ELSEVIER

Social Networks 23 (2001) 261–283

SOCIAL NETWORKS

www.elsevier.com/locate/socnet

Peer influence groups: identifying dense clusters in large networks

James Moody*

Department of Sociology, The Ohio State University, 372 Bricker Hall, 300 North Oval Mall, Columbus, OH 43210, USA

Abstract

Sociologists have seen a dramatic increase in the size and availability of social network data. This represents a poverty of riches, however, since many of our analysis techniques cannot handle the resulting large (tens to hundreds of thousands of nodes) networks. In this paper, I provide a method for identifying dense regions within large networks based on a peer influence model. Using software familiar to most sociologists, the method reduces the network to a set of *m* position variables that can then be used in fast cluster analysis programs. The method is tested against simulated networks with a known small-world structure showing that the underlying clusters can be accurately recovered. I then compare the performance of the procedure with other subgroup detection algorithms on the MacRea and Gagnon prison friendship data and a larger adolescent friendship network, showing that the algorithm replicates other procedures for small networks and outperforms them on the larger friendship network.

Keywords: Friendship; Social network; Peer influence; Cohesion; Methods

1. Introduction

Early social network theorists argued that the power of social networks lies in large-scale connectivity (Pool and Kochen, 1978; Rapoport and Horvath, 1961). The extended effects of social networks are clear when we consider the spread of diseases, such as HIV/AIDS, that have crossed the globe (mainly) through an intimate but far-reaching social network. A tradition of work on the small-world problem similarly rests on large-scale connectivity, which has been shown to have potentially important consequences for information and large scale coordination (Watts and Strogatz, 1998; Kochen, 1989; Milgram, 1969). Work on large social networks promises to provide new empirical support for visions of social

* Tel.: +1-614-292-1772; fax: +1-614-292-6687.
E-mail address: moody.77@osu.edu (J. Moody).

0378-8733/01/$ – see front matter
PII: S0378-8733(01)00042-9

class and structure that rest on interaction patterns (marriage, work relations, and informal association) instead of nominal categories (Warner et al., 1963; White, 1965).

For the 30 years or so since these early insights, however, the vast majority of empirical network research has focused on small (less than 100, usually less than 50 nodes) networks (for discussion, Wellman, 1988). Recently, our ability to collect data on large social networks has outstripped our capacity to meaningfully analyze such networks. Most tools developed to analyze networks were developed for small networks and run into significant computational barriers in large networks. Working on graphs with over 10,000 nodes, for example, is cumbersome with most social network packages. [1]

Two classical social network theories provide insights that can help analyze large networks. First, the small-world literature has shown that while most of our acquaintances tend to be acquainted with each other, short acquaintanceship chains (relative to the size of the network) link most pairs in the network. This high degree of local clustering suggests that a practical approach to studying the structure of large networks would involve first identifying local clusters and then analyzing the relations within or between clusters. Second, we know from work on peer influence that people tend to be similar to each other. Based on an endogenous influence process, close friends tend to converge on similar attitudes (Friedkin, 1998) and thus clusters in a small-world network should be similar along multiple dimensions.

In what follows, I show how one can use an endogenous peer influence model to identify clusters of closely related actors in large networks. The resulting algorithm is computationally efficient and can be implemented with programs familiar to most social scientists. After providing a set of definitions for terms used throughout the paper, I provide a short background to the problem of identifying dense sub-regions within networks, explaining why these procedures tend not to be useful for large networks. I then review the relevant details of a peer influence model and present the resulting recursive neighborhood mean (RNM) algorithm for identifying peer groups. I then test the RNM procedure on large simulated networks with a known structure and observed networks of prisoner and adolescent friendships.

2. Background

2.1. Definitions

I represent a social network as a finite graph, $\boldsymbol{G}(\boldsymbol{V}, \boldsymbol{E})$, where people are represented by $\boldsymbol{V}$, the set of $|v|$ vertices, and relations by $\boldsymbol{E}$, the set of edges composed of pairs of vertices. An actor i is adjacent to actor j if $(v_i, v_j) \in \boldsymbol{E}$. [2] The set of all nodes adjacent to node i is that

[1] This limitation is becoming less serious as new network packages develop. PAJEK (Batagelj and Mrvar, 2001) is designed to work with very large networks, and while the standard PC version of NEGOPY is hard-coded to 1000 nodes, the authors can extend the program to handle networks of more than 30,000 nodes. Multinet (Richards and Seary, 2000) and UCINET (Borgatti et al., 1999) both have data capacity limited to computer memory, with size limitations that vary by analysis technique.

[2] I assume that actors do not relate to themselves and thus $(v_i, v_j) \notin \boldsymbol{E}$.

actor's neighborhood. A path in the network is defined as an alternating sequence of distinct nodes and edges, beginning and ending with nodes, in which each edge is incident with its preceding and following nodes. Actor i can reach actor j, if there is a path in the graph starting with i and ending with j. The length of the path from i to j is equal to the number of edges in the path, and the shortest path connecting any two vertices is the geodesic. If there is a path linking every pair of actors in the network then the network is connected. In general, a set is maximal with respect to a given property if it has the property but no proper superset does. A component of a graph $\boldsymbol{G}$ is a maximal connected subgraph of $\boldsymbol{G}$. A bicomponent is a maximal connected subgraph of $\boldsymbol{G}$ in which every pair of nodes is connected by at least two paths that overlap by only the start and end nodes. A clique is a maximal subgraph of $\boldsymbol{G}$ in which every pair of actors in the subgraph is adjacent. The level of clustering in a graph relates to how uniformly ties are distributed throughout a network. When ties are concentrated within subgraphs, the network is clustered. A network is said to have a small-world structure when it is clustered and the average distance among all pairs (the characteristic path length) is close to that of a random graph of similar size and density.

2.2. The problem

If most large networks admit to a small-world structure, then a reasonable analysis strategy for large networks involves first identifying the local clusters. Once such clusters have been identified, one can then analyze the internal structures of the clusters or relations among the clusters to get a picture of the global network structure. While we suspect that most social networks are highly clustered, and can show that large graphs conform to such a structure with respect to certain global parameters (such as path length), identifying the relevant local clusters is daunting. We thus need an efficient group detection technique that will provide a reasonable first cut on the structure of these large networks.

The simplest form of network clustering is based on connectivity (Harary, 1969; White, 1998). Substantively, components and bicomponents are minimum requirements for primary groups, ensuring that the identified groups are connected and, if at least a bicomponent, structurally cohesive (Moody and White, 2001). Non-polynomial time algorithms are available to identify components and bicomponents, and exist for tricomponents. While low-polynomial time algorithms exist for identifying higher k-connected components (Moody and White, 2001), they are still impracticably time consuming for networks with tens of thousands of nodes. Usually, a single giant component (or bicomponent) that contains almost every node in the network dominates large graphs beyond a certain density (Palmer, 1985). [3] Thus, while identifying components in the graph is a necessary first step in any analysis, it often does not meaningfully reduce the complexity of large networks.

Most extant procedures for identifying dense clusters beyond low-level connectivity in graphs either search the network for specific graph–theoretic features (such as cliques, k-cores, or k-plexes), or iteratively assign nodes to groups until an optimum index of network clustering (such as the ratio of ties within groups to ties between groups) is found (Alba, 1973; Borgatti et al., 1999; Fershtman, 1997; Frank, 1995). These techniques tend not to

[3] In fact, the reachability properties associated with small-world graphs depend on the fact that a small number of random links in the overall network results in a relatively high global connectivity.

be useful for large networks because they are either computationally inefficient or do not substantively identify primary groups (or both). A more computationally efficient method would use summaries of the network structure to cluster nodes within an analytic space, such that nodes with many common partners are situated close to each other. One can then use pattern recognition algorithms or cluster analysis to identify groups based on these position (Richards, 1995).

Most graph theoretic approaches to identifying dense clusters in a network start with fully connected cliques. Cliques are not useful for large networks; being both computationally inefficient (requiring exponential running time to find) and often identifying groups that either heavily overlap or miss substantively important groups that are not completely connected. While the overlapping structure of cliques can be used to identify more loosely connected groups in small networks, the analysis procedure is cumbersome for large networks (Freeman, 1996).

These limitations have led methodologists to relax the clique requirement. [4] An obvious starting point for relaxing the requirement of cliques is to focus on the number of people to which each node is connected. A *k*-core is a maximal subgraph in which every node is adjacent to *k* other nodes in the subgraph. While computationally efficient, when the network has a small-world structure, ties between dense local regions confound *k*-cores by cutting loosely connected members of subgroups from the group and placing highly connected members of different subgroups together. Other graph-theoretic measures, such as *k*-plexes, or *k*-clans, that are based on degree or path criteria, are similarly limited (and more costly to calculate) making them unwieldy for large networks.

