This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

SAND2019-7000C

Generalized Blockmodeling of Multi-Valued Networks

Abstract

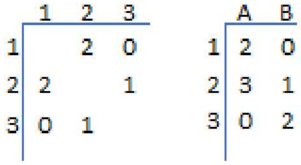
This research presents an extension to generalized blockmodeling where there are more than two types of objects to be clustered based on valued network data. We use the ideas in homogeneity blockmodeling to develop an optimization model to perform the clustering of the objects and the resulting partitioning of the ties so as to minimize the inconsistency of an empirical block with an ideal block. The ideal block types used in this modeling are null (all zeros), complete (all ones) and valued. Two case studies are presented: the Southern Women dataset and a larger example using a subset of the IMDb movie dataset.

1. Introduction

The goal of blockmodeling is the identification of clusters of objects and the partitioning of the set of ties between those objects into blocks ([1], [2], and [3]). Much of the early work in blockmodeling focused on structural equivalence as the basis for partitioning. [4] extended that basis to include regular equivalence. [5] suggested using the network data directly to perform this clustering and partitioning rather than summarizing the network information into similarity or dissimilarity matrices and appealing to a generic clustering algorithm. Generalized blockmodeling extends these concepts to include a wide array of block types and the explicit use of optimization to perform the partitioning on the network data directly.

The vast majority of the generalized blockmodeling literature focuses on a single matrix where all the entries are binary. Research including [2], [5], [6], and [7] extend these ideas to valued matrices allowing for the representation of the strength of ties. While much of the literature focuses on matrices for which the row and column objects are the same, several authors extend those ideas to matrices for which the row and columns refer to different types of objects, thereby representing two-mode network data (e.g., [8] and [9]). [17] describes methods for modeling multilevel networks including the combination of one-mode and two-mode networks into a single optimization. The goal of [17] is essentially the same as ours, however, we use three object types where that paper only uses two.

Our focus is the extension of value-based generalized blockmodeling tools (for both one and two-mode network data) to identify the underlying structure when more than two types of objects are under consideration and where information on the strength of the ties can be included. Figure 1 gives an illustration of our focus. In this example, there are two types of objects, people (labeled 1, 2, and 3) and locations (labeled A and B). The goal of the technique described in this paper is to support the development of conclusions of the nature "individuals 1 and 2 form a group and that group is associated with location A; whereas, individual 3 is a singleton group and associated with location B". Notice that we could illustrate the first matrix as a one-mode social network and the second network as a two-mode social network and apply existing blockmodeling tools to each matrix separately. Our focus is the simultaneous analysis of multiple networks of both types.



	1	2	3
1		2	0
2	2		1
3	0	1	

	A	B
1	2	0
2	3	1
3	0	2

Figure 1. Example Networks

[10] and [11] apply the ideas in blockmodeling to computational biology and show that the clustering of molecules is substantially improved by using both protein to protein interactions as well as protein sequence similarities rather than focusing on either separately. Their analysis is similar in structure to that given in Figure 1 with an important exception; their research focuses on partitioning a single type of object into clusters. Conceptually, our focus is the extension of that research augmented to multiple types of objects where the relationship between objects of different types are given in matrices. In the example given in Figure 1 above, there are two types of objects. In the IMDb movie data that we use [16], there are three types of objects.

The next section develops an explicit optimization model to determine these groups (and blocks). The third section describes a solution procedure for that

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

model. The fourth section applies the model and solution procedure to several illustrative examples. The fifth section gives opportunities for future research.

2. Model Formulation

A key element in the development of the mathematical model to identify these groups is the development of a criterion function. The criterion function provides a mechanism to understand the degree of inconsistency of a block with an ideal block. Remember, when objects are partitioned into groups we can examine the nature of the interaction of one group with another by considering the relevant block formed by the rows of one of the groups and the columns associated with the other. Suppose for each matrix m, an ideal block either has entries which are each equal to or below some value d_m or equal to or above some value e_m and $e_m > d_m$. Based on this definition, we can compute the inconsistency of that block from either ideal and simply assume the ideal that generates the smallest amount of inconsistency is appropriate.

Consider the example in Figure 1. Grouping individuals 1 and 2 together and leaving individual 3, and locations A and B each as singleton sets, provides several interesting blocks to assess each for their level of inconsistency from an ideal. The intersection of individuals 1 and 2 with location A, based on the right-hand matrix in Figure 1, creates a 2x1 vector that indicates that individuals 1 and 2 visited location A twice and three times, respectively. Suppose d_m is 1 and e_m is 2. Perhaps the two ideals could be characterized as the presence or absence of an association between individuals 1 and 2 and location A. Therefore, the inconsistency from the ideal of no association is then 2-1+3-1=3 whereas the inconsistency from the ideal of an association is 0 (since both entries are equal to or greater than 2).

It is useful to notice that if we have a binary matrix and set d_m equal to 0 and e_m equal to 1, these inconsistency computations match that commonly used for binary blockmodeling. These computations for block inconsistency are similar to those given by [7] except we allow for a different critical value to distinguish “associated” from “not associated”. They use restrictions that certain values must be zero, a user-defined value or some function of the entries in the row (or column) must be at least some value.

Suppose there are *N* types of objects to cluster and m matrices to support that clustering. Also, let $\lambda(m)$ equal 0 if the rows and columns of matrix *m* correspond to the same object type and one otherwise. We assume that if the interactions, as given by the matrix, are between objects of the same type, the ideal for the level of interaction between objects in the same cluster is defined by e_m whereas the ideal for the level of interaction between objects in different clusters should be defined by dm. When the rows and columns of the matrix correspond to different types of objects we make no assumption as to which ideal is correct. For the applications to be supported by this formulation, we commonly expect one of the object types to be individuals. Since modeling the interactions among individuals is so important in social network analysis, we provide the capability to assume an ideal for interactions between objects of the same type.

