Pathfinder Associative Networks: Studies in Knowledge Organization

ABLEX SERIES IN COMPUTATIONAL SCIENCES

Derek Partridge, University of Exeter

Series Editor

Pathfinder Associative Networks: Studies in Knowledge Organization
Roger W. Schvaneveldt (Editor)

In Preparation

A New Guide to Artificial Intelligence
Derek Partridge

Artificial Intelligence and Software Engineering
Derek Partridge (Editor)

Binding Time – Six Studies in Programming Technology and Milieu
Mark Halpern

Artificial Intelligence and Business Management
Derek Partridge and K.M. Hussain

Intelligent Systems
Eric Dietrich and Chris Fields

Database in Practice Rather Than in Theory
K.G. Jeffrey

New Generation Architectures and Languages
Stephen J. Turner

Computer Analysis of English: Lexical Semantics and Preference Semantics Analysis
Brian Slator

Pathfinder Associative Networks: Studies in Knowledge Organization

Edited by

Roger W. Schvaneveldt
Department of Psychology and
Computing Research Laboratory
New Mexico State University

ABLEX PUBLISHING CORPORATION
Norwood, New Jersey 07648

Printed in the United States of America.

Library of Congress Cataloging-in-Publication Data

Pathfinder associative networks : studies in knowledge organization / edited by Roger W. Schvaneveldt.

p. cm. – (Ablex series in computational sciences)
Includes bibliographical references.
ISBN 0-89391-624-2
1. Expert systems (Computer science). 2. Artificial intelligence. 3. Graph theory.
4. Knowledge acquisition (Expert systems).
I. Schvaneveldt, Roger W. II. Series.
QA76.76.E95P28 1989
006.3'3–dc20 89-18219
CIP

ABLEX Publishing Corporation
355 Chestnut Street
Norwood, New Jersey 07648

Contents

Contributors

Russell J. Branaghan
Department of Psychology and
Computing Research Laboratory
New Mexico State University
Las Cruces, NM 88003

Timothy J. Breen
Knowledge Systems Laboratory
Boeing Computer Services
P.O. Box 24346, MS 7L-64
Seattle WA 98124-0346

Kathy A. Coggins
Department of Psychology
University of Oklahoma
Norman, OK 73019

Nancy Jaworski Cooke
Department of Psychology
Rice University
Box 1892
Houston, TX 77251

Daniel M. Davenport
P. O. Box 5800
Sandia National Laboratories
Albuquerque, NM 87185

Tom Dayton
Department of Psychology
University of Oklahoma
Norman, OK 73019

Donald W. Dearholt
Department of Computer Science and
Computing Research Laboratory
New Mexico State University
Las Cruces, NM 88003

Francis T. Durso
Department of Psychology
University of Oklahoma
Norman, OK 73019

Chris Esposito
Boeing Advanced Technology Center
P.O. Box 24346 MS 7L-64
Seattle, WA 98124

Richard H. Fowler
Department of Mathematics and
Computer Science
Pan American University
Edinburg, TX 78539

John G. Gammack
MRC Applied Psychology Unit
15 Chaucer Road
Cambridge, CB2 2EF
England

Timothy E. Goldsmith
Department of Psychology
University of New Mexico
Albuquerque, NM 87131

Peder J. Johnson
Department of Psychology
University of New Mexico
Albuquerque, NM 87131

Wendy A. Kellogg
User Interface Institute
IBM Thomas J. Watson Research
Center
P. O. Box 704
Yorktown Heights, NY 10598

Deborah R. McDonald
Department of Psychology
New Mexico State University
Las Cruces, NM 88003

James E. McDonald
Department of Psychology and
Computing Research Laboratory
New Mexico State University
Las Cruces, NM 88003

Lisa A. Onorato
Department of Psychology
Hartwick College
Oneonta, NY 13820

Kenneth R. Paap
Department of Psychology and
Computing Research Laboratory
New Mexico State University
Las Cruces, NM 88003

Tony A. Plate
Department of Computer Science
University of Toronto
Toronto M5S 1A1
Canada

Renate J. Roske-Hofstrand
Aerospace Human Factors Research
Division
Nasa-Ames Research Center
Mailstop 239-21
Moffett Field, CA 94035