The second common approach is more direct, seeking to identify dense regions of the network by searching through various group assignments until a parameter that summarizes the clique structure (such as the ratio of within-group to between group contact) is maximized (Borgatti et al., 1999; Frank, 1995). While theoretically appealing, such procedures require assigning nodes to classes and then repeatedly moving nodes from one group to another until an optimum partition is identified. The iterative nature of these procedures is often very time consuming. [5]

Given the wealth of new data on large social networks, researchers have the opportunity to test the initial global connectivity insights of early network theorists and extend social network research beyond the local group to larger communities. To do so, however, we need a computationally efficient and substantively accurate procedure to partition a network into dense regions. The most analytically useful procedures for large graphs would assign nodes to single groups in a manner that is consistent with what we know about small-world networks. The goal is to provide researchers with a first cut on the network: once these groups have been identified, one can then apply the previously developed measures to the smaller regions of the graph and identify subtle distinctions (cluster overlaps, bridges, etc.) in the global network structure.

[4] See Moody and White (2001) for an extended discussion of many of these measures.

[5] For example, clustering a 300 node network from Add Health with UCINET V's FACTIONS routine took about 7 h. An attempt to identify clusters in 1000 node network failed to converge in less than 3 days and was cancelled.

2.3. A peer-influence analogy

Much social network research has focused on peer group similarity (Billy et al., 1984; Cohen, 1983; Kandel, 1978; McPherson and Smith-Lovin, 1987), finding that members of close peer groups tend to be similar along multiple dimensions. While multiple selection and focal factors likely account for some of this observed similarity (Feld, 1981), much is likely due to a dynamic and endogenous influence process (Friedkin, 1998; Friedkin and Cook, 1990; Friedkin and Johnsen, 1997). This peer influence model suggests that people adjust their opinions and attitudes based on the opinions and attitudes of their close associates. Consequently, groups of people that are tightly connected within clusters tend to have similar opinions.

This process can be modeled as an iterative adjustment sequence, where each person takes account of the attitudes of their peers and adjusts their own accordingly. Formally, the system can be modeled with two equations:

$$\boldsymbol{Y}^{(\mathrm{I})} = \boldsymbol{XB} \tag{1}$$

$$\boldsymbol{Y}^{(t)} = \boldsymbol{AWY}^{(t-1)} + (\boldsymbol{I} - \boldsymbol{A})\boldsymbol{Y}^{1} \tag{2}$$

where $\boldsymbol{Y}$ is an $N \times m$ matrix of opinions, $\boldsymbol{X}$ an $N \times k$ matrix of k exogenous variables that influence opinions through the set of $\boldsymbol{B}$ coefficients, $\boldsymbol{A}$ the diagonal matrix that represents the relative weights of endogenous interpersonal influence, and $\boldsymbol{W}$ is an $N \times N$ matrix of interpersonal influence based on the network contact structure (Friedkin, 1998, Chapter 2). If we ignore the exogenous influences (as I will for the purposes of the algorithm below), the above reduces to:

$$\boldsymbol{Y}^{(t)} = \boldsymbol{WY}^{(t-1)} \tag{3}$$

The most common way to construct $\boldsymbol{W}$ is to row normalize the adjacency matrix such that each of ego's contacts has influence proportional to contact volume (Friedkin and Cook, 1990). In systems, where there are local clusters of relations, the process defined in Eq. (3) will generate homogeneity within groups. If the initial opinions are also uncorrelated, then within group homogeneity across multiple dimensions will yield unique opinion combinations that define particular groups. Thus, if we simulate the peer influence process in a network, but set the initial opinions to be random, then each dense region of the graph will come to occupy a unique position in the m-dimensional space defined by $\boldsymbol{Y}$, the set of opinion variables.

Alternative peer influence models are available in the literature that incorporate differences in the information available to actors or how they know it (Frank and Fahrbach, 1999; Mark, 1998; Chaiken and Stangor, 1987). Frank and Fahrbach argue, for example, that if actors are influenced by exposure to information, as opposed to normative pressure, then only those paths that bring new information are salient. This approach leads to an influence model that focuses on particular paths that convey information at a particular time, reducing the potential redundancy found in simpler models. As with other alternative peer influence models, Frank and Fahrbach's approach avoids predicting too much within-group similarity. Using an attenuation parameter (δ), they are able to decrease the extent of interpersonal influence and anchor attitudes in a manner that prevents extreme opinion formation (p. 262).

This attenuation and anchoring allows the models to more accurately capture within group attitude variance. However, while model details vary, all agree that actors who share salient ties tend toward similarity, which is the property I wish to exploit for finding groups in a network. [6] To be most effective at uncovering dense regions in the network, any computationally efficient model that generates maximal within-group similarity will work. [7]

3. An influence based algorithm

3.1. Recursive neighborhood means (RNMs)

The algorithm I propose mimics Friedkin's endogenous influence process for a set of random variables, and is presented in the following Box 1. [8]

Box 1. Recursive neighborhood mean (RNM) algorithm

1. Assign each person in the network a random number on each of m variables, $\boldsymbol{Y}$.
2. Do t times.
3. Reset each person's value(s) for $\boldsymbol{Y}$ to the mean of their contacts.

Initially, each node is assigned a uniform random value between 0 and 1 for each of the m columns in $\boldsymbol{Y}$. At every iteration, each node's value (Y_{imt}) is replaced by the mean of their contact's values:

$$Y_{imt} = \frac{\sum_L Y_{imt-1}}{|L|} \tag{4}$$

where i indexes nodes, m indexes dimensions, t indexes the iteration number and L is the set of $|L|$ people ego is adjacent to in the graph. Note that one can introduce multiple dimensions by expanding m. If the network is stored as an adjacency list and the m "opinions" stored in a similarly sorted dataset, $\boldsymbol{Y}_t$ can be calculated quickly (Gibbons, 1985). Each iteration of the RNM procedure visits each node once, pulling the values of $\boldsymbol{Y}$ for the node's neighbors. This means that the number of operations for each iteration is Nd, where N is the number of nodes and d is the average degree in the network. [9]

[6] Setting Frank and Fahrbach's attenuation parameter to one recovers the Friedkin model (Frank and Fahrbach, 1999, p. 261).

[7] In some circumstances, one might have network data that could trace information flow, such as email exchanges in a large corporation. When such data are available, an alternative model based on information exposure might be of more interest.

[8] All examples of this algorithm used in this paper were done in SAS, and sample code can be found at http:/www.sociology.ohio-stat.edu/jwm/largeClusters/index.htm.

[9] It is possible to get the resulting values based on the peer influence equations, since $\boldsymbol{Y}^{\infty} = \boldsymbol{W}^{\infty}\boldsymbol{Y}^{1}$, which under some conditions can be estimated based on Eq. 2.11 given in Friedkin (1998). However, for large graphs, it is faster to simulate $\boldsymbol{Y}$ than to store and manipulate the matrices required to estimate $\boldsymbol{Y}^{\infty}$. A similar approach is used in the first pass of NEGOPY (Richards, 1995). NEGOPY, however, differs in that it (a) limits the assignment procedure to one dimension and (b) uses a weighted mean based on relationship strength and the number of two-step links connecting each pair of nodes. This type of weighting could be incorporated in the RNM algorithm.

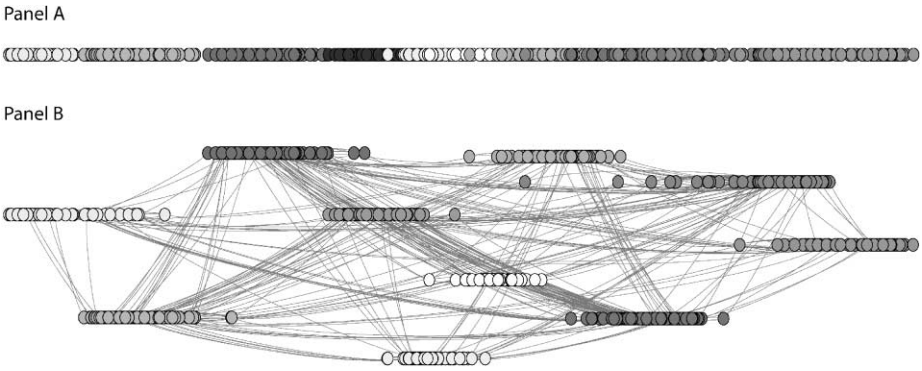

Fig. 1. One-dimensional RNM layout.