Let *m(r)* and *m(c)* be the clusters associated with the row and column objects, respectively. Therefore, the goal of the clustering is to minimize the following objective.

$$\sum_{m|\lambda(m)=0}\left[\sum_{\substack{m(r)\\m(c)}|m(r)=m(c)}\ \sum_{\substack{i\epsilon m\,(r)\\ j\epsilon m\,(c)}i\neq j} max\left(e_m - m_{ij}, 0\right)\right.$$

$$\left. + \sum_{\substack{m(r)\\m(c)}|m(r)\neq m(c)}\ \sum_{\substack{i\epsilon m\,(r)\\ j\epsilon m\,(c)}} max\left(m_{ij} - d_m, 0\right)\right] \quad +$$

$$\sum_{m|\lambda(m)=1}\ \sum_{\substack{m(r)\\m(c)}} \min\left[\sum_{\substack{i\epsilon m\,(r)\\ j\epsilon m\,(c)}} max\left(e_m - m_{ij}, 0\right), \sum_{\substack{i\epsilon m\,(r)\\ j\epsilon m\,(c)}} max\left(m_{ij} - d_m, 0\right)\right]$$

$$(1)$$

where m_{ij} is the entry in the *(i,j)* position in matrix *m*. The first term in the objective is the penalty associated with matrices that have the same objects for the rows and columns. The second term focuses on matrices for which the object type differs between the rows and the columns. Let’s focus on the first component of the first term. There are two situations which are considered, and those situations are represented by the first and second components in the brackets, respectively; namely when the cluster for the row objects is the same as the cluster for the column objects and when they are different. When they are the same, we assume that the ideal is the higher level of interaction given by parameter e_m. Therefore, for all entries in the block that represents the cluster with itself, between unique objects, we simply take the maximum of e_m minus the entry and zero. If the interaction is equal to or higher than the minimum given by the ideal, the penalty is assumed to be zero. If the interaction is lower than this minimum, a positive penalty is assessed. When the cluster for the rows is different than that for the columns, the ideal interaction is at the level of d_m or lower. Hence, we take the entry for each pair of objects, one from each cluster and subtract the "allowable" level of interaction. If this interaction is exceeded, a penalty is assessed. It is useful to notice that this penalty structure can be considered to be a generalization of that described by [7].

The second terms focus on matrices for which the objects that comprise the rows and columns are different. When they are different we must test which ideal is closer to the entries in the block. That is, is the ideal associated with e_m or that associated with d_m a better representation for the interaction between the clusters? Hence, this second term requires the minimum function. The first component within the brackets for this second term computes the penalty if the ideal for the interaction is an association and the second term computes the penalty if the ideal for the interaction is the absence of an association. The minimum function simply selects the penalty to apply for the ideal that is closest to the block values.

The minimum and maximum functions can be replaced by additional variables. However, we do not do this substitution in the interest of clarity and brevity. This conversion is not needed by the solution procedure either; which is a tabu search and described in the next section.

Suppose the set of object types is indexed by n $(n=1,...N)$, the set of objects of type n is indexed by t_n $(t_n=1_n,...T_n)$ and the set of clusters which contain objects of type n is indexed by k_n $(k_n=1_n,...K_n)$. Each object must belong to one and only one cluster where that cluster only contains objects of that type. This restriction is given by the following equation.

$$\sum_{k_n} \gamma_{nk_nt_n} = 1 \qquad \forall n, t_n \qquad (2)$$

where $\gamma_{nk_nt_n}$ is a binary variable that is one if object t_n of object type n belongs to cluster k_n and zero otherwise. Notice that this formulation assumes that the number of clusters for each object type is known. It also allows clusters to be empty, if that leads to a better objective value.

To illustrate this formulation, consider the example given in Figure 1. In this example, there are two matrices so m ranges from 1 to 2. There are also two types of objects: people and locations, where n is 1 for people and 2 for locations. Suppose we may have up to 2 clusters of people and 2 clusters of locations. Further suppose that e_1 is 2 and d_1 is 1. That is, the ideal for communication between people within the same cluster is 2 or more and the ideal for communication between people in different clusters is 0 or 1. Finally, suppose that e_2 is 2 and d_2 is 1; ideally people that are associated with a location visit that location at least twice and people not associated with a location ideally no more than once.

Now, suppose individuals 1 and 2 are in one cluster and individual 3 is in another cluster. Further, suppose each facility forms a singleton cluster. The computations associated with the objective function for this grouping are as follows. First, consider the first matrix (communication between individuals). There are four blocks for which the penalty stemming from this clustering of individuals is needed. Two blocks are associated with the first component in the first term in the objective (cluster 1 with itself and cluster 2 with itself) and two with the second component in this same term (cluster 1 with cluster 2 and cluster 2 with cluster 1). The penalty associated with the block formed by cluster 1 with itself is zero because the one pair of individuals has a level of communications which is equal to the minimum allowed as given by e_m. The penalty formed by cluster 2 with itself is zero for the same reason. The penalty associated with the block formed by cluster 1 with cluster 2 is zero because the communication between individual 3 and individuals 1 and 2 does not exceed the maximum allowed as given by d_m. Similarly, the cluster 2 to cluster 1 block also produces no penalty.

3. Solution Procedure

This section describes a Tabu Search (TS) algorithm to solve the optimization problems described above. A key element of defining a TS algorithm is to identify what constitutes a neighborhood for a solution *S*, *π(S)*, where a solution is a mapping of each object to a single cluster for that type of object. Our assumption is that a neighboring solution is exactly the same except a single object has moved from one cluster to another cluster. Rather than investigating all solutions in the current neighborhood, the user specifies the number of nodes to examine for each local search (MaxNodes). Each node is selected at random, to avoid getting trapped in local optima, and its movement to each available cluster is tested against a proxy objective. The proxy objective evaluates the sum of common link connections with each cluster node differenced with the sum of all differing link connections. This method is more efficient than a full objective evaluation and helps the algorithm progress towards better solutions when there are different but inferior moves that produce the same overall objective value.