David C. Rubin
Department of Psychology
Duke University
Durham, NC 27706

Roger W. Schvaneveldt
Department of Psychology and
Computing Research Laboratory
New Mexico State University
Las Cruces, NM 88003

Jack D. Shepard
Department of Psychology
University of Oklahoma
Norman, OK 73019

Preface

The chapters in this book represent a sampling of theoretical, empirical, and applied work with *Pathfinder networks*. These networks began in 1981 (Schvaneveldt & Durso, 1981) as an attempt to develop a *network* model for *proximity* data. The intervening years have seen several developments of that original work. A theoretical paper relating Pathfinder networks to fundamental concepts in graph theory (Schvaneveldt, Dearholt, & Durso, 1988) grew out of a conference organized by Frank Harary and Keith Phillips. The chapters in this book represent a wide range of applications for network models.

The original motivation for developing Pathfinder grew out of our realization that although network representations abound in theoretical work in cognitive psychology and artificial intelligence, there were few methods for arriving at a network representation from empirical data. Proximity data offer a convenient starting point for networks. Indeed, proximity data serve as the building block for several interesting structural models such as multidimensional scaling (MDS) and cluster analysis. Essentially, Pathfinder networks are determined by identifying the proximities that provide the most efficient connections between the entities by considering the indirect connections provided by *paths* through other entities. The resulting networks have several interesting properties (see Chapter 1), and they have also proven to be useful in a variety of applications. There are now various algorithms for deriving PFNETs in several computer languages running on several different computers.[1]

There are a few features of this book that should be helpful to readers without much background knowledge of graph theory. A brief primer on graph theory and Pathfinder and a glossary can be found at the back of the book. References from all of the chapters are compiled in a single reference section at the back of the book. Chapter 1 reviews some definitions and properties of Pathfinder networks as well as some algorithms for deriving these networks from proximity data. Chapter 2 is a general review of empirical work with Pathfinder in cognitive modeling and an exploration of potential applications in social networks. The other chapters relate to several major themes.

Chapters 3, 4, and 5 address some methodological issues. Esposito (Chapter 3) develops and evaluates a version of Pathfinder that takes variability of proximity data into account. Roske-Hofstrand and Paap (Chapter 4) analyze some properties of proximity data obtained by ratings and the implications for Pathfinder networks. Goldsmith and Davenport (Chapter 5) present some measures of the similarity of two networks.

Chapters 6 through 10 report investigations of some basic phenomena in human memory. Esposito (Chapter 6) analyzes the relation between human judgments of the goodness of categories and various formal characteristics of graphs. Cooke (Chapter 7) examines the time required to judge that two concepts belong to the same category. Branaghan (Chapter 8) analyzes the ease with which lists of associations are learned. Rubin (Chapter 9) investigates the strategies people use to search memory. Schvaneveldt (Chapter 10) examines the representation of schemata in Pathfinder and connectionist style networks.

---

[1]Programs have been written in Pascal, C, LISP, and APL. Various versions of the programs run on IBM PC, Apple Macintosh, and SUN Microsystems computers. Information on obtaining programs is available from: Interlink, Inc., P.O. Box 4086 UPB, Las Cruces, NM 88003-4086.

Chapters 11 through 16 address applications of Pathfinder to problems in knowledge elicitation, information retrieval, and interface design. McDonald, Plate, and Schvaneveldt (Chapter 11) extract associative information from text and use this information to resolve word ambiguity. Fowler and Dearholt (Chapter 12) address the classic problem of retrieving information from large collections as in libraries. Kellogg and Breen (Chapter 13) compare the models of systems to mental models of users. McDonald, Paap, and McDonald (Chapter 14) attack the problem of establishing connections in Hypertext. Gammack (Chapter 15) analyzes the use of different techniques for eliciting proximity information from an expert. Cooke (Chapter 16) develops a method for identifying the nature of the relations between linked concepts in a network.

