

Robert A. Hanneman and Mark Riddle

Introduction to social network methods

Table of contents

About this book

This on-line textbook introduces many of the basics of formal approaches to the analysis of social networks. The text relies heavily on the work of Freeman, Borgatti, and Everett (the authors of the UCINET software package). The materials here, and their organization, were also very strongly influenced by the text of Wasserman and Faust, and by a graduate seminar conducted by Professor Phillip Bonacich at UCLA. Many other users have also made very helpful comments and suggestions based on the first version. Errors and omissions, of course, are the responsibility of the authors.

You are invited to use and redistribute this text freely -- but please acknowledge the source.

Hanneman, Robert A. and Mark Riddle. 2005. Introduction to social network methods. Riverside, CA: University of California, Riverside (published in digital form at <http://faculty.ucr.edu/~hanneman/>)

Table of contents:

[Preface](#)

- [1. Social network data](#)
- [2. Why formal methods?](#)
- [3. Using graphs to represent social relations](#)
- [4. Working with Netdraw to visualize graphs](#)
- [5. Using matrices to represent social relations](#)
- [6. Working with network data](#)
- [7. Connection](#)
- [8. Embedding](#)
- [9. Ego networks](#)
- [10. Centrality and power](#)
- [11. Cliques and sub-groups](#)
- [12. Positions and roles: The idea of equivalence](#)
- [13. Measures of similarity and structural equivalence](#)
- [14. Automorphic equivalence](#)
- [15. Regular equivalence](#)
- [16. Multiplex networks](#)
- [17. Two-mode networks](#)
- [18. Some statistical tools](#)

[After word](#)

[Bibliography](#)

Introduction to social network methods

Preface

This page is part of an on-line text by [Robert A. Hanneman](#) ([Department of Sociology](#), [University of California, Riverside](#)) and Mark Riddle (Department of Sociology, University of Northern Colorado). Feel free to use and distribute this textbook, with citation. Your comments and suggestions are very welcome. [Send me e-mail](#).

This book began as a set of reading notes as Hanneman sought to teach himself the basics of social network analysis. It then became a set of lecture notes for students in his undergraduate course in social network analysis. Through a couple extensions and revisions, it has evolved to cover more of the basic approaches to the analysis of social network data. Its current form, written in 2005, covers most of the algorithms and approaches that are collected in the computer package UCINET, version 6.85. Mark Riddle has added expertise in the statistical modeling of network data, study questions and problems, and connections to a variety of empirical literature that uses the techniques discussed here.

Our goal in preparing this book is to provide a very basic introduction to the core ideas of social network analysis, and how these ideas are implemented in the methodologies that many social network analysts use. The book is distributed free on the Internet in the hope that it may reach a diverse audience, and that the core ideas and methods of this field may be of interest. The book may also be suitable as course-support for undergraduate or introductory graduate training in social network analysis. While this text is not a user's guide to UCINET (which has excellent documentation in its help files), it may be of assistance to users working with that particular software package.

We hope that you will find things here that may stimulate your imagination. Social network analysis is a continuously and rapidly evolving field, and is one branch of the broader study of networks and complex systems. The concepts and techniques of social network analysis are informed by, and inform the evolution of these broader fields. We hope that this text will serve as a starting point.

[table of contents of the book](#)

Introduction to social network methods

1. Social network data

This page is part of an on-line text by [Robert A. Hanneman](#) ([Department of Sociology, University of California, Riverside](#)) and Mark Riddle ([Department of Sociology, University of Northern Colorado](#)). Feel free to use and distribute this textbook, with citation. Your comments and suggestions are very welcome. [Send me e-mail](#).

Contents of chapter 1: Social network data

- [Introduction: What's different about social network data?](#)
- [Nodes](#)
 - [Populations, samples, and boundaries](#)
 - [Modality and levels of analysis](#)
- [Relations](#)
 - [Sampling ties](#)
 - [Multiple relations](#)
- [Scales of measurement](#)
- [A note on statistics and social network data](#)

Introduction: What's different about social network data?

On one hand, there really isn't anything about social network data that is all that unusual. Social network analysts do use a specialized language for describing the structure and contents of the sets of observations that they use. But, network data can also be described and understood using the ideas and concepts of more familiar methods, like cross-sectional survey research.

On the other hand, the data sets that social network analysts develop usually end up looking quite different from the conventional rectangular data array so familiar to survey researchers and statistical analysts. The differences are quite important because they lead us to look at our data in a different way -- and even lead us to think differently about how to apply statistics.

"Conventional" social science data consist of a rectangular array of measurements. The rows of the array are the cases, or subjects, or observations. The columns consist of scores (quantitative or qualitative) on attributes, or variables, or measures. A simple example is shown as figure 1.1. Each cell of the array then describes the score of some actor (row) on some attribute (column). In some cases, there may be a third dimension to these arrays, representing panels of observations or multiple groups.

Figure 1.1. Example of rectangular data array

<i>Name</i>	<i>Sex</i>	<i>Age</i>	<i>In-Degree</i>
Bob	Male	32	2
Carol	Female	27	1
Ted	Male	29	1
Alice	Female	28	3

The fundamental data structure is one that leads us to compare how actors are similar or dissimilar to each other across attributes (by comparing rows). Or, perhaps more commonly, we examine how variables are

similar or dissimilar to each other in their distributions across actors (by comparing or correlating columns).

"Network" data (in their purest form) consist of a square array of measurements. The rows of the array are the cases, or subjects, or observations. The columns of the array are -- and note the key difference from conventional data -- the same set of cases, subjects, or observations. In each cell of the array describes a relationship between the actors. A simple example is shown as figure 1.2, which describes the network of friendship relations among four people.

Figure 1.2. Example of square array of network data

Who reports liking whom?				
	Choice:			
Chooser:	Bob	Carol	Ted	Alice
Bob	---	0	1	1
Carol	1	---	0	1
Ted	0	1	---	1
Alice	1	0	0	---

We could look at this data structure the same way as with attribute data. By comparing rows of the array, we can see which actors are similar to which other actors in whom they choose. By looking at the columns, we can see who is similar to whom in terms of being chosen by others. These are useful ways to look at the data, because they help us to see which actors have similar positions in the network. This is the first major emphasis of network analysis: seeing how actors are located or "embedded" in the overall network.

But a network analyst is also likely to look at the data structure in a second way -- holistically. The analyst might note that there are about equal numbers of ones and zeros in the matrix. This suggests that there is a moderate "density" of liking overall. The analyst might also compare the cells above and below the diagonal to see if there is reciprocity in choices (e.g. Bob chose Ted, did Ted choose Bob?). This is the second major emphasis of network analysis: seeing how the whole pattern of individual choices gives rise to more holistic patterns.

It is quite possible to think of the network data set in the same terms as "conventional data." One can think of the rows as simply a listing of cases, and the columns as attributes of each actor (i.e. the relations with other actors can be thought of as "attributes" of each actor). Indeed, many of the techniques used by network analysts (like calculating correlations and distances) are applied exactly the same way to network data as they would be to conventional data.

While it is possible to describe network data as just a special form of conventional data (and it is), network analysts look at the data in some rather fundamentally different ways. Rather than thinking about how an actor's ties with other actors describes the attributes of "ego," network analysts instead see a structure of connections, within which the actor is embedded. Actors are described by their relations, not by their attributes. And, the relations themselves are just as fundamental as the actors that they connect.

The major difference between conventional and network data is that conventional data focuses on actors and attributes; network data focus on actors and relations. The difference in emphasis is consequential for the choices that a researcher must make in deciding on research design, in conducting sampling, developing measurement, and handling the resulting data. It is not that the research tools used by network analysts are different from those of other social scientists (they mostly are not). But the special purposes and emphases of network research do call for some different considerations.

In this chapter, we will take a look at some of the issues that arise in design, sampling, and measurement for social network analysis. Our discussion will focus on the two parts of network data: nodes (or actors) and edges (or relations). We will try to show some of the ways in which network data are similar to, and different from more familiar actor by attribute data. We will introduce some new terminology that makes it easier to describe the special features of network data. Lastly, we will briefly discuss how the differences between network and actor-attribute data are consequential for the application of statistical tools.

[table of contents](#)

Nodes

Network data are defined by actors and by relations (or "nodes" and "edges"). The nodes or actors part of network data would seem to be pretty straight-forward. Other empirical approaches in the social sciences also think in terms of cases or subjects or sample elements and the like. There is one difference with most network data, however, that makes a big difference in how such data are usually collected -- and the kinds of samples and populations that are studied.

Network analysis focuses on the relations among actors, and not individual actors and their attributes. This means that the actors are usually not sampled independently, as in many other kinds of studies (most typically, surveys). Suppose we are studying friendship ties, for example. John has been selected to be in our sample. When we ask him, John identifies seven friends. We need to track down each of those seven friends and ask them about their friendship ties, as well. The seven friends are in our sample because John is (and vice-versa), so the "sample elements" are no longer "independent."

The nodes or actors included in non-network studies tend to be the result of independent probability sampling. Network studies are much more likely to include all of the actors who occur within some (usually naturally occurring) boundary. Often network studies don't use "samples" at all, at least in the conventional sense. Rather, they tend to include all of the actors in some population or populations. Of course, the populations included in a network study may be a sample of some larger set of populations. For example, when we study patterns of interaction among students in a classrooms, we include all of the children in a classroom (that is, we study the whole population of the classroom). The classroom itself, though, might have been selected by probability methods from a population of classrooms (say all of those in a school).

The use of whole populations as a way of selecting observations in (many) network studies makes it important for the analyst to be clear about the boundaries of each population to be studied, and how individual units of observation are to be selected within that population. Network data sets also frequently involve several levels of analysis, with actors embedded at the lowest level (i.e. network designs can be described using the language of "nested" designs).

[table of contents](#)

Populations, samples, and boundaries

Social network analysts rarely draw samples in their work. Most commonly, network analysts will identify some population and conduct a census (i.e. include all elements of the population as units of observation). A network analyst might examine all of the nouns and objects occurring in a text, all of the persons at a birthday party, all members of a kinship group, of an organization, neighborhood, or social class (e.g. landowners in a region, or royalty).

Survey research methods usually use a quite different approach to deciding which nodes to study. A list is made of all nodes (sometimes stratified or clustered), and individual elements are selected by probability methods. The logic of the method treats each individual as a separate "replication" that is, in a sense,

interchangeable with any other.

Because network methods focus on relations among actors, actors cannot be sampled independently to be included as observations. If one actor happens to be selected, then we must also include all other actors to whom our ego has (or could have) ties. As a result, network approaches tend to study whole populations by means of census, rather than by sample (we will discuss a number of exceptions to this shortly, under the topic of [sampling ties](#)).

The populations that network analysts study are remarkably diverse. At one extreme, they might consist of symbols in texts or sounds in verbalizations; at the other extreme, nations in the world system of states might constitute the population of nodes. Perhaps most common, of course, are populations of individual persons. In each case, however, the elements of the population to be studied are defined by falling within some boundary.

The boundaries of the populations studied by network analysts are of two main types. Probably most commonly, the boundaries are those imposed or created by the actors themselves. All the members of a classroom, organization, club, neighborhood, or community can constitute a population. These are naturally occurring clusters, or networks. So, in a sense, social network studies often draw the boundaries around a population that is known, *a priori*, to be a network. Alternatively, a network analyst might take a more "demographic" or "ecological" approach to defining population boundaries. We might draw observations by contacting all of the people who are found in a bounded spatial area, or who meet some criterion (having gross family incomes over \$1,000,000 per year). Here, we might have reason to suspect that networks exist, but the entity being studied is an abstract aggregation imposed by the investigator -- rather than a pattern of institutionalized social action that has been identified and labeled by its participants.

Network analysts can expand the boundaries of their studies by replicating populations. Rather than studying one neighborhood, we can study several. This type of design (which could use sampling methods to select populations) allows for replication and for testing of hypotheses by comparing populations. A second, and equally important way that network studies expand their scope is by the inclusion of multiple levels of analysis, or modalities.

[table of contents](#)

Modality and levels of analysis

The network analyst tends to see individual people nested within networks of face-to-face relations with other persons. Often these networks of interpersonal relations become "social facts" and take on a life of their own. A family, for example, is a network of close relations among a set of people. But this particular network has been institutionalized and given a name and reality beyond that of its component nodes. Individuals in their work relations may be seen as nested within organizations; in their leisure relations they may be nested in voluntary associations. Neighborhoods, communities, and even societies are, to varying degrees, social entities in and of themselves. And, as social entities, they may form ties with the individuals nested within them, and with other social entities.

Often network data sets describe the nodes and relations among nodes for a single bounded population. If I study the friendship patterns among students in a classroom, I am doing a study of this type. But a classroom exists within a school - which might be thought of as a network relating classes and other actors (principals, administrators, librarians, etc.). And most schools exist within school districts, which can be thought of as networks of schools and other actors (school boards, research wings, purchasing and personnel departments, etc.). There may even be patterns of ties among school districts (say by the exchange of students, teachers, curricular materials, etc.).

Most social network analysts think of individual persons as being embedded in networks that are embedded in networks that are embedded in networks. Network analysts describe such structures as "multi-modal." In our school example, individual students and teachers form one mode, classrooms a second, schools a third, and so on. A data set that contains information about two types of social entities (say persons and organizations) is a two mode network.

Of course, this kind of view of the nature of social structures is not unique to social network analystst. Statistical analysts deal with the same issues as "hierarchical" or "nested" designs. Theorists speak of the macro-meso-micro levels of analysis, or develop schema for identifying levels of analysis (individual, group, organization, community, institution, society, global order being perhaps the most commonly used system in sociology). One advantage of network thinking and method is that it naturally predisposes the analyst to focus on multiple levels of analysis simultaneously. That is, the network analyst is always interested in how the individual is embedded within a structure and how the structure emerges from the micro-relations between individual parts. The ability of network methods to map such multi-modal relations is, at least potentially, a step forward in rigor.

Having claimed that social network methods are particularly well suited for dealing with multiple levels of analysis and multi-modal data structures, it must immediately be admitted that social network analysis rarely actually takes much advantage. Most network analyses does move us beyond simple micro or macro reductionism -- and this is good. Few, if any, data sets and analyses, however, have attempted to work at more than two modes simultaneously. And, even when working with two modes, the most common strategy is to examine them more or less separately (one exception to this is the conjoint analysis of two mode networks). In chapter 17, we'll take a look at some methods for multi-mode networks.

[table of contents](#)

Relations

The other half of the design of network data has to do with what ties or relations are to be measured for the selected nodes. There are two main issues to be discussed here. In many network studies, all of the ties of a given type among all of the selected nodes are studied -- that is, a census is conducted. But, sometimes different approaches are used (because they are less expensive, or because of a need to generalize) that sample ties. There is also a second kind of sampling of ties that always occurs in network data. Any set of actors might be connected by many different kinds of ties and relations (e.g. students in a classroom might like or dislike each other, they might play together or not, they might share food or not, etc.). When we collect network data, we are usually selecting, or sampling, from among a set of kinds of relations that we might have measured.

[table of contents](#)

Sampling ties

Given a set of actors or nodes, there are several strategies for deciding how to go about collecting measurements on the relations among them. At one end of the spectrum of approaches are "full network" methods. This approach yields the maximum of information, but can also be costly and difficult to execute, and may be difficult to generalize. At the other end of the spectrum are methods that look quite like those used in conventional survey research. These approaches yield considerably less information about network structure, but are often less costly, and often allow easier generalization from the observations in the sample to some larger population. There is no one "right" method for all research questions and problems.

Full network methods require that we collect information about each actor's ties with all other actors. In

essence, this approach is taking a census of ties in a population of actors -- rather than a sample. For example we could collect data on shipments of copper between all pairs of nation states in the world system from International Monetary Fund records; we could examine the boards of directors of all public corporations for overlapping directors; we could count the number of vehicles moving between all pairs of cities; we could look at the flows of e-mail between all pairs of employees in a company; we could ask each child in a play group to identify their friends.

Because we collect information about ties between all pairs or dyads, full network data give a complete picture of relations in the population. Most of the special approaches and methods of network analysis that we will discuss in the remainder of this text were developed to be used with full network data. Full network data is necessary to properly define and measure many of the structural concepts of network analysis (e.g. between-ness).

Full network data allows for very powerful descriptions and analyses of social structures. Unfortunately, full network data can also be very expensive and difficult to collect. Obtaining data from every member of a population, and having every member rank or rate every other member can be very challenging tasks in any but the smallest groups. The task is made more manageable by asking respondents to identify a limited number of specific individuals with whom they have ties. These lists can then be compiled and cross-connected. But, for large groups (say all the people in a city), the task is practically impossible.

In many cases, the problems are not quite as severe as one might imagine. Most persons, groups, and organizations tend to have limited numbers of ties -- or at least limited numbers of strong ties. This is probably because social actors have limited resources, energy, time, and cognitive capacity -- and cannot maintain large numbers of strong ties. It is also true that social structures can develop a considerable degree of order and solidarity with relatively few connections.