As an example, consider the network presented in Fig. 1.[10] The network in Fig. 1 contains 1800 nodes in a highly clustered network. Clusters range in size from 100 to 300 nodes. In Fig. 1A, the network is arranged based on one dimension of the process outlined in Box 1, distributing cases along a single line. This process is similar to that used to array initial clusters in NEGOPY. Some clear clustering is evident, as can be seen by the gap in the left side of the network. In Fig. 1B, the horizontal dimension is maintained, but the known clusters are arbitrarily distributed across the vertical axis, allowing one to see how well known clusters converge on the same area of the line. Clearly, most cluster members are close to each other in the horizontal dimension, indicating that the RNM process generated agreement within clusters. However, while internal consistency is high, many clusters occupy a similar position on the line. As such, applying a pattern recognition algorithm to a single dimension would be unable to satisfactorily distinguish two clusters from each other since there is simply no information that uniquely identifies the groups.

In Fig. 2, a second RNM dimension is added to the figure. The horizontal axis is the same as that presented above, but the vertical axis is determined by the second RNM position variable.

Because the initial random variables input into the RNM procedure are uncorrelated, the probability that any two clusters will converge on the same portion of the resulting space is small. Instead, the resulting dimensions clearly separate nodes into distinct regions of the variable space. While two dimensions are sufficient for the 10 clusters in the comparatively simple network above, as the size increases by orders of magnitude, a problem similar to the one-dimensional case can arise, and we need to increase the number of dimensions used to identify clusters.[11]

[10] All network figures are drawn using PAJEK (Batagelj and Mrvar, 2001). Full color versions of the figures can be found at http://www.sociology.ohio-state.edu/jwm/largeClusters/index.htm.

[11] The graphical representation in Fig. 2 suggests a natural link between this procedure and multi-dimensional scaling techniques (Weller and Romney, 1990). When the number of nodes is small, it may be useful to use the graphical representation to identify dense regions in the graph, though this quickly becomes unwieldy as the number of nodes increases.

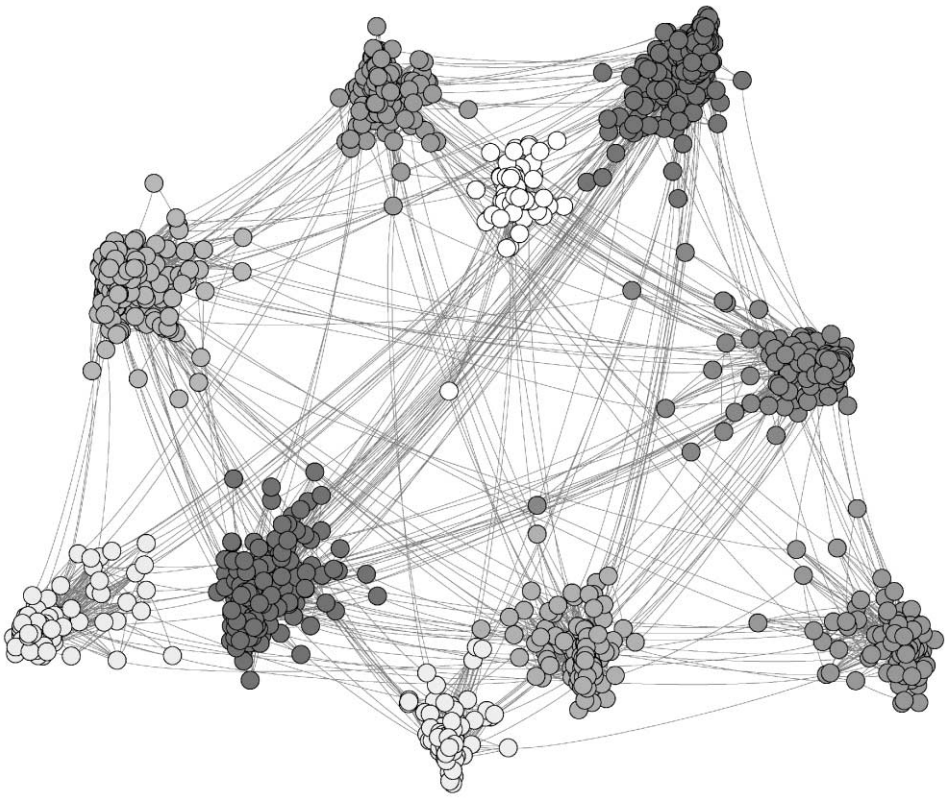

Fig. 2. Two-dimensional RNM layout.

3.2. Identifying clusters

Once one has generated $\boldsymbol{Y}$, clusters can be identified using cluster analysis (see Wasserman and Faust, 1994, pp. 381–385, for similar examples). This raises two related questions: Are there any clusters in the data? And, if so, how many clusters are there? In general, theory should guide answering the first question, as the algorithm is intended to apply in settings where a small-world structure is suspected. Empirically, one can often determine whether a graph has a small-world structure with the clustering coefficient, $\boldsymbol{C}$, defined as the average fraction of pairs of neighbors of a node which are also neighbors of each other, and the characteristic path length, ℓ, defined as the average distance between pairs of nodes in the network (Watts, 1999), both of which can be compared to random expectations for particular graph models (Newman, 2000). Alternatively, one can use the triad census and corresponding random graph distributions (Wasserman, 1977) to determine the underlying topology of the network (Johnsen, 1985, 1986). Recent software advances have made computing the triad census for large graphs straightforward (Batagelj and Mrvar, 2001; Moody, 1998). Finally, when the graph is small enough to make generating multiple random graphs

feasible, one can use a statistical model to compare observed cluster solutions to solutions from similar random graphs (Frank, 1995).

Assuming the network contains clusters, one must determine the number of clusters, which can be tricky. Wasserman and Faust (1994) make a pragmatic argument, saying,"... the 'trick' is to choose the point along the series [cluster partition sequence] that gives a useful and interpretable partition ... Theory is the best guide." (p. 383). In the absence of strong theoretical expectations for a particular number of clusters, some statistical guides are available within the cluster analysis literature (for a review of multiple criteria, see Milligan and Cooper, 1985; Milligan, 1996; Koehly, 2001), but these often result in a small number of very large clusters. In many cases, the dendrogram produced by the clustering procedure will suggest a natural number of clusters, which is often the most pragmatic solution.

Alternatively, one can explore multiple partitions based on the cluster hierarchy and choose a partition that optimizes a network clustering index, such as Freeman's (1972) segregation index. Hierarchical clustering methods place individuals who are close to each other in $\boldsymbol{Y}$ together in a group. These groups are then joined together based on how close they are to each other creating a new group, and so forth until all members are held in a final group. This results in a tree, starting at the bottom with each individual and ending at the top with every individual in one set. At each stage, one can evaluate the extent of clustering for each group. Starting low in the partition tree, we walk up each branch, and at each joining point ask whether combining the two groups into one improve the fit for each. If so, then join them, else stop along that branch of the tree.

Freeman's segregation index is a useful measure for such an approach. Freeman defined network segregation as the difference between the number of observed cross-group ties and the number expected under random mixing. A network with many dense clusters would be highly segregated. For any group A, the segregation index, SEG(A) would be:

$$\mathrm{SEG(A)} = \frac{E(X) - X}{E(X)} \tag{5}$$

where $E(X)$ is the expected number of contacts between group A and not A, calculated based on the marginal values of the group mixing matrix, and X is the observed number of contacts between group A and not A.[12] If all ties were sent within group, then the index would $= 1$, if ties were sent between groups at a level equal to random chance, then the index $= 0$. Fig. 3 provides an example of this tree-walking procedure on a starting partition of 20 groups. Values within the nodes indicate the segregation index, and shaded nodes illustrate where we would identify a group.

This approach to identifying the number of clusters in the data does not assume that groups are formed equally well at a given level of the partition hierarchy. Thus, we may cut the tree at lower levels along one branch than we do another. This is useful in situations where groups are of different sizes or densities; elements that confound many standard stopping rules.

[12] An alternative to the segregation index is the group mixing odds ratio (Mosteller, 1968), which has the advantage of being margin free, which would be useful if you suspect groups of very different sizes or interaction levels. One could also follow Frank (1995), and use a log-linear framework to test model improvements at each level of the hierarchy.

