Clusterwise p* Models for Social Network Analysis

Douglas Steinley[1]*, Michael J. Brusco[2] and Stanley Wasserman[3]

[1] *Department of Psychological Sciences, University of Misouri, Columbia, MO*

[2] *Department of Marketing, Florida State University, Tallahassee, FL*

[3] *Department of Psychology and Statistics, Indiana University, Bloomington, IN*

Received 25 October 2010; revised 11 August 2011; accepted 12 August 2011
DOI:10.1002/sam.10139
Published online in Wiley Online Library (wileyonlinelibrary.com).

Abstract: Clusterwise p^* models are developed to detect differentially functioning network models as a function of the subset of observations being considered. These models allow the identification of subgroups (i.e., clusters) of individuals who are 'structurally' different from each other. These clusters are different from those produced by standard blockmodeling of social interactions in that the goal is not necessarily to find dense subregions of the network; rather, the focus is finding subregions that are functionally different in terms of graph structure. Furthermore, the clusterwise p^* approach allows for local estimation of network regions, avoiding some of the common degeneracy problems that are rampant in p^* (e.g., exponential random graph) models. © 2011 Wiley Periodicals, Inc. Statistical Analysis and Data Mining 4: 487–496, 2011

Keywords: cluster analysis; blockmodeling; social network analysis; p^* models

1. INTRODUCTION

To establish the context in which the models of this paper are to be discussed, we first note that the 'social networks' we are discussing are nothing more than graphs (directed or undirected) that have a set of N vertices, $\mathcal{V} = \{1, 2, \ldots, N\}$, and an associated set of edges, $\mathcal{E}$. For the set of vertices (called 'actors' in social network analysis), any number of relations, R, that specify how the actors are related to each other can be defined; however, it is often the case that $R = 1$ and that the relational tie can assume one of two values: presence or absence. This latter assumption, along with defining a particular relation, $\mathcal{R}$, on the actors allows the representation of the social relation in the binary adjacency matrix, $\mathbf{A}_{N\times N} = \{A_{ij}\}$, where $A_{ij} = 1 \Leftrightarrow (i, j) \in \mathcal{R}$ and $A_{ij} = 0$ otherwise.

Once $\mathbf{A}$ has been established, various analytic approaches can be used to uncover the general structure of the social network (see ref. 1). One recurring goal over the last 50 years is to identify differential structural patterns that may be present within the same network. Often, capturing heterogeneity within the context of social network analysis can be a difficult task. Early approaches relied on classic clustering techniques (e.g., hierarchical clustering) that were designed to find 'dense' regions of the network by directly clustering the adjacency matrix that defined the network. Hierarchical clustering naturally led to closely related seriation techniques, such as CONCOR (see ref. 1, Chapter 9), that relied on the permutation of rows and columns of the adjacency matrix. More recent approaches fall under the general terminology of 'blockmodeling', where the goal is create blocks (i.e., clusters) that correspond to known types of equivalence relationships (e.g., structural equivalence, regular equivalence, etc.).

1.1. Blockmodeling Via Combinatorial Optimization

The earliest approach to blockmodeling for finding dense regions of networks relied on combinatorial optimization approaches, rather than statistical models, for finding blocks (named for the resultant blocks of '1's' on the diagonal of the rearranged adjacency matrix) of actors (see ref. 2, for an early description). These early approaches were encouraged by techniques for graph partitioning [3] with modern approaches (see ref. 4, for a review) becoming

Correspondence to: Douglas Steinley (steinleyd@missouri.edu)

more reliant on various optimization techniques. Examples of these algorithmic approaches include: permutations of row and column objects to reveal structure [5], Boolean decompositions of the adjacency matrix [6], variable-neighborhood search techniques [7], integer programming [8], and tabu search [9] among others. Many of these approaches to blockmodeling rely on notions of cluster density and compactness that have a history in classic clustering algorithms, such as K-means clustering [10] and p-median clustering [11].

1.2. Stochastic Blockmodeling

While the literature is rife with examples from the combinatorial optimization perspective, there is also no shortage of techniques that use a more statistical approach in modeling network structure to find groups. Wasserman and Anderson [12] describe the difference between stochastic and 'regular' (e.g., approaches rooted in combinatorial optimization) blockmodeling as that the sought after block structure is revealed during the modeling process for stochastic models. The first approach for formalizing this type of modeling for exponential random graph models (ERGMs, originally termed p^* models and discussed in more detail in the subsequent section) is fairly recent [13]. These models have been extended to mixed-membership models by Airoldi *et al.* [14] where each actor has different probabilities of belonging to each of the underlying blocks (e.g., clusters). This probabilistic membership contrasts with traditional blockmodeling, which similar to traditional clustering procedures, require an observation either to be a member of one (and only one) block or not a member. Other models utilizing probabilistic membership include the latent space models developed by Handcock *et al.* [15] and Hoff *et al.* [16].

As mentioned, both approaches to blockmodeling are usually designed to find either dense regions of the network or highly connected regions of the network. Here, we move away from the approach of looking for clusters based on either density or various kinds of equivalence. Instead, the problem is reformulated as one where the relational ties result in a set of random variables (e.g., **A** and its elements), allowing us to create a dependence graph to understand the possible (of which there are many) graph probability distributions [17]. As it turns out, any observed relational network may be regarded as a realization $\mathbf{a} = [a_{ij}]$ of a random two-way binary array **A**, which has an associated dependence graph $\mathcal{D}$ with vertices that are elements of the index set $\mathcal{V}_{\mathcal{D}} = \{(i, j); i, j \in V, i \neq j\}$ for the random variables in **A**. The edges of $\mathcal{D}$ signify pairs of the random variables that are assumed to be conditionally dependent, such that $\mathcal{E}_{\mathcal{D}} = \{((i, j), (k, l)),$ where A_{ij} and A_{kl} are not conditionally independent$\}$. Within the auspices of the dependence graph, Wasserman and Pattison [18] noted that there were three major types: Bernoulli graphs, dyadic dependence distributions, and p^* (e.g., exponential random graphs). It is the latter on which we focus our attention.