Snowball methods begin with a focal actor or set of actors. Each of these actors is asked to name some or all of their ties to other actors. Then, all the actors named (who were not part of the original list) are tracked down and asked for some or all of their ties. The process continues until no new actors are identified, or until we decide to stop (usually for reasons of time and resources, or because the new actors being named are very marginal to the group we are trying to study).

The snowball method can be particularly helpful for tracking down "special" populations (often numerically small sub-sets of people mixed in with large numbers of others). Business contact networks, community elites, deviant sub-cultures, avid stamp collectors, kinship networks, and many other structures can be pretty effectively located and described by snowball methods. It is sometimes not as difficult to achieve closure in snowball "samples" as one might think. The limitations on the numbers of strong ties that most actors have, and the tendency for ties to be reciprocated often make it fairly easy to find the boundaries.

There are two major potential limitations and weaknesses of snowball methods. First, actors who are not connected (i.e. "isolates") are not located by this method. The presence and numbers of isolates can be a very important feature of populations for some analytic purposes. The snowball method may tend to overstate the "connectedness" and "solidarity" of populations of actors. Second, there is no guaranteed way of finding all of the connected individuals in the population. Where does one start the snowball rolling? If we start in the wrong place or places, we may miss whole sub-sets of actors who are connected -- but not attached to our starting points.

Snowball approaches can be strengthened by giving some thought to how to select the initial nodes. In many studies, there may be a natural starting point. In community power studies, for example, it is common to begin snowball searches with the chief executives of large economic, cultural, and political organizations. While such an approach will miss most of the community (those who are "isolated" from the elite network), the approach is very likely to capture the elite network quite effectively.

Ego-centric networks (with alter connections)

In many cases it will not be possible (or necessary) to track down the full networks beginning with focal nodes (as in the snowball method). An alternative approach is to begin with a selection of focal nodes (egos), and identify the nodes to which they are connected. Then, we determine which of the nodes identified in the first stage are connected to one another. This can be done by contacting each of the nodes; sometimes we can ask ego to report which of the nodes that it is tied to are tied to one another.

This kind of approach can be quite effective for collecting a form of relational data from very large populations, and can be combined with attribute-based approaches. For example, we might take a simple random sample of male college students and ask them to report who are their close friends, and which of these friends know one another. This kind of approach can give us a good and reliable picture of the kinds of networks (or at least the local neighborhoods) in which individuals are embedded. We can find out such things as how many connections nodes have, and the extent to which these nodes are close-knit groups. Such data can be very useful in helping to understand the opportunities and constraints that ego has as a result of the way they are embedded in their networks.

The ego-centered approach with alter connections can also give us some information about the network as a whole, though not as much as snowball or census approaches. Such data are, in fact, micro-network data sets -- samplings of local areas of larger networks. Many network properties -- distance, centrality, and various kinds of positional equivalence cannot be assessed with ego-centric data. Some properties, such as overall network density can be reasonably estimated with ego-centric data. Some properties -- such as the prevalence of reciprocal ties, cliques, and the like can be estimated rather directly.

Ego-centric networks (ego only)

Ego-centric methods really focus on the individual, rather than on the network as a whole. By collecting information on the connections among the actors connected to each focal ego, we can still get a pretty good picture of the "local" networks or "neighborhoods" of individuals. Such information is useful for understanding how networks affect individuals, and they also give a (incomplete) picture of the general texture of the network as a whole.

Suppose, however, that we only obtained information on ego's connections to alters -- but not information on the connections among those alters. Data like these are not really "network" data at all. That is, they cannot be represented as a square actor-by-actor array of ties. But doesn't mean that ego-centric data without connections among the alters are of no value for analysts seeking to take a structural or network approach to understanding actors. We can know, for example, that some actors have many close friends and kin, and others have few. Knowing this, we are able to understand something about the differences in the actors places in social structure, and make some predictions about how these locations constrain their behavior. What we cannot know from ego-centric data with any certainty is the nature of the macro-structure or the whole network.

In ego-centric networks, the alters identified as connected to each ego are probably a set that is unconnected with those for each other ego. While we cannot assess the overall density or connectedness of the population, we can sometimes be a bit more general. If we have some good theoretical reason to think about alters in terms of their social roles, rather than as individual occupants of social roles, ego-centered networks can tell us a good bit about local social structures. For example, if we identify each of the alters connected to an ego by a friendship relation as "kin," "co-worker," "member of the same church," etc., we can build up a picture of the networks of social positions (rather than the networks of individuals) in which egos are embedded. Such an approach, of course, assumes that such categories as "kin" are real and meaningful determinants of patterns of interaction.

[table of contents](#)

Multiple relations

In a conventional actor-by-trait data set, each actor is described by many variables (and each variable is realized in many actors). In the most common social network data set of actor-by-actor ties, only one kind of relation is described. Just as we often are interested in multiple attributes of actors, we are often interested in multiple kinds of ties that connect actors in a network.

In thinking about the network ties among faculty in an academic department, for example, we might be interested in which faculty have students in common, serve on the same committees, interact as friends outside of the workplace, have one or more areas of expertise in common, and co-author papers. The positions that actors hold in the web of group affiliations are multi-faceted. Positions in one set of relations may re-enforce or contradict positions in another (I might share friendship ties with one set of people with whom I do not work on committees, for example). Actors may be tied together closely in one relational network, but be quite distant from one another in a different relational network. The locations of actors in multi-relational networks and the structure of networks composed of multiple relations are some of the most interesting (and still relatively unexplored) areas of social network analysis.

When we collect social network data about certain kinds of relations among actors we are, in a sense, sampling from a population of possible relations. Usually our research question and theory indicate which of the kinds of relations among actors are the most relevant to our study, and we do not sample -- but rather select -- relations. In a study concerned with economic dependency and growth, for example, I could collect data on the exchange of performances by musicians between nations -- but it is not really likely to be all that relevant.

If we do not know what relations to examine, how might we decide? There are a number of conceptual approaches that might be of assistance. Systems theory, for example, suggests two domains: material and informational. Material things are "conserved" in the sense that they can only be located at one node of the network at a time. Movements of people between organizations, money between people, automobiles between cities, and the like are all examples of material things which move between nodes -- and hence establish a network of material relations. Informational things, to the systems theorist, are "non-conserved" in the sense that they can be in more than one place at the same time. If I know something and share it with you, we both now know it. In a sense, the commonality that is shared by the exchange of information may also be said to establish a tie between two nodes. One needs to be cautious here, however, not to confuse the simple possession of a common attribute (e.g. gender) with the presence of a tie (e.g. the exchange of views between two persons on issues of gender).

Methodologies for working with multi-relational data are not as well developed as those for working with single relations. Many interesting areas of work such as network correlation, multi-dimensional scaling and clustering, and role algebras have been developed to work with multi-relational data. For the most part, these topics are beyond the scope of the current text, and are best approached after the basics of working with single relational networks are mastered. We will look at some methods for multi-relational (a.k.a. "multiplex" network data in chapter 16).

[table of contents](#)

Scales of measurement

Like other kinds of data, the information we collect about ties between actors can be measured (i.e. we can assign scores to our observations) at different "levels of measurement." The different levels of measurement

are important because they limit the kinds of questions that can be examined by the researcher. Scales of measurement are also important because different kinds of scales have different mathematical properties, and call for different algorithms in describing patterns and testing inferences about them.

It is conventional to distinguish nominal, ordinal, and interval levels of measurement (the ratio level can, for all practical purposes, be grouped with interval). It is useful, however, to further divide nominal measurement into binary and multi-category variations; it is also useful to distinguish between full-rank ordinal measures and grouped ordinal measures. We will briefly describe all of these variations, and provide examples of how they are commonly applied in social network studies.

Binary measures of relations: By far the most common approach to scaling (assigning numbers to) relations is to simply distinguish between relations being absent (coded zero), and ties being present (coded one). If we ask respondents in a survey to tell us "which other people on this list do you like?" we are doing binary measurement. Each person from the list that is selected is coded one. Those who are not selected are coded zero.

Much of the development of graph theory in mathematics, and many of the algorithms for measuring properties of actors and networks have been developed for binary data. Binary data is so widely used in network analysis that it is not unusual to see data that are measured at a "higher" level transformed into binary scores before analysis proceeds. To do this, one simply selects some "cut point" and re-scores cases as below the cut-point (zero) or above it (one). Dichotomizing data in this way is throwing away information. The analyst needs to consider what is relevant (i.e. what is the theory about? is it about the presence and pattern of ties, or about the strengths of ties?), and what algorithms are to be applied in deciding whether it is reasonable to recode the data. Very often, the additional power and simplicity of analysis of binary data is "worth" the cost in information lost.

Multiple-category nominal measures of relations: In collecting data we might ask our respondents to look at a list of other people and tell us: "for each person on this list, select the category that describes your relationship with them the best: friend, lover, business relationship, kin, or no relationship." We might score each person on the list as having a relationship of type "1" type "2" etc. This kind of a scale is nominal or qualitative -- each person's relationship to the subject is coded by its type, rather than its strength. Unlike the binary nominal (true-false) data, the multiple category nominal measure is multiple choice.

The most common approach to analyzing multiple-category nominal measures is to use it to create a series of binary measures. That is, we might take the data arising from the question described above and create separate sets of scores for friendship ties, for lover ties, for kin ties, etc. This is very similar to "dummy coding" as a way of handling multiple choice types of measures in statistical analysis. In examining the resulting data, however, one must remember that each node was allowed to have a tie in at most one of the resulting networks. That is, a person can be a friendship tie or a lover tie -- but not both -- as a result of the way we asked the question. In examining the resulting networks, densities may be artificially low, and there will be an inherent negative correlation among the matrices.

This sort of multiple choice data can also be "binarized." That is, we can ignore what kind of tie is reported, and simply code whether a tie exists for a dyad, or not. This may be fine for some analyses -- but it does waste information. One might also wish to regard the types of ties as reflecting some underlying continuous dimension (for example, emotional intensity). The types of ties can then be scaled into a single grouped ordinal measure of tie strength. The scaling, of course, reflects the predispositions of the analyst -- not the reports of the respondents.

Grouped ordinal measures of relations: One of the earliest traditions in the study of social networks asked respondents to rate each of a set of others as "liked" "disliked" or "neutral." The result is a grouped ordinal scale (i.e., there can be more than one "liked" person, and the categories reflect an underlying rank order of

intensity). Usually, this kind of three point scale was coded -1, 0, and +1 to reflect negative liking, indifference, and positive liking. When scored this way, the pluses and minuses make it fairly easy to write algorithms that will count and describe various network properties (e.g. the structural balance of the graph).

Grouped ordinal measures can be used to reflect a number of different quantitative aspects of relations. Network analysts are often concerned with describing the "strength" of ties. But, "strength" may mean (some or all of) a variety of things. One dimension is the frequency of interaction -- do actors have contact daily, weekly, monthly, etc. Another dimension is "intensity," which usually reflects the degree of emotional arousal associated with the relationship (e.g. kin ties may be infrequent, but carry a high "emotional charge" because of the highly ritualized and institutionalized expectations). Ties may be said to be stronger if they involve many different contexts or types of ties. Summing nominal data about the presence or absence of multiple types of ties gives rise to an ordinal (actually, interval) scale of one dimension of tie strength. Ties are also said to be stronger to the extent that they are reciprocated. Normally we would assess reciprocity by asking each actor in a dyad to report their feelings about the other. However, one might also ask each actor for their perceptions of the degree of reciprocity in a relation: Would you say that neither of you like each other very much, that you like X more than X likes you, that X likes you more than you like X, or that you both like each other about equally?

Ordinal scales of measurement contain more information than nominal. That is, the scores reflect finer gradations of tie strength than the simple binary "presence or absence." This would seem to be a good thing, yet it is frequently difficult to take advantage of ordinal data. The most commonly used algorithms for the analysis of social networks have been designed for binary data. Many have been adapted to continuous data -- but for interval, rather than ordinal scales of measurement. Ordinal data, consequently, are often binarized by choosing some cut-point and re-scoring. Alternatively, ordinal data are sometimes treated as though they really were interval. The former strategy has some risks, in that choices of cut-points can be consequential; the latter strategy has some risks, in that the intervals separating points on an ordinal scale may be very heterogeneous.

Full-rank ordinal measures of relations: Sometimes it is possible to score the strength of all of the relations of an actor in a rank order from strongest to weakest. For example, I could ask each respondent to write a "1" next to the name of the person in the class that you like the most, a "2" next to the name of the person you like next most, etc. The kind of scale that would result from this would be a "full rank order scale." Such scales reflect differences in degree of intensity, but not necessarily equal differences -- that is, the difference between my first and second choices is not necessarily the same as the difference between my second and third choices. Each relation, however, has a unique score (1st, 2nd, 3rd, etc.).

Full rank ordinal measures are somewhat uncommon in the social networks research literature, as they are in most other traditions. Consequently, there are relatively few methods, definitions, and algorithms that take specific and full advantage of the information in such scales. Most commonly, full rank ordinal measures are treated as if they were interval. There is probably somewhat less risk in treating fully rank ordered measures (compared to grouped ordinal measures) as though they were interval, though the assumption is still a risky one. Of course, it is also possible to group the rank order scores into groups (i.e. produce a grouped ordinal scale) or dichotomize the data (e.g. the top three choices might be treated as ties, the remainder as non-ties). In combining information on multiple types of ties, it is frequently necessary to simplify full rank order scales. But, if we have a number of full rank order scales that we may wish to combine to form a scale (i.e. rankings of people's likings of other in the group, frequency of interaction, etc.), the sum of such scales into an index is plausibly treated as a truly interval measure.

Interval measures of relations: The most "advanced" level of measurement allows us to discriminate among the relations reported in ways that allow us to validly state that, for example, "this tie is twice as strong as that tie." Ties are rated on scales in which the difference between a "1" and a "2" reflects the same

amount of real difference as that between "23" and "24."

True interval level measures of the strength of many kinds of relationships are fairly easy to construct, with a little imagination and persistence. Asking respondents to report the details of the frequency or intensity of ties by survey or interview methods, however, can be rather unreliable -- particularly if the relationships being tracked are not highly salient and infrequent. Rather than asking whether two people communicate, one could count the number of email, phone, and inter-office mail deliveries between them. Rather than asking whether two nations trade with one another, look at statistics on balances of payments. In many cases, it is possible to construct interval level measures of relationship strength by using artifacts (e.g. statistics collected for other purposes) or observation.

Continuous measures of the strengths of relationships allow the application of a wider range of mathematical and statistical tools to the exploration and analysis of the data. Many of the algorithms that have been developed by social network analysts, originally for binary data, have been extended to take advantage of the information available in full interval measures. Whenever possible, connections should be measured at the interval level -- as we can always move to a less refined approach later; if data are collected at the nominal level, it is much more difficult to move to a more refined level.

Even though it is a good idea to measure relationship intensity at the most refined level possible, most network analysis does not operate at this level. The most powerful insights of network analysis, and many of the mathematical and graphical tools used by network analysts were developed for simple graphs (i.e. binary, undirected). Many characterizations of the embeddedness of actors in their networks, and of the networks themselves are most commonly thought of in discrete terms in the research literature. As a result, it is often desirable to reduce even interval data to the binary level by choosing a cutting -point, and coding tie strength above that point as "1" and below that point as "0." Unfortunately, there is no single "correct" way to choose a cut-point. Theory and the purposes of the analysis provide the best guidance. Sometimes examining the data can help (maybe the distribution of tie strengths really is discretely bi-modal, and displays a clear cut point; maybe the distribution is highly skewed and the main feature is a distinction between no tie and any tie). When a cut-point is chosen, it is wise to also consider alternative values that are somewhat higher and lower, and repeat the analyses with different cut-points to see if the substance of the results is affected. This can be very tedious, but it is very necessary. Otherwise, one may be fooled into thinking that a real pattern has been found, when we have only observed the consequences of where we decided to put our cut-point.

[table of contents](#)

A note on statistics and social network data

Social network analysis is more a branch of "mathematical" sociology than of "statistical or quantitative analysis," though social network analysts most certainly practice both approaches. The distinction between the two approaches is not clear-cut. Mathematical approaches to network analysis tend to treat the data as "deterministic." That is, they tend to regard the measured relationships and relationship strengths as accurately reflecting the "real" or "final" or "equilibrium" status of the network. Mathematical types also tend to assume that the observations are not a "sample" of some larger population of possible observations; rather, the observations are usually regarded as the population of interest. Statistical analysts tend to regard the particular scores on relationship strengths as stochastic or probabilistic realizations of an underlying true tendency or probability distribution of relationship strengths. Statistical analysts also tend to think of a particular set of network data as a "sample" of a larger class or population of such networks or network elements -- and have a concern for the results of the current study would be reproduced in the "next" study of similar samples.

In the chapters that follow in this text, we will mostly be concerned with the "mathematical" rather than the "statistical" side of network analysis (again, it is important to remember that I am over-drawing the

differences in this discussion). Before passing on to this, we should note a couple main points about the relationship between the material that you will be studying here, and the main statistical approaches in sociology. In chapter 18, we will explore some of the basic ways in which statistical tools have been adapted to study social network data.

