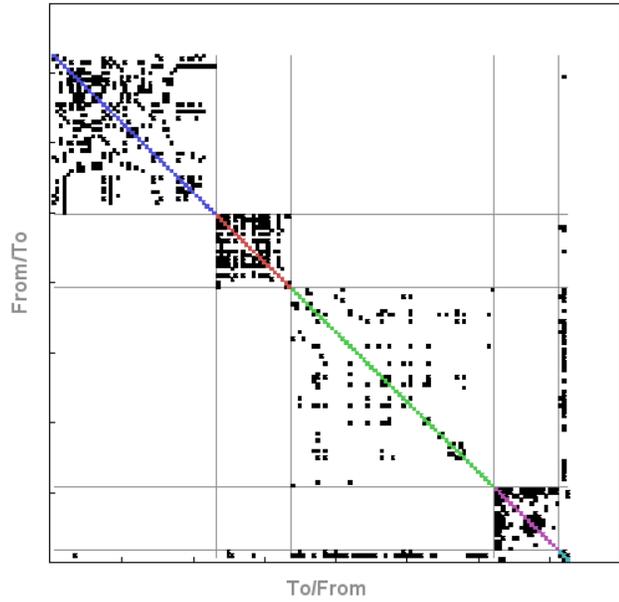
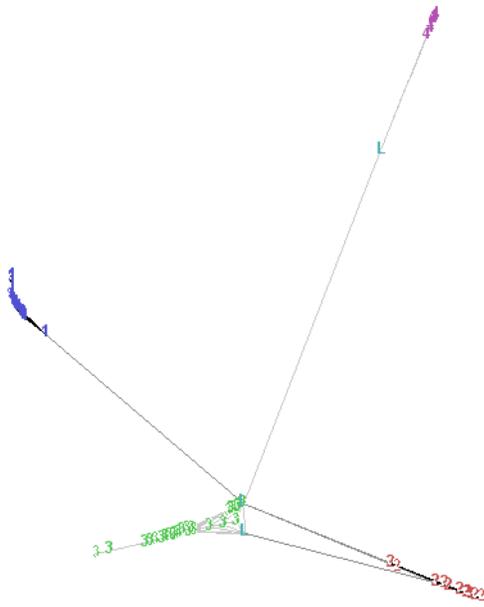
Eigenvectors of \mathbf{L} *do not* behave well under Kronecker product.



c) Using eigenvectors of \mathbf{N} . Lines are slightly curved, so the cube is slightly distorted.

Largest negative eigenvector of \mathbf{N} captures bipartiteness perfectly.

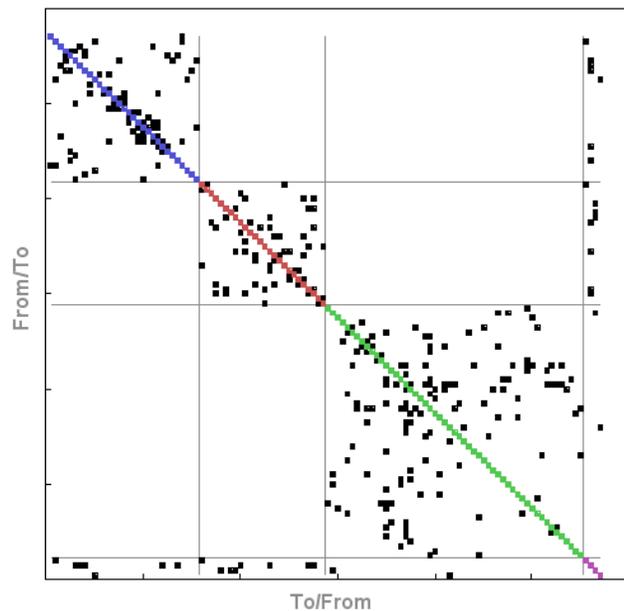
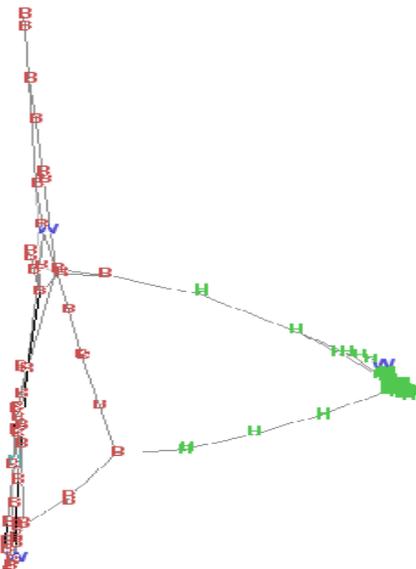
Figure 1. Six views of a 6 x 8 x 10 grid. Left: as the Cartesian sum of three paths producing a three-dimensional grid. Right: as the Kronecker product producing a bipartite graph.