Chapters 17, 18, and 19 are concerned with still other aspects of knowledge representation. Goldsmith and Johnson (Chapter 17) investigate the use of networks and MDS spaces to assess classroom learning. Onorato (Chapter 18) analyzes the ways in which people organize information depending on the purpose of the information. Dayton, Durso, and Shepard (Chapter 19) examine the differences in the way solvers and nonsolvers organize problem-relevant information.

Obviously, there are many interrelations among the various chapters. As an aid to seeing these relations and as an initial illustration of the use of Pathfinder, I constructed Figure 1. This figure shows a Pathfinder network depicting the close associations among the chapters.

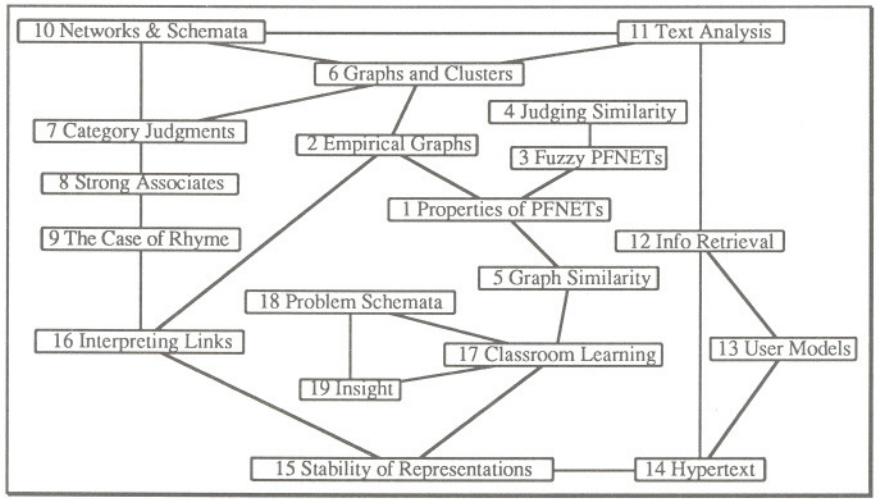


Figure 1. The modified PFNET($r = \infty$, $q = n$–1) for the chapters in this book.

To construct Figure 1, I first made a list of the three chapters most closely related to each chapter. This ordered list of associations was used to construct a matrix of proximity data where an entry was 1, 2, or 3 if the chapter on the column was the first , second, or third most associated with the chapter on the row. Other entries were treated as infinite. This matrix was non-symmetrical and the Pathfinder network that resulted from analyzing the matrix had directed links. However, I was not able to interpret the directions of the links so I made all of the links undirected as shown in the figure.

This figure can be used to find chapters that are closely related to other chapters. Several groups of interrelated chapters can be identified in addition to the one I used to order the chapters. It is obviously impossible to capture all of these relations in the linear ordering enforced on a book.

The development of Pathfinder and much of the research reported in this book have been supported by the National Science Foundation (IST-8506706), the Air Force Human Resources Laboratory (F33615-84-C-0072 and F33615-80-C-0004), Texas Instruments, Inc., and the National Aeronautic and Space Administration (NAG 2-453). Such support has been invaluable in the development of the methods and research.

I gratefully acknowledge the assistance of several people in assembling this book. Derek Partridge encouraged me to undertake the project in the first place. Most of the authors reviewed one or more chapters in addition to writing their own. Douglas Nelson and David Farwell also provided very helpful reviews. My associates here at New Mexico State were invaluable in their assistance with all of the details and in the defense of conceptual coherence. I am particularly grateful to Bob Fiegel, Tarra Fiegel, Rebecca Gomez, and Paula Moreland for their help. Thanks also to my wife, Ann, and daughter, Susan, for their love and support.