2. *P** MODELS

The set of p^* models was first introduced by Wasserman and Pattison [19] and extended by Pattison and Wasserman [20] and Robins *et al.* [21]. These models find their basis in the Hammersley–Clifford theorem [17], which establishes a probability model for **A** that only depends on the cliques of the associated dependence graph $\mathcal{D}$. Specifically, the form is given by

$$Pr(\mathbf{A} = \mathbf{a}) = \frac{1}{\kappa}\exp\left(\sum_{S \subseteq \mathcal{V}_{\mathcal{D}}} \lambda_S \prod_{(i,j) \subseteq S} a_{ij}\right), \qquad (1)$$

where κ is a normalizing quantity; $\mathcal{D}$ is the dependence graph for **A**, the summation is overall subsets S of vertices of $\mathcal{D}$; the product term is the sufficient statistic corresponding to the parameter λ_S; $\lambda_S = 0$ whenever the subgraph induced by S is not a clique of $\mathcal{D}$. Consequently, the nonzero parameters in the associated probability distribution depend on the maximal cliques of $\mathcal{D}$. By definition, while not maximal, all subgraphs of a complete subgraph are also complete. So, if S is a maximal clique, then it and all of its subgraphs will have nonzero parameters associated with it. Given this nested nature of complete subgraphs, the number of parameters can become staggering, requiring the examination of either simple dependence structures or making reasonable assumptions about the parameters. A common assumption (see ref. 20) is homogeneity—isomorphic configurations of vertices are equated.

A standard practice (see ref. 22) has been to fit models with either the full set (or various subsets) of the 16 triadic configurations (see Fig. 1). Additionally, it is clear that, depending on the nature of the structure one is interested in, several different configurations can be modeled in the p^* framework. Configurations consisting of more than three nodes have been fit as well. For instance, Pattison and Robins [23] fit configurations consisting of 4-nodes that each followed what they deemed the 'three-path' model (i.e., all pairs of edges lie on a path of length 3). Returning our attention to Fig. 1, it can be seen that some of the triads can be constructed from simpler triads, making them conditional on the simpler triad classes being present as well. This nested nature of lower-order configurations within higher-order configurations can make interpretation of the associated parameters difficult. The easiest interpretation is that a significant positive (negative) parameter for

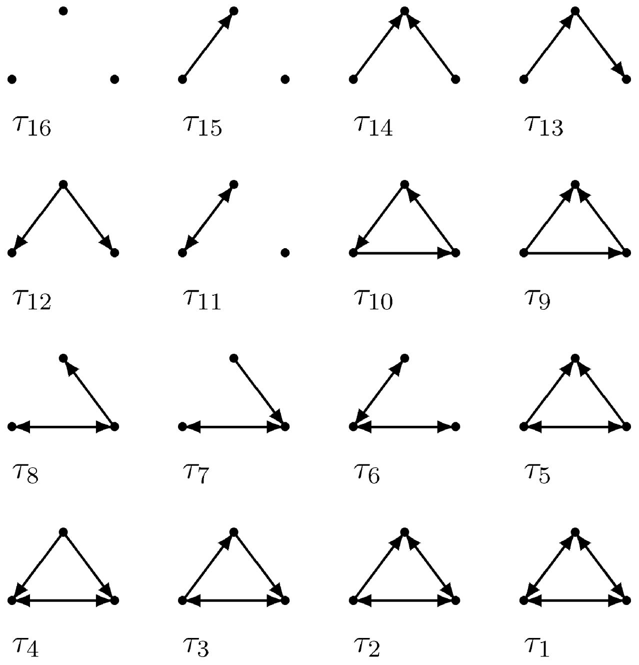

Fig. 1 This figure depicts the 16 possible triad isomorphisms that exist among three nodes with asymmetric ties.

a configuration suggests, given the number of other configurations in the dependence graph, there are more (less) of those configurations present than one would expect to occur by chance alone.

In general, there have been two common approaches to estimating p^* models: maximum likelihood estimation and pseudolikelihood estimation. The former is done through Markov Chain Monte Carlo estimation (MCMC; see refs 24–26), while the latter is done via logistic regression (see refs 19,27–30).

One may question whether pseudolikelihood estimation should be pursued now that MCMC procedures have been developed for maximum likelihood estimation. In fact, the primary problem with p^* models is the potential of either model degeneracy or inferential degeneracy. Model degeneracy refers to degeneracy that is related directly to the model, rather than the estimation process, and it has been shown that model degeneracy is more likely to occur when there are particular graph structures present. For instance, in cases where there are large numbers of transitive triads (τ_9) model degeneracy is more likely to occur [31]. Handcock [25] defines a graph distribution to be degenerate, or near degenerate, if there are only a few graphs that have nonzero probabilities.