In one way, there is little apparent difference between conventional statistical approaches and network approaches. Univariate, bi-variate, and even many multivariate descriptive statistical tools are commonly used in the describing, exploring, and modeling social network data. Social network data are, as we have pointed out, easily represented as arrays of numbers -- just like other types of sociological data. As a result, the same kinds of operations can be performed on network data as on other types of data. Algorithms from statistics are commonly used to describe characteristics of individual observations (e.g. the median tie strength of actor X with all other actors in the network) and the network as a whole (e.g. the mean of all tie strengths among all actors in the network). Statistical algorithms are very heavily used in assessing the degree of similarity among actors, and in finding patterns in network data (e.g. factor analysis, cluster analysis, multi-dimensional scaling). Even the tools of predictive modeling are commonly applied to network data (e.g. correlation and regression).

Descriptive statistical tools are really just algorithms for summarizing characteristics of the distributions of scores. That is, they are mathematical operations. Where statistics really become "statistical" is on the inferential side. That is, when our attention turns to assessing the reproducibility or likelihood of the pattern that we have described. Inferential statistics can be, and are, applied to the analysis of network data. But, there are some quite important differences between the flavors of inferential statistics used with network data, and those that are most commonly taught in basic courses in statistical analysis in sociology.

Probably the most common emphasis in the application of inferential statistics to social science data is to answer questions about the stability, reproducibility, or generalizability of results observed in a single sample. The main question is: if I repeated the study on a different sample (drawn by the same method), how likely is it that I would get the same answer about what is going on in the whole population from which I drew both samples? This is a really important question -- because it helps us to assess the confidence (or lack of it) that we ought to have in assessing our theories and giving advice.

To the extent the observations used in a network analysis are drawn by probability sampling methods from some identifiable population of actors and/or ties, the same kind of question about the generalizability of sample results applies. Often this type of inferential question is of little interest to social network researchers. In many cases, they are studying a particular network or set of networks, and have no interest in generalizing to a larger population of such networks (either because there isn't any such population, or we don't care about generalizing to it in any probabilistic way). In some other cases we may have an interest in generalizing, but our sample was not drawn by probability methods. Network analysis often relies on artifacts, direct observation, laboratory experiments, and documents as data sources -- and usually there are no plausible ways of identifying populations and drawing samples by probability methods.

The other major use of inferential statistics in the social sciences is for testing hypotheses. In many cases, the same or closely related tools are used for questions of assessing generalizability and for hypothesis testing. The basic logic of hypothesis testing is to compare an observed result in a sample to some null hypothesis value, relative to the sampling variability of the result under the assumption that the null hypothesis is true. If the sample result differs greatly from what was likely to have been observed under the assumption that the null hypothesis is true -- then the null hypothesis is probably not true.

The key link in the inferential chain of hypothesis testing is the estimation of the standard errors of statistics. That is, estimating the expected amount that the value of a statistic would "jump around" from one sample to the next simply as a result of accidents of sampling. We rarely, of course, can directly observe or calculate such standard errors -- because we don't have replications. Instead, information from our sample is used to

estimate the sampling variability.

With many common statistical procedures, it is possible to estimate standard errors by well validated approximations (e.g. the standard error of a mean is usually estimated by the sample standard deviation divided by the square root of the sample size). These approximations, however, hold when the observations are drawn by independent random sampling. Network observations are almost always non-independent, by definition. Consequently, conventional inferential formulas do not apply to network data (though formulas developed for other types of dependent sampling may apply). It is particularly dangerous to assume that such formulas do apply, because the non-independence of network observations will usually result in under-estimates of true sampling variability -- and hence, too much confidence in our results.

The approach of most network analysts interested in statistical inference for testing hypotheses about network properties is to work out the probability distributions for statistics directly. This approach is used because: 1) no one has developed approximations for the sampling distributions of most of the descriptive statistics used by network analysts and 2) interest often focuses on the probability of a parameter relative to some theoretical baseline (usually randomness) rather than on the probability that a given network is typical of the population of all networks.

Suppose, for example, that I was interested in the proportion of the actors in a network who were members of cliques (or any other network statistic or parameter). The notion of a clique implies structure -- non-random connections among actors. I have data on a network of ten nodes, in which there are 20 symmetric ties among actors, and I observe that there is one clique containing four actors. The inferential question might be posed as: how likely is it, if ties among actors were purely random events, that a network composed of ten nodes and 20 symmetric ties would display one or more cliques of size four or more? If it turns out that cliques of size four or more in random networks of this size and degree are quite common, I should be very cautious in concluding that I have discovered "structure" or non-randomness. If it turns out that such cliques (or more numerous or more inclusive ones) are very unlikely under the assumption that ties are purely random, then it is very plausible to reach the conclusion that there is a social structure present.

But how can I determine this probability? The method used is one of simulation -- and, like most simulation, a lot of computer resources and some programming skills are often necessary. In the current case, I might use a table of random numbers to distribute 20 ties among 10 actors, and then search the resulting network for cliques of size four or more. If no clique is found, I record a zero for the trial; if a clique is found, I record a one. The rest is simple. Just repeat the experiment several thousand times and add up what proportion of the "trials" result in "successes." The probability of a success across these simulation experiments is a good estimator of the likelihood that I might find a network of this size and density to have a clique of this size "just by accident" when the non-random causal mechanisms that I think cause cliques are not, in fact, operating.

This may sound odd, and it is certainly a lot of work (most of which, thankfully, can be done by computers). But, in fact, it is not really different from the logic of testing hypotheses with non-network data. Social network data tend to differ from more "conventional" survey data in some key ways: network data are often not probability samples, and the observations of individual nodes are not independent. These differences are quite consequential for both the questions of generalization of findings, and for the mechanics of hypothesis testing. There is, however, nothing fundamentally different about the logic of the use of descriptive and inferential statistics with social network data.

The application of statistics to social network data is an interesting area, and one that is, at the time of this writing, at a "cutting edge" of research in the area. Since this text focuses on more basic and commonplace uses of network analysis, we won't have very much more to say about statistics beyond this point. You can think of much of what follows here as dealing with the "descriptive" side of statistics (developing index numbers to describe certain aspects of the distribution of relational ties among actors in networks). For those with an interest in the inferential side, a good place to start is with the second half of the excellent

Wasserman and Faust textbook.

[table of contents](#)

[table of contents of the book](#)

Introduction to Social Network Methods

2. Why formal methods?

This page is part of an on-line text by [Robert A. Hanneman](#) ([Department of Sociology, University of California, Riverside](#)) and Mark Riddle (Department of Sociology, University of Northern Colorado). Feel free to use and distribute this textbook, with citation. Your comments and suggestions are very welcome. [Send me e-mail.](#)

Contents of chapter 2: Why formal methods?

- [Introduction](#)
 - [Efficiency](#)
 - [Using computers](#)
 - [Seeing patterns](#)
 - [Summary](#)
-

Introduction:

The basic idea of a social network is very simple. A social network is a set of actors (or points, or nodes, or agents) that may have relationships (or edges, or ties) with one another. Networks can have few or many actors, and one or more kinds of relations between pairs of actors. To build a useful understanding of a social network, a complete and rigorous description of a pattern of social relationships is a necessary starting point for analysis. That is, ideally we will know about all of the relationships between each pair of actors in the population.

The amount of information that we need to describe even small social networks can be quite great. Managing these data, and manipulating them so that we can see patterns of social structure can be tedious and complicated. All of the tasks of social network methods are made easier by using tools from mathematics. For the manipulation of network data, and the calculation of indexes describing networks, it is most useful to record information as matrices. For visualizing patterns, graphs are often useful.

Efficiency

One reason for using mathematical and graphical techniques in social network analysis is to represent the descriptions of networks compactly and systematically. This also enables us to use computers to store and manipulate the information quickly and more accurately than we can by hand. For small populations of actors (e.g. the people in a neighborhood, or the business firms in an industry), we can describe the pattern of social relationships that connect the actors rather completely and effectively using words. To make sure that our description is complete, however, we might want to list all logically possible pairs of actors, and describe each kind of possible relationship for each pair. This can get pretty tedious if the number of actors and/or number of kinds of relations is large. Formal representations ensure that all the necessary information is systematically represented, and provides rules for doing so in ways that are much more efficient than lists.

Using computers

A related reason for using (particularly mathematical) formal methods for representing social networks is that mathematical representations allow us to apply computers to the analysis of network data. Why this is important will become clearer as we learn more about how structural analysis of social networks occurs.

Suppose, for a simple example, we had information about trade-flows of 50 different commodities (e.g. coffee, sugar, tea, copper, bauxite) among the 170 or so nations of the world system in a given year. Here, the 170 nations can be thought of as actors or nodes, and the amount of each commodity exported from each nation to each of the other 169 can be thought of as the strength of a directed tie from the focal nation to the other. A social scientist might be interested in whether the "structures" of trade in mineral products are more similar to one another than, the structure of trade in mineral products are to vegetable products. To answer this fairly simple (but also pretty important) question, a huge amount of manipulation of the data is necessary. It could take, literally, years to do by hand; it can be done by a computer in a few minutes.

Seeing patterns

The third, and final reason for using "formal" methods (mathematics and graphs) for representing social network data is that the techniques of graphing and the rules of mathematics themselves suggest things that we might look for in our data — things that might not have occurred to us if we presented our data using descriptions in words. Again, allow me a simple example.

Suppose we were describing the structure of close friendship in a group of four people: Bob, Carol, Ted, and Alice. This is easy enough to do with words. Suppose that Bob likes Carol and Ted, but not Alice; Carol likes Ted, but neither Bob nor Alice; Ted likes all three of the other members of the group; and Alice likes only Ted (this description should probably strike you as being a description of a very unusual social structure).

We could also describe this pattern of liking ties with an actor-by-actor matrix where the rows represent choices by each actor. We will put in a "1" if an actor likes another, and a "0" if they don't. Such a matrix would look like figure 2.1.

Figure 2.1. Matrix representation of "liking" relation among four actors

	Bob	Carol	Ted	Alice
Bob	---	1	1	0
Carol	0	---	1	0
Ted	1	1	---	1
Alice	0	0	1	---

There are lots of things that might immediately occur to us when we see our data arrayed in this way, that we might not have thought of from reading the description of the pattern of ties in words. For example, our eye is led to scan across each row; we notice that Ted likes more people than Bob, than Alice and Carol. Is it possible that there is a pattern here? Are men are more likely to report ties of liking than women are (actually, research literature suggests that this is not generally true). Using a "matrix representation" also immediately raises a question: the locations on the main diagonal (e.g. Bob likes Bob, Carol likes Carol) are empty. Is this a reasonable thing? Or, should our description of the pattern of liking in the group include some statements about "self-liking"? There isn't any right answer to this question. My point is just that using a matrix to represent the pattern of ties among actors may let us see some patterns more easily, and may cause us to ask some questions (and maybe even some useful ones) that a verbal description doesn't stimulate.

Summary

There are three main reasons for using "formal" methods in representing social network data:

- Matrices and graphs are compact and systematic: They summarize and present a lot of information quickly and easily; and they force us to be systematic and complete in describing patterns of social relations.
- Matrices and graphs allow us to apply computers to analyzing data: This is helpful because doing systematic analysis of social network data can be extremely tedious if the number of actors or number of types of relationships among the actors is large. Most of the work is dull, repetitive, and uninteresting, but requires accuracy; exactly the sort of thing that computers do well, and we don't.
- Matrices and graphs have rules and conventions: Sometimes these are just rules and conventions that help us communicate clearly. But sometimes the rules and conventions of the language of graphs and mathematics themselves lead us to see things in our data that might not have occurred to us to look for if we had described our data only with words.

So, we need to learn the basics of representing social network data using matrices and graphs. The next several chapters (3, 4, 5, and 6) introduce these basic tools.

[table of contents](#)

[table of context of the book](#)

Introduction to social network methods

3. Using graphs to represent social relations

This page is part of an on-line text by [Robert A. Hanneman](#) ([Department of Sociology, University of California, Riverside](#)) and Mark Riddle (Department of Sociology, University of Northern Colorado). Feel free to use and distribute this textbook, with citation. Your comments and suggestions are very welcome. [Send me e-mail.](#)

Contents of chapter 3: Using graphs to represent social relations

- [Introduction: Representing networks with graphs](#)
 - [Graphs and sociograms](#)
 - [Kinds of graphs:](#)
 - [Levels of measurement:](#) Binary, signed, and valued graphs
 - [Directed or "bonded" Ties](#) in the graph
 - [Simplex or multiplex relations](#) in the graph
 - [Summary](#)
 - [Study questions](#)
-

Introduction: Representing networks with graphs

Social network analysts use two kinds of tools from mathematics to represent information about patterns of ties among social actors: graphs and matrices. On this page, we will learn enough about graphs to understand how to represent social network data. On the next page, we will look at matrix representations of social relations. With these tools in hand, we can understand most of the things that network analysts do with such data (for example, calculate precise measures of "relative density of ties").

There is a lot more to these topics than we will cover here; mathematics has whole sub-fields devoted to "graph theory" and to "matrix algebra." Social scientists have borrowed just a few things that they find helpful for describing and analyzing patterns of social relations.

A word of warning: there is a lot of specialized terminology here that you do need to learn. its worth the effort, because we can represent some important ideas about social structure in quite simple ways, once the basics have been mastered.

[table of contents](#)

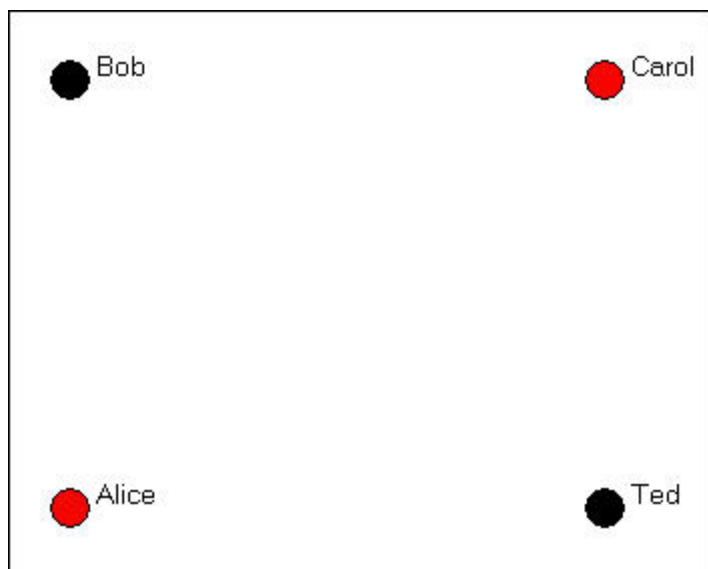
Graphs and Sociograms

There are lots of different kinds of "graphs." Bar-charts, pie-charts, line and trend charts, and many other things are called graphs and/or graphics. Network analysis uses (primarily) one kind of graphic display that consists of points (or nodes) to represent actors and lines (or edges) to represent ties or relations. When sociologists borrowed this way of graphing things from the mathematicians, they re-named their graphics "socio-grams." Mathematicians know the kind of graphic displays by the names of "directed graphs" "signed graphs" or simply "graphs."

There are a number of variations on the theme of socio-grams, but they all share the common feature of using a labeled circle for each actor in the population we are describing, and line segments between pairs of actors to represent the observation that a tie exists between the two. Let's suppose that we are interested in

summarizing who nominates whom as being a "friend" in a group of four people (Bob, Carol, Ted, and Alice). We would begin by representing each actor as a "node" with a label (sometimes nodes are represented by labels in circles or boxes). Figure 3.1 shows a graph with four labeled nodes, but no connections.

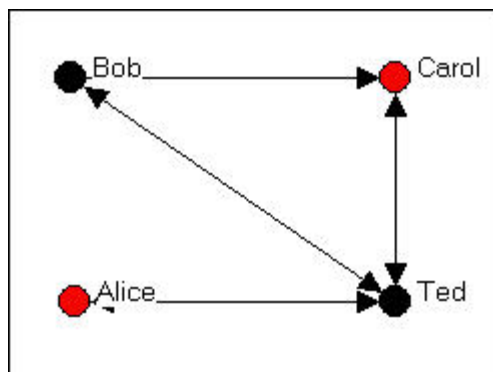
Figure 3.1. Nodes for a simple graph



In this example, we've also indicated an "attribute" of each actor by coloring the node (black for males, red for females). Coloring, shading, or different shapes and sizes are often used to represent attributes of the individual nodes.

We collected our data about friendship ties by asking each member of the group (privately and confidentially) who they regarded as "close friends" from a list containing each of the other members of the group. Each of the four people could choose none to all three of the others as "close friends." As it turned out, in our (fictitious) case, Bob chose Carol and Ted, but not Alice; Carol chose only Ted; Ted chose Bob and Carol and Alice; and Alice chose only Ted. We would represent this information by drawing an arrow from the chooser to each of the chosen, as in figure 3.2.

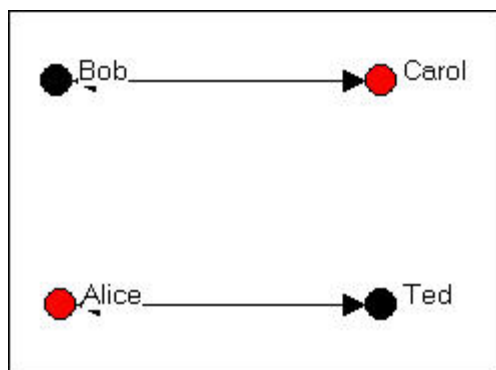
Figure 3.2. A directed graph of friendship ties



To reduce visual clutter, a double-headed arrow has been used when the relationship between two nodes is "reciprocated" (i.e. each actor chooses the other).