a) The first 3 eigenvectors of \mathbf{N} produce clusters of nodes. These are labelled with the resulting partition into 4 blocks, along with the liaisons. The central liaison is of high degree and holds the network together. The figure is slightly rotated to show the clusters.

b) Adjacency matrix permuted by partition numbers. The blocks have no interconnections, and the network is held together by the liaisons (right and bottom).

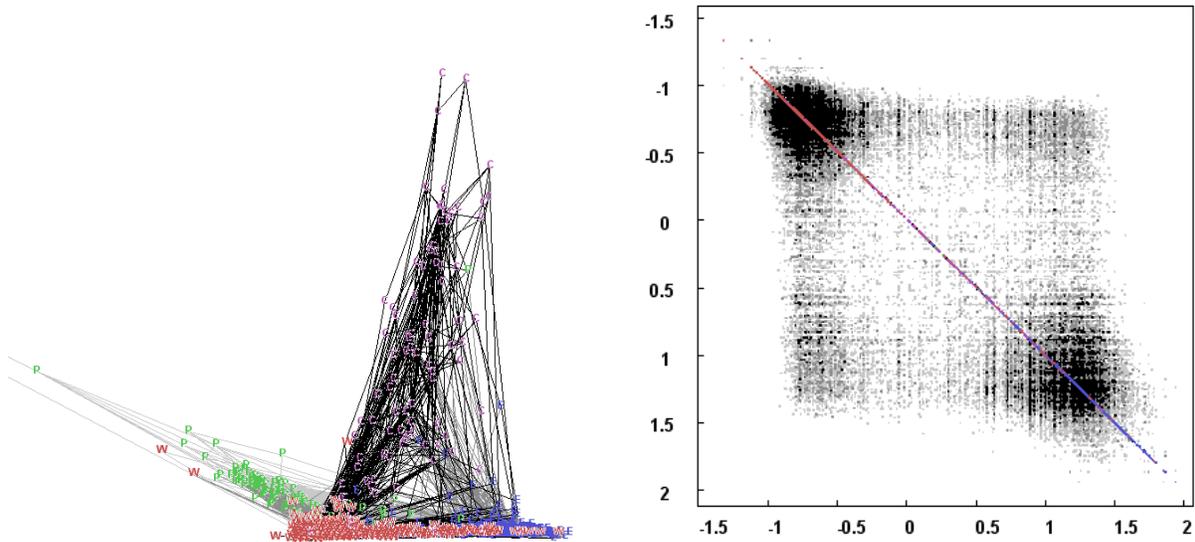
Figure 2. Two views of a small social network with 145 nodes. (Data Source: L. Koehly)



a) The first 3 eigenvectors of \mathbf{N} . The 3rd eigenvalue is negative, since there are very few triangles in the rightmost cluster. The labelling is by ethnic group, which shows a close relation to structure. There are only two connections between the two main ethnic groups.

b) Adjacency matrix permuted by partition numbers. In this case the liaisons are of *low* degree.