Inferential degeneracy is related to the shortcomings of the pseudolikelihood estimation procedure. As Robins *et al.* [31] indicate, one of the primary advantages of maximum likelihood estimation is the ability to obtain stable standard errors for the estimates. In one of the few comparisons of the two types of estimates, Robins *et al.* [31] (p. 212) conclude: 'Probably the best that can be said is that PL [pseudolikelihood] estimates that suggest "significance" according to PL [pseudolikelihood] standard errors may be indicative of the effects needed to model the data'. However, as we see below for computational reasons, pseudolikelihood estimation becomes necessary for the proposed clusterwise p^* models. As such, we include an automated process for screening models that may be degenerate, either at the model or inferential level.

3. CLUSTERWISE REGRESSION

The goal of the clusterwise p^* model is to find subgroups of vertices that have different functional relationships between the dependent variable (e.g., the observed value of the ties in **A**) and the independent variables (e.g., the various vertex configurations being modeled). Given that pseudolikelihood estimation will be implemented, the process then becomes one of extending traditional clusterwise regression models (see refs 32–34) to a logistic regression setting. To describe the clusterwise p^* model, we adopt the following notation from Brusco *et al.* [35]:

$N =$ the number of objects to be clustered, indexed $1 \leq i \leq N$;

$V =$ the number of independent variables (i.e., vertex configurations), indexed $1 \leq v \leq V$;

$x_{iv} =$ the measurement of predictor variable v for the ith object;

$y_i =$ the measurement of the response variable for the ith object;

$K =$ the number of clusters, indexed $1 \leq k \leq K$;

$P_K = \{C_1, C_2, \ldots, C_K\}$ a feasible partition of the N objects into K clusters, where C_k represents the set of objects assigned to the kth cluster;

$N_k =$ the number of objects in the kth cluster;

$b_{0k} =$ the intercept for the logistic regression model in the kth cluster;

$b_{vk} =$ the slope coefficient for the v^{th} predictor variable in the kth cluster.

If one were concerned with just conducting a traditional cluster analysis, such as k-means clustering (see ref. 10, for a review), then any partition of the data set, P_K, has an associated within-cluster sums-of-squares error

$$\text{WCSS}(P_K) = \sum_{k=1}^{K} \sum_{i \in C_k} (y_i - \overline{y}_k)^2, \qquad (2)$$

which leads to the natural total sums-of-squares decomposition

$$\text{TSS } = \text{BCSS}(P_K) + \text{WCSS}(P_K), \tag{3}$$

where $\text{BCSS}(P_K)$ is the resultant between-cluster sums-of-squares and is represented as

$$\text{BCSS}(P_K) = \sum_{k=1}^{K} N_k(\overline{y}_k - \overline{y})^2. \tag{4}$$

The $\text{BCSS}(P_K)$ is the amount of variation explained by the clustering process. Adding an additional K functional models (i.e., one for each cluster) that relates the independent variables to the dependent variable serves to reduce the variation not explained by the clustering process (e.g., $\text{WCSS}(P_K)$). Specifically, $\text{WCSS}(P_K)$ can be decomposed as

$$\text{WCSS}(P_K) = \left[\sum_{k=1}^{K}\sum_{i\in C_k}\left(\overline{y}_k - \hat{y}_i\right)^2\right] + \left[\sum_{k=1}^{K}\sum_{i\in C_k}\left(y_i - \hat{y}_i\right)^2\right], \tag{5}$$

where the first bracketed term represents the within-cluster variation explained by the regression model, $\text{SSR}(P_K)$, and the second bracketed term represents the residual error in the clusters, $\text{SSE}(P_K)$. Taking everything together, the TSS can be decomposed as

$$\text{TSS} = \text{BCSS}(P_K) + \text{SSR}(P_K) + \text{SSE}(P_K). \tag{6}$$

The obvious goal [32] then becomes to minimize $\text{SSE}(P_K)$ subject to P_K being a feasible partition of the N objects into K clusters. A standard objective to maximize is a normalized version of $\text{SSE}(P_K)$

$$\Phi(P_K) = 1 - \frac{\text{SSE}(P_K)}{\text{TSS}}, \tag{7}$$

which is analogous to an R^2 measure that is created from the variance accounted for from both the clustering of the observations, $\text{BCSS}(P_K)$, and the fitted values from the clusterwise regression models, $\text{SSR}(P_K)$.

Generally, this optimization is conducted through an exchange algorithm [32] that begins with an initial partition of the object set. An iterative process of object relocation is initiated by considering reassigning all objects to the clusters of which the object is not currently a member. If no reassignment results in an increase in Eq. (7), then the object remains in its current cluster; otherwise, the object is reassigned to the cluster that yields the greatest improvement. The relocation algorithm proceeds until no relocation can result in an improvement. Upon termination, the solution is guaranteed to be locally optimal with respect to all possible relocations of a single object; however, a global optimum is not guaranteed. Consequently, with these types of algorithms, it is often recommended that the exchange process is repeated with several random initializations [36,37].

3.1. Adapting Clusterwise Regression to Pseudomaximum Likelihood Estimation

Pseudolikelihood estimation proceeds by treating each binary tie (regardless of whether it is present or absent) in the adjacency matrix, A_{ij}, as a 'case' in the vector that represents the dependent variable (e.g., $\mathbf{y}$). The associated independent variables are represented by the parameters in the model, as established in Eq. (1), such as the isomorphic configurations (e.g., stars and triads of various types). For each case, the statistic associated with the an independent variable is the difference—often termed change statistics—in the number of the relevant configurations between the graph with $a_{ij} = 1$ and $a_{ij} = 0$. Then, standard logistic regression can be applied to the data; however, it is important to realize that there are dependencies within the data that prevent the strict adherence to the usual tests of model fit. Nonetheless, measures of fit can be taken as heuristic guides and used for within model selection and/or building. For details on the pseudolikelihood estimation process, we refer readers to Pattison and Wasserman [20].