In order to reduce the likelihood of cycling we maintain a tabu list of moves that have occurred over the last tabu tenure iterations. The entries on this list are simply a list of the objects and the clusters they have moved from and into. This allows us to create rules based on this information that minimize the chance of cycling.

This algorithm is initially seeded with a greedy solution which clusters nodes together that have the strongest common links. The number of tabu search steps determines the number of times the local search procedure is executed on the current best solution. Each local search is then executed a fixed number of times to improve the current and global best solution. If during this progression the current solution fails to improve for a predetermined number of tabu search steps, a new random or greedy solution (at approximately a 50/50 split) is selected as a new starting point. For each tabu step, we keep the best solution found. The solution reported is then the best identified over all steps.

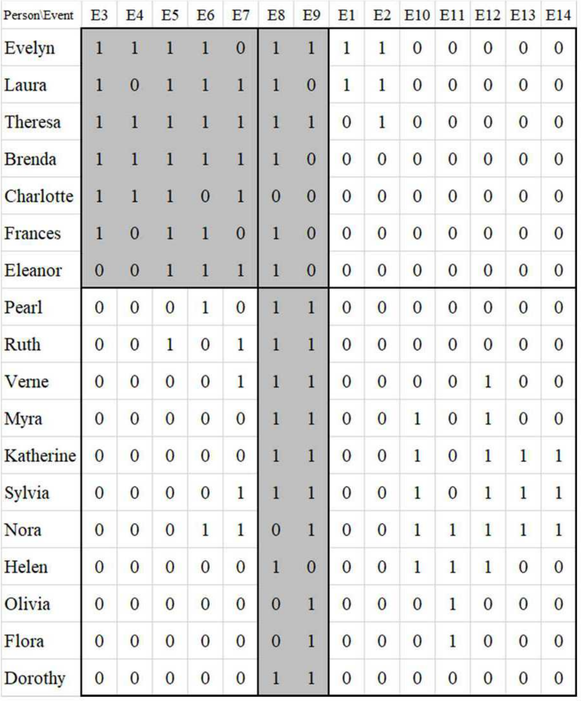
4. Illustrative Examples

In this section, we focus on three examples, each with different characteristics. First, we focus on a single matrix for which the columns and rows represent different types of objects, and the relationship between each pair of objects (where each object in the pair is of a different type) is binary. This example demonstrates how the above modeling approach can be used to perform generalized block modeling of two-mode network data. Next, we focus on a matrix of one-mode network data but for which the relationships are valued. This example illustrates the use of the modeling approach to perform valued generalized block modeling. Finally, we turn to an example which involves three types of objects for which one type of object forms the rows of one matrix and the columns of another. This example illustrates how blockmodeling can be used to simultaneously address matrices of different structure.

4.1. Blockmodeling with two object types

Our first and second examples are based on the dataset described in [12] focused on Southern Women and their participation in social events. For an interesting and detailed discussion of this data set see [13]. The mapping of individuals to the events they attended is given in Table 1, where a one indicates attendance at the event and a zero indicates that the person did not attend the event. Table 1 also illustrates the clustering produced using blockmodeling when two women clusters and three event clusters have been specified.

Table 1. Matrix of Southern Women Data with 2 Clusters for Women and 3 for Events

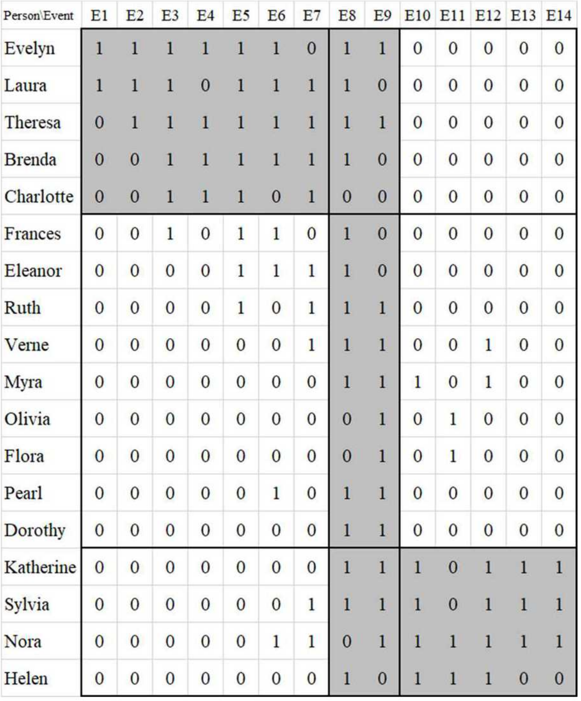


Person\Event	E3	E4	E5	E6	E7	E8	E9	E1	E2	E10	E11	E12	E13	E14
Evelyn	1	1	1	1	0	1	1	1	1	0	0	0	0	0
Laura	1	0	1	1	1	1	0	1	1	0	0	0	0	0
Theresa	1	1	1	1	1	1	1	0	1	0	0	0	0	0
Brenda	1	1	1	1	1	1	0	0	0	0	0	0	0	0
Charlotte	1	1	1	0	1	0	0	0	0	0	0	0	0	0
Frances	1	0	1	1	0	1	0	0	0	0	0	0	0	0
Eleanor	0	0	1	1	1	1	0	0	0	0	0	0	0	0
Pearl	0	0	0	1	0	1	1	0	0	0	0	0	0	0
Ruth	0	0	1	0	1	1	1	0	0	0	0	0	0	0
Verne	0	0	0	0	1	1	1	0	0	0	0	1	0	0
Myra	0	0	0	0	0	1	1	0	0	1	0	1	0	0
Katherine	0	0	0	0	0	1	1	0	0	1	0	1	1	1
Sylvia	0	0	0	0	1	1	1	0	0	1	0	1	1	1
Nora	0	0	0	1	1	0	1	0	0	1	1	1	1	1
Helen	0	0	0	0	0	1	0	0	0	1	1	1	0	0
Olivia	0	0	0	0	0	0	1	0	0	0	1	0	0	0
Flora	0	0	0	0	0	0	1	0	0	0	1	0	0	0
Dorothy	0	0	0	0	0	1	1	0	0	0	0	0	0	0