Let's suppose that we had also taken note of a second kind of relation - whether persons share the relationship "spouse" with one another. In our example, Bob and Carol are spouses, as are Ted and Alice. We can also represent this kind of a "bonded tie" with a directed graph as in figure 3.3.

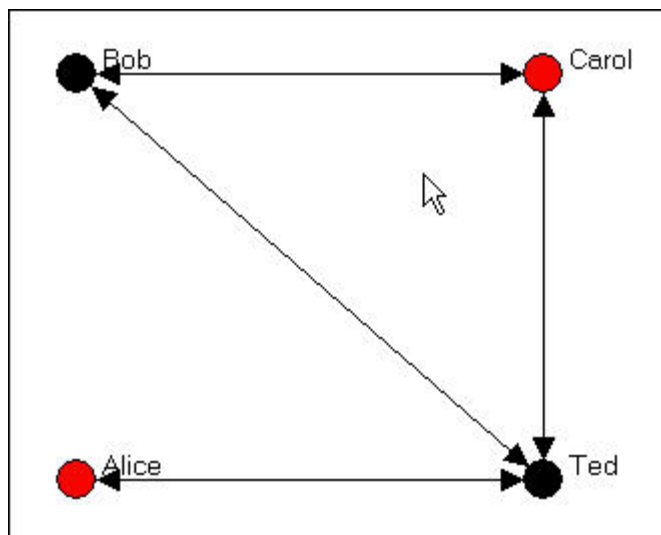
Figure 3.3. A directed graph of spousal ties



Where a tie is necessarily reciprocated (see the discussion of "bonded ties, below), a "simple" graph is often used instead of a "directed" graph. In a simple graph, relations are simply present or absent, and the relations are indicated by lines without arrow heads.

We can also represent multiple relations (multiplex relations) using graphs -- though with larger numbers of actors or many relations, the results may not be very easy to read. Let's combine the graphs of both "friendship" and "spousal" relations, as in figure 3.4.

Figure 3.4. A directed graph of multiplex relations (friendship and spouse)



In this figure, a tie is shown between two nodes whenever there is either a friendship tie, or a spousal tie, or both. This helps us to see that Bob, Carol, and Ted form a "clique" (i.e. each is connected to each of the others), and Alice is a "pendant" (tied to the group by only one connection).

This particular way for drawing the multiplex relation, however, loses some information about which ties connect which actors. As an alternative, one might want to superimpose the two single-relation graphs, and show multiple lines (or different color lines, or some dashed lines) to show the different kinds of connections.

[table of contents](#)

Kinds of Graphs

Now we need to introduce some terminology to describe different kinds of graphs. Figure 3.2 is an example of a *binary* (as opposed to a signed or ordinal or valued) and *directed* (as opposed to a co-occurrence or co-

presence or bonded-tie) graph. Figure 3.3 is an example of a "co-occurrence" or "co-presence" or "bonded-tie" graph that is *binary* and *undirected* (or simple). The social relations being described here are also *simplex* (in figures 3.2 and 3.3). Figure 3.4 is an example of one method of representing *multiplex* relational data with a single graph.

Let's take a moment to review some of this terminology in a little more detail.

[table of contents](#)

Levels of Measurement: Binary, Signed, and Valued Graphs

In describing the pattern of who describes whom as a close friend, we could have asked our question in several different ways. If we asked each respondent "is this person a close friend or not," we are asking for a binary choice: each person is or is not chosen by each interviewee. Many social relationships can be described this way: the only thing that matters is whether a tie exists or not. When our data are collected this way, we can graph them simply: an arrow represents a choice that was made, no arrow represents the absence of a choice. But, we could have asked the question a second way: "for each person on this list, indicate whether you like, dislike, or don't care." We might assign a + to indicate "liking," zero to indicate "don't care" and - to indicate dislike. This kind of data is called "signed" data. The graph with signed data uses a + on the arrow to indicate a positive choice, a - to indicate a negative choice, and no arrow to indicate neutral or indifferent. Yet another approach would have been to ask: "rank the three people on this list in order of who you like most, next most, and least." This would give us "rank order" or "ordinal" data describing the strength of each friendship choice. Lastly, we could have asked: "on a scale from minus one hundred to plus one hundred - where minus 100 means you hate this person, zero means you feel neutral, and plus 100 means you love this person - how do you feel about...". This would give us information about the value of the strength of each choice on a (supposedly, at least) ratio level of measurement. With either an ordinal or valued graph, we would put the measure of the strength of the relationship on the arrow in the diagram.

[table of contents](#)

Directed or "bonded" ties in the graph

In our example, we asked each member of the group to choose which others in the group they regarded as close friends. Each person (ego) then is being asked about ties or relations that they themselves direct toward others (alters). Each alter does not necessarily feel the same way about each tie as ego does: Bob may regard himself as a good friend to Alice, but Alice does not necessarily regard Bob as a good friend. It is very useful to describe many social structures as being composed of "directed" ties (which can be binary, signed, ordered, or valued). Indeed, most social processes involve sequences of directed actions. For example, suppose that person A directs a comment to B, then B directs a comment back to A, and so on. We may not know the order in which actions occurred (i.e. who started the conversation), or we may not care. In this example, we might just want to know that "A and B are having a conversation." In this case, the tie or relation "in conversation with" necessarily involves both actors A and B. Both A and B are "co-present" or "co-occurring" in the relation of "having a conversation." Or, we might also describe the situation as being one of an the social institution of a "conversation" that by definition involves two (or more) actors "bonded" in an interaction (Berkowitz).

"Directed" graphs use the convention of connecting nodes or actors with arrows that have arrow heads, indicating who is directing the tie toward whom. This is what we used in the graphs above, where individuals (egos) were directing choices toward others (alters). "Simple" or "Co-occurrence" or "co-presence" or "bonded-tie" graphs use the convention of connecting the pair of actors involved in the relation with a simple line segment (no arrow head). Be careful here, though. In a directed graph, Bob could choose Ted, and Ted

choose Bob. This would be represented by headed arrows going from Bob to Ted, and from Ted to Bob, or by a double-headed arrow. But, this represents a different meaning from a graph that shows Bob and Ted connected by a single line segment without arrow heads. Such a graph would say "there is a relationship called close friend which ties Bob and Ted together." The distinction can be subtle, but it is important in some analyses.

[table of contents](#)

Simplex or multiplex relations in the graph

Social actors are often connected by more than one kind of relationship. In our simple example, we showed two graphs of simple (sometimes referred to as "simplex" to differentiate from "multiplex") relations. The friendship graph (figure 3.2) showed a single relation (that happened to be binary and directed). The spouse graph (figure 3.3) showed a single relation (that happened to be binary and un-directed). Figure 3.4 combines information from two relations into a "multiplex" graph.

There are, potentially, different kinds of multiplex graphs. We graphed a tie if there was either a friendship or spousal relation. But, we could have graphed a tie only if there were both a friendship and spousal tie (what would such a graph look like?).

We also combined the information about multiple ties into a single line. Alternatively, one might use different symbols, colors, line widths, or other devices to keep all of the information about multiple relations visible in a multiplex graph -- but the result can often be too complicated to be useful.

[table of contents](#)

Summary

A graph (sometimes called a sociogram) is composed of nodes (or actors or points) connected by edges (or relations or ties). A graph may represent a single type of relations among the actors (simplex), or more than one kind of relation (multiplex). Each tie or relation may be directed (i.e. originates with a source actor and reaches a target actor), or it may be a tie that represents co-occurrence, co-presence, or a bonded-tie between the pair of actors. Directed ties are represented with arrows, bonded-tie relations are represented with line segments. Directed ties may be reciprocated (A chooses B and B chooses A); such ties can be represented with a double-headed arrow. The strength of ties among actors in a graph may be nominal or binary (represents presence or absence of a tie); signed (represents a negative tie, a positive tie, or no tie); ordinal (represents whether the tie is the strongest, next strongest, etc.); or valued (measured on an interval or ratio level). In speaking the position of one actor or node in a graph to other actors or nodes in a graph, we may refer to the focal actor as "ego" and the other actors as "alters."

[table of contents](#)

Review questions

1. What are "nodes" and "edges"? In a sociogram, what is used for nodes? for edges?
2. How do valued, binary, and signed graphs correspond to the "nominal" "ordinal" and "interval" levels of measurement?
3. Distinguish between directed relations or ties and "bonded" relations or ties.

4. How does a reciprocated directed relation differ from a "bonded" relation?
5. Give an example of a multi-plex relation. How can multi-plex relations be represented in graphs?

Application questions

1. Think of the readings from the first part of the course. Did any studies present graphs? If they did, what kinds of graphs were they (that is, what is the technical description of the kind of graph or matrix). Pick one article and show what a graph of its data would look like.
2. Suppose that I was interested in drawing a graph of which large corporations were networked with one another by having the same persons on their boards of directors. Would it make more sense to use "directed" ties, or "bonded" ties for my graph? Can you think of a kind of relation among large corporations that would be better represented with directed ties?
3. Think of some small group of which you are a member (maybe a club, or a set of friends, or people living in the same apartment complex, etc.). What kinds of relations among them might tell us something about the social structures in this population? Try drawing a graph to represent one of the kinds of relations you chose. Can you extend this graph to also describe a second kind of relation? (e.g. one might start with "who likes whom?" and add "who spends a lot of time with whom?").
4. Make graphs of a "star" network, a "line" and a "circle." Think of real world examples of these kinds of structures where the ties are directed and where they are bonded, or undirected. What does a strict hierarchy look like? What does a population that is segregated into two groups look like?

[table of contents](#)

[table of contents of the book](#)

Introduction to social network methods

4. Working with NetDraw to visualize graphs

This page is part of an on-line text by [Robert A. Hanneman](#) ([Department of Sociology, University of California, Riverside](#)) and Mark Riddle (Department of Sociology, University of Northern Colorado). Feel free to use and distribute this textbook, with citation. Your comments and suggestions are very welcome. [Send me e-mail.](#)

Contents of this chapter:

- [Introduction: A picture is worth...](#)
 - [Node attributes](#)
 - [Relation properties](#)
 - [Location, location, location](#)
 - [Highlighting parts of the network](#)
 - [A few hints on data handling with NetDraw](#)
-

Introduction: A picture is worth...

As we saw in chapter 3, a graph representing the information about the relations among nodes can be an very efficient way of describing a social structure. A good drawing of a graph can immediately suggest some of the most important features of overall network structure. Are all the nodes connected? Are there many or few ties among the actors? Are there sub-groups or local "clusters" of actors that are tied to one another, but not to other groups? Are there some actors with many ties, and some with few?

A good drawing can also help us to better understand how a particular "ego" (node) is "embedded" (connected to) its "neighborhood" (the actors that are connected to ego, and their connections to one another) and to the larger graph (is "ego" an "isolate" a "pendant"?). By looking at "ego" and the "ego network" (i.e. "neighborhood"), we can get a sense of the structural constraints and opportunities that an actor faces; we may be better able to understand the role that an actor plays in a social structure.

There is no single "right way" to represent network data with graphs. There are a few basic rules, and we reviewed these in the previous chapter. Different ways of drawing pictures of network data can emphasize (or obscure) different features of the social structure. It's usually a good idea to play with visualizing a network, to experiment and be creative. There are a number of software tools that are available for drawing graphs, and each has certain strengths and limitations. In this chapter, we will look at some commonly used techniques for visualizing graphs using NetDraw (version 4.14, which is distributed along with UCINET). There are many other packages though, and you might want to explore some of the tools available in Pajek, and Mage (look for software at the web-site of the International Network of Social Network Analysts - INSNA).

Of course, if there are a large number of actors or a large number of relations among them, pictures may not help the situation much; numerical indexes describing the graph may be the only choice. Numerical approaches and graphical approaches can be used in combination, though. For example, we might first calculate the "between-ness centrality" of the nodes in a large network, and then use graphs that include only those actors that have been identified as "important."

[table of contents](#)

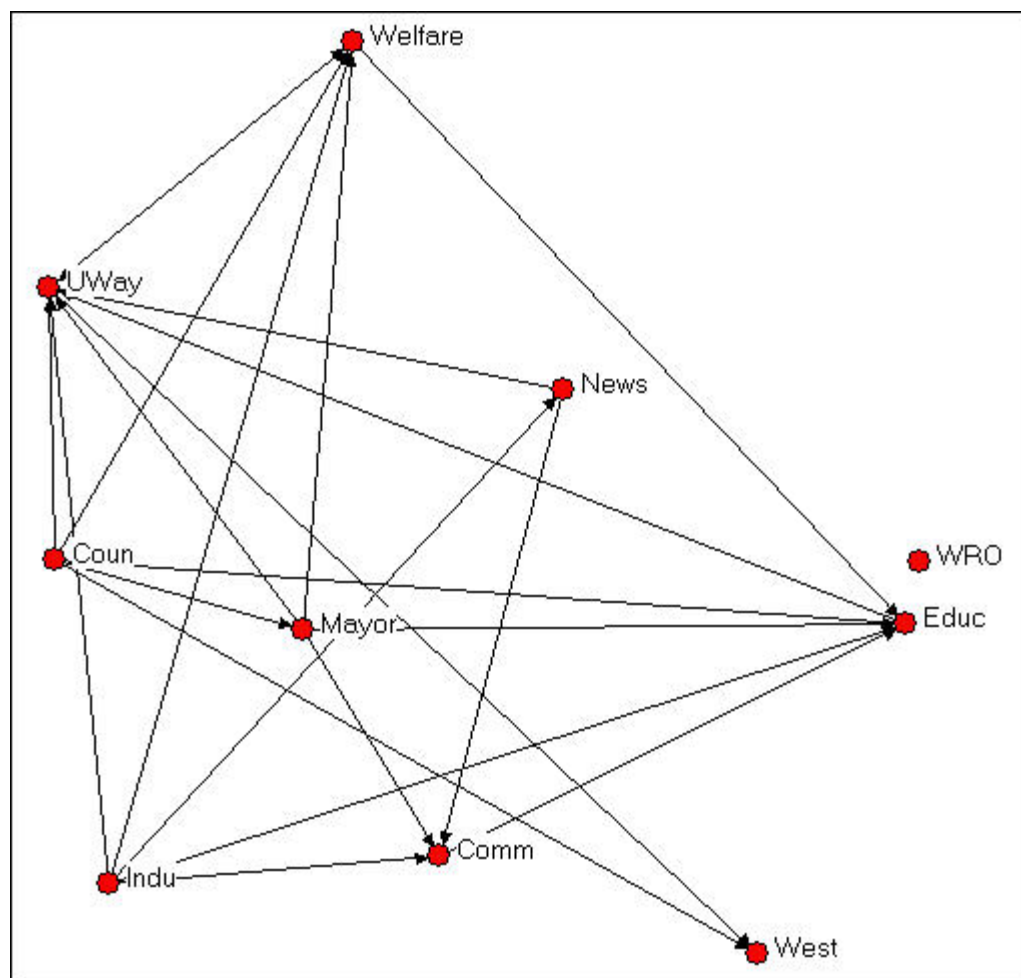
Node attributes

Differences of kind: We often have information available about some attributes of each the actors in our network. In the Bob, Carol, Ted and Alice example, we noted that two of the actors were male and two female. The scores of the cases (Bob, Carol, Ted, Alice) on the variable "sex" are a nominal dichotomy. It is also pretty common to be able to divide actors in a "multiple-choice" way; that is, we can record an attribute as a nominal polyotomy (for example, if we knew the religious affiliation of each actor, we might record it as "Christian," "Muslim," "Jewish," "Zoroastrian," or whatever).

It is often the case that the structure of a network depends on the attributes of the actors embedded in it. If we are looking at the network of "spouse" ties among Bob, Carol, Ted, and Alice, one would note that ties exist for male-female pairs, but not (in our example) for female-female or male-male pairs. Being able to visualize the locations of different types of actors in a graph can help us see patterns, and to understand the nature of the social processes that generated the tie structure.

Using colors and shapes are useful ways of conveying information about what "type" of actor each node is. Figures 4.1 and 4.2 provide an example. The data here describe the exchange of information among ten organizations that were involved in the local political economy of social welfare services in a Midwestern city (from a study by David Knoke; the data are one of the data sets distributed with UCINET). In Figure 4.1, NetDraw has been used to render a directed graph of the data. This is done by opening [Netdraw>File>Open>UCInet dataset>Network](#), and locating the data file. NetDraw produces a basic graph that you can then edit.