Obviously, as the dependent variable is binary, we turn to logistic regression to model the relationships between the various isomorphic configurations and the presence (absence) of a tie in the observed network. Thus, $\hat{y}$ becomes

$$\hat{y} = \frac{e^{b_{0k}+\sum b_{vk}x_{iv}}}{1 + e^{b_{0k}+\sum b_{vk}x_{iv}}} \tag{8}$$

and we utilize a standard 'r-squared' measure of fit as described in Eq. (7). While, the r-squared measure can be artificially low (even for well-fitting models), Hosmer and Lemeshow [38, p. 167] indicate that 'they may be helpful in the model building stage as a statistic to evaluate competing models'. In the present application, the measure is used to determine which cluster to assign an observation; thus, the measure is only utilized for model building and never as the final assessment of the model itself. Likewise, and equally relevant, is assuring that the minimization of within-cluster variance is a reasonable objective for the clustering of binary data. In fact, Brusco [39] showed that this objective function worked quite well for binary data in a wide-range of simulations.

Finally, the last consideration is the nature of the relocation algorithm and recalculation of the independent variables. Specifically, if the observed adjacency matrix is $N \times N$ then the dependent variable will be a binary vector with length of $N(N-1)$ (note: the self-links are ignored). During the application of the relocation algorithm, if an observation is moved to a different cluster, then all $(N-1)$ binary ties are moved with it. To make it slightly more complicated, the binary tie between the ith and jth observation is not possible if the two observations are in different clusters. The overall effect is the need to recompute K adjacency matrices (one for each cluster), making the overall length of the 'full' dependent variable vector $\sum N_k(N_k-1)$, which will always be less than length of the vector if only one p^* model (rather than K) were fit to the data. Additionally, as the independent variables reflect structural properties of each cluster's adjacency matrix, they will have to be recomputed upon the evaluation of the effect of assigning each observation to one of the K clusters. To evaluate the potential assignment of an observation to a cluster, the p^* model will have to be fit, resulting in a total of $N \times K$ p^* models on *each* pass through the relocation algorithm. As Steinley [10,36] and Steinley and Brusco [40] showed, these types of algorithms tend to have numerous local optima (potentially in the hundreds, and maybe even in the thousands) so the general recommendation is to fit the model with several thousand random initializations. Coupled with the fact that there may be up to a few hundred iterations before convergence for any one random initialization, the fitting of a clusterwise p^* model to a single data set could result in the estimation of several million p^* models. It is for this reason that pseudolikelihood estimation is required, as true maximum likelihood estimation would be too costly terms of computation time and defeat the overall purpose of exploratory data analysis.

The algorithm for estimating the clusterwise p^* model proceeds as:

1. The user chooses the number of clusters, K.
2. The data are partitioned into K clusters where each cluster must be connected.
3. In turn, each observation is considered to be part of each cluster. For each consideration, the network statistics (i.e., the network isomorphisms of interest) are computed as if that observation were part of that cluster.
4. For each of the clusters, the observation is assigned to the cluster which resulted in the highest overall model R^2.
5. Steps 3 and 4 are repeated until no observations change clusters.

3.1.1. Step 1

While K must be fixed prior to estimating the clusterwise p^* model, a common way to determine the value of K is to fit several different models assuming $K = 1, \ldots K_{\max}$ and choose the value of K such that the incremental increase in overall model fit is negligible. This approach is analogous to using a screen plot to determine the number of components in principal component analysis. For the present application we have adjusted the standard, overall model R^2 to be

$$\Phi^{'}(P_K) = \sum_{k=1}^{K} \left(\frac{N_k}{N}\right)\left(\frac{N_k - V_k - 1 - K}{N_k - 1}\right) R_k^2. \qquad (9)$$

This formulation has several advantages. The first term weights the overall contribution to the model fit by the relative size of each cluster, insuring that exceptionally well-fitting, small clusters are not over-weighted (as would be the case with a simple arithmetic mean of the cluster R^2 values). The second term favors parsimonious models in two regards: (i) there is a penalty for the number of predictors, V_k, for each cluster, and (ii) there is an explicit penalty for too many clusters. Finally, the third term measures the goodness-of-fit for the kth cluster. The final partition is chosen to maximize $\Phi^{'}(P_K)$.

3.1.2. Step 2

The initialization scheme for the algorithm is to partition the data into K clusters. This partitioning may be done in several manners; most naturally, either randomly, a graph partitioning algorithm, or a blockmodeling algorithm, or any combination thereof. The only constraint being that each of the K clusters must be connected. Under this rule, isolate nodes would be considered their own cluster; consequently, a natural prescreening method would be to remove isolates prior to the analysis.

3.1.3. Step 3

In general, this step serves as a standard core component for the majority of clusterwise regression procedures; however, with the p^* model, degeneracy is a serious concern. Normally, in a standard OLS setting for clusterwise regression, parameters that were unimportant for a particular cluster would have values close to zero and merely be insignificant. Contrarily, the inclusion of extra network isomorphisms (especially, if their counts are too low or too high) can cause degeneracy within the model estimation step. Thus, depending on the configuration of the nodes within a particular cluster, specific models may become

Table 1. Pseudolikelihood parameter estimates for Sampson's monk data.