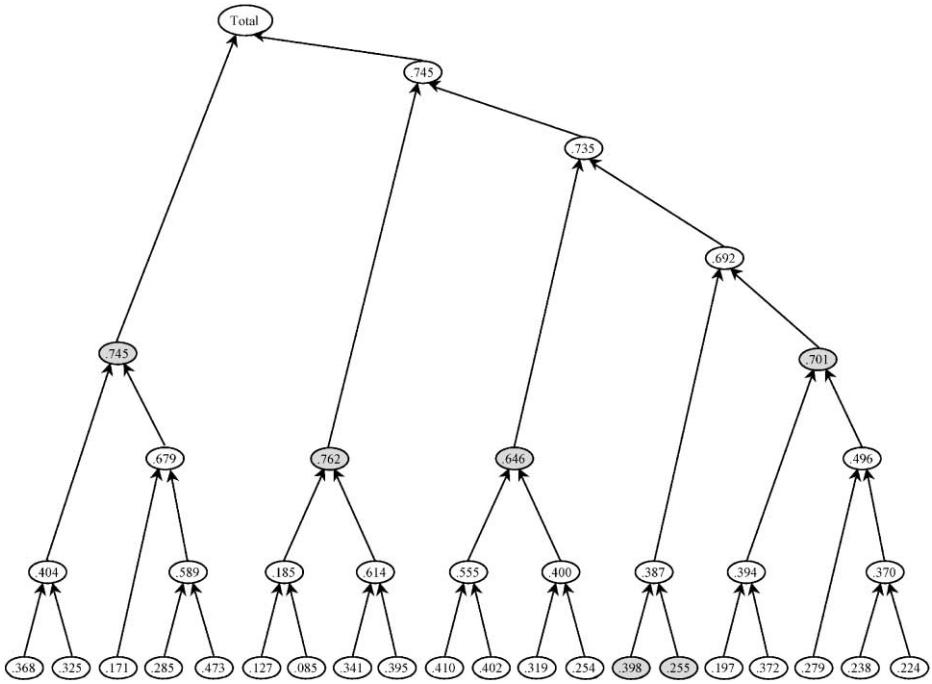

Fig. 3. Cluster partition tree.

4. Evaluation

How does the procedure outlined above perform with networks that have a small-world structure? To test the algorithm, I first simulate a set of large networks with a known cluster structure and see if the procedure can recover the known clusters. The simulation is designed to answer two questions. First, can the RNM procedure successfully differentiate nodes sufficiently for a fast cluster analysis program to identify dense clusters within the network? Second, since the algorithm is an iterative procedure across multiple dimensions, what is the optimum number of dimensions and iterations?

4.1. Simulating small-world networks

The key substantive feature of a small-world network is that people's relations tend to fall within a small number of close associates, some of whom have ties to people outside the primary group. Furthermore, most networks likely have a nested structure, as departments are nested within universities, neighborhoods within cities within states, and so forth. As such, most ties sent outside the primary group are likely not sent to the population at random, but fall within a wider secondary group. To test the RNM algorithm, I construct networks that have a three-layer structure. Each node is embedded within a small primary group, which is embedded within a larger secondary group, which is embedded within the total population. The networks are constructed by identifying the number of ties each person sends to each

Table 1
Simulated network descriptive statistics

Structure	Network		Degree	PG degree	SG degree	Pop degree
Primary group size: 50	1	Mean	9.13	8.74	0.36	0.033
Secondary group size: 400		S.D.	2.28	2.19	0.58	0.18
Population size: 20,000		Range	1–20	1–19	0–4	0–2
	2	Mean	9.08	7.80	1.25	0.030
		S.D.	2.43	2.19	1.06	0.17
		Range	1–20	1–18	0–7	0–2
	3	Mean	9.21	5.92	2.61	0.67
		S.D.	2.54	2.09	1.34	0.74
		Range	1–20	1–20	0–10	0–4

type of alter (primary group member, secondary group member and population at large), based on draws from a random normal distribution with specified mean, standard deviation and range. No further structure is implied within primary groups or between secondary groups. Table 1 presents the mixing statistics for each of the simulated networks.

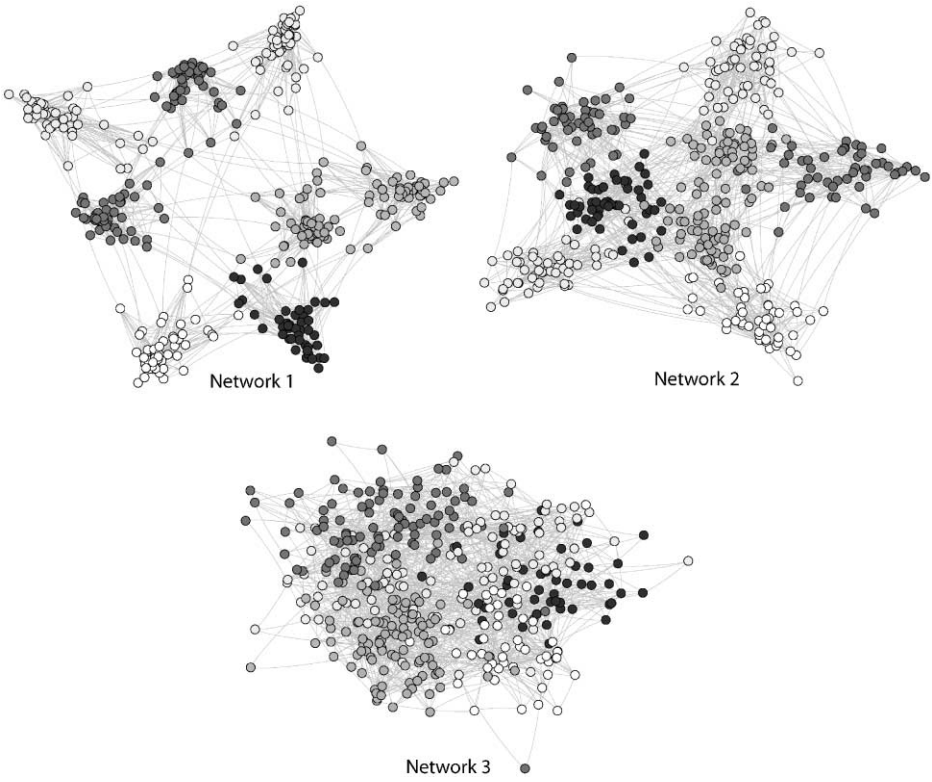

Fig. 4. Examples of relations within one secondary group for three test networks.

The networks differ in the extent of within-group mixing. In the first network, almost all ties fall within the primary group and only 3% of the population have contacts that extend beyond their secondary group. These graphs have a tight clustering, which can easily be seen when one of the secondary groups is plotted, as in Fig. 4. The other networks relax the extent of within group mixing, expanding both the number of ties to the secondary group and the number of ties to the population at large.

4.2. Monte Carlo design

The two elements of the algorithm that can vary are the amount of time the peer influence process operates (the number of iterations) and the number of positional variables (m) to construct. Under most circumstances, the peer influence model suggests that opinions will tend toward equilibrium. Fig. 5 shows the correlation between time t and time $t + 1$ vectors for a network similar to two, showing that for these networks opinion values tend to equilibrate within seven or eight iterations.

The Monte Carlo design varies the number of dimensions upon which to calculate neighborhood means (from one to nine) and the amount of time to let the procedure continue (from two to nine iterations). For each combination of dimensions and iterations, variables are produced that summarize each node's position in the space. These variables are then put into two clustering algorithms (SAS PROC FASTCLUS and Ward's minimum variance) to see if the program can correctly recover known primary groups. [13]

I use the adjusted Rand statistic (Morey and Agresti, 1984) to calculate the fit between observed and known cluster membership. Rand's statistic is based on two types of agreement: (1) whether two cases belong to the same cluster in both partitions and (2) whether two cases do not belong to the same cluster (Rand, 1971). Substantively, the Rand statistic can be defined as the probability that a randomly selected pair is classified in agreement. Morey and Agresti adjust Rand's statistic for chance, giving a more conservative estimate for how closely two partitions match. The Rand statistic can be calculated based on the cluster mixing matrix as:

$$\Omega = \frac{\sum\sum n_{ij}^2 - \left(\sum n_{i+}^2 \sum n_{+j}^2\right)/n^2}{(1/2)\left(\sum n_{i+}^2\right) + (1/2)\left(\sum n_{j+}^2\right) - \left(\sum n_{i+}^2 \sum n_{j+}^2\right)/n^2} \tag{6}$$

where n_{ij} is the number of cases observed in cell cluster i in the first partition and cluster j in the second partition, $n = \sum\sum n_{ij}$, n_{i+}^2 is the sum of the squared elements of cluster i and

[13] Details on the clustering procedures can be found at the SAS website (www.sas.com). FASTCLUS is based on a nearest centroid clustering algorithm. Both are fast procedures, with FASTCLUS running in time proportional to $nvcp$, where n is the number of observations, v the number of variables, c the number of clusters requested and p the number of passes over the data. For the runs reported in this paper, FASTCULS returned clusters in under 10 s for each network. The Ward's minimum variance routine is slower running in time roughly proportional to n^2. The seeming handicap of a polynomial clustering time can be easily overcome by using FASTCLUS as a first pass through the data to identify many small clusters (say 4000), and using those as the seeds for Ward's method. The combined runtime for FASTCLUS and Ward's for these analyses was less than 3 min. Early results show that within cell variance was very small, and thus only one iteration of each combination was used.