After employing a range of analytical procedures, [13] concluded that the data suggest that the women should be partitioned into two groups where membership in the first group is Evelyn, Laura, Theresa, Brenda, Charlotte, Frances, Eleanor, Pearl and Ruth with the remainder in the second group. [13] also presents a consensus analysis using 21 procedures. All 21 procedures suggested that all pairs of women from the set Evelyn, Laura, Theresa, Brenda, Charlotte and Frances belonged together. Further, they also suggested that all pairs of women from the set Myra, Katherine, Sylvia, Nora and Helen also belonged together. [13] does not suggest a partitioning of the events.

We apply the formulation and solution procedure described in the previous two sections to this dataset when the maximum number of person clusters is two and the maximum number of event clusters is three. Table 1 illustrates the suggested clustering. The number of inconsistencies associated with this solution is 51. That is, of the 252 entries in the matrix in Table 1, 51 are not consistent with the clustering suggested by the model. The key difference in the assignment of women to clusters suggested by this model and that discussed in [13] is that Pearl and Ruth are part of the second cluster based on this model. This assignment stems from the fact that women in cluster one are associated with events E3-E7 but Ruth and Pearl each only attended one of those events. The second cluster of women is associated with the second cluster of events (E8 and E9), which both Pearl and Ruth also attend. In addition, the first cluster of women is also associated with these events.

Table 2. Matrix of Southern Women Data with 3 Clusters for Women and 3 for Events



Person\Event	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14
Evelyn	1	1	1	1	1	1	0	1	1	0	0	0	0	0
Laura	1	1	1	0	1	1	1	1	0	0	0	0	0	0
Theresa	0	1	1	1	1	1	1	1	1	0	0	0	0	0
Brenda	0	0	1	1	1	1	1	1	0	0	0	0	0	0
Charlotte	0	0	1	1	1	0	1	0	0	0	0	0	0	0
Frances	0	0	1	0	1	1	0	1	0	0	0	0	0	0
Eleanor	0	0	0	0	1	1	1	1	0	0	0	0	0	0
Ruth	0	0	0	0	1	0	1	1	1	0	0	0	0	0
Verne	0	0	0	0	0	0	1	1	1	0	0	1	0	0
Myra	0	0	0	0	0	0	0	1	1	1	0	1	0	0
Olivia	0	0	0	0	0	0	0	0	1	0	1	0	0	0
Flora	0	0	0	0	0	0	0	0	1	0	1	0	0	0
Pearl	0	0	0	0	0	1	0	1	1	0	0	0	0	0
Dorothy	0	0	0	0	0	0	0	1	1	0	0	0	0	0
Katherine	0	0	0	0	0	0	0	1	1	1	0	1	1	1
Sylvia	0	0	0	0	0	0	1	1	1	1	0	1	1	1
Nora	0	0	0	0	0	1	1	0	1	1	1	1	1	1
Helen	0	0	0	0	0	0	0	1	0	1	1	1	0	0

In Table 1, the third cluster of events (E1, E2 and E10-E14) does not appear to tell much of a story with respect to either group of women. This suggests that looking for a solution that has three clusters of women might be useful. That solution is illustrated in Table 2. The number of inconsistencies associated with this solution is 41 (a reduction of 10 over the previous solution) which translates into about 16% of the entries in the matrix in Table 1. This solution removes Frances and Eleanor from the first cluster of women in the previous solution and groups them with Ruth, Verne, Myra, Olivia, Flora, Pearl and Dorothy. The third group of women is composed of Katherine, Sylvia, Nora, and Helen. Under this clustering, the first group of women is associated with events E1-E7. All three clusters of women are associated with E8 and E9 and the third cluster is associated with events E10-E14.

It is useful to notice that this solution differs from the consensus analysis given in [13] in that all pairs of women from the set Evelyn, Laura, Theresa, Brenda, Charlotte and Frances are concluded to belong together and that all pairs of women from the set Myra, Katherine, Sylvia, Nora and Helen also are concluded to belong together. The motivation from this model to omit Frances from the first cluster (which contains the other five women) is that she only attends 3 of the events E1 through E7 so the penalty is lower by one if she is placed in the second cluster (rather than the first). As for Myra, she only attended 2 of the 5 events in the third event cluster (E10-E14) so the penalty is one less to place her in the second person cluster rather than the third (person cluster).

It is useful to notice that we did not have to pre-specify any block types to produce solutions that are consistent with the literature [14]. The model concluded whether the block ideal should be a complete block (all ones) or a null block (all zeros). Also, it is very easy to see the motivation behind the groupings the model has suggested.

Next, to explore the application of this formulation to valued generalized blockmodeling, we convert the two-mode network data associated with the Southern Women dataset into a one-mode representation where the objects are the women and the relationships are the number of events that pairs of women attended. That data is given in Table 3 formatted to illustrate the clusters (max of two allowed) and the blocks (within cluster minimum value of 3 and between cluster maximum value of 2). This is the same clustering suggested by [13]. This solution is rather insensitive to the within cluster minimum value and the between cluster maximum value, so it is a very stable solution using this formulation.