		Parameter estimates (standard error)							
		Complete	Transitivity	Cyclicity	Reciprocity	Outstar	Mixed star	Instar	Choice
K(Φ')	N_k	τ_1	τ_9	τ_{10}	τ_{11}	τ_{12}	τ_{13}	τ_{14}	τ_{15}
1 (0.260)	18	0.35 (0.69)	0.42 (0.12)	−0.08 (0.34)	−1.78 (0.42)		−0.34 (0.13)		
2 (0.589)	7		−3.16 (1.21)	−4.44 (2.02)	−3.60 (1.88)		3.63 (1.49)	3.20 (1.13)	
	11	1.57 (1.26)	0.32 (0.27)		1.87 (0.96)	−4.33 (0.88)			
3 (0.278)	7		−3.16 (1.21)	−4.44 (2.02)	−3.60 (1.88)		3.63 (1.49)	3.20 (1.13)	
	7		0.65 (0.39)	−1.11 (0.56)	1.26 (0.74)			−0.30 (0.43)	
	4		0.28 (0.73)					−0.75 (1.04)	

degenerate and necessitates the need to have different predictor variables for each of the clusters.

To address this concern, for every cluster, a mini-model comparison technique is conducted to insure the fidelity of the model and avoid potential degeneracies that are often rampant in p^* models. For each cluster, all $2^V - 1$ models are evaluated in terms of Φ'; by extension, we recommend that the user chooses a set of theory driven graph statistics (probably less than 10 because of the number of possible models) rather than a 'shotgun' approach where everything and the kitchen sink is included as a predictor. A standard 'red flag' for degeneracy in p^* models is the presence of very large or near zero standard errors of the parameter estimates. A simple and efficient screening mechanism is to ignore all solutions with large standard errors, defined here as being an order of magnitude larger than the actual parameter estimate, or with extremely small standard errors (e.g., <0.001). These rules have served well in practice, with the former more likely to indicate inferential degeneracy and the latter more likely to indicate general model degeneracy.

An additional protection against degeneracy is the minimization of the sum-of-squared residuals. As discussed in the rich literature on K-means clustering (see ref. 10), minimizing within-cluster variance estimates often results in clusters that are equally sized. In this setting, the result is to create clusters that are close to the same size, minimizing the occurrence of clusters that are too large and likely to suffer from a degree of heterogeneity that can be the culprit of model degeneracy. Thus, we are relying on a series of 'local' network models rather than trying to model the network in full.

3.1.4. Steps 4 and 5

Observations are moved to the cluster, provided that the cluster will remain a connected component, to which model fit is most increased. To this end, the procedure is like most greedy clustering approaches and is myopic in nature. However, the iterative nature often results in solutions that compare favorably with more complex, likelihood driven approaches (see ref. 41).

4. EXAMPLES

4.1. Sampson's Monk Data

A classic example used in social network analysis is Sampson's monastery study [42,43] with 18 monks based on the 'whom do you like' sociometric relation. The network consists of 56 ties, and Sampson originally partitioned the data into three subgroups—a partitioning that is originally assumed to be the 'true structure' in terms of finding coherent, dense groups within the network. First, we performed the variable selection procedure described above for when $K = 1$ and using the graph statistics (and their common terminology in parentheses): τ_1 (complete), τ_9 (transitivity), τ_{10} (cyclicity), τ_{11} (reciprocity), τ_{12} (outstar), τ_{13} (mixed star), τ_{14} (instar), and τ_{15} (choice) as they are the most commonly used in this type of modeling.[1] In this instance, there were 192 models that were deemed degenerate in terms of their standard errors. The best fitting model, in terms of Φ' is provided in Table 1 and will serve as the baseline for determining if it is beneficial to fragment the network into clusters with different p^* models for each cluster.

Comparing the values of Φ', it is seen that the two cluster solution is the best in terms of predicting the existing data set most parsimoniously. Additionally, while the within-group models fit well, neither of the models are nondegenerate when applied to the other group. Furthermore, it is clear that the clusterwise p^* approach differs from other blockmodeling attempts (both stochastic and traditional) in that it is not driven to find coherent or dense subgroups;

[1] While these statistics are the most commonly used, they are also used here for demonstration purposes only. Clearly, for a theoretically motivated question, it would be reasonable and expected to choose a set of graph statistics that would aid in interpretability of the phenomenon under study.