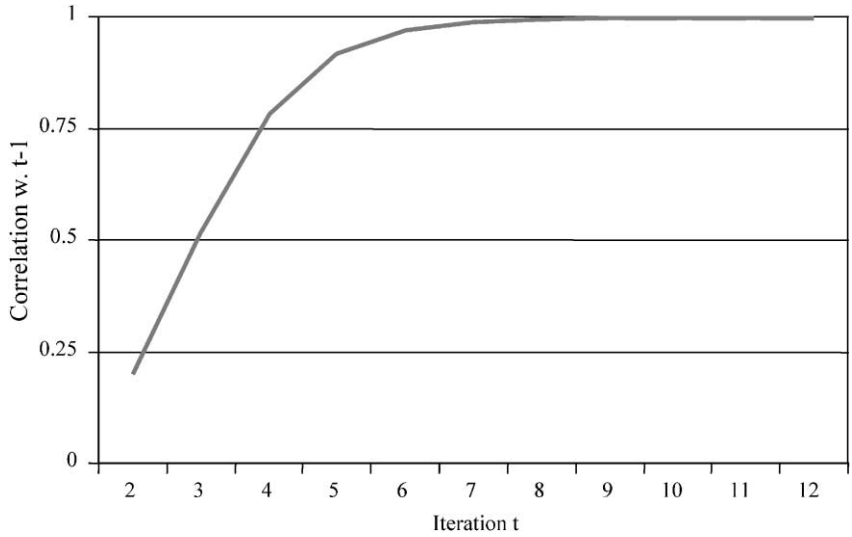

Fig. 5. Convergence of RNM values.

n_{j+}^2 is the sum of the squared elements of cluster j. Values of Ω approaching zero imply chance agreement, while values >0 "represent the proportion of the maximum possible difference obtained between the probability of agreement and the probability of chance agreement." (p. 35). When the statistic $= 1$, the two partitions match exactly.

4.3. Simulation results

The Monte Carlo results are presented in the set of surface plots in Fig. 6. In each surface plot, the z-axis reports the adjusted Rand statistic, the x-axis the number of dimensions and the y-axis the number of iterations. The rows correspond to each of the three input networks, and the two columns represent the fit statistics for the FASTCLUS and Ward's minimum variance solutions, respectively.

In general, the cluster algorithms accurately uncover the primary groups for both the highly separated and moderately separated clusters, with adjusted Rand statistics reaching values between 0.9 and 1.0 quickly. In all cases, Ward's minimum variance clustering procedure slightly outperforms the FASTCLUS procedure. The marginal returns to iterating beyond seven iterations are small, as the fit tends to level off, which is what we would expect given the converging values presented in Fig. 5. [14]

Returns to additional dimensions do not level out as dramatically, especially for the moderate and weakly clustered networks. The first point to note is the remarkably poor performance when only a single dimension is used. This suggests that programs using a single

[14] While the values change slowly at higher iterations, they do change. Letting the program iterate 100 times, for example, tends to blur distinctions between primary groups as the between-group influence process starts to slowly shift the position of the entire group.

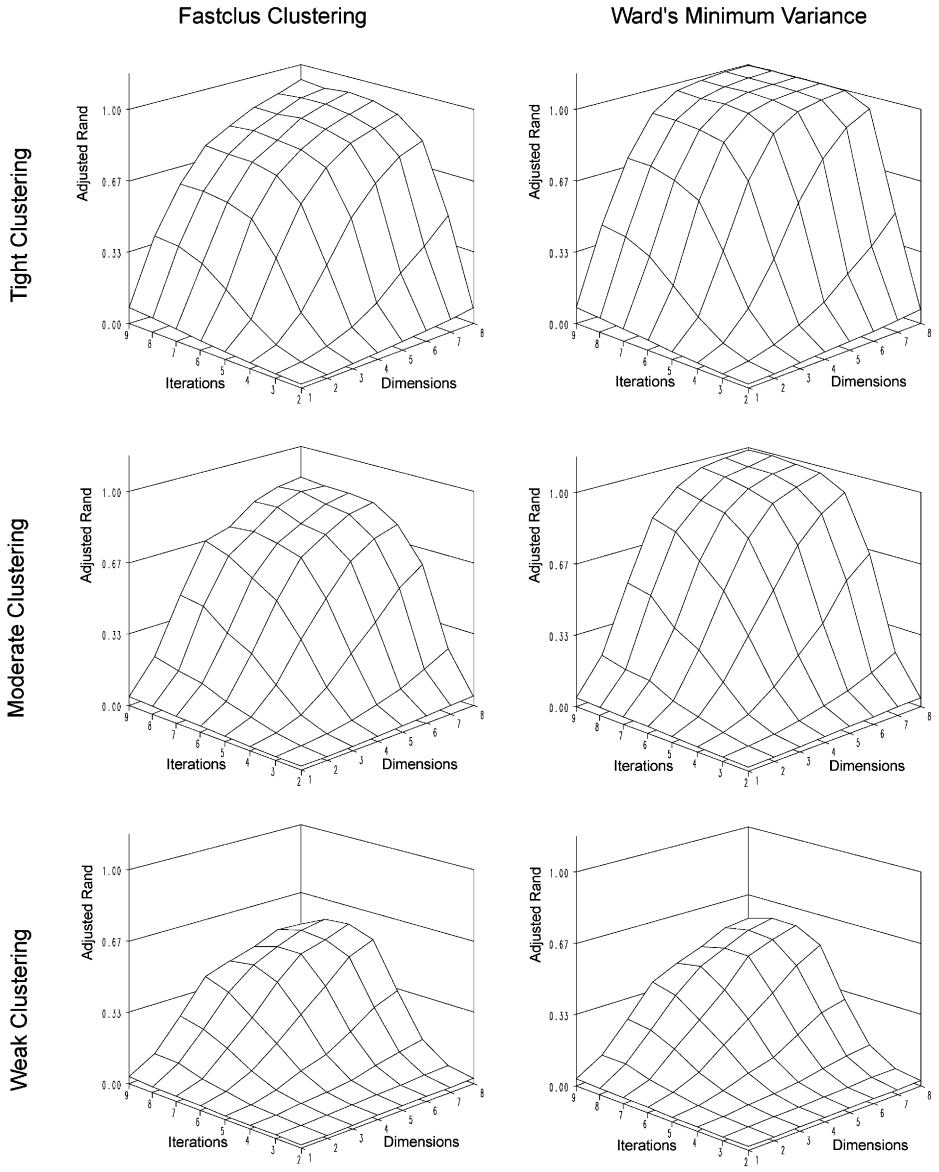

Fig. 6. Primary cluster recovery.

dimension as starting values for iterative search procedures are not getting much of a head start. When the network is highly clustered, Ward's minimum variance provides a perfect fit at six dimensions, and in all cases, a reasonably good fit with either six or seven dimensions.

For weakly clustered networks, the algorithm struggles to separate primary clusters. The increased number of ties between groups means that the influence process generates higher

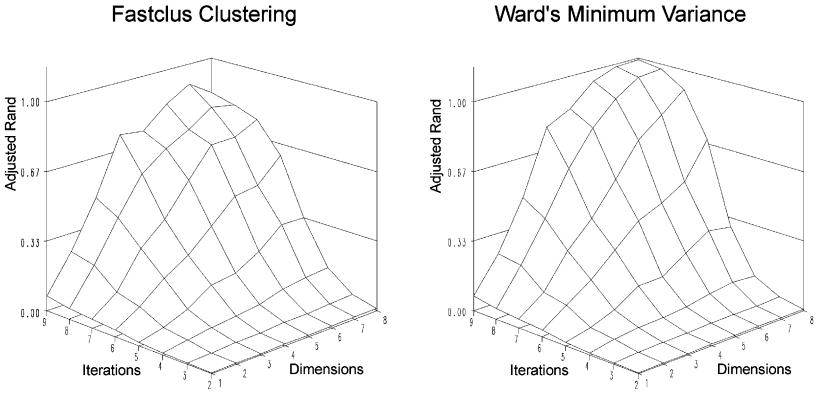

Fig. 7. Secondary group recovery weakly connected primary groups.

levels of homogeneity within secondary groups, blurring the distinction between primary and secondary groups.[15] This suggests that one ought to search for secondary groups, and Fig. 7 provides the Monte Carlo profiles for the secondary groups in the same weak primary-group network. Here we see an excellent recovery of the secondary groups. If one were to have applied the algorithm to a real network with weak primary clustering, we still would have accurately uncovered the secondary groups, at which point one could identify primary groups by analyzing the relations within the secondary groups.