Table 4 gives the suggested clustering when 3 clusters are allowed. Notice that the third cluster results from combining Pearl from the previous first cluster and Olivia, Flora and Dorothy from the previous second cluster. This third cluster is made up of individuals that do not attend very many events (in comparison to the other women) and what events they do attend tend to be somewhat common among them. For example, all four women attended event E9. Two of the four women attended E8 and E11. Of the events for which at least one of the four attended, E6 had the minimum attendance from the group with only Pearl attending.

Table 3. One-mode analysis of Southern Women data with two clusters

Person	Evelyn	Laura	Theresa	Brenda	Charlotte	Frances	Eleanor	Ruth	Pearl	Verne	Myra	Katherine	Sylvia	Nora	Helen	Olivia	Flora	Dorothy
Evelyn	-	6	7	5	3	4	3	3	3	2	2	2	2	2	1	1	1	2
Laura	6	-	6	5	3	4	4	3	2	2	1	1	2	2	2	0	0	1
Theresa	7	6	-	6	4	4	4	4	3	3	2	2	3	3	2	1	1	2
Brenda	5	5	6	-	4	4	4	3	2	2	1	1	2	2	2	0	0	1
Charlotte	3	3	4	4	-	2	2	2	0	1	0	0	1	1	1	0	0	0
Frances	4	4	4	4	2	-	3	2	2	1	1	1	1	1	1	0	0	1
Eleanor	3	4	4	4	2	3	-	3	2	2	1	1	2	2	2	0	0	1
Ruth	3	3	4	3	2	2	3	-	2	3	2	2	3	2	2	1	1	2
Pearl	3	2	3	2	0	2	2	2	-	2	2	2	2	2	1	1	1	2
Verne	2	2	3	2	1	1	2	3	2	-	3	3	4	3	3	1	1	2
Myra	2	1	2	1	0	1	1	2	2	3	-	4	4	3	3	1	1	2
Katherine	2	1	2	1	0	1	1	2	2	3	4	-	6	5	3	1	1	2
Sylvia	2	2	3	2	1	1	2	3	2	4	4	6	-	6	4	1	1	2
Nora	2	2	3	2	1	1	2	2	2	3	3	5	6	-	4	2	2	1
Helen	1	2	2	2	1	1	2	2	1	3	3	3	4	4	-	1	1	1
Olivia	1	0	1	0	0	0	0	1	1	1	1	1	1	2	1	-	2	1
Flora	1	0	1	0	0	0	0	1	1	1	1	1	1	2	1	2	-	1
Dorothy	2	1	2	1	0	1	1	2	2	2	2	2	2	1	1	1	1	-

Table 4. One-mode analysis of Southern Women data with three clusters

Person	Evelyn	Laura	Theresa	Brenda	Charlotte	Frances	Eleanor	Ruth	Verne	Myra	Katherine	Sylvia	Nora	Helen	Olivia	Flora	Pearl	Dorothy
Evelyn	-	6	7	5	3	4	3	3	2	2	2	2	2	1	1	1	3	2
Laura	6	-	6	5	3	4	4	3	2	1	1	2	2	2	0	0	2	1
Theresa	7	6	-	6	4	4	4	4	3	2	2	3	3	2	1	1	3	2
Brenda	5	5	6	-	4	4	4	3	2	1	1	2	2	2	0	0	2	1
Charlotte	3	3	4	4	-	2	2	2	1	0	0	1	1	1	0	0	0	0
Frances	4	4	4	4	2	-	3	2	1	1	1	1	1	1	0	0	2	1
Eleanor	3	4	4	4	2	3	-	3	2	1	1	2	2	2	0	0	2	1
Ruth	3	3	4	3	2	2	3	-	3	2	2	3	2	2	1	1	2	2
Verne	2	2	3	2	1	1	2	3	-	3	3	4	3	3	1	1	2	2
Myra	2	1	2	1	0	1	1	2	3	-	4	4	3	3	1	1	2	2
Katherine	2	1	2	1	0	1	1	2	3	4	-	6	5	3	1	1	2	2
Sylvia	2	2	3	2	1	1	2	3	4	4	6	-	6	4	1	1	2	2
Nora	2	2	3	2	1	1	2	2	3	3	5	6	-	4	2	2	2	1
Helen	1	2	2	2	1	1	2	2	3	3	3	4	4	-	1	1	1	1
Olivia	1	0	1	0	0	0	0	1	1	1	1	1	2	1	-	2	1	1
Flora	1	0	1	0	0	0	0	1	1	1	1	1	2	1	2	-	1	1
Pearl	3	2	3	2	0	2	2	2	2	2	2	2	2	1	1	1	-	2
Dorothy	2	1	2	1	0	1	1	2	2	2	2	2	1	1	1	1	2	-

4.2. Blockmodeling with three object types

In this section, we focus on an example which demonstrates the core contribution of this paper – the simultaneous analysis of multiple matrices, each of different structure. Our example includes three two-mode matrices and three object types which form the rows and columns of the matrices.

We use a subset from 10 years of IMDb data [16], filtered down to the top 1000 movies from 2006 to 2016. We then selected 37 actors and only kept movies that had at least two actors on the list which resulted in 31 movies being selected. Only those genres which are associated with these movies were included, resulting in 12 genres (with "music" being removed since it was only associated with a single movie). Additionally, at most three genres and four actors are associated with a single movie. The resulting dataset is shown in Tables 5 and 6. Given these restrictions, it's not surprising that many of the movies are part of a series (e.g., the *Harry Potter* series). To better demonstrate how the blockmodeling algorithm works, we include a few movies which share actors from two different series and have some differing genres from those series. Given this structure, we created a three-type dataset consisting of three two-mode matrices: Actor-Movie, Movie-Genre, and Genre-Actor. We apply blockmodeling to this dataset and compare the results to analyzing just the two-mode Movie-Actor relationships.