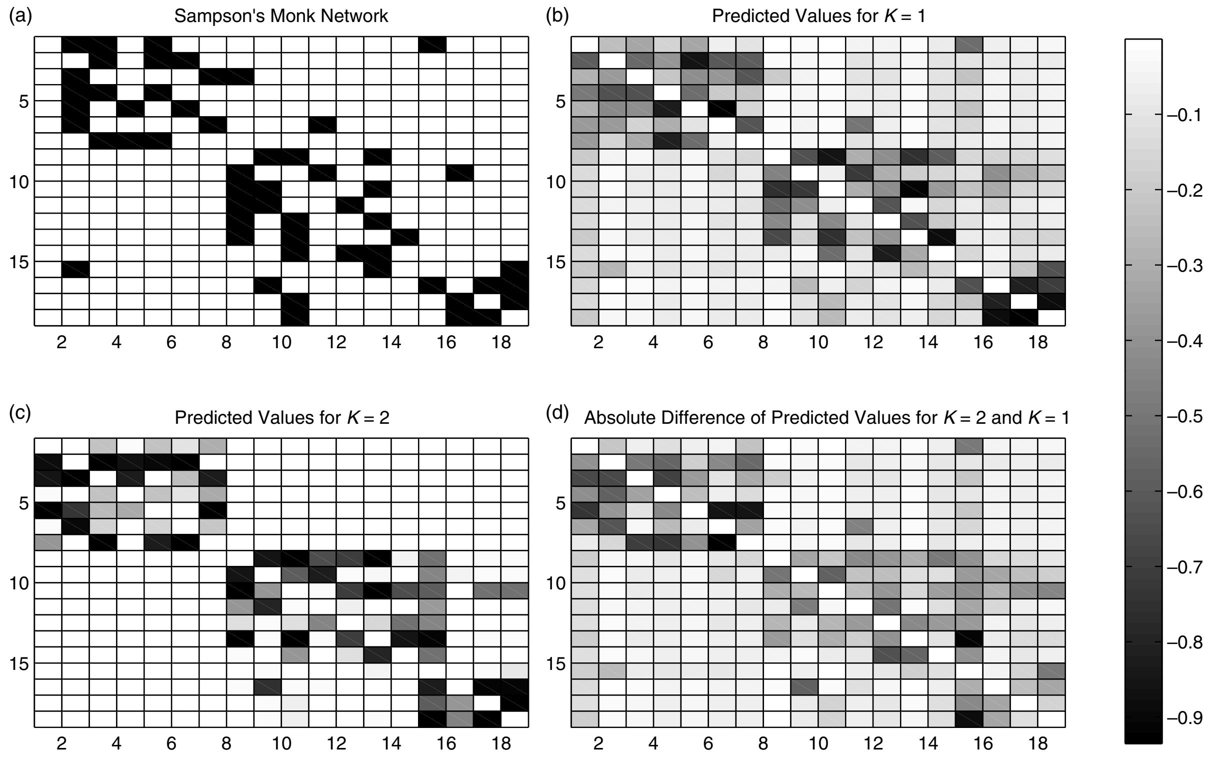

Fig. 2 (a) Ties are represented by black boxes, while white boxes indicate there is not a tie presence. (b) The shading of the box indicates the p^* model ($K = 1$) derived probability of a tie, with white being a probability of zero and black being a probability of unity. (c) The p^* model derived probabilities for $K = 2$. (d) The absolute difference between the p^* model derived probabilities when $K = 1$ and $K = 2$, with white indicating the same prediction and black indicating opposite prediction of links between the two models.

rather, the overall goal is prediction. If one were to choose the three group solution, it is equivalent to that originally found by Sampson [42] and replicated in other blockmodeling work (see ref. 14); whereas, the consistency between the selected two group solution and the three group solution is only moderate with an adjusted Rand index (ARI; which is equal to zero when there is chance agreement and unity when there is perfect agreement [44]) of 0.63.

Figure 2 helps illustrate the difference between the predictive power of the one group versus the two group solution. Figure 2(a) is a representation of the original network, with the solid blocks indicating a tie between two monks. Figure 2(b) shows the model predicted values of the one group p^* model, while Fig. 2(c) shows the predicted values of the two group p^* model. Finally, Fig. 2(d) shows the absolute difference in predicted values between the two models, clearly indicating that the two group model has more predictive power. In this instance, the predictive power is not hindered too much by ignoring the ties between the clusters because the network as a whole is fairly sparse.

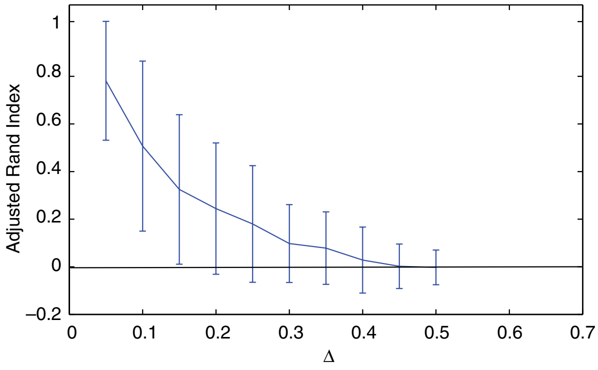

Fig. 3 The x-axis indicates the percentage of ties that are randomly perturbed (either from zero to one or from one to zero) in Sampson's monk data, while the y-axis denotes the agreement between the partition derived from the unperturbed data and the partition derived at a given level of perturbation. The plotted values are the averages of 1,000 data sets, with accompanying standard error bars. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

To test the sensitivity of the clusterwise p^* approach, we randomly perturbated the existing network and refit the two-class model, comparing the resultant partitions with that under the unperturbated data (see Fig. 3). For each tie, Δ represents the random probability that the tie was changed to absent if previously present or to present if previously absent, where Δ ranges from 0.05 to 0.5, with 1000 replications being conducted at each level of Δ. Figure 3 indicates the expected decrease in agreement between the partitions as the data becomes more perturbed, with the associated error bars crossing chance levels (e.g., ARI = 0) when $\Delta = 0$. The fact that changing, on average, 20% of the links completely eliminates any meaningful agreement between the partitions should not be too surprising given the sparseness (only 56 ties out of 306 possible ties) of the original network.

4.2. Graph and DiGraph Glossary

The second data set represents a network of graph theoretic terms which contains a link from term A to term B if and only if Term B is used to describe the meaning of term A [45]. The data set contains 72 terms with 122 links; once again, a rather sparse network that contains only about 2% of the possible ties. Repeating the analytic procedure above, and testing from $K = 1$ to $K = 4$, the associated fit statistics were: $\Phi^{'}(P_1) = 0.111$, $\Phi^{'}(P_2) = 0.130$, $\Phi^{'}(P_3) = 0.158$, and $\Phi^{'}(P_4) = 0.151$, with continued decreasing fit as K increases.