5. Comparing RNM and other subgroup methods

How well do other clustering algorithms compare with the RNM procedure outlined above? There are two ways to answer this question. First, I apply the computationally feasible measures to the large simulated networks. Second, I use data on two smaller networks, a portion of the MacRea–Gagnon (MacRea, 1960) prison network ($g = 39$) and a private Northeastern high school ($g = 790$) to compare with other group detection methods.

5.1. Large network comparison

The only computationally feasible graph theoretic group detection algorithms for networks of this scale are connectivity (components and bicomponets) and k-core partitions. Neither components nor bicomponents do a good job of uncovering the primary or secondary groups in the large networks. In every case, the graph is connected and thus every person is a member of the largest component. Bicomponents also failed to substantively reduce the network, since only three nodes were excluded from of the largest bicomponent for the tightly and moderately clustered networks, and only one for the weakly clustered network. In all cases then, components and bicomponents do not sufficiently identify the small-world structure of the graph.

[15] This effect might decrease if we weight ties by two-step paths, as in NEGOPY.

Recall that a k-core is a maximal subgraph where every node is connected to k other nodes. Since we know that every member of the test networks is embedded within a primary group and that some members have only a few ties (<2), it would be impossible for a k-core with $k > 2$ to provide a perfect match. Still, it may be the case that at high levels of k the graph falls into smaller components that are the heart of each subgroup. If so, one could extract the highest k-cores from the network, and identify components within this set to uncover the hearts of the true clusters in the network.

The k-cores are most likely to succeed in the tight cluster network. If the method fails, it will do so because members of clusters have ties to other clusters, which is minimized in the tight network. Using PAJEK's core procedure, there are 9345 nodes, or about 47% of the total graph, involved in 7-cores. If we look within this set, we find it breaks into nine small bicomponents of between 42 and 320 nodes and one large bicomponent of 8490 nodes. Each of the small seven-core groups comes from a different secondary group, meaning that in these nine cases one would have identified the highest degree actors in the secondary groups, but in all but one case, the k-cores cross primary groups. The largest biconnected k-core, however, contains members from every other secondary group, and thus does not adequately reproduce the underlying small-world structure of the network. [16]

Showing that the RNM procedure can recover known clusters is an important validation of the procedure, but how well does the procedure compare with other well-known methods on real networks? Here we must restrict ourselves to smaller networks, since there are no other subgroup algorithms for large networks. I provide two examples. First, I use data taken from MacRea (1960), based on work of Gagnon, on friendships in a prison. This example provides a small network that is not overly clustered and thus allows me to compare a non-obvious case between three different methods. Second, I use data from the National Longitudinal Survey of Adolescent Health (Add Health) to compare the partitions from RNM with NEGOPY for a larger friendship network.

5.2. *The MacRea and Gagnon prison data*

The first example uses sociometric friendship data collected by Gagnon and published in an early methods paper by MacRea (1960), which is one of the standard datasets in UCINET. The original data consists of friendship choices among 67 prisoners. From these 67 people, I first identified the largest strongly connected component, containing 39 people, and use the underlying undirected graph to compare the performance of UCINET V's FACTIONS routine, NEGOPY and the RNM procedure. [17] The sociogram for this network is presented in Fig. 8, with the cluster partitions for each method indicated by dotted circles.

NEGOPY (Richards, 1995) is a network analysis program designed to identify the subgroup structure of a network. The program starts with a procedure similar to RNM, but based on a single dimension. After arraying nodes spatially, the program uses the link structure and

[16] If we were to restrict our attention only to these 9345 nodes, we get an adjusted Rand of 0.012 between the secondary group partition and the 7-core partition.

[17] For the purposes of this comparison, I treated the graph as undirected. This results in a less clustered graph than would be the case if we used only reciprocated arcs or weighted reciprocated nominations more strongly than unreciprocated nominations (which can be done). As such, it is more difficult for the clustering algorithms to find dense sub-regions in the graph and is thus a stronger test.

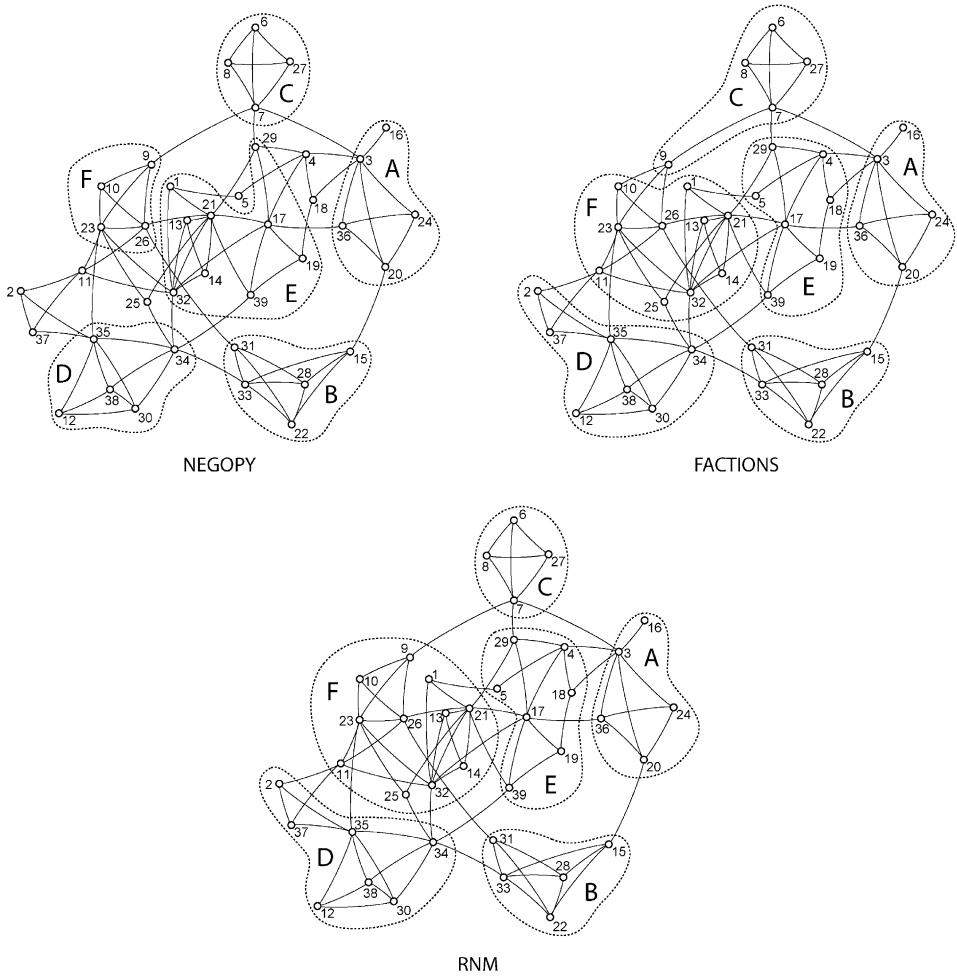

Fig. 8. Strong component of the Gagnon-MacRea prison network.

a pattern recognition algorithm to identify connected groups. NEGOPY is sociologically sophisticated in that it distinguishes between multiple roles in the network based on the pattern of relations, differentiating group members from multiple types of isolates, bridges and tree nodes. Among the three procedures, NEGOPY is the only one that automatically identifies the number of groups in the network. NEGOPY found six groups in the data and assigned eight people to non-group roles.

The FACTIONS routine in UCINET V uses a Tabu search procedure to identify groups by minimizing a function that describes the 'clique like' structure of the network. For this example, I used an option that maximizes the relative within-group density. Based on the initial NEGOPY results, FACTIONS was asked to produce six clusters. I applied the RNM procedure to the prison data to construct a set of eight positional variables that were then

Table 2
Correspondence among three network clustering routines

NEGOPY	FACTIONS	RNM
Cramer's V	0.854	0.883
Adjusted Rand (total)	0.548	0.557
Adusted Rand (group)	0.548	0.725
FACTIONS	–	0.972
		0.939
		0.916

entered into Ward's minimum variance clustering routine. Visual inspection of the resulting dendrogram for the cluster analysis confirms that six appears to be a reasonable number of clusters in this particular network.