When analyzed *without* the genre data, the Movie-Actor relationships produce a very clean clustering structure when both are broken into 11 clusters each. This cluster value was discovered by increasing the maximum allowed until the solution produced what a human would consider to be an intuitive solution. Though solutions with 12 Actor and 11 Movie clusters exist that produce the same objective value, our assumption is that a solution with fewer clusters is better. For the most part, each Movie cluster is associated with a single Actor cluster as indicated by the alternating shading applied to the various row groupings in Table 6 (e.g., all of the actors in the four *Harry Potter* movies comprise the actor cluster associated with the *Harry Potter* movie cluster). The few exceptions occur when actors cross multiple Movie clusters as shown in table 7 and noted below:

- *Transcendence* gets included in the cluster with *Dark Shadows* and *Sweeney Todd* even though one of the actors, Morgan Freeman, is clustered with the actors from another series (in this case, the *Fallen* series).

- *Death Race* is placed in its own movie cluster since it includes an actor from the *Transformers* (Tyrese Gibson) and an actor from one of the *Fast* movies (Jason Statham). Since Jason Statham is only in one of the *Fast* movies, he is placed in a cluster by himself and *Death Race* is only associated with his cluster.
- The actors in the *Alice* series are split into two clusters with Anne Hathaway (AH) and Mia Wasikowska (MW) in one cluster and Helena Bonham Carter (HBC) and Johnny Depp (JD_1) in a second cluster. This is due to the fact that HBC and JD_1 are separately associated with the *Dark-Sweeney-Trans* movie cluster.

Table 5. 37 actors and 12 genres from IMDb dataset

Actor	ID		Genre	ID
Aaron Eckhart	AE		Action	Act
Angela Bassett	AB		Adventure	Adv
Anne Hathaway	AH		Comedy	Com
Cynthia Nixon	CN		Crime	Cri
Daniel Radcliffe	DR		Drama	Dra
Devon Bostick	DB		Family	Fam
Dwayne Johnson	DJ		Fantasy	Fan
Emma Watson	EW		Horror	Hor
Gerard Butler	GB		Mystery	Mys
Helena Bonham Carter	HBC		Romance	Rom
James Franco	JF		Sci-Fi	Sci
Jason Statham	JS		Thriller	Thr
Jennifer Lawrence	JL			
Johnny Depp	JD_1			
Jordana Brewster	JB			
Josh Duhamel	JD_2			
Josh Hutcherson	JH			
Kim Cattrall	KC			
Kristin Davis	KD			
Liam Hemsworth	LH			
Megan Fox	MF_1			
Mia Wasikowska	MW			
Michael Gambon	MG			
Michelle Rodriguez	MR			
Morgan Freeman	MF_2			
Paul Walker	PW			
Rachael Harris	RH			
Robert Capron	RC			
Rupert Grint	RG			
Sarah Jessica Parker	SJP			
Seth Rogen	SR			
Shia LaBeouf	SL			
Steve Zahn	SZ			
Tyrese Gibson	TG			
Vin Diesel	VD			
Woody Harrelson	WH			
Zachary Gordon	ZG			

Table 6. Filtered IMDb dataset with actor-movie clustering ignoring genre information

Movie	ID	Genre	Actors
Alice in Wonderland	Alice_1	Adv, Fam, Fan	AH, JD_1, HBC, MW
Alice Through the Looking Glass	Alice_2	Adv, Fam, Fan	AH, JD_1, HBC, MW
Dark Shadows	Dark	Com, Fan, Hor	HBC, JD_1
Sweeney Todd: The Demon Barber of Fleet Street	Sweeney	Dra, Hor	HBC, JD_1
Transcendence	Trans	Dra, Mys, Rom	JD_1, MF_2
London Has Fallen	Fall_1	Act, Cri, Dra	AB, AE, GB, MF_2,
Olympus Has Fallen	Fall_2	Act, Thr	AB, AE, GB, MF_2
Transformers	Xform_1	Act, Adv, Sci	JD_2, MF_1, SL, TG
Transformers: Dark of the Moon	Xform_2	Act, Adv, Sci	JD_2, SL, TG
Transformers: Revenge of the Fallen	Xform_3	Act, Adv, Sci	JD_2, MF_1, SL, TG
Death Race	Race	Act, Sci, Thr	JS, TG
Fast & Furious	Fast_1	Act, Cri, Thr	JB, MR, PW, VD
Fast Five	Fast_2	Act, Cri, Thr	DJ, JB, PW, VD
Furious 6	Fast_3	Act, Cri, Thr	DJ, MR, PW, VD
Furious Seven	Fast_4	Act, Cri, Thr	DJ, JS, PW, VD
Diary of a Wimpy Kid	Diary_1	Com, Fam	RC, RH, SZ, ZG
Diary of a Wimpy Kid: Dog Days	Diary_2	Com, Fam	DB, RC, SZ, ZG
Diary of a Wimpy Kid: Rodrick Rules	Diary_3	Com, Fam	DB, RC, RH, ZG
Harry Potter and the Deathly Hallows: Part 1	Potter_1	Adv, Fam, Fan	DR, EW, RG
Harry Potter and the Deathly Hallows: Part 2	Potter_2	Adv, Dra, Fan	DR, EW, RG, MG
Harry Potter and the Half-Blood Prince	Potter_3	Adv, Fam, Fan	DR, EW, RG, MG
Harry Potter and the Order of the Phoenix	Potter_4	Adv, Fam, Fan	DR, EW, RG
Sex and the City	City_1	Com, Dra, Rom	CN, KC, KD, SJP
Sex and the City 2	City_2	Com, Dra, Rom	CN, KC, KD, SJP
The Hunger Games	Hung_1	Adv, Sci, Thr	JL, JH, LH
The Hunger Games: Catching Fire	Hung_2	Act, Adv, Mys	JL, JH, LH
The Hunger Games: Mockingjay - Part 1	Hung_3	Act, Adv, Sci	JL, JH, LH, WH
The Hunger Games: Mockingjay - Part 2	Hung_4	Act, Adv, Sci	JL, JH, LH, WH
The Interview	Inter	Com	JF, SR
Pineapple Express	Pine	Act, Com, Cri	JF, SR
This Is the End	End	Com, Fan	JF, SR