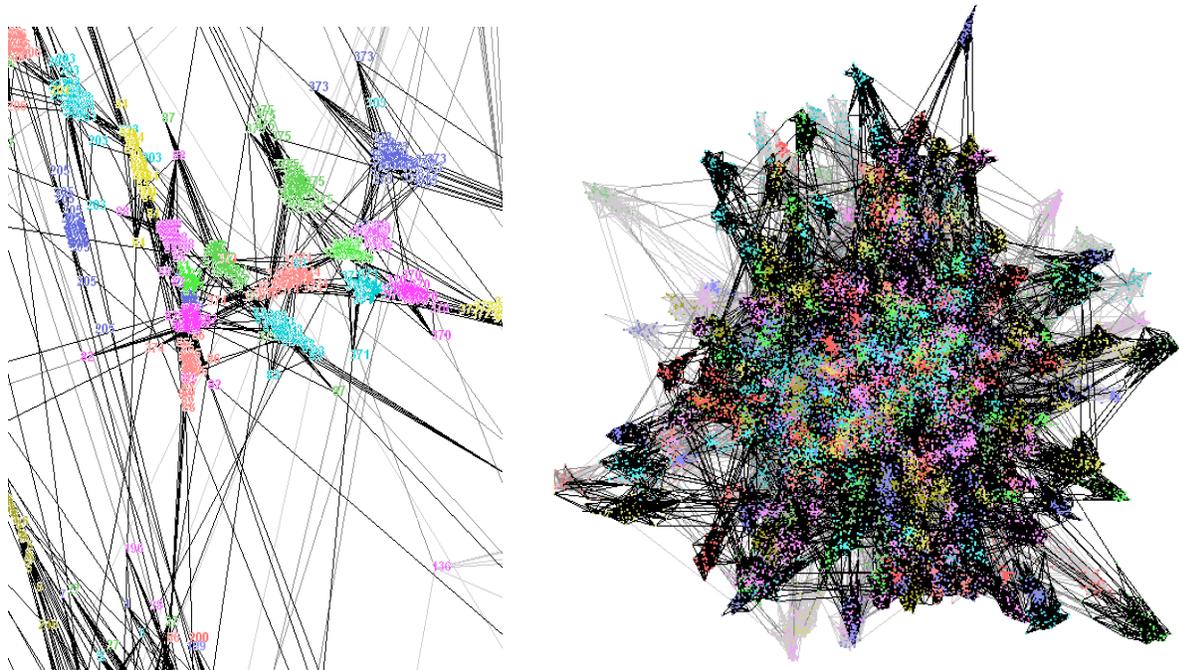
Figure 3. Two views of a small social network of drug users with 114 nodes. (Data Source: Scott Clair)



a. This figure shows the close relation between the first 3 eigenvectors of \mathbf{N} and four needle exchange sites, which are used to label the nodes. The figure is rotated to make this clear.

b. This figure is like an Adjacency matrix, except links are located by the *coordinates* of the first eigenvector of \mathbf{N} . The network is dominated by exchanges *within* sites E and W.

Figure 4. Two views of a needle exchange network. (Data Source: T. Valente and R. Foreman). This network is moderately large ($N=2736$) and roughly *scale-free* ($k \approx 1.7$). The eigenvectors of \mathbf{A} are dominated by nodes of high degree (>100).



a) Close-up of clusters formed by the first 3 eigenvectors of \mathbf{N} , labelled by construction.

b) Clusters formed by placing each node at the centroid of its neighbours, iterating 8 times with 3 random starts (multiple partial iteration). Labelled by construction.

Figure 5. A moderately large ($N=20,000$) artificial network (Data Source: J. Moody) constructed for testing purposes. The network was constructed from tightly connected groups of 50 nodes, each group then loosely connected in sets of 8, with 400 of these even more loosely connected into a single component.

APPENDIX (Glossary, examples, facts and definitions)

adjacency matrix: A network (graph) may be represented by a matrix of zeros and ones, with a one indicating that two nodes are connected (adjacent), and a zero otherwise. In a weighted graph, the ones may be replaced by other positive numbers (e.g., a distance or cost).