Table 2 provides the cluster correspondence statistics for each partition. The correspondence measures show that the NEGOPY, FACTIONS and RNM solutions overlap considerably, with the lowest Cramer's V of 0.85 and Rand statistic of 0.55 between the NEGOPY and FACTIONS partitions. The three partitions agree perfectly for two groups of five nodes each (RNM groups A and B), and agree almost completely for RNM group C, with the FACTIONS and RNM procedures differing only in their assignment of node 9. Substantively, it seems reasonable that node 9 is grouped with 10, 26 and 23 (the people 9 sends most of his ties to), and thus the RNM and NEGOPY assignments seem sensible. FACTIONS and RNM match perfectly on the remaining assignments. The major differences between the NEGOPY and RNM partition center around the three high degree nodes at the center of the sociogram. NEGOPY's group E consists of people close to node 21 and 17, while FACTIONS and RNM separate 17 and 21 into two separate groups. To do so, NEGOPY has to assign nodes 4 and 18 to bridging positions between two groups (as well as nodes 11 and 25, to separate the $\{9, 10, 26\}$ clique from the rest of the group). While only deeper knowledge of the setting could identify which of these two partitions is more appropriate, the exact correspondence of FACTIONS and RNM is reassuring. The overall high correspondence between the three methods provides a nice small-network confirmation of the ability of the RNM procedure to identify dense sub-regions in the network.

5.3. An adolescent friendship network

Add Health asked students from a national sample of 140 schools to fill out an in-school friendship nomination questionnaire. Each student was allowed to nominate their five best male and five best female friends (for general information on the Add Health data see http://www.cpc.unc.edu/addhealth/). For detailed descriptions of the school networks, see Moody (1999). For this comparison, I use data from the largest connected component in one private northeast high school. The network contains 790 nodes. I generated eight position variables using the RNM procedure and submitted them to a Ward's minimum variance cluster analysis. I use the tree-walk procedure outlined in section 3.2 to determine the number of clusters, resulting in 19 groups ranging in size from 3 to 179 students.

NEGOPYs default parameters for the same graph resulted in a single large group. Following Richard's (1995) suggestions (pp. 123–125), I increased the transitive bias in the

Table 3
Correspondence between NEGOPY and RNM partition[a]

	0	3	4	16	17	18	25	26	29	31	33	34	36	38	43	46	52	56	Total
1							8												8
2									6										6
3										6									6
4																**3**			3
5						4													4
6									5										5
7														7					7
8									9										9
9										6									6
10											6								6
11		1		3		2	3			49	1		7		5				71
12										3									3
13															5				5
14										3									3
15		5																	5
16									13										13
17					4														4
18	2	5	8		48	39	8	12	96	78	64		2	23	68		10		463
19																		**5**	5
20									3					5					8
21										1					6				7
22												**5**							5
23															4				4
24															3				3
25							7												7
26											5								5
Total	2	11	8	3	52	45	26	12	132	146	76	5	9	35	91	3	10	5	671

[a] Adjusted Rand: 0.03.

network (since we would expect friends to be friends with each other in high schools), the sensitivity of the group detection algorithm, and lowered the size of the window used to initially detect groups. After these adjustments, NEGOPY assigned 671 people to 26 groups ranging in size from 3 to 463 people. [18] The overlap between the two clustering procedures is presented in Table 3.

The two programs overlap exactly in three small groups (underlined in Table 3), but the most overwhelming result is that the RNM procedure assigns nodes to groups that NEGOPY does not disentangle. This is most clear in that the one large cluster returned by NEGOPY has members in 14 of the 19 groups detected by RNM. The adjusted Rand statistic for the two partitions is 0.03, indicating a very poor correspondence.

To make an intuitive judgement about the type of groups found in the two procedures, Fig. 9 displays three of the RNM groups that were subsumed under one cluster by NEGOPY.

[18] The difference between 790 and 671 is the number of people NEGOPY assigned to non-group roles in the network.

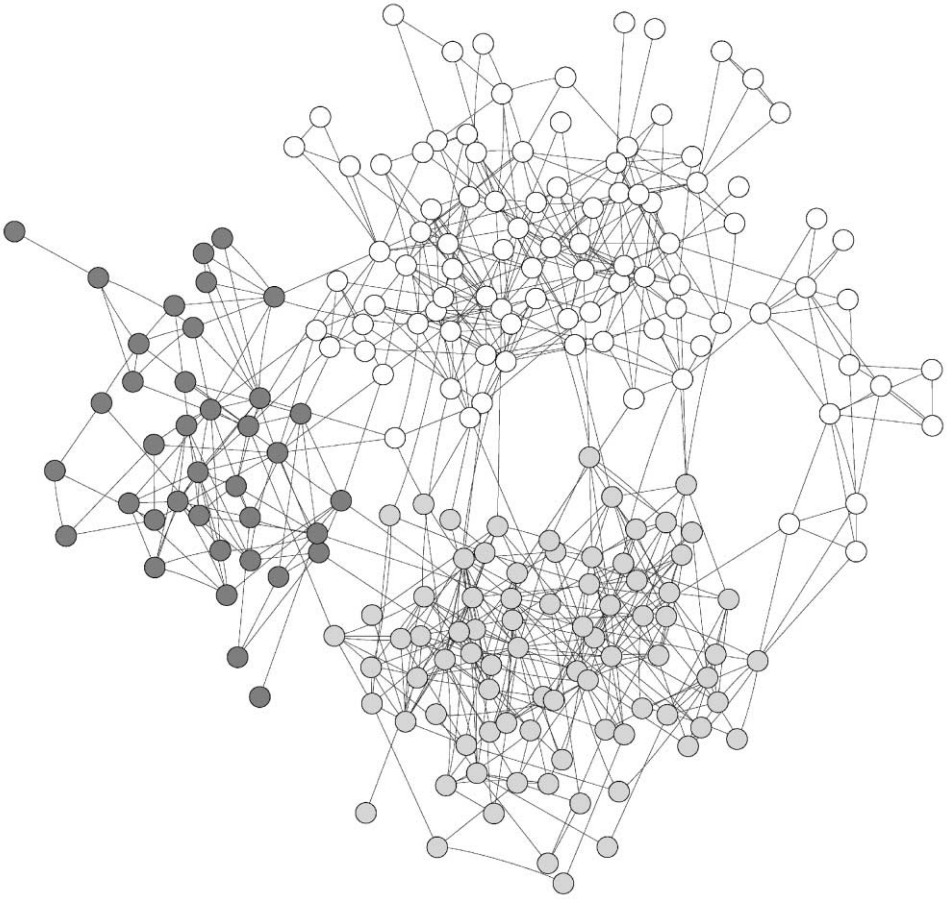

Fig. 9. Three RNM clusters from a large high school.

The three groups are clearly distinct and seem to have few internal divisions. There is a break within the white group, in that 13 nodes on the far right are connected by only three ties to the remainder of the group. This is an error introduced by the tree-walk process for determining the number of clusters. If I were to move down the clustering hierarchy, these 13 nodes would form a distinct group.

The procedure is not perfect and has clear limitations. Since the procedure reduces the graph to a set of position variables, the link structure is not used directly for finding the groups, though the resulting position variables are created based solely on the link structure. As such, there is no guarantee that the groups uncovered using the RNM procedure will be connected. Similarly, high connectivity but low overall density — similar but more extreme than the Add Health examples produced above — should produce fairly high levels of overall consistency with little subdivisions between groups. Still, the procedure does remarkably well given its simplicity, and should provide a nice first step for any

subgroup analysis of large networks, where more direct techniques are computationally cumbersome.

6. Conclusion

In this paper, I presented an efficient procedure for identifying dense regions of large networks that rests on expected properties of social networks. First, much evidence suggests that many large networks have a small-world structure, with dense local clusters and sparse connections between subgroups in the network. Second, theory predicts that peers within such high-density regions of the graph will be similar along multiple dimensions, due at least in part to endogenous peer influence. The RNM procedure uses a peer-influence process to identify dense regions of the graph and uncover its small-world structure. Starting with a set of m random variables ($\boldsymbol{Y}$), the procedure simulates a dynamic influence process, resulting in a set of variables that describe each node's position in the m-dimensional space. Because the initial random variables are uncorrelated, groups converge on unique positions in the space, which can then be easily recovered using well-known fast cluster analysis routines.

The Monte Carlo results of this paper show that the algorithm can successfully uncover known clusters in large networks (tested here on 20,000 node networks, but run successfully on graphs with over 50,000 nodes). The procedure seems to converge within about seven iterations and can correctly identify groups with between seven and eight dimensions. The number of dimensions needed likely increases as the level of clustering in the network decreases, since each dimension provides greater power for the detection program. Computationally, extra dimensions take little time to produce, though they increase the amount of time needed in the cluster analyses.