R. Schvaneveldt
April 11, 1989
Las Cruces, New Mexico

Chapter 1

Properties of Pathfinder Networks

Donald W. Dearholt and Roger W. Schvaneveldt

Network models have played important roles in various areas of cognitive science and computer science. In cognitive psychology and artificial intelligence, network representations of concepts stored in semantic memory have been used in models of memory retrieval and human performance (e.g., Anderson, 1983; Collins & Loftus, 1975; Collins & Quillian, 1969; Friendly, 1977, 1979; Meyer & Schvaneveldt, 1976; Rumelhart & McClelland, 1986), scene description and analysis (e.g., Brooks & Binford, 1980; Waltz, 1972; Winston, 1970), natural-language processing (e.g., Bobrow & Webber, 1980; Fillenbaum & Rapoport, 1971; Kintsch, 1974; Quillian, 1967, 1969; Schank, 1972; Woods, Kaplan, & Nash-Webber, 1972), and knowledge representation (e.g., Brachman, 1977, 1979; Fahlman, 1979; Fikes & Hendrix, 1977; Griffith, 1982; Novak, 1977; Schmolze & Lipkis, 1983; Sowa, 1984; Woods, 1975).

In database systems, a network data model often results in efficient representations of sets of concepts (Date, 1981; Ullman, 1982). Thus far, the network model incorporated in database systems has been constructed with two primary objectives: providing efficient data access for the anticipated user environment and making the most of the rather severe limitations imposed by present computer operations and architecture. Although the network data model used most frequently in database models (CODASYL, 1971) can support abstractions of essentially any type, there are constraints (for the purpose of modularity, simplicity of definition, and hardware support) that must be circumvented by artificial programming devices. Networks identifying relationships between data items have been proposed for designing the logical schema of a database system (e.g., see Martin, 1977, Chapter 6) by means of bubble charts. The bubble charts are used to indicate relationships between data items (e.g., functional dependencies, primary keys, and secondary keys). The bubble charts are usually viewed as an intermediate step in the development of a logical schema. Clustering strategies for data items have been investigated and proposed for improving expected retrieval time, based on the estimated likelihood of retrieval of data items contingent upon the retrieval of other data items (e.g., Navathe & Fry, 1976; Schkolnick, 1977).

Recently developed techniques from our laboratory and elsewhere allow researchers to derive networks from the same proximity data employed by multidimensional scaling (Dearholt, Schvaneveldt, & Durso, 1985; Hutchinson, 1989; Schvaneveldt, Dearholt, & Durso, 1988; Schvaneveldt & Durso, 1981; Schvaneveldt, Durso, & Dearholt, 1989).

Networks and Proximity Data

Hutchinson's NETSCAL procedure (Hutchinson, 1981, 1989), which makes *ordinal* data assumptions, is based on a theorem of Hakimi and Yau (1964) regarding the *distance* matrix of a *graph* and its realizability. The distance metric used by Hakimi and Yau is the

sum of *edge weights* along a path, so that the distance between *nodes* is the (minimum) sum of the weights (distances) of the edges along a path between the nodes. This measure of *path length* is appropriate for *ratio*-scale data. Hutchinson, however, also used a distance metric in which the distance between nodes is the smallest maximum weight along any of the paths between the two nodes. This path-length measure is appropriate for ordinal as well as ratio-scale data. A serious shortcoming of Hutchinson's work is that his Corollary I considers *triangle inequalities* of only two-link paths. That is, the triangle inequality can be violated in paths having three or more links in Hutchinson's networks. This seems to be an unfortunate limitation, inappropriate for the scaling of data, and perhaps also for cognitive modeling, although psychological proximity may not always obey the triangle inequality (Tversky, 1977; Tversky & Gati, 1982).

The triangle inequality can be viewed in three different domains. The first, and the sparsest, is in euclidean space, as addressed by Hakimi and Yau (1964), in which the triangle inequality must always be satisfied. The second is the class of problems in which measures of similarity or "distance" are measured objectively by set intersections; for most such problems, there is no expectation of transitivity holding, so that there is likewise no expectation that the triangle inequality will be satisfied, either. That is, if we know the intersections of sets A and B, and of B and C, we do not generally know anything about the intersection of sets A and C. The information retrieval application to be discussed in detail in the chapter by Fowler and Dearholt (Chapter 12, this volume) is an example of such a problem in which the triangle inequality may be violated. The third domain is that of subjective estimates of similarity, in which data frequently show violations of the triangle inequality. Philosophically, it is attractive to use geodetic distance measures, in which the distance between each pair of nodes is considered to be the length of the shortest path available between those nodes; indeed, in graph theory this has been the usual definition of distance. Then, a violation of some triangle inequality is never a part of a path between any pair of nodes, because a shorter alternative path is always available. Thus the omission of the edges which violate some triangle inequality in a network assures the preservation of the (geodetic) distances between all pairs of nodes and provides a simpler structure which possesses precisely those edges which are responsible for the most economical paths (Schvaneveldt, Dearholt, & Durso, 1988).