Table 2 provides the terms by group assignments, where the three groups seem to be related to (broadly speaking): graph traversal and components, general definitions, and trees, respectively. Figure 4 provides the three clusters solution with the nodes color coded and grouped by cluster membership. First, we see that there is still a substantial number of connections between the two largest groups; however, we emphasize that this is part of what makes clusterwise p^* models different from standard blockmodels (although, the relative within-cluster densities are still more pronounced than the between cluster densities). Namely, the goal is not to necessarily find the densest subgroups. The models that define the three groups have the following relevant parameters: Group 1 (cyclicity, instar, outstar, mixed star), Group 2 (cyclicity, instar, outstar, mixed star, choice), and Group 3 (cyclicity, instar, mixed star, choice). Additionally sensitivity analysis was conducted as in the prior example; however, the results are not displayed as they are nearly identical in nature to Sampson's data set.

Table 2. Clusterwise p^* group membership for glossary terms.

Group	Terms
Group 1	Acyclic, Adjacency structure, Adjacent, Bridge, Clique, Complete, Connected component, Connected Cycle diameter, Distance, Forest, Hamiltonian, Path, Spanning Subgraph, Spanning tree, Subgraph Trail, Tree, Walk
Group 2	Adjacency matrix, Ancestor, Arc, Arc List, Bipartite Graph, Binary Code, Chromatic number, Closure Condensed graph, Degree, Degree sequence, Descendant, Digraph, Edge, Graph, Homeomorphic, Incidence matrix Internal vertex, Isomorphic, k-colorable, Label, Leaf, Loop, Matching, Neighborhood, Node, Order, Orientation Parent, Pendant vertex, Prefix code, Perfect matching, Planar, Reduced graph, Regular, Saturated vertex Sibling, Size, Strongly connected, Topological order, Tournament, Underlying graph, Vertex
Group 3	Binary search tree, Child, Decision tree, Height, Level, m-ary tree, Offspring, Ordered tree, Rooted tree

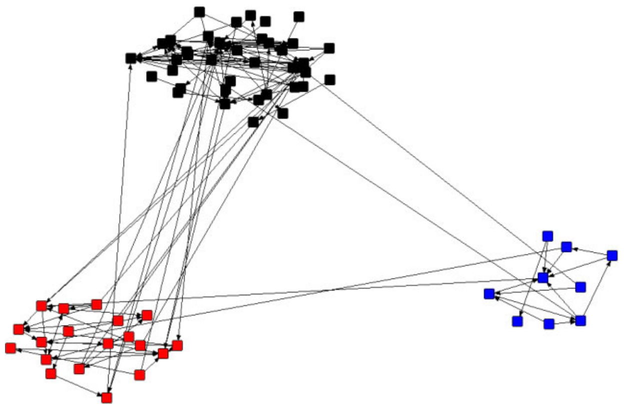

Fig. 4 The three cluster solution of the graph term glossary data. The blue cluster represents terms related to 'trees'; the red cluster represents terms related to 'graph traversal'; the black cluster represents general graph definitions. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

5. FUTURE DIRECTIONS

To our knowledge, this is the first attempt to cluster network data in a manner that is not based on finding dense clusters and/or 'blocks' with specific types of patterns. However, while the current manuscript introduces an exploratory procedure for finding subregions of a network that may contain differing structures, the main difficulties of combining partitioning and model fitting remain. Namely, that it is easy to take advantage of sample variation in order to extract clusters.

One potential advantage is that the clusterwise p^* approach can help guard against degenerate models by finding sets of local models that are not degenerate. Furthermore, this enhances the local prediction of the network by introducing flexibility into how different subregions of the network are modeled, enhancing understanding of differential functionality across the network. One drawback of the current work is that we only are modeling ties within each block, relying on the modeling process itself to ignore sparse regions of the networks that often occur *between* blocks. Future research will focus on incorporating within-cluster network models, likely as described herein, with between-cluster network models.

ACKNOWLEDGMENT

This manuscript was partially supported by the following grants to the first author: National Institute of Health Grant #1-K25-A017456-03 and Army Research Institute Grant #W91WAW-08-C-0034.

REFERENCES

[1] S. Wasserman and K. Faust, Social Network Analysis: Methods and Applications, New York, Cambridge, 1994.

[2] F. Lorrain and H. C. White, Structural equivalence of individuals in social networks, J Math Sociol 1 (1971), 49–80.

[3] L. Hubert, Data analysis implications of some concepts related to the cuts of a graph, J Math Psychol 15 (1977), 199–208.

[4] P. Doreian, V. Batagelj, and A. Ferligoj, Generalized Blockmodeling, Cambridge University Press, Cambridge, 2005.

[5] M. J. Brusco and D. Steinley, Inducing a blockmodel structure on two-mode data using seriation procedures, J Math Psychol 50 (2006), 468–477.

[6] P. E. Pattison and R. L. Breiger, Lattices and dimensional representations: Matrix decompositions and ordering structures, Social Netw 24 (2002), 423–444.

[7] M. J. Brusco and D. Steinley, An evaluation of a variable-neighborhood search method for blockmodeling of two-mode binary matrices based on structural equivalence, J Math Psychol 51 (2007), 325–338.

[8] M. J. Brusco and D. Steinley, Integer programs for one- and two-mode blockmodeling based on prespecified image matrices for structural and regular equivalence, J Math Psychol 53 (2009), 577–585.

[9] M. J. Brusco and D. Steinley, A tabu search heuristic for deterministic two-mode blockmodeling of binary network matrices, Psychometrika, in press.

[10] D. Steinley, K-means clustering: a half-century synthesis, Brit J Math Stat Psychol 59 (2006), 1–34.