Figure 4.1. Knoke information exchange network



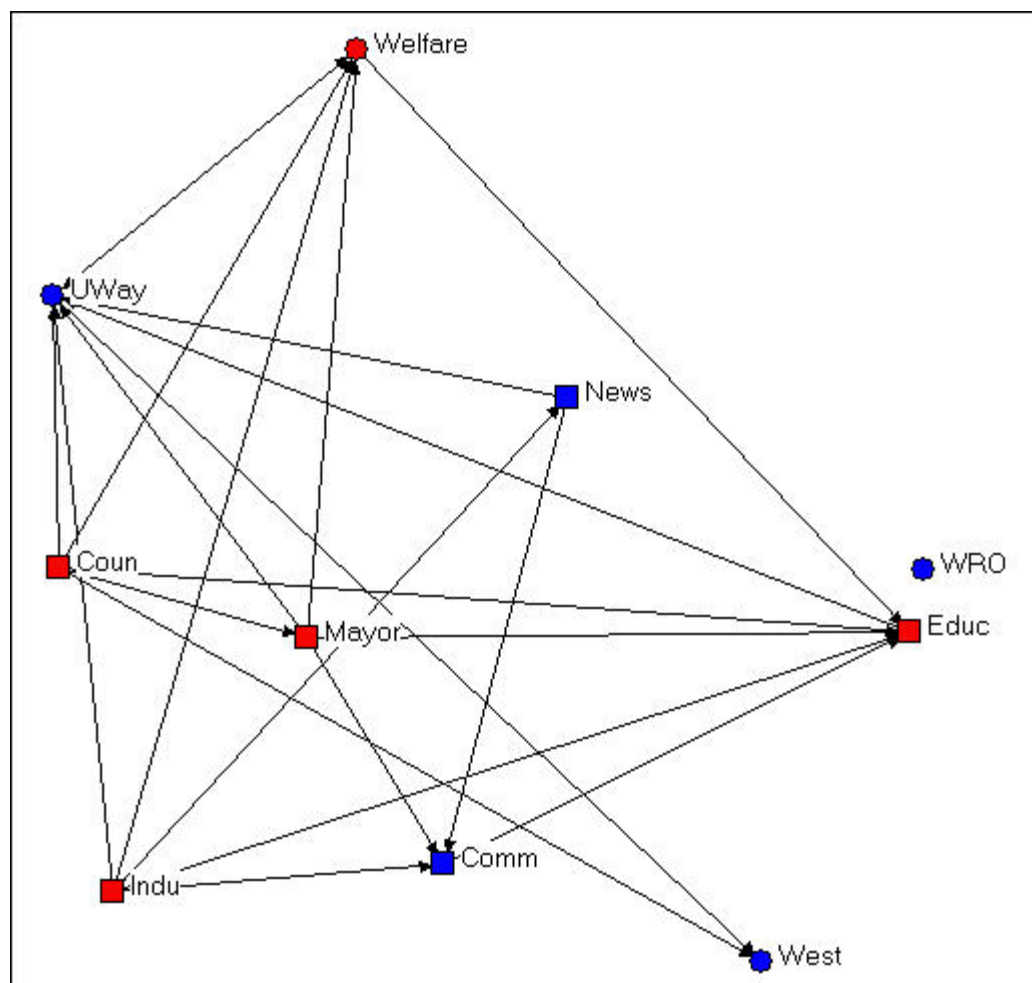
Institutional theory might suggest that information exchange among organizations of the same "type" would be more common than information exchange between organizations of different types. Some of the organizations here are governmental (Welfare, Coun, Educ, Mayor, Indu), some are non-governmental (UWay, News, WRO, Comm, West).

In Netdraw, we used the *Transform>mode attribute* editor to assign a score of "1" to each node if it was governmental, and "0" if it was not. We then used *Properties>nodes>color>attribute-based* to select the government attribute, and assign the color red to government organizations, and blue to non-government organizations. You could also create an attribute data file in UCINET using the same nodes as the network data file, and creating one or more columns of attributes. *NetDraw>File>Open>UCInet dataset>Attribute data* can then be used to open the attributes, along with the network, in NetDraw.

Ecological theory of organizations suggests that a division between organizations that are "generalists" (i.e. perform a variety of functions and operate in several different fields) and organizations that are "specialists" (e.g. work only in social welfare) might affect information-sharing patterns.

In Netdraw, we used the *Transform>mode attribute* editor to create a new column to hold information about whether each organization was a "generalist" or a "specialist." We assigned the score of "1" to "generalists" (e.g. the Newspaper, Mayor) and a score of "0" to "specialists" (e.g. the Welfare Rights Organization). We then used *Properties>nodes>shape>attribute-based* to assign the shape "square" to generalists and "circle" to specialists. The result of these operations is shown in figure 4.2.

Figure 4.2. Knoke information exchange with government/non-government and specialist/generalist codes

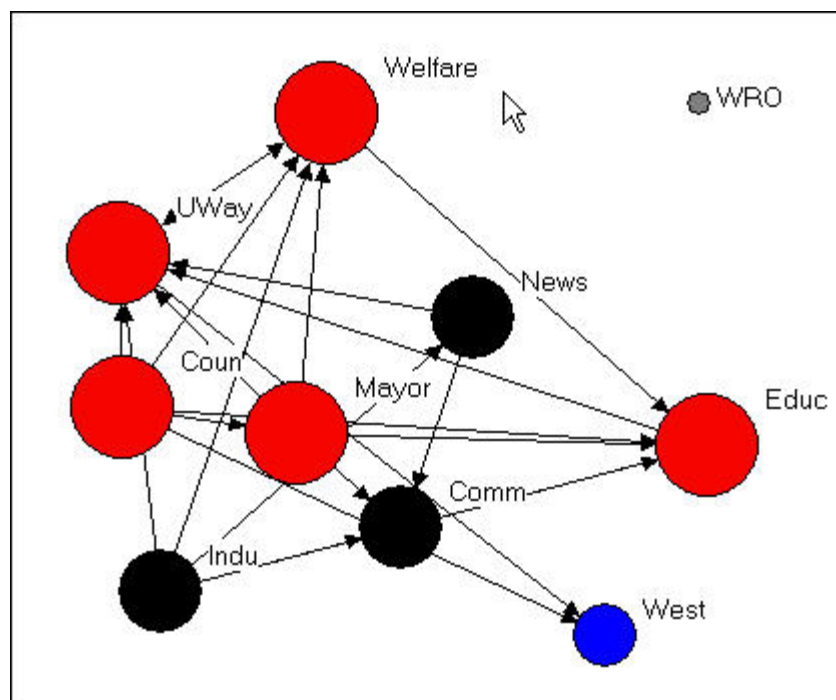


A visual inspection of the diagram with the two attributes highlighted by node color and shape is much more

informative about the hypotheses of differential rates of connection among red/blue and among circle/square. It doesn't look like this diagram is very supportive of either of our hypotheses.

Identifying types of nodes according to their attributes can be useful to point out characteristics of actors that are based on their position in the graph. Consider the example in the next figure, 4.3.

Figure 4.3 Knoke information exchange with k-cores



This figure was created by using the [Analysis>K-core](#) tool that is built into NetDraw. We'll talk about the precise definition of a k-core in a later chapter. But, generally, a k-core is a set of nodes that are more closely connected to one another than they are to nodes in other k-cores. That is, a k-core is one definition of a "group" or "sub-structure" in a graph.

Figure 4.3 shows four sub-groups, which are colored to identify which nodes are members of which group (the "West" group and the "WRO" group each contain only a single node). In addition, the size of the nodes in each K-core are proportional to the size of the K-core. The largest group contains government members (Mayor, County Government, Board of Education), as well as the main public (Welfare) and private (United Way) welfare agencies. A second group, colored in black, groups together the newspaper, chamber of commerce, and industrial development agency. Substantively, this actually makes some sense!

This example shows that color and shape of nodes to represent qualitative differences among actors can be based on classifying actors according to their position in the graph, how they are embedded, rather than on some inherent feature of the actor itself (e.g. governmental or non-governmental). One can use UCINET (or other programs) to identify "types" of actors based on their relations (e.g. where are the cliques?), and then enter this information into the attribute editor of NetDraw ([Transform>node attribute editor>edit>add column](#)). The groupings that are created by using [Analysis](#) tools already built-in to NetDraw are automatically added to the node attribute data base).

Differences of amount: Figure 4.3 also uses the size of the nodes ([Properties>nodes>size>attribute-based](#)) to display an index of the number of nodes in each group. This difference of amount among the nodes is best indicated, visually, by assigning the size of the node to values of some attribute. In example 4.3, NetDraw has done this automatically for an amount that was computed by its [Analysis>K-cores](#) tool. One can easily compute other variables reflecting differences in amount among actors (e.g. how many "out

degrees" or arrows from each actor are there?) using UCINET or other programs. Once these quantities are computed, they can be added to NetDraw (*Transform>node attribute editor>edit>add column*), and then added to the graph (*Properties>nodes>size>attribute-based*).

Differences of amount among the nodes could also reflect an attribute that is inherent to an actor alone. In the welfare organizations example, we might know the annual budget or number of employees of each organization. These measures of organizational "size" might be added to the node attributes, and used to scale the size of the nodes so that we could see whether information sharing patterns differ by organizational size.

Color, shape, and size of the nodes in a graph then can be used to visualize information that we have about the attributes of the actors. These attributes can be based on either "external" information about inherent differences in kind and amount among the actors; or, the attributes can be based on "internal" information about differences in kind and amount that describe how the actor is embedded in the relational network.

[table of contents](#)

Relation properties

A graph, as we discussed in the last chapter, is made up of both the actors and the relations among the actors. The relations among the actors (the line segments in a simple graph or the arrows in a directed graph) can also have "attributes." Sometimes it can be very helpful to use color and size to indicate difference of kind and amount among the relations. You can be creative with this idea to explore and display patterns in the connections among the actors in a network. Here are a few ideas of things that you could do.

Suppose that you wanted to highlight certain "types" of relations in the graph. For example, in a sociogram of friendship ties you might want to show how the patterns of ties among persons of the same sex differed from the patterns of ties among persons of different sexes. You might want to show the ties in the graph as differing in color or shape (e.g. dashed or solid line) depending on the type of relation. If you have recorded the two kinds of relations (same sex, different sex) as two relations in a multiplex graph, then you can use NetDraw's *Properties>Lines>Color* from the menu. Then select *Relations*, and choose the color for each of the relations you want to graph (e.g. red for same-sex, blue for different-sex).

Line colors (but not line shapes) can be used to highlight links within actors of the same type or between actors of different types, or both, using NetDraw. First select *Properties>Lines>Node-attribute* from the menus. Then select whether you want to color the ties among actors of the same attribute type ("*within*") or ties among actors of different types ("*between*"), or both. Then use the drop-down menu to select the attribute that you want to graph.

If you have measured each tie with an ordinal or interval level variable (usually reflecting the "strength" of the tie), you can also assign colors to ties based on tie strength (*Properties>Lines>Color>Tie-strength*). But, when you have information on the "value" of the relations, a different method would usually be preferred.

Where the ties among actors have been measured as a value (rather than just present-absent), the magnitude of the tie can be suggested by using thicker lines to represent stronger ties, and thinner lines to represent weaker ties. Recall that sometimes ties are measured as negative-neutral-positive (recorded as -1, 0, +1), as grouped ordinal (5=very strong, 4=strong, 3= moderate, 2=weak, 1=very weak, 0=absent), full-rank order (10=strongest tie of 10, 9=second strongest tie of 10, etc.), or interval (e.g. dollars of trade flowing from Belgium to the United States in 1975).

Since the value of the tie is already recorded in the data set, it is very easy to get NetDraw to visualize it in the graph. From the menus, select *Properties>Lines>Size*. Then, select *Tie-Strength* and indicate which

relation you want graphed. You can select the amount of gradation in the line widths (e.g. from 0 to 5 for a grouped ordinal variable with 6 levels; or from 5 to 10 if you want really thick lines).

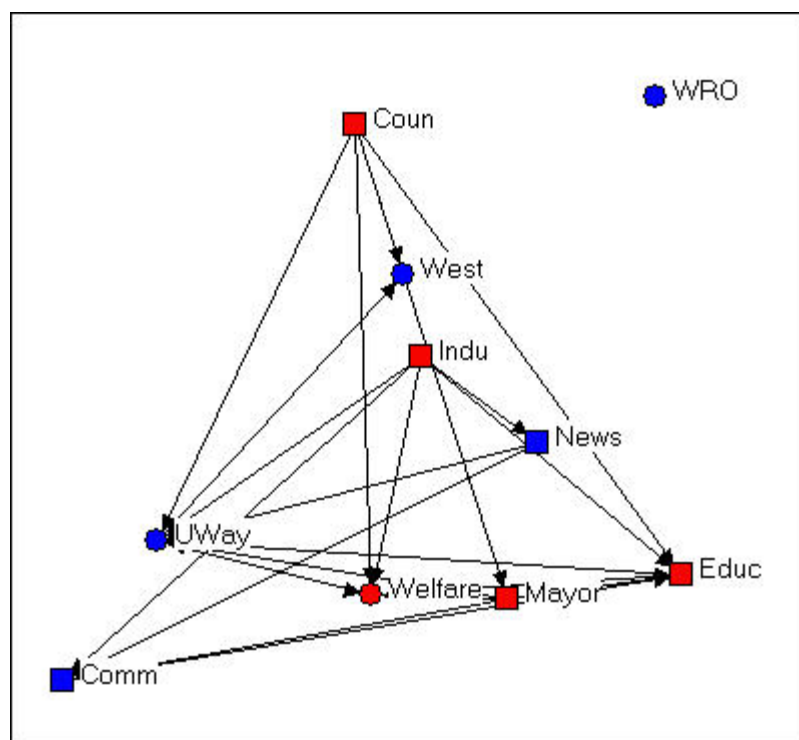
Using line colors and thickness, you can highlight certain types of ties and varying strength of ties among actors in the network. This can be combined with visual highlighting of attributes of the actors to make a compelling presentation of the features of the graph that you want to emphasize. There is a lot of art to this, and you will have to play and experiment. Node and line attributes can obscure as well as reveal; they can mis-represent, as well as represent. Sometimes, they add to the confusion of an already too-complicated graph.

[table of contents of this page](#)

Location, location, location

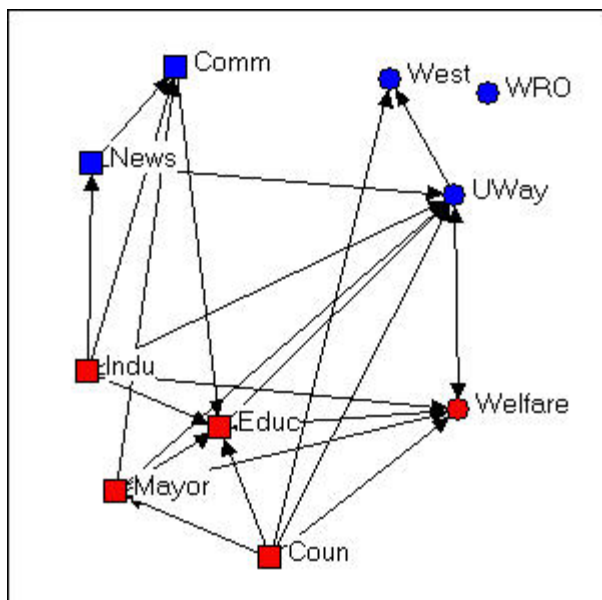
Most graphs of networks are drawn in a two-dimensional "X-Y axis" space (Mage and some other packages allow 3-dimensional rendering and rotation). Where a node or a relation is drawn in the space is essentially arbitrary -- the full information about the network is contained in its list of nodes and relations. The figures below are exactly the same network (Knoke's money flow network) that has been rendered in several different ways.

Figure 4.4 Random configuration of Knoke's money network



This drawing was created using NetDraw's *Layout>Random* on the graph that we had previously "colored" (blue for non-government, red for government; circles for welfare specialists, squares for generalists). This algorithm locates the nodes randomly in the X-Y space (you can use other tools to change the size of the graphic, rotate it, etc.). Since the X and Y directions don't "mean" anything, the location of the nodes and relations don't provide any particular insight.

Figure 4.5 Free-hand grouping by attributes configuration of Knoke's money network



In figure 4.5, we've used the "drag and drop" method ("grab" a node with the cursor, and drag it to a new location) to relocate the nodes so that organizations that share the same combinations of attributes are located in different quadrants of the graph. Since we had hypothesized that organizations of like "kind" would have higher rates of connection, this is a useful (but still arbitrary) way to locate the points. If the hypothesis were strongly supported (and it's not) most of the arrows would be located within each the four quadrants, and there would be few arrows between quadrants. NetDraw has a built-in tool that allows the user to assign the X and Y dimensions of the graph to scores on attributes (either categorical or continuous): [Layout>Attributes as Coordinates](#), and then select attributes to be assigned to X or Y or both. This can be a very useful tool for exploring how patterns of ties differ within and between "partitions" (or types of nodes).