A sample adjacency matrix is shown below. See *link list*

	a	b	c	d	e	f	g	h	
a	0	1	1	1	0	0	0	0	a has connections to b,c,d
b	1	0	1	1	0	0	0	0	b has connections to a,c,d
c	1	1	0	1	0	0	0	0	...etc...
d	1	1	1	0	0	0	0	0	
e	0	0	0	0	0	0	1	1	
f	0	0	0	0	0	0	1	1	
g	0	0	0	0	1	1	0	0	
h	0	0	0	0	1	1	0	0	

Adjacency spectrum: The adjacency matrix of a graph, like any matrix, may be subject to an eigen decomposition. In graph theory, the resulting set of eigenvalues is referred to as *the* graph spectrum, in analogy to the continuous spectrum from continuous spectral analysis methods such as Fourier analysis. In Fourier analysis, the spectrum is understood to refer to the weighting of sines and cosines, whereas the discrete graph spectrum (eigenvalues) are weights of eigenvectors with unknown functional form. We sometimes use the term eigenpair refer to both eigenvalues and eigenvectors. Since there are other spectra associated with graphs, we refer to this one as the Adjacency or Standard spectrum.

block: A block may be contrasted with a clique in the sense that the former are defined as sets of nodes that have similar patterns of links to nodes in other sets, while the latter is a set of nodes that have most of their links to other nodes in their set. All cliques are blocs, but some blocks are not cliques. One of the aims of blockmodelling is to identify roles by clustering the nodes so that those with similar patterns of connections are next to one another in the matrix. The members of each block perform similar roles in the network.

block model: a higher-level description of a network, where roles (or blocks) are represented by a simplified graph. For the matrix above, a block model would be:

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{matrix}$$

Cartesian sum: A form of graph composition, which forms more complex graphs from simpler ones. Cartesian sum may be expressed in terms of Kronecker product as:

$$\mathbf{A}_1 \oplus \mathbf{A}_2 = \mathbf{A}_1 \otimes \mathbf{I}_2 + \mathbf{A}_2 \otimes \mathbf{I}_1 \quad (\text{where } \mathbf{I}_1 \text{ and } \mathbf{I}_2 \text{ are identity matrices of appropriate size})$$

As an example, the Cartesian sum of two paths is a rectangular grid.

clique: In graph theory, a clique is a sub-graph in which all nodes are connected to each other. In social networks, a clique is a set of nodes with most of their connections with other members of the clique. This would generally correspond to an informal role (e.g, friendship). In the above matrix {a,b,c,d} form a clique.

cluster: A collection of points that are “close” to each other in some sense. Many definitions (and related techniques) are available. For networks, we should also insist that the points share connections, either within the cluster (clique) or with another cluster (see block model).

component: If a graph is connected, it consists of a single component. A disconnected graph does not have a path between any pair of nodes, and may consist of several components.

connected: If there is a path between every pair of nodes in a graph, the graph is said to be connected. A disconnected graph does not have a path between any pair of nodes, and so distances (and diameters) cannot be defined, except within each component.

distance: For graphs, the distance between nodes is defined as the smallest number of links connecting them. Also called geodesic distance.

diameter: the largest geodesic distance between any pair of nodes in a graph

gap: The term gap or spectral gap refers to large distances in the spectrum of eigenvalues, particularly between 0 and the second-smallest (Laplacian) or between 1 and the second-largest in absolute value (Normal). A small gap means that a graph can be disconnected with few edge-cuts; a large gap means there are many paths between sets of nodes.

global vs local methods: In graph theory, a local method is one that examines only a few neighbours of a node. A global method is one which examines the entire graph, such as an eigendecomposition.

Kronecker product: A form of graph composition, which forms more complex graphs from simpler ones. An example of the Kronecker product is:

$$\begin{matrix} 1 & 1 & & 1 & 0 \\ 1 & 0 & \otimes & 0 & 1 \end{matrix} = \begin{matrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{matrix}$$

where every 1 in the first matrix has been replaced by a complete copy of the second matrix. In this example the first matrix is a block model, not a graph.

Lanczos iteration: a generalization of the power method which allows calculation of a specified number of eigenpairs without loss of precision or orthogonality. Currently one of the best methods for eigendecomposition of large systems.