The RNM procedure was also tested on two smaller real-world networks. The first comparison showed that the RNM procedure replicates results of two well-known procedures (UCINET's FACTIONS and Richard's NEGOPY), showing that the types of groups uncovered by RNM are similar to those that would be found with the more sophisticated procedures. In the cases where they disagree, the RNM procedure seems to produce reasonable groups, and may be preferred if the purpose of the analysis is to uncover peer influence groups. The second example shows that for larger, and perhaps less clustered settings, RNM outperforms NEGOPY by identifying divisions within the network that NEGOPY missed. Combined, the Monte Carlo and empirical examples show that this simple approach to identifying subgroups in a network can accurately uncover dense regions within the network. Substantively, this procedure should provide especially interesting results for people working in cultural or ideational diffusion. Since the algorithm mimics a peer influence process, if such a process is active in the network, then groups that emerge should have relatively homogeneous opinions and attitudes.

Acknowledgements

This work is supported by NSF Grant IIS no. 0080860 "ITR/SOC: "The Structure and Dynamics of Electronic Social Networks." This research uses data from Add Health, a

program project designed by J. Richard Udry (PI) and Peter Bearman, and funded by Grant P01-HD31921 from the National Institute of Child Health and Human Development to the Carolina Population Center, University of North Carolina at Chapel Hill, with cooperative funding participation by the National Cancer Institute; the National Institute of Alcohol Abuse and Alcoholism; the National Institute on Deafness and Other Communication Disorders; the National Institute on Drug Abuse; the National Institute of General Medical Sciences; the National Institute of Mental Health; the National Institute of Nursing Research; the Office of AIDS Research, NIH; the Office of Behavior and Social Science Research, NIH; the Office of the Director, NIH; the Office of Research on Women's Health, NIH; the Office of Population Affairs, DHHS; the National Center for Health Statistics, Centers for Disease Control and Prevention, DHHS; the Office of Minority Health, Centers for Disease Control and Prevention, DHHS; the Office of Minority Health, Office of Public Health and Science, DHHS; the Office of the Assistant Secretary for Planning and Evaluation, DHHS; and the National Science Foundation. Persons interested in obtaining data files from The National Longitudinal Study of Adolescent Health should contact Jo Jones, Carolina Population Center, 123 West Franklin Street, Chapel Hill, NC 27516-3997 (E-mail: addhealth@unc.edu). Thanks to Susanne Bunn, Jill Burkart and the Social Networks reviewers for helpful comments and suggestions on earlier drafts of this paper.

References

Alba, R.D., 1973. A graph-theoretic definition of a sociometric clique. Journal of Mathematical Sociology 3, 113–126.

Batagelj, V., Mrvar, A., 2001. PAJEK. Version 0.71.

Billy, J.O., Rodgers, J.L., Udry, J.R., 1984. Adolescent sexual behavior and friendship choice. Social Forces 62, 653–678.

Borgatti, S., Everett, M.G., Freeman, L.C., 1999. UCINET V for Windows: Software for Social Network Analysis, Version 5.2.0.1. Analytic Technologies, Natick, MA.

Chaiken, S., Stangor, C., 1987. Attitudes and attitude change. Annual Review of Psychology 38, 575–629.

Cohen, J.M., 1983. Peer influence on college aspirations. American Sociological Review 48, 728–734.

Feld, S.L., 1981. The focused organization of social ties. American Journal of Sociology 86, 1015–1035.

Fershtman, M., 1997. Cohesive group detection in a social network by the segregation matrix index. Social Networks 19, 193–207.

Frank, K.A., 1995. Identifying cohesive subgroups. Social Networks 17, 27–56.

Frank, K.A., Fahrbach, K., 1999. Organization culture as a complex system: balance and information in models of influence and selection. Organization Science 10, 253–277.

Freeman, L.C., 1972. Segregation in social networks. Sociological Methods and Research 6, 411–430.

Freeman, L.C., 1996. Cliques, galois lattices, and the structure of human social groups. Social Networks 18, 173–187.

Friedkin, N.E., 1998. A Structural Theory of Social Influence. Cambridge University Press, Cambridge.

Friedkin, N.E., Cook, K.S., 1990. Peer group influence. Sociological Methods and Research 19 (1), 122–143.

Friedkin, N.E., Johnsen, E.C., 1997. Social positions in influence networks. Social Networks 19, 209–222.

Gibbons, A., 1985. Algorithmic Graph Theory. Cambridge University Press, Cambridge.

Harary, F., 1969. Graph Theory. Addison-Wesley, Reading, MA.

Johnsen, E.C., 1985. Network macrostructure models for the davis-leinhardt set of empirical sociomatrices. Social Networks 7, 203–224.

Johnsen, E.C., 1986. Structure and process: agreement models for friendship formation. Social Networks 8, 257–306.

Kandel, D.B., 1978. Homophily, selection, and socialization in adolescent friendships. American Journal of Sociology 84, 427–436.
Kochen, M., 1989. The Small World. Ablex Publishing Corporation, Norwood, NJ.
Koehly, L.K., 2001. How do I choose the optimal number of clusters in cluster analysis? Journal of Consumer Psychology 10, 102–103.
MacRea Jr., D., 1960. Direct factor analysis of socimetric data. Sociometry 23, 360–371.
Mark, N., 1998. Beyond individual differences: social differentiation from first principles. American Sociological Review 63, 309–330.
McPherson, J.M., Smith-Lovin, L., 1987. Homophily in voluntary organizations: status distance and the composition of face-to-face groups. American Sociological Review 52, 370–379.
Milgram, S., 1969. The small world problem. Psychology Today 22, 61–67.
Milligan, G.W., 1996. Clustering validation: results and implications for applied analyses. In: Arabie, P., Hubers, L., DeSoete, G. (Eds.), Clustering and Classification. World Scientific, River Edge, NJ, pp. 341–375.
Milligan, G.W., Cooper, M.C., 1985. An examination of procedures for determining the number of clusters in a data set. Psychometrika 50 (2), 159–179.
Moody, J., 1998. Matrix methods for calculating the triad census. Social Networks 20, 291–299.
Moody, J., 1999. The structure of adolescent social relations: modeling friendship in dynamic social settings. Dissertation, University of North Carolina, Chapel Hill, NC.
Moody, J., White, D.R., 2001. Social Cohesion and Embeddedness: a Hierarchical Conception of Social Groups. The Ohio State University, OH, USA, in preparation.
Morey, L.C., Agresti, A., 1984. The measurement of classification agreement: an adjustment to the Rand statistic for chance agreement. Educational and Psychological Measurement 44, 33–37.
Mosteller, F., 1968. Association and estimation in contingency tables. Journal of the American Statistical Association 63, 1–28.
Newman, M.E.J., 2000. Models of the Small World. Sante Fe Institute Technical Paper. HTTP://WWW.SANTAFE.EDU/SFI/PUBLICATIONS/Abstract/99-12-080abs.html (forthcoming in complexity).
Palmer, E.N., 1985. Graphical Evolution: an Introduction to the Theory of Random Graphs. Wiley, New York.
Pool, I.D.S., Kochen, M., 1978. Contacts and influence. Social Networks 1, 5–51.
Rand, W.M., 1971. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 66, 846–850.
Rapoport, A., Horvath, W.J., 1961. A study of a large sociogram. Behavioral Science 6, 279–291.
Richards, W.D., 1995. NEGOPY, Version 4.30. Simon Fraser University. Brunaby, BC, Canada.
Richards, W.D., Seary, A.J., 2000. MultiNet for Windows, Version 1.2.
Warner, W.L., Low, J.O., Lunt, P.S., Srole, L., 1963. Yankee City. Yale University Press, New Haven.
Wasserman, S., 1977. Random directed graph distributions and the triad census in social networks. Journal of Mathematical Sociology 5, 61–86.
Wasserman, S., Faust, K., 1994. Social Network Analysis. Cambridge University Press, Cambridge.
Watts, D.J., 1999. Small Worlds: the Dynamics of Networks Between Order and Randomness. Princeton University Press, Princeton.
Watts, D.J., Strogatz, S.H., 1998. Collective dynamics of 'small-world' networks. Nature 393, 440–442.
Weller, S.C., Romney, A.K., 1990. Metric Scaling: Correspondence Analysis. Sage, Beverly Hills.
Wellman, B., 1988. Structural analysis from method and metaphor to theory and substance. In: Wellman, B., Berkowitz, S.D. (Eds.), Social Structures: a Network Approach. Cambridge University Press, Cambridge.
White, H.C., 1965. Notes on the Constituents of Social Structure. Social Relations Department, Harvard University, Harvard.
White, D.R., 1998. Concepts of Cohesion, Old and New: Which Are Valid Which Are Not? University of California, Irvine, in preparation.