Figure 4.6 Circle configuration of Knoke's money network

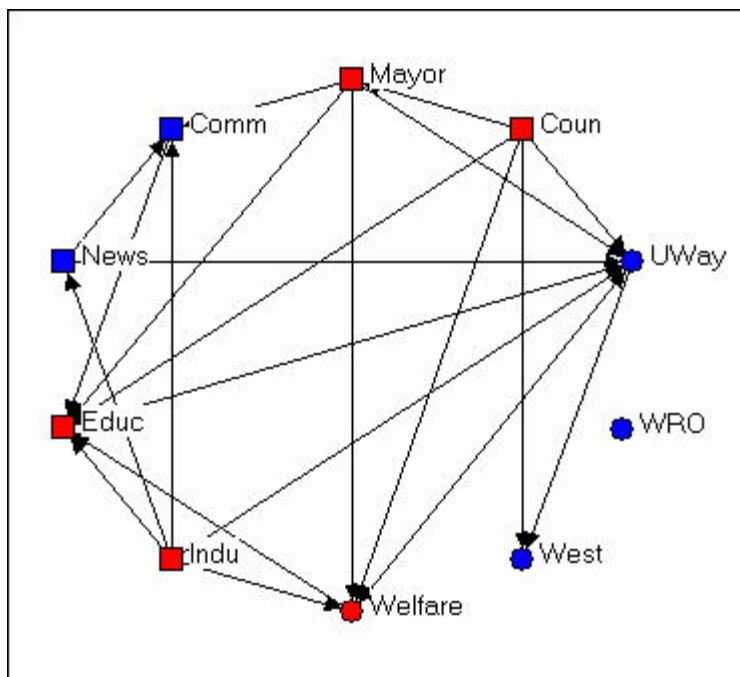


Figure 4.6 shows the same graph using [Layout>Circle](#), and selecting the "generalist-specialist" (i.e. the circle or square node type) as the organizing criterion. Circle graphs are commonly used to visualize which nodes are most highly connected. The nodes are located at equal distances around a circle, and nodes that are highly connected are very easy to quickly locate (e.g. UWay, Educ) because of the density of lines.

When nodes sharing the same attribute are located together in a segment of the circle, the density of ties within and between types is also quite apparent.

In each of the different layouts we've discussed so far, the distances between the nodes are arbitrary, and can't be interpreted in any meaningful way as "closeness" of the actors. And, the "directions" X and Y have no meaning -- we could rotate any of the graphs any amount, and it would not change a thing about our interpretation. There are several other commonly used graphic layouts that do try to make the distances and/or directions of locations among the actors somewhat more meaningful.

Figure 4.7 Non-metric multi-dimensional scaling configuration of Knoke's money network

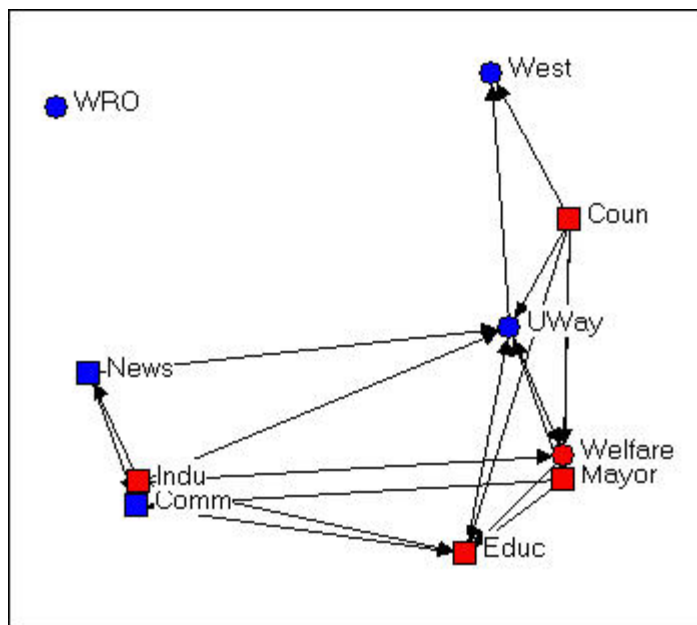


Figure 4.7 was generated using the [Layout>Graph Theoretic Layout>MDS](#) tool of NetDraw. MDS stands for (non-metric, in this case) "Multi-Dimensional Scaling." MDS is a family of techniques that is used (in network analysis) to assign locations to nodes in multi-dimensional space (in the case of the drawing, a 2-dimensional space) such that nodes that are "more similar" are closer together. There are many reasonable definitions of what it means for two nodes to be "similar." In this example, two nodes are "similar" to the extent that they have similar shortest paths (geodesic distances) to all other nodes. There are many, many ways of doing MDS, but the default tools chosen in NetDraw can often generate meaningful renderings of graphs that provide insights. NetDraw has several built-in algorithms for generating coordinates based on similarity (metric and non-metric two-dimensional scaling, and principle components analysis).

One very important difference between figure 4.7 and the earlier graphs is that the distances between the nodes is interpretable. The "Welfare" and "Mayor" nodes are very similar in their geodesic distances from other actors. "West" and "Educ" have very different patterns of ties to the other nodes.

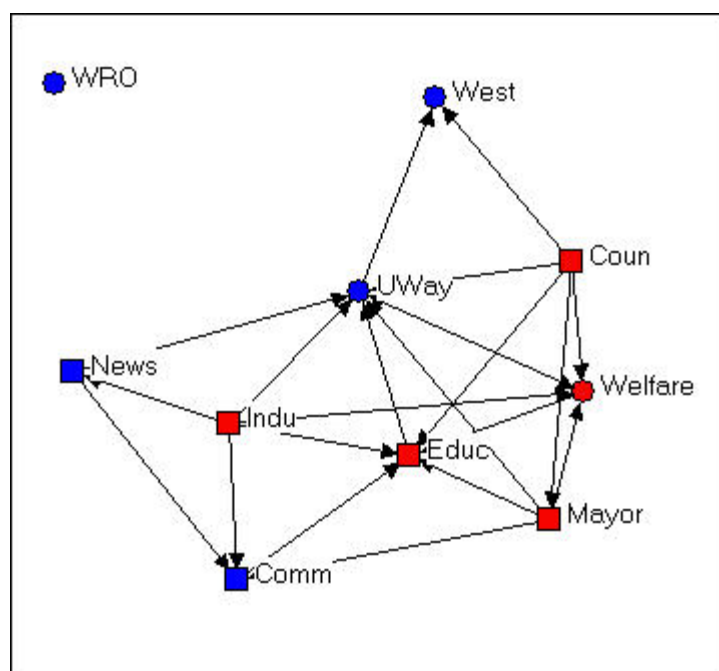
The other important difference between figure 4.7 and the earlier graphs is that direction may be meaningful. Notice that there is a cluster of nodes at the left (News, Indu, Comm) that are all pretty much not welfare organizations themselves, while the nodes at the right are (generally) more directly involved in welfare service provision. The upper left-hand quadrant contains mostly "blue" nodes, while the lower right quadrant contains mostly "red" ones -- so one "direction" might be interpreted as "non-government/government."

Let me emphasize that different applications of MDS (and other scaling tools) to different definitions of what it means for nodes to be "similar" can generate wildly different looking graphs. These are (almost always) exploratory techniques, and there is (usually) no single "correct" interpretation. But, a graphic that uses both distance and direction to summarize something about the structure of the network can provide considerable

insight when compared to graphs (like figures 4.4, 4.5, and 4.6) that don't.

Consider one last example of rendering the same data, this time using NetDraw's unique built-in algorithm for locating points (*Layout>Graph Theoretic Layout>Spring Embedding*).

Figure 4.8 "Spring-embedding" configuration of Knoke's money network



You might immediately notice that this graph is fairly similar to the MDS solution. The algorithm uses iterative fitting (i.e. start with a random graph, measure "badness" of fit; move something, measure "badness" and if it's better keep going in that direction...) to locate the points in such a way as to put those with smallest path lengths to one another closest in the graph. This approach can often locate points very close together, and make for a graph that is hard to read. In the current example, we've also selected the optional "node repulsion" criterion that creates separation between objects that would otherwise be located very close to one another. We've also used the optional criterion of seeking to make the paths of "equal edge length" so that the distances between adjacent objects are similar.

The result is a graph that preserves many of the features of the dimensional scaling approach (distances are still somewhat interpretable; directions are often interpretable), but is usually easier to read -- particularly if it matters which specific nodes are where (rather than node types or clusters).

There is no one "right way" to use space in a graph. But one can usually do much better than a random configuration -- particularly if one has some prior hypotheses or research questions about the kinds of patterns that would be meaningful.

[table of contents of this page](#)

Highlighting parts of the network

Large networks (those that contain many actors, many kinds of relations, and/or high densities of ties) can be very difficult to visualize in any useful way -- there is simply too much information. Often, we need to clear away some of the "clutter" to see main patterns more clearly.

One of the most interesting features of social networks -- whether small or large -- is the extent to which we

can locate "local sub-structures." We will discuss this topic a good bit more in a later chapter. Highlighting or focusing attention on sub-sets of nodes in a drawing can be a powerful tool for visualizing sub-structures.

In this section, we will briefly outline some approaches to rendering network drawings that can help to simplify complex diagrams and locate interesting sub-graphs (i.e. collections of nodes and their connections).

Clearing away the underbrush

Social structures can be composed of multiple relations. Bob, Carol, Ted, and Alice in our earlier example are a multi-plex structure of people connected by both friendship and by spousal ties. Graphs that display all of the connections among a set of nodes can be very useful for understanding how actors are tied together -- but they can also get so complicated and dense that it is difficult to see any patterns. There are a couple approaches that can help.

One approach is to combine multiple relations into an index. For example, one could combine the information on friendship and spousal ties using an "and" rule: if two nodes have both a friendship and spousal tie, then they have a tie - otherwise they do not (i.e. if they have no tie, or only one type of tie). Alternatively, we could create an index that records a tie when there is either a friendship tie or a spousal tie. If we had measured relations with values, rather than simple presence-absence, multiple relations could be combined by addition, subtraction, multiplication, division, averaging, or other methods. UCINET has tools for these kinds of operations, that are located at: [Transform>matrix operations>within dataset>aggregations](#).

The other approach is to simplify the data a bit. NetDraw has some tools that can be of some help.

Rather than examining the information on multiple kinds of ties in one diagram, one can look at them one at a time, or in combination. If the data have been stored as a UCINET or NetDraw data file with multiple relations, then the [Options>View>Relations Box](#) opens a dialog box that lets you select which relations you want to display. Suppose that we had a data set in which we had recorded the friendship ties among a number of people at intervals over a period of time. By first displaying the first time point, and then adding subsequent time point, we can visualize the evolution of the friendship structure.

It isn't unusual for some of the nodes in a graph of a social network to not be connected to the others at all. Nodes that aren't connected are called "isolates." Some nodes may be connected to the network by a single tie. These nodes sort of "dangle" from the diagram; they are called "pendants." One way of simplifying graphs is to hide isolates and/or pendants to reduce visual clutter. Of course, this does mis-represent the structure, but it may allow us to focus more attention where most of the action is. NetDraw has both button-bar tools and a menu item ([Analysis>Isolates](#)) to hide these less-connected nodes.

Finding and visualizing local sub-structures

One of the common questions in network analysis is whether a graph displays various kinds of "sub-structures." For example, a "clique" is a sub-structure that is defined as a set of nodes where every element of the set is connected to every other member. A network that has no cliques might be a very different place than a network that has many small cliques, or one that has one clique and many nodes that are not part of the clique. We'll take a closer look at UCINET tools for identifying sub-structures in a later chapter.

NetDraw has built-in a number of tools for identifying sub-structures, and automatically coloring the graph to identify them visually.

[Analysis>components](#) locates the parts of graph that are completely disconnected from one another, and colors each set of nodes (i.e. each component). In our Bob-Carol-Ted-Alice example, the entire graph is one component, because all the actors are connected. In the welfare bureaucracies example, there are two

components, one composed of only WRO (which does not receive ties from any other organization) and the other composed of the other nine nodes. In NetDraw, executing this command also creates a variable in the database of node attributes -- as do all the other commands discussed here. These attributes can then be used for selecting cases, changing color, shape, and size, etc.

Analysis>Blocks and Cutpoints locates parts of the graph that would become disconnected components if either one node or one relation were removed (the blocks are the resulting components; the cutpoint is the node that would, if removed, create the dis-connect). NetDraw graphs these sub-structures, and saves the information in the node-attribute database.

Analysis>K-cores locates parts of the graph that form sub-groups such that each member of a sub-group is connected to N-K of the other members. That is, groups are the largest structures in which all members are connected to all but some number (K) of other members. A "clique" is a group like this where all members are connected to all other members; "fuzzier" or "looser" groups are created by increasing "K." NetDraw identifies the K-cores that are created by different levels of K, and provides colored graphs and data-base entries.

Analysis>Subgroups>block based. Sorry, but I don't know what this algorithm does! Most likely, it creates sub-structures that would become components with differing amounts of nodes/relations removed.

Analysis>Subgroups>Hierarchical Clustering of Geodesic Distances. The geodesic distance between two nodes is the length of the shortest path between them. A hierarchical clustering of distances produces a tree-like diagram in which the two nodes that are most similar in their profile of distances to all other points are joined into a cluster; the process is then repeated over and over until all nodes are joined. The resulting graphic is one way of understanding which nodes are most similar to one another, and how the nodes may be classified into "types" based on their patterns of connection to other nodes. The graph is colored to represent the clusters, and database information is stored about the cluster memberships at various levels of aggregation. A hierarchical clustering can be very interesting in understanding which groups are more homogeneous (those that group together at early stages in the clustering) than others; moving up the clustering tree diagram, we can see a sort of a "contour map" of the similarity of nodes.

Analysis>Subgroups>Factions (select number). A "faction" is a part of a graph in which the nodes are more tightly connected to one another than they are to members of other "factions." This is quite an intuitively appealing idea of local clustering or sub-structure (though, as you can see, only one such idea). NetDraw asks you how many factions you would like to find (always explore various reasonable possibilities!). The algorithm then forms the number of groups that you desire by seeking to maximize connection within, and minimize connection between the groups. Points are colored, and the information about which nodes fall in which partitions (i.e. which cases are in which factions) is saved to the node attributes database.

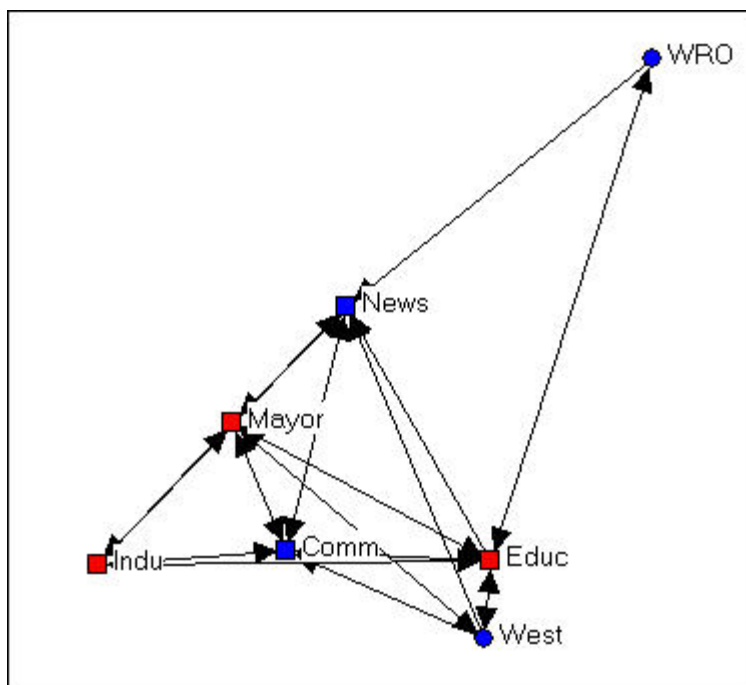
Analysis>Subgroups>Newman-Girvan. This is another numerical algorithm that seeks to create clusters of nodes that are closely connected within, and less connected between clusters. The approach is that of "block modeling." Rows and columns are moved to try to create "blocks" where all connections within a block are present, and all connections between blocks are absent. This algorithm will usually produce results similar to the factions algorithm. Importantly, though, the Newman-Girvan algorithm also produces measures of goodness-of-fit of the configuration for two blocks, three blocks, etc. This allows you to get some sense of what division into blocks is optimal for your needs (there isn't one "right" answer).

Ego Networks (neighborhoods)

A very useful way of understanding complicated network graphs is to see how they arise from the local connections of individual actors. The network formed by selecting a node, including all actors that are connected to that node, and all the connections among those other actors is called the "ego network" or (1-

step) neighborhood of an actor. Figure 4.9 is an example from the Knoke bureaucracies information network, where we select as our "ego" the board of education.