[11] H.-F. Köhn, D. Steinley, and M. J. Brusco, The p-median model as a tool for clustering psychological data, Psychol Meth 15 (2010), 87–95.

[12] S. Wasserman and C. A. Anderson, Stochastic a posteriori blockmodels: construction and assessment, Social Netw 9 (1987), 1–36.

[13] K. Nowicki and T. A. B. Snijders, Estimation and prediction for stochastic blockstructures, J Am Stat Assoc 96 (2001), 1077–1087.

[14] E. M. Airoldi, D. Blei, S. Fienberg, and E. P. Xing, Mixed membership stochastic blockmodels, J Mach Learn Res 9 (2008), 1981–2014.

[15] M. S. Handcock, A. E. Raftery, and J. Tantrum, Model-based clustering for social networks (with discussion), J R Stat Soc A 170 (2007), 301–354.

[16] P. D. Hoff, A. E. Raftery, and M. S. Handcock, Latent space approaches to social network analysis, J Am Stat Assoc 97 (2002), 1090–1098.

[17] J. E. Besag, Spatial interaction and the statistical analysis of lattice systems (with discussion), J R Stat Assoc B 36 (1974), 192–236.

[18] S. Wasserman and P. Pattison, Statistical models for social networks, In Studies in Classification, Data Analysis, and Knowledge Organization, H. Kiers, J. Rasson, P. Groenen, and M. Schader, eds. Heidelberg, Springer, 2000.

[19] S. Wasserman and P. Pattison, Logit models and logistic regressions for social networks: I. An introduction to Markov random graphs and p^*, Psychometrika 60 (1996), 401–426.

[20] P. Pattsion and S. Wasserman, Logit models and logistic regressions for social networks: II. Multivariate relations, Brit J Math Stat Psychol 52 (1999), 169–193.

[21] G. L. Robins, P. Pattison, and S. Wasserman, Logit models and logistic regressions for social networks: III. Valued relations, Psychometrika 64 (1999), 371–394.

[22] G. L. Robins, P. Pattison, and P. Elliott, Network models for social influence processes, Psychometrika 66 (2001), 161–190.

[23] P. Pattison and G. L. Robins, Neighbourhood-based models for social networks, Sociol Methodol 32 (2002), 301–337.

[24] B. Crouch and S. Wasserman, Fitting p^*: Monte Carlo maximum likelihood estimation, In Paper presented at International Conference on Social Networks, Sitges, Spain, May 1999, 28–31.

[25] M. S. Handcock, Statistical models for social networks: inference and degeneracy, In Dynamic Social Network Modeling and Analysis, R. Breiger, K. Carley, and P. Pattison, eds. Washington, DC, National Academies Press, 2003, 220–240.

[26] T. A. B. Snijders, Markov chain Monte Carlo estimation of exponential random graph models, J Social Struct 3 (2002), 2.

[27] J. E. Besag, Statistical analysis of non-lattice data, The Statistician 24 (1975), 179–195.

[28] J. E. Besag, Efficiency of pseudo-likelihood estimation for simple Gaussian random fields, Biometrika 64 (1977), 616–618.

[29] D. Strauss, On a general class of models for interaction, SIAM Rev 28 (1986), 513–527.

[30] D. Strauss and M. Ikeda, Pseduolikelihood estimation for social networks, J Am Stat Assoc 85 (1990), 204–212.

[31] G. L. Robins, T. A. B. Snijders, P. Wang, M. S. Handcock, and P. E. Pattison, Recent developments in exponential random graph (p^*) models for social networks, Social Netw 29 (2007), 192–215.

[32] H. Späth, Algorithm 39: clusterwise linear regression, Computing 22 (1979), 367–373.

[33] H. Späth, Algorithm 48: a fast algorithm for clusterwise linear regression, Computing 29 (1982), 175–181.

[34] H. Späth, Mathematical Algorithms for Linear Regression, San Diego, CA, Academic, 1991.

[35] M. J. Brusco, J. D. Cradit, D. Steinley, and G. Fox, Cautionary remarks on the use of clusterwise regression, Multivariate Behav Res 43 (2008), 29–49.

[36] D. Steinley, Local optima in K-means clustering: What you don't know may hurt you, Psychol Methods 8 (2003), 294–304.

[37] D. Steinley and M. J. Brusco, A new variable weighting and selection procedure for K-means cluster analysis, Multivariate Behav Res 43 (2008), 77–108.

[38] D. W. Hosmer and S. Lemeshow, Applied Logistic Regression, (2nd ed.), New York, Wiley, 2004.

[39] M. J. Brusco, Clustering binary data in the presence of masking variables, Psych Meth 9 (2004), 510–523.

[40] D. Steinley and M. J. Brusco, Initializing K-means batch clustering: a critical evaluation of several techniques, J Classif 24 (2007), 99–121.

[41] D. Steinley and M. J. Brusco, Evaluating the performance of model-based clustering: recommendations and cautions, Psychol Meth 16 (2011), 63–79.

[42] F. S. Sampson, A novitiate in a period of change: An experimental and case study of social relationships, Unpublished Doctoral Dissertation, Cornell University, 1968.

[43] R. Breiger, S. Boorman, and P. Arabie, An algorithm for clustering relational data with applications to social network analysis and comparison with multidimensional scaling, J Math Psychol 12 (1975), 328–383.

[44] D. Steinley, Properties of the Hubert–Arabie adjusted Rand index, Psychol Meth 9 (2004), 386–396.

[45] V. Batagelj and A. Mrvar, Pajek datasets. 2006. http://vlado.fmf.uni-lj.si/pub/networks/data/. [Last Accessed June 2011].