Table 7. Movie-Genre-Actor cluster associations when the number of clusters is limited to 11

Movie Cluster	Associated Genre Clusters	Missing/ (Extra) Genres	Associated Actor Clusters
Alice	Adventure, Family, Fantasy	none	[AH, MW], [HBC, JD_1]
Dark-Sweeney-Trans	Drama, Horror	Comedy, Fantasy, Romance	HBC, JD_1
Fall	Action, Drama, [Crime-Thriller]	none	AB, AE, GB, MF_2,
Xform	Action, Adventure, Sci-Fi	none	JD_2, MF_1, SL, TG
Race	Action, Sci-Fi, [Crime-Thriller]	(Crime)	JS
Fast	Action, [Crime-Thriller]	none	DJ, JB, MR, PW, VD
Diary	Comedy, Family	none	DB, RC, SZ, ZG
Potter	Adventure, Family, Fantasy	Drama	DR, EW, RG, MG
City	Comedy, Drama, Romance	none	CN, KC, KD, SJP
Hung	Action, Adventure, Sci-Fi	Thriller, Mystery	JL, JH, LH, WH
Inter-Pine-End	Comedy	Action, Crime, Fantasy	JF, SR

When we include genre information with a limit of 11 clusters for each type, we get the same actor and movie clusters as without genre. This result is primarily due to there being only 12 genres total, with only Crime and Thriller being paired since they appear together in all four of the *Fast* movies. This clustering allows the genres to associate quite well, with a few exceptions, as shown in table 7. This table delineates the associations between the Movie and Genres clusters as well as the associations from Actors to Movies, which creates a full three-way association from Movie to Genre to Actor. For all but four of the Movie clusters, all genres are properly associated with the following exceptions noted below:

- *Race* has an incorrect association to Crime, however, this is due to Crime and Thriller being clustered together.
- Since Drama is only associated with one of the four *Potter* movies (Potter_2), the series cluster is not associated to Drama.
- Since Thriller and Mystery are each only associated with one of the four *Hunger* movies (Hung_1 and Hung_2), the series cluster is not associated to either Thriller or Mystery.
- Since Comedy is the only genre that appears within all three movies in the *Inter-Pine-End*, cluster, the other singleton genres (Action, Crime, and Fantasy) are not associated.

To see the effect that genre associations have, we need to examine a sub-optimal configuration which forces the algorithm to cluster items that are not as distinctly similar as our 11-cluster example. We do this by limiting the maximum number of types in the Actor and Movie clusters to eight but keep the number of Genre clusters at 11. The change in Movie-Actor clusters between ignoring versus including genre information is significant in a few instances as shown in table 8. In both cases, most of the Movie and Actor clusters are maintained, with the exception of the highlighted instances.

Table 8. Movie-Actor associations when the number of clusters is limited to eight

No Genre Information		With Genre Information	
Movie Cluster (no/ Genre)	Actor Cluster	Movie Cluster	Actor Cluster
Alice, Dark, Sweeney, Trans	AH, MW, HBC, JD_1	Alice, Dark	AH, MW, HBC, JD_1
Fall, Race, Inter, Pine, End	DJ, JB, MR, PW, VD, JS, AB, AE, GB, MF_2,	Fast, Fall, Race, Sweeney	DJ, JB, MR, PW, VD, JS, AB, AE, GB, MF_2,
City	CN, KC, KD, SJP	City, Trans	CN, KC, KD, SJP
Fast	DJ, JB, MR, PW, VD		
Xform	JD_2, MF_1, SL, TG	Xform	JD_2, MF_1, SL, TG
Diary	DB, RC, SZ, ZG	Diary	DB, RC, SZ, ZG
Potter	DR, EW, RG, MG	Potter	DR, EW, RG, MG
Hung	JL, JH, LH, WH	Hung	JL, JH, LH, WH
		Inter-Pine-End	JF, SR

When ignoring genre information, the Movie and Actor clustering in table 8 is less intuitive than when it is included. Some examples of the impact of genre are:

- Combining the *Death Race* and the *Fall* series with *Inter-Pine-End* is counterintuitive since the latter movies have very little overlap in terms of group genre which is primarily Comedy.

- Combining the *Fast* and *Fall* series is logical since both are associated with the Action, Crime and Thriller genres.
- Since *Race* overlaps with the Action and Thriller genres and *Sweeney* overlaps the Drama genre with *Fall_1*, combining these with the *Fast* and *Fall* series also makes sense.
- The *Alice* series, is appropriately associated with *Dark* both in the Fantasy genre overlap as well as the common actors, but less so *Sweeney* and *Trans* (*Transcendence*) since they have no common genres with *Alice*.
- The *City* series is appropriately associated with *Transcendence* (*Trans*) in the overlap of the Drama and Romance genres.

This example demonstrates that when the number of clusters must be less than an optimal number, the blockmodeling algorithm can still make reasonable choices, which can be improved if more information is included (in this case genre associations). If the two-mode/three-type blockmodeling algorithm were to be used to categorize similar movies (as in the case of a movie recommender), it could do so based on both Actor and Genre information. When relying on just the actor associations, clustering of movies can be problematic if collections of movies are related by genre but not by actor.