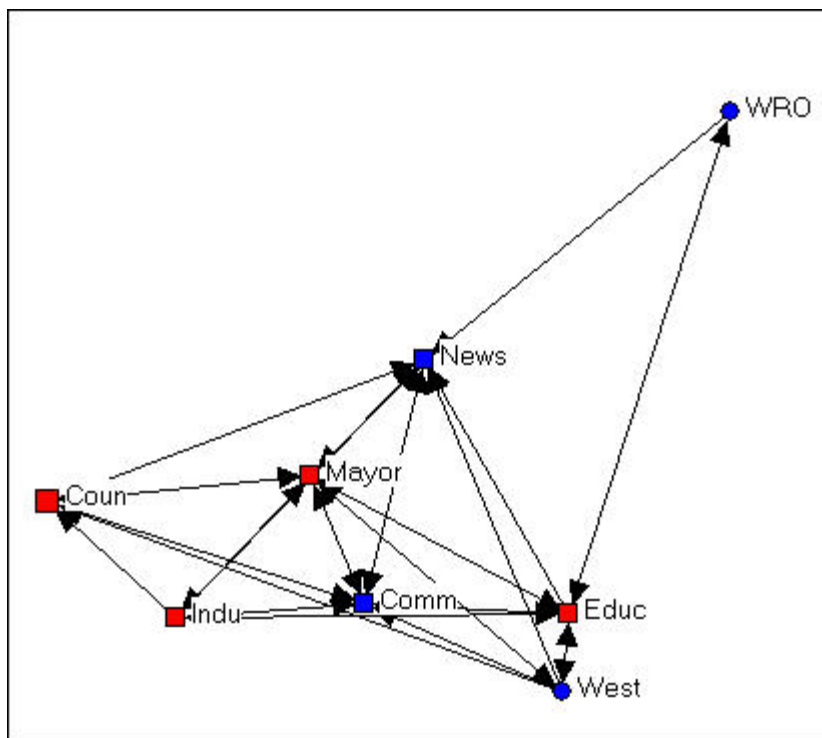
Figure 4.9. Ego network of Educ in Knoke information network



We note that the ego-network of the board of education is fairly extensive, and that the density of connection among the actors in the ego-network is fairly high. This is to say say the the board of education is quite "embedded" in a dense local sub-structure.

Next, let's add the ego network of the "West" agency, in figure 4.10.

Figure 4.10. Ego networks of Educ and West in Knoke information network



The two ego networks combined result in a fully connected structure. We note that one connection between Educ and Coun is mediated by West.

One can often gain considerable insight about complicated networks by "building" them starting with one actor and adding others. Or, one can begin with the whole network, and see what happens as individual's ego networks are removed.

The network of each individual actor may also be of considerable interest. Who's most connected? How dense are the neighborhoods of particular actors?

NetDraw has useful tools for visualizing and working with ego-networks. The *Layout>Egonet* command presents a dialog box that lets you select which ego's networks are to be displayed. You can start with all the actors and delete; or start with focal actors and build up the full network.

[table of contents of this page](#)

A few hints on data handling with NetDraw

Input

There are several ways to get data into NetDraw. Probably the simplest is to import data from UCINET or Pajek. The *File>Open* command lets you read a UCINET text (DL) file (discussed elsewhere), and existing UCINET dataset, or a Pajek dataset. This menu also is used to access data that have been stored in the native data format of the NetDraw program (.VNA format). Once the data has been imported with the *Open* command, the node and line attribute editors of NetDraw can be used to create a diagram that can be saved with colors, shapes, locations, etc.

A second method is to build a dataset within NetDraw itself. Begin by creating a random network (*File>Random*). This creates an arbitrary network of 20 nodes. You can then use the node attributes editor (*Transform>Node Attribute editor*) and the link editor (*Transform>Link Editor*) to modify the nodes (and add or delete nodes) and their attributes; and to create connections among nodes. This is great for small, simple networks; for more complicated data, it's best to create the basic data set elsewhere and import it.

The third method is to use an external editor to create a NetDraw dataset (a .vna file) directly. This file is a plain ascii text file (if you use a word processor, be sure to save as ascii text). The contents of the file is pretty simple, and is discussed in the brief tutorial to NetDraw. Here is part of the file for the Knoke data, after we have created some of the diagrams we've seen.

```
*Node data
"ID", "General", "Size", "Govt"
  "1" "1" "3" "1"
  "2" "1" "2" "0"
.
.
  "9" "0" "2" "1"
  "10" "0" "1" "0"
*Node properties
ID x y color shape size shortlabel labelsize labelcolor active
"1" 51 476 255 2 16 "Coun" 11 0 TRUE
"2" 451 648 16711680 2 11 "Comm" 11 0 TRUE
.
.
"9" 348 54 255 1 11 "Welfare" 11 0 TRUE
"10" 744 801 16711680 1 6 "West" 11 0 TRUE
*Tie data
FROM TO "KNOKI" "KNOKM"
```

```

"1" "2" 1 0
"1" "5" 1 1
.
.
"9" "3" 0 1
"9" "8" 0 1
*Tie properties
FROM TO color size active
"1" "2" 0 1 TRUE
"1" "5" 0 1 TRUE
.
.
"9" "3" 0 1 FALSE
"9" "8" 0 1 FALSE

```

There are four sections of code here (not all are needed, but the `*node data` and `*tie data` are, to define the network structure). `*Node data` lists variables describing the nodes. An ID variable is necessary, the other variables in the example describe attributes of each node. The (optional) `*Node properties` section lists the variables, and gives values for ID, location on the diagram (X and Y coordinates from the upper left corner), shape, size, color, etc. Usually, one will not create this code; rather you input the data, use NetDraw to create a diagram, and save the result as a file -- and this section (and the `*Tie properties`) is created for you.

The `*Tie data` section is necessary to define the connections among the nodes. There is one data line for each relation in the graph. Each data line is described by its origin and destination, and value. Here, since there are two relations, "KNOKI" and "KNOKM" there are two values -- each of which happens, in our example, to be binary (but they could be valued).

The `*Tie properties` section is probably best created by using NetDraw and saving the resulting file. Each tie is identified by origin and destination, and its color and size are set. Here, certain ties are not to be visible in the drawing (the "active" property is set to "FALSE").

Output

When you are working with NetDraw, it is a good idea to save a copy of your work in the format (.vna, above) that is native to the program ([File>Save Data As>Vna](#)). This format keeps all of the information about your diagram (what's visible and not, node and line attributes, locations) so that you can re-open the diagram looking exactly as you left it.

You may also want to save datasets created with NetDraw to other program's formats. You won't be able to save all of the information about node and line properties and locations, but you can save the basic network (what are the nodes, which is connected to which) and node attributes. [File>Save Data As>Pajek](#) lets you save the network, partitions of it (which record attributes), and clusterings in Pajek format. [File>Save Data As>UCINET](#) lets you save the basic network information for binary or valued networks (UCINET needs to know which) and attributes (which are stored in a separate file in UCINET).

The whole point of making more interesting drawings of graphs, of course, is to be able to use them to illustrate your ideas. There are several possibilities.

Screen capture programs (I used SnagIT) can take pictures of your graphics that can be then saved in any number of formats, and edited further by external graphics editors (perhaps to add titles, annotations, and other highlights).

[File>print](#), of course, does just that.

[File>Save Diagram As](#) let's you save your diagram in three of the most common graphics formats (Windows

Metafile, bitmap BMP, or JPEG). Once saved, these file formats can be further edited with graphics editing programs to be inserted into web or hard-copy documents.

[table of contents of this page](#)

Conclusions

A lot of the work that we do with social networks is primarily descriptive and/or exploratory, rather than confirmatory hypothesis testing. Using some of the tools described in this chapter can be particularly helpful because they may let you see patterns that you might not otherwise have seen. The tools can be used to explore tentative empirical generalizations and provide crude first examinations of hypotheses about patterns that may be present in the data.

Some of the tools are also very helpful for dealing with the complexity of social network data, which may involve many actors, many ties, and several types of ties. Hiding, highlighting, and locating parts of the data can be a big help in making sense of the data. In some cases (like ego networks and the evolution of networks over time) hiding and revealing parts of the data are critical to understanding and describing the construction and evolution of the social structures.

Finally, working with drawings can be a lot of fun, and a bit of an outlet for your creative side. A really good graphic can also be far more effective in sharing your insights than any number of words.

[table of contents of this page](#)

[table of contents of the textbook](#)

Introduction to social network methods

5. Using matrices to represent social relations

This page is part of an on-line text by [Robert A. Hanneman](#) ([Department of Sociology, University of California, Riverside](#)) and Mark Riddle (Department of Sociology, University of Northern Colorado). Feel free to use and distribute this textbook, with citation. Your comments and suggestions are very welcome. [Send me e-mail.](#)

Contents of this chapter:

- [What is a matrix?](#)
 - [The "adjacency" matrix](#)
 - [Matrix permutation, blocks, and images](#)
 - [Doing mathematical operations on matrices](#)
 - [Transposing a matrix](#)
 - [Taking the inverse of a matrix](#)
 - [Matrix addition and matrix subtraction](#)
 - [Matrix multiplication and Boolean matrix multiplication](#)
 - [Summary](#)
 - [Study questions](#)
-

Introduction

Graphs are very useful ways of presenting information about social networks. However, when there are many actors and/or many kinds of relations, they can become so visually complicated that it is very difficult to see patterns. It is also possible to represent information about social networks in the form of matrices. Representing the information in this way also allows the application of mathematical and computer tools to summarize and find patterns. Social network analysts use matrices in a number of different ways. So, understanding a few basic things about matrices from mathematics is necessary. We'll go over just a few basics here that cover most of what you need to know to understand what social network analysts are doing. For those who want to know more, there are a number of good introductory books on matrix algebra for social scientists.

[table of contents](#)

What is a matrix?

To start with, a matrix is nothing more than a rectangular arrangement of a set of elements (actually, it's a bit more complicated than that, but we will return to matrices of more than two dimensions in a little bit). Rectangles have sizes that are described by the number of rows of elements and columns of elements that they contain. A "3 by 6" matrix has three rows and six columns; an "l by j" matrix has l rows and j columns. A matrix that has only one row is called a "row vector." A matrix that has only one column is called a "column vector."

Figure 5.1 shows a two-by-four matrix. Figure 5.2 shows a four by two matrix. Just for the moment, ignore the contents of the cells (e.g. 1,1).

Figure 5.1. Example of a "two-by-four" matrix



1,1	1,2	1,3	1,4
2,1	2,2	2,3	2,4

Figure 5.2. Example of a "four-by-two" matrix

1,1	1,2
2,1	2,2
3,1	3,2
4,1	4,2

The elements (cells) of a matrix are identified by their "addresses." Element 1,1 is the entry in the first row and first column; element 13,2 is in the 13th row and is the second element of that row. The cell addresses have been entered as matrix elements in the two examples above.

Matrices are often represented as arrays of elements surrounded by vertical lines at their left and right, or square brackets at the left and right. In web pages it's easier to use "tables" to represent matrices. Matrices can be given names; these names are usually presented as capital bold-faced letters. Social scientists using matrices to represent social networks often dispense with the mathematical conventions, and simply show their data as an array of labeled rows and columns. The labels are not really part of the matrix, but are simply for clarity of presentation. The matrix in figure 5.3 for example, is a 4 by 4 matrix, with additional labels.

Figure 5.3. Four-by-four matrix with additional row and column labels

	A	B	C	D
A	---	1	0	0
B	1	---	1	0
C	1	1	---	1
D	0	0	1	---

The matrices used in social network analysis are frequently "square." That is, they contain the same number of rows and columns. But "rectangular" matrices are also used, as are row and column vectors. The same conventions apply to all these variations.

Occasionally, social network analysts will use a "3-dimensional" matrix. A three dimensional matrix has rows, columns, and "levels" or "slices." Each "slice" has the same rows and columns as each other slice. UCINET thinks about these more complicated 3-dimensional arrays of data as a collection of two-dimensional matrices.

[table of contents](#)

The "adjacency" matrix

The most common form of matrix in social network analysis is a very simple square matrix with as many rows and columns as there are actors in our data set. The "elements" or scores in the cells of the matrix record information about the ties between each pair of actors.

The simplest and most common matrix is binary. That is, if a tie is present, a one is entered in a cell; if there

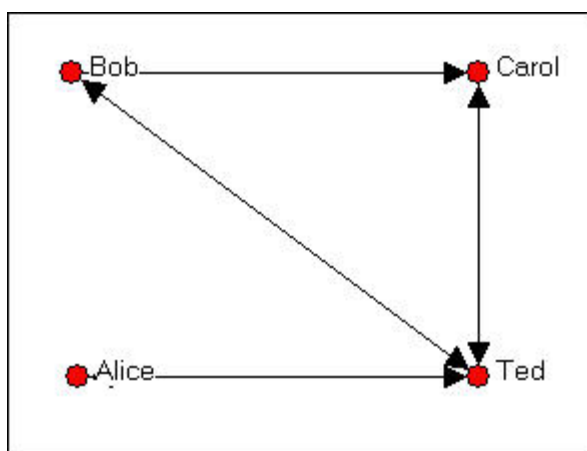
is no tie, a zero is entered. This kind of a matrix is the starting point for almost all network analysis, and is called an "adjacency matrix" because it represents who is next to, or adjacent to whom in the "social space" mapped by the relations that we have measured.

An adjacency matrix may be "symmetric" or "asymmetric." Social distance can be either symmetric or asymmetric. If Bob and Carol are "friends" they share a "bonded tie" and the entry in the $X_{i,j}$ cell will be the same as the entry in the $X_{j,i}$ cell.

But social distance can be a funny (non-Euclidean) thing. Bob may feel close to Carol, but Carol may not feel the same way about Bob. In this case, the element showing Bob's relationship to Carol would be scored "1," while the element showing Carol's relation to Bob would be scored "0." That is, in an "asymmetric" matrix, $X_{i,j}$ is not necessarily equal to $X_{j,i}$.

By convention, in a directed (i.e. asymmetric) matrix, the sender of a tie is the row and the target of the tie is the column. Let's look at a simple example. The directed graph of friendship choices among Bob, Carol, Ted, and Alice is shown in figure 5.4.

Figure 5.4 Bob, Carol, Ted, and Alice



We can since the ties are measured at the nominal level (that is, the data are binary choice data), we can represent the same information in a matrix that looks like:

Figure 5.5. Asymmetric adjacency matrix of the graph shown in figure 5.4.

	Bob	Carol	Ted	Alice
Bob	---	1	1	0
Carol	0	---	1	0
Ted	1	1	---	1
Alice	0	0	1	---

Remember that the rows represent the source of directed ties, and the columns the targets; Bob chooses Carol here, but Carol does not choose Bob. This is an example of an "asymmetric" matrix that represents directed ties (ties that go from a source to a receiver). That is, the element i,j does not necessarily equal the element j,i . If the ties that we were representing in our matrix were "bonded-ties" (for example, ties representing the relation "is a business partner of" or "co-occurrence or co-presence," (e.g. where ties represent a relation like: "serves on the same board of directors as") the matrix would necessarily be symmetric; that is element i,j would be equal to element j,i .

Binary choice data are usually represented with zeros and ones, indicating the presence or absence of each logically possible relationship between pairs of actors.

Signed graphs are represented in matrix form (usually) with -1, 0, and +1 to indicate negative relations, no or neutral relations, and positive relations. "Signed" graphs are actually a specialized version of an ordinal relation.

When ties are measured at the ordinal or interval level, the numeric magnitude of the measured tie is entered as the element of the matrix. As we discussed earlier, other forms of data are possible (multi-category nominal, ordinal with more than three ranks, full-rank order nominal). These other forms, however, are rarely used in sociological studies, and we won't give them very much attention.

In representing social network data as matrices, the question always arises: what do I do with the elements of the matrix where $i = j$? That is, for example, does Bob regard himself as a close friend of Bob? This part of the matrix is called the *main diagonal*. Sometimes the value of the main diagonal is meaningless, and it is ignored (and left blank or filled with zeros or ones). Sometimes, however, the main diagonal can be very important, and can take on meaningful values. This is particularly true when the rows and columns of our matrix are "super-nodes" or "blocks." More on that in a minute.

It is often convenient to refer to certain parts of a matrix using shorthand terminology. If I take all of the elements of a row (e.g. who Bob chose as friends: ---,1,1,0) I am examining the "*row vector*" for Bob. If I look only at who chose Bob as a friend (the first column, or ---,0,1,0), I am examining the "*column vector*" for Bob. It is sometimes useful to perform certain operations on row or column vectors. For example, if I summed the elements of the column vectors in this example, I would be measuring how "popular" each node was (in terms of how often they were the target of a directed friendship tie). So a "vector" can be an entire matrix (1 x ... or ...x 1), or a part of a larger matrix.

return to the [table of contents of this page](#)

Matrix permutation, blocks, and images

It is also helpful, sometimes, to rearrange the rows and columns of a matrix so that we can see patterns more clearly. Shifting rows and columns (if you want to rearrange the rows, you must rearrange the columns in the same way, or the matrix won't make sense for most operations) is called "permutation" of the matrix.

Our original data look like figure 5.6:

Figure 5.6. Asymmetric adjacency matrix

	Bob	Carol	Ted	Alice
Bob	---	1	1	0
Carol	0	---	1	0
Ted	1	1	---	1
Alice	0	0	1	---

Let's rearrange (permute) this so that the two males and the two females are adjacent in the matrix. *Matrix permutation* ([Data>Permute](#)) simply means to change the order of the rows and columns. Since the matrix is symmetric, if I change the position of a row, I must also change the position of the corresponding column. The result is shown in figure 5.7.