Laplacian spectrum: The eigenvalues (and eigenvectors) of a matrix formed by subtracting the adjacency matrix from a diagonal matrix of node degrees. The eigenvalues are non-negative, with a "trivial" (constant) eigenvector of eigenvalue 0. This discrete analogue of the continuous Laplacian shares a great many of its important properties. For this reason, it has become the focus of much research in the last decade.

liaisons: according to NEGOPY, these come in two types. *Direct* liaisons are individuals who have most of their interaction with members of groups, but not with members of any one group. They provide direct connections between the groups they are connected to. *Indirect* liaisons are individuals who do not have most of their interaction with members of groups. They provide indirect or 'multi-step' connections between groups by connecting Direct Liaisons, who have direct connections with members of groups (Richards, 1995).

link: A pair of nodes with some connection between them. In graph theory, links are also called edges or lines. In social networks, links are often called ties.

link list: A sparse format for storing information in a network. Only the pairs of nodes that are connected are in the link list. For symmetric graphs, only one pair is needed for each link. For weighted graphs, a third column may be used to hold the weights. For the symmetric adjacency matrix shown above, the link list is:

- 1 2
- 1 3
- 1 4
- 2 3
- ...and so on...

localized: As applied to an eigenvector means that most of the coordinates are near zero, and only a few have large values. Coordinates may be either positive or negative, and the eigenvectors are normalized to

make the sum of squares of components 1, so the sum of 4th powers is generally used as a measure of localization. If this sum is near 1 only a small number of coordinates are important. If it is near 1/m, then all nodes contribute to the eigenvector.

neighbourhood: all the nodes which are connected to a given node. May be extended to all nodes connected to a *set* of nodes, but not including the original set.

NEGOPY: (NEGative entrOPY) (Richards and Rice, 1981, Richards, 1995) is a computer program designed to find clique structures. It uses a random starting vector, and multiplies it by the row-normalised adjacency matrix, subtracting off row means. Usually 6-8 such iterations are performed, resulting in a vector which is a mixture of the the important Normal eigenvectors (Richards and Seary, 1997). This vector is then scanned for potential clique structures, which are tested against the original network and for some statistical properties (e.g., variance in the node degrees). Sparse matrix methods are used throughout, allowing large networks to be analysed rapidly.

node: An object that may have some kind of connection (link) to another object. In some cases, nodes are people, organizations, companies, countries, etc. In graph theory nodes are also called vertices and points. In social networks, nodes are often called actors.

normal spectrum: The eigenvalues (and eigenvectors) of a row-normalised adjacency matrix. This matrix is row-stochastic, and similar to a symmetric matrix, so its eigenvalues are real and less than or equal to 1 in absolute value. It is closely related to the Laplacian (indeed, it may be defined to be the Laplacian in the χ^2 metric defined by the node degrees).

partition: A partition of a graph is a division of the nodes into a collection of non-empty mutually exclusive sets. A partition of the adjacency matrix shown above could be: {a,b,c,d}, {e,f,g,h}, so that there are no links between the nodes in each part of the partition.

sparse matrix techniques: In analysis of networks with more than 50 or 60 members, it is usually the case that each node is connected to only a fraction of the others. The adjacency matrix for such networks contain mostly zeroes, which indicates the absence of links. In these situations, it far is easier to work with a list of the links (link list) that are present, rather than the whole matrix which contains many times more numbers. Any array (such as an adjacency matrix) which consists mostly of some default number (usually zero) may be treated as a sparse matrix. Since this value is known, it does not need to be stored as part of the array. This allows the array to be stored in a much more efficient manner, e.g., for an adjacency matrix, we only need to store the links (pairs of nodes) when they exist. For a weighted adjacency matrix, we also need to store the values of the weights, one for each link. Many matrix operations (e.g., multiplying a matrix by a vector) can utilize this more efficient storage to run much faster as well. Sparse matrix techniques are those which avoid any manipulation of the matrix that would affect the sparseness property (e.g., taking the inverse will generally do this, as will correlating each row or column with all the others). It is quite possible to find eigenvalues and eigenvectors using sparse techniques.

spectral analysis or methods: Loosely speaking, another term for eigendecomposition. Mathematically speaking, a general term referring to any re-statement of some function in terms of a set of basis functions (e.g. sines and cosines for Fourier analysis). The spectrum is the weights of these basis functions. The Fourier transform is especially useful in mathematical physics since the sines and cosines (or e^z for complex z) are eigenfunctions of the ubiquitous derivative and integral operators. The terms function, operator and eigenfunction have the discrete analogues of vector, matrix and eigenvector.

Standard spectrum: see Adjacency spectrum