5. Conclusions

This paper describes an extension to generalized blockmodeling where there are more than two types of objects to be clustered based on valued network data. The ideas in homogeneity block modeling are used to develop an optimization model to perform the clustering of the objects and the resulting partitioning of the ties so as to minimize the inconsistency of an empirical block with an ideal block. The ideal block types used in this modeling were null, complete and a new type that is related to that used in [7]. A tabu search solution procedure was developed to solve the resultant optimization.

This modeling approach for valued network data is dependent on two parameters e_m and d_m where the ideal for the level of interaction between objects in the same cluster is at least e_m and the ideal for the level of interaction between objects in different clusters is assumed to be no more than d_m. These two parameters provide more flexibility to tailor the analysis to application then that given in [7], which relies on a single parameter for this purpose.

Two case studies using the formulation and solution procedure were described, two based on the Southern Women dataset [12] and a third based on IMDb movie data [16]. The clustering suggested by this formulation for the Southern Women dataset is consistent with that given in [13]. As for the movie analysis, the formulation identified clusters of similar movies based on both associated genres and actors. This hybrid approach using three different two-mode matrices provides a more intuitive clustering of movies than using just actor or just genre associations and could be the basis for a movie recommender system similar to those employed by movie streaming services.

There are opportunities for future work in at least two complementary directions. One opportunity focuses on the explicit introduction of uncertainly. For example, as these ideas are used in practice, some of the information available on the ties between objects could be subject to some uncertainty. As an illustration, if one were to apply these tools to attempt to understand the activities of a market competitor (industrial competition), and there were observations as to who is frequenting different locations as an indicator of the character of the activities undertaken at that location, that data might be limited by the ability to collect this information. One mechanism to include this in the analysis is to simply ignore relationships in the computation of the objective function value that are subject to these issues. Alternatively, weights could be associated with each tie to indicate the quality of the information that lead to that estimate of the relationship. Further research to explore what mechanisms to use in order to incorporate this information into the analysis is very important.

A second opportunity is to address data collected over time. For example, suppose we had the same types of information collected at multiple points in time; it would be useful to identify a clustering and partitioning of ties that departs as little of possible from block ideals over all time periods. We might require that the solution include membership in clusters that is invariant over time or we might allow the membership to change, but with a penalty. Allowing the membership to change over time is useful in that organizational structures are often fluid and understanding the nature of the fluidity is very useful.

6. Acknowledgements

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. The views expressed in the article do not necessarily represent the views of the U.S. Department of Energy or the United States Government. SAND NO. 2019-xxxx C

7. References

[1] F. Lorrain and H.C. White, "Structural Equivalence of Individuals in Social Networks", Journal of Mathematical Sociology 1, 1971, pp. 49–80.

[2] R.L. Breiger, S. Boorman, and P. Arabie, "An Algorithm for Clustering Relational Data with Applications to Social Network Analysis", Journal of Mathematical Psychology 12, 1975, pp. 329–383.

[3] R.S. Burt, "Positions in Networks", Social Forces 55, 1976, pp. 93–122.

[4] D.R. White and K.P. Reitz, "Graph and Semigroup Homomorphisms on Networks of Relations", Social Networks 5 (1), 1983, pp. 143-234.

[5] V. Batagelj, A. Ferligoj, and P. Doreian, "Direct and Indirect Methods for Structural Equivalence", Social Networks 14, 1992, pp. 63–90.

[6] R.L. Breiger and J.W. Mohr, "Institutional Logics from the Aggregation of Organizational Networks: Operational Procedures for the Analysis of Counted Data", Computational and Mathematical Organization Theory 10, 2004, pp. 17–43.

[7] A. Žiberna, "Generalized Blockmodeling of Valued Networks", Social Networks 29, 2007, pp. 105-126.

[8] M. Brusco, and D. Steinley, "A Variable Neighborhood Search Method for Generalized Blockmodeling of Two-mode Binary Matrices", Journal of Mathematical Psychology 51 (5), 2007, pp. 325-328.

[9] M. Brusco, and D. Steinley, "Integer Programs for One- and Two-mode Blockmodeling Based on Prespecified Image Matrices for Structural and Regular Equivalence", Journal of Mathematical Psychology 53, 2009, pp. 577-585.

[10] S. Pinkert, J. Schultz, and J. Reichardt, "Protein Interaction Networks—More Than Mere Modules", Public Library of Science, Computational Biology, 6 (1), 2010.

[11] Y. Wang and X. Qian, "Joint Clustering of Protein Interaction Networks by Blockmodeling", IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP), Florence, Italy, 2014, pp. 116-1620

[12] A. Davis, B. Gardner, M.R. Gardner, Deep South, University of Chicago Press, Chicago, 1941.

[13] L.C. Freeman, "Finding Social Groups: A Meta-Analysis of the Southern Women Data", Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers, R. Breiger, C. Carley, P. Pattison (Eds.), National Research Council, The National Academies Press, Washington, DC, 2003, pp. 39–97.

[14] P. Doreian, V. Batagelj, and A. Ferligoj, "Generalized Blockmodeling of Two-Mode Network Data", Social Networks 26, 2004, pp. 29-53.

[15] V.D. Blondel, J. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast Unfolding of Communities in Large Networks", Journal of Statistical Mechanics: Theory and Experiment (10), 2008, P10008. arXiv:0803.0476.

[16] IMDb data from 2006 to 2016: A data set of 1,000 popular movies on IMDb in the last 10 years, https://www.kaggle.com/PromptCloudHQ/imdb-data, accessed 2018.

[17] A. Žiberna, "Blockmodeling of Multilevel Networks", Social Networks 39, 2014, pp. 46-61.