Figure 5.7. Permuted matrix

	Bob	Ted	Carol	Alice
Bob	---	1	1	0
Ted	1	---	1	1
Carol	0	1	---	0
Alice	0	1	0	---

None of the elements have had their values changed by this operation or rearranging the rows and columns, we have just shifted things around. We've also highlighted some sections of the matrix. Each colored section is referred to as a *block*. Blocks are formed by passing dividing lines through the matrix (e.g. between Ted and Carol) rows and columns. Passing these dividing lines through the matrix is called *partitioning the matrix*. Here we have partitioned by the actor by their sex. Partitioning is also sometimes called "blocking the matrix," because partitioning produces blocks.

This kind of grouping of cells is often done in network analysis to understand how some sets of actors are "embedded" in social roles or in larger entities. Here, for example, we can see that all occupants of the social role "male" choose each other as friends; no females choose each other as friends, and that males are more likely to choose females (3 out of 4 possibilities are selected) than females are to choose males (only 2 out of 4 possible choices). We have grouped the males together to create a "partition" or "super-node" or "social role" or "block." We often partition social network matrices in this way to identify and test ideas about how actors are "embedded" in social roles or other "contexts."

We might wish to dispense with the individual nodes altogether, and examine only the positions or roles. If we calculate the proportion of all ties within a block that are present, we can create a *block density matrix*. In doing this, we have ignored self-ties in figure 5.8.

Figure 5.8. Block density matrix

	Male	Female
Male	1.00	0.75
Female	0.50	0.00

We may wish to summarize the information still further by using *block image* or *image matrix*. If the density in a block is greater than some amount (we often use the average density for the whole matrix as a cut-off score, in the current example the density is .58), we enter a "1" in a cell of the blocked matrix, and a "0" otherwise. This kind of simplification is called the "image" of the blocked matrix, as in figure 5.9.

Figure 5.9. Image matrix of sex blocked data, using overall mean density as the cut-off

	Male	Female
Male	1	1
Female	0	0

Images of blocked matrices are powerful tools for simplifying the presentation of complex patterns of data. Like any simplifying procedure, good judgment must be used in deciding how to block and what cut-offs to use to create images -- or we may lose important information.

UCINET includes tools that make permuting and blocking matrices rather easy.

Transform>Block allows you to select a matrix to be blocked, a row and/or column partition, and a method for calculating the entries in the resulting blocks.

To use this command, you need to first create separate files that describe the row partition and the column partition. These files are simply vectors (either one row, or one column) that identify which actors are to fall into which partition. For example, if actors 1, 2, and 5 were to form group A, and actors 3 and 4 were to form group B, my column partition data set would read: 1 1 2 2 1. These partitions or blockings are simply regular UCINET data files with one row or one column.

The command asks for a method of summarizing the information within each block. You may take the average of the values in the block (if the data are binary, taking the average is the same thing as calculating the density), sum the values in the block, select the highest value or the lowest value, or select a measure of the amount of variation among the scores in the block -- either the sums of squares or the standard deviation.

The command outputs two new matrices. The "Prelmage" data set contains the original scores, but permuted; the "Reduced image dataset" contains a new block matrix containing the block densities.

Transform>Collapse allows you to combine rows and/or columns by specifying (detailed instructions are given on the command window) which elements are to be combined, and how. We might select, for example, to combine columns 1, 2, and 5, and rows 1, 2, and 5 by taking the average of the values (we could also select the maximum, minimum, or sum). The command creates a new matrix that has collapsed the desired rows or columns using the summary operation you selected.

The data menu also gives you some tools for this kind of work:

Data>Permute allows you to re-arrange the rows and/or columns and/or matrices (if your data set contains multiple matrices representing multiple relations, like the Knoke bureaucracies "information" and "money" relations). You simply specify the new order with a list. If I wanted to group rows 1, 2, and 5 to be new rows 1, 2, and 3; and rows 3 and 4 to be new rows 4 and 5, I would enter 1 2 4 5 3.

Data>Sort re-arranges the rows, columns, or both of the matrix according to a criterion you select. If you data are valued (i.e. represent tie strength) you might want to sort the rows and columns in ascending or descending order (this could make sense for binary data, too). If you want a more complicated sort (say "all the 3's first, then all the 1's, then all the 2's) you can use an external UCINET data file to specify this as a vector (i.e. the data set would just be: 3 1 2).

Data>Transpose re-arranges the data in a way that is very commonly used in matrix algebra -- by taking the "transpose." A transpose is, very simply, switching the rows and columns of a matrix for one another.

[table of contents](#)

Doing mathematical operations on matrices

Representing the ties among actors as matrices can help us to see patterns by performing simple manipulations like summing row vectors or partitioning the matrix into blocks. Social network analysts use a number of other mathematical operations that can be performed on matrices for a variety of purposes (matrix addition and subtraction, transposes, inverses, matrix multiplication, and some other more exotic stuff like determinants and eigenvalues). Without trying to teach you matrix algebra, it is useful to know at least a little bit about some of these mathematical operations, and what they are used for in social network analysis.

UCINET has built-in functions for doing most matrix algebra functions. Look under the *Tools>Matrix Algebra*

menu. If you do know some matrix algebra, you will find that this tool lets you do almost anything to matrix data that you may desire. But, you do need to know what you are doing. The help screen for this command shows how to identify the matrix or matrices that are to be manipulated, and the algorithms that can be applied.

Transposing a matrix

This simply means to exchange the rows and columns so that i becomes j , and *vice versa*. If we take the transpose of a directed adjacency matrix and examine its row vectors (you should know all this jargon by now!), we are looking at the sources of ties directed at an actor. The degree of similarity between an adjacency matrix and the transpose of that matrix is one way of summarizing the degree of symmetry in the pattern of relations among actors. That is, the correlation between an adjacency matrix and the transpose of that matrix is a measure of the degree of reciprocity of ties (think about that assertion a bit). Reciprocity of ties can be a very important property of a social structure because it relates to both the balance and to the degree and form of hierarchy in a network. This command is also available as [Data>Transpose](#).

Taking the inverse of a matrix

This is a mathematical operation that finds a matrix which, when multiplied by the original matrix, yields a new matrix with ones in the main diagonal and zeros elsewhere (which is called an identity matrix). Without going any further into this, you can think of the inverse of a matrix as being sort of the "opposite of" the original matrix. Matrix inverses are used mostly in calculating other things in social network analysis. They are sometimes interesting to study in themselves, however. It is sort of like looking at black lettering on white paper versus white lettering on black paper: sometimes you see different things. Inverses are calculated with [Tools>Matrix Algebra](#).

Matrix addition and matrix subtraction

These are the easiest of matrix mathematical operations. One simply adds together or subtracts each corresponding i, j element of the two (or more) matrices. Of course, the matrices that this is being done to have to have the same numbers of i and j elements (this is called "conformable" to addition and subtraction) - and, the values of i and j have to be in the same order in each matrix.

Matrix addition and subtraction are most often used in network analysis when we are trying to simplify or reduce the complexity of multiplex (multiple relations recorded as separate matrices or slices) data to simpler forms. If I had a symmetric matrix that represented the tie "exchanges money" and another that represented the relation "exchanges goods" I could add the two matrices to indicate the intensity of the exchange relationship. Pairs with a score of zero would have no relationship, those with a "1" would be involved in either barter or commodity exchange, and those with a "2" would have both barter and commodity exchange relations. If I subtracted the "goods" exchange matrix from the "money exchange" matrix, a score of -1 would indicate pairs with a barter relationship; a score of zero would indicate either no relationship or a barter and commodity tie; a score of +1 would indicate pairs with only a commodified exchange relationship. For different research questions, either or both approaches might be useful. [Tools>Matrix Algebra](#) are one way of doing these sorts of data transformations.

Matrix multiplication and Boolean matrix multiplication

Matrix multiplication is a somewhat unusual operation, but can be very useful for the network analyst. You will have to be a bit patient here. First we need to show you how to do matrix multiplication and a few important results (like what happens when you multiply an adjacency matrix times itself, or raise it to a power). Then, we will try to explain why this is useful.

To multiply two matrices, they must be "conformable" to multiplication. This means that the number of rows in the first matrix must equal the number of columns in the second. Usually network analysis uses adjacency matrices, which are square, and hence, conformable for multiplication. Multiplying a matrix by itself (i.e. raising it to a power) and multiplying a square matrix by its transpose are obviously "conformable." Unlike regular multiplication of individual numbers $X*Y$ is not the same thing as $Y*X$ in matrix multiplication -- the order matters!

To multiply two matrices, begin in the upper left hand corner of the first matrix, and multiply every cell in the first row of the first matrix by the values in each cell of the first column of the second matrix, and sum the results. Proceed through each cell in each row in the first matrix, multiplying by the column in the second. To perform a Boolean matrix multiplication, proceed in the same fashion, but enter a zero in the cell if the multiplication product is zero, and one if it is not zero. An example helps. Suppose we wanted to multiply the two matrices in figure 5.10.

Figure 5.10. Two matrices to be multiplied.

0	1
2	3
4	5

times

6	7	8
9	10	11

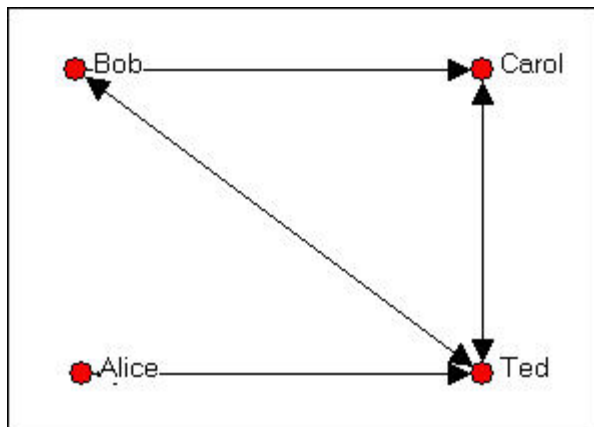
The result is shown in figure 5.11.

Figure 5.11. Result of matrix multiplication.

$(0*6)+(1*9)$	$(0*7)+(1*10)$	$(0*8)+(1*11)$
$(2*6)+(3*9)$	$(2*7)+(3*10)$	$(2*8)+(3*11)$
$(4*6)+(5*9)$	$(4*7)+(5*10)$	$(4*8)+(5*11)$

The mathematical operation in itself doesn't interest us here (any number of programs can perform matrix multiplication). But, the operation is useful when applied to an adjacency matrix. Consider our four friends again, in figure 5.12.

Figure 5.12. Directed graph of friendship relations among Bob, Carol, Ted, and Alice



The adjacency matrix for the four actors B, C, T, and A (in that order) is shown as figure 5.13.

Figure 5.13. Adjacency matrix for graph in figure 5.12.

---	1	1	0
0	---	1	0
1	1	---	1
0	0	1	---

Another way of thinking about this matrix is to notice that it tells us whether there is a path from each actor to each actor. A one represents the presence of a path, a zero represents the lack of a path. The adjacency matrix is exactly what its name suggests -- it tells us which actors are adjacent, or have a direct path from one to the other.

Now suppose that we multiply this adjacency matrix times itself (i.e. raise the matrix to the 2nd power, or square it). We will treat "self-ties" as zeros, which, effectively, ignores them. The calculation of the matrix squared is shown as figure 5.14.

Figure 5.14. Squaring matrix 5.13.

$(0*0)+(1*0)+(1*1)+(0*0)$	$(0*1)+(1*0)+(1*1)+(0*0)$	$(0*1)+(1*1)+(1*0)+(0*1)$	$(0*0)+(1*0)+(1*1)+(0*0)$
$(0*0)+(0*0)+(1*1)+(0*0)$	$(0*1)+(0*0)+(1*1)+(0*0)$	$(0*1)+(0*1)+(1*0)+(0*1)$	$(0*0)+(0*0)+(1*1)+(0*0)$
$(1*0)+(1*0)+(0*1)+(1*0)$	$(1*1)+(1*0)+(0*1)+(1*0)$	$(1*1)+(1*1)+(0*0)+(1*1)$	$(1*0)+(1*0)+(0*1)+(1*0)$
$(0*0)+(0*0)+(1*1)+(0*0)$	$(0*1)+(0*0)+(1*1)+(0*0)$	$(0*1)+(0*1)+(1*0)+(0*1)$	$(0*0)+(0*0)+(1*1)+(0*0)$

equals:

1	1	1	1
1	1	0	1
0	1	3	0
1	1	0	1

This matrix (i.e. the adjacency matrix squared) counts the number of pathways between two nodes that are of length two. Stop for a minute and verify this assertion (go back to the graph and find the paths). For example, note that actor "B" is connected to each of the other actors by a pathway of length two; and that there is no more than one such pathway to any other actor. Actor T is connected to himself by pathways of

length two, three times. This is because actor T has reciprocal ties with each of the other three actors. There is no pathway of length two from T to B (although there is a pathway of length one).

So, the adjacency matrix tells us how many paths of length one are there from each actor to each other actor. The adjacency matrix squared tells us how many pathways of length two are there from each actor to each other actor. It is true (but we won't show it to you) that the adjacency matrix cubed counts the number of pathways of length three from each actor to each other actor. And so on...

If we calculated the Boolean product, rather than the simple matrix product, the adjacency matrix squared would tell us whether there was a path of length two between two actors (not how many such paths there were). If we took the Boolean squared matrix and multiplied it by the adjacency matrix using Boolean multiplication, the result would tell us which actors were connected by one or more pathways of length three. And so on...

Now, finally: why should you care?

Some of the most fundamental properties of a social network have to do with how connected the actors are to one another. Networks that have few or weak connections, or where some actors are connected only by pathways of great length may display low solidarity, a tendency to fall apart, slow response to stimuli, and the like. Networks that have more and stronger connections with shorter paths among actors may be more robust and more able to respond quickly and effectively. Measuring the number and lengths of pathways among the actors in a network allow us to index these important tendencies of whole networks.

Individual actor's positions in networks are also usefully described by the numbers and lengths of pathways that they have to other actors. Actors who have many pathways to other actors may be more influential with regard to them. Actors who have short pathways to more other actors may be more influential or central figures. So, the number and lengths of pathways in a network are very important to understanding both individual's constraints and opportunities, and for understanding the behavior and potentials of the network as a whole.

There are many measures of individual position and overall network structure that are based on whether there are pathways of given lengths between actors, the length of the shortest pathway between two actors, and the numbers of pathways between actors. Indeed, most of the basic measures of networks, measures of centrality and power, and measures of network groupings and substructures are based on looking at the numbers and lengths of pathways among actors.

For most analyses, you won't have to manipulate matrices -- UCINET and other programs have already built algorithms that have the compute do these operations. Most of the computational work in network analysis is done with matrix mathematics though, so in order to understand what is going on, it's useful to understand the basics.

[table of contents](#)

Summary

Matrices are collections of elements into rows and columns. They are often used in network analysis to represent the adjacency of each actor to each other actor in a network. An adjacency matrix is a square actor-by-actor ($i=j$) matrix where the presence of pair wise ties are recorded as elements. The main diagonal, or "self-tie" of an adjacency matrix is often ignored in network analysis.

Sociograms, or graphs of networks can be represented in matrix form, and mathematical operations can then be performed to summarize the information in the graph. Vector operations, blocking and partitioning, and

matrix mathematics (inverses, transposes, addition, subtraction, multiplication and Boolean multiplication), are mathematical operations that are sometimes helpful to let us see certain things about the patterns of ties in social networks.

Social network data are often multiplex (i.e. there are multiple kinds of ties among the actors). Such data are represented as a series of matrices of the same dimension with the actors in the same position in each matrix. Many of the same tools that we can use for working with a single matrix (matrix addition and correlation, blocking, etc.) Are helpful for trying to summarize and see the patterns in multiplex data.

Once a pattern of social relations or ties among a set of actors has been represented in a formal way (graphs or matrices), we can define some important ideas about social structure in quite precise ways using mathematics for the definitions. In the remainder of the book, we will look at how social network analysts have formally translated some of the core concepts that social scientists use to describe social structures.

[table of contents](#)

Review questions

1. A matrix is "3 by 2." How many columns does it have? How many rows?
2. Adjacency matrices are "square" matrices. Why?
3. There is a "1" in cell 3,2 of an adjacency matrix representing a sociogram. What does this tell us?
4. What does it mean to "permute" a matrix, and to "block" it?

Application questions

1. Think of the readings from the first part of the course. Did any studies present matrices? If they did, what kinds of matrices were they (that is, what is the technical description of the kind of graph or matrix). Pick one article, and show what the data would look like, if represented in matrix form.
2. Think of some small group of which you are a member (maybe a club, or a set of friends, or people living in the same apartment complex, etc.). What kinds of relations among them might tell us something about the social structures in this population? Try preparing a matrix to represent one of the kinds of relations you chose. Can you extend this matrix to also describe a second kind of relation? (e.g. one might start with "who likes whom?" and add "who spends a lot of time with whom?").
3. Using the matrices you created in the previous question, does it make sense to leave the diagonal "blank," or not, in your case? Try permuting your matrix, and blocking it.
4. Can you make an adjacency matrix to represent the "star" network? what about the "line" and "circle." Look at the ones and zeros in these matrices -- sometimes we can recognize the presence of certain kinds of social relations by these "digital" representations. What does a strict hierarchy look like? What does a population that is segregated into two groups look like?

[table of contents](#)

[table of contents of the book](#)