The links that are omitted include those due to differences on two or more separable dimensions, in which the triangle inequality is expected to be violated, as discussed by Tversky and Gati (1982). That is, if A and B are judged to be similar because of feature x, and B and C are judged to be similar because of feature y, then A and C will normally be judged to be less similar than the triangle inequality would indicate. Thus if salient associations are linked in a graph paradigm, the absence of a *link* can denote a difference in the basis for judged similarity.

We have developed a procedure called Pathfinder (several equivalent procedures, actually) to generate a class of networks called PFNETs, which are based on estimates or measures of distances between pairs of entities. This procedure allows a spectrum of assumptions to be made about the data, including ordinal and ratio properties. The data required are either similarities or distances. Similarities can be obtained either from a subject's estimates of the similarity of each pair of entities in the set or from a measure of set intersections. Distances can be obtained in some domains by estimating or computing appropriate differences between all pairs of entities. The result of the Pathfinder procedure is a network which is either a directed graph (if the similarity or distance matrix is not symmetrical) or an undirected graph otherwise. Each entity in the set is represented as a node in the network, and each link that is entered in the network has a weight value determined

by the distance between the two entities so linked. Our network generation procedure incorporates two parameters. The first, the Minkowski r-metric, determines how distance between two nodes not directly linked is computed. The weight of a path with weights w_1, w_2,..., w_k is:

$$W(P)=\left(\sum_{i=1}^{k} w_i^r\right)^{1/r}$$

For $r = 1$, the path weight is the sum of the link weights along the path; for $r = 2$, the path weight is computed as euclidean distance is computed; and for $r = \infty$, the path weight is the same as the maximum weight associated with any link along the path. We will use "distance" in this chapter to mean the Minkowski distance (geodetic), which depends upon the value of the r-metric. The second parameter is the *q parameter*, which is a limit on the number of links in the paths examined in constructing a network. Its value determines the maximum number of links in paths in which the triangle inequalities are guaranteed to be satisfied in the resulting network. Our procedure generates families of PFNETs, and we can generate Hutchinson's (1989) networks as a special case with $r = \infty$ and $q = 2$.

The links omitted from a PFNET are omitted because they violate a triangle inequality involving q or fewer links. These omissions preserve all (geodetic) distances from the original data, however, and because not all links are present in most PFNETs, structural features are easier to ascertain. If a distance between two nodes not directly linked must be computed from the PFNET, it is computed using the Minkowski metric, resulting in a computed distance less than that given explicitly in the original data.

Advantages of PFNETs include (1) the capability of directly modeling asymmetrical relationships (Hutchinson, 1981; Tversky, 1977), which is more difficult with multidimensional scaling (Constantine & Gower, 1978; Harshman, Green, Wind, & Lundy, 1982; Krumhansl, 1978); (2) the provision for a complementary alternative to multidimensional scaling which often provides a more accurate representation of local data relationships than does multidimensional scaling; since multidimensional scaling must move data points to minimize a global error criterion, the resulting relationships between neighboring points is often significantly different than the original data would indicate; (3) the fact that hierarchical constraints in most cluster analysis techniques do not apply to PFNETs; (4) the representation of the most "salient" relationships present in the data; (5) the provision for a new paradigm in studying models of classification; and (6) the provision for a more quantitative paradigm for some of the issues in which networks have been invoked qualitatively or designed intuitively.

From the viewpoint of cognitive modeling, a disadvantage of PFNETs in the present state of development is that we have no way of knowing the features upon which similarity judgments are made. Thus the semantic content of links is not easily discernible (but see Cooke, Chapter 16, this volume). The empirical data we have collected, however, should be viewed as similarity estimates having components which may be unknown; but the use of such data seems important in bridging the gap between the more standard semantic networks (in which the researcher labels links according to his preferences or beliefs at the time) and a more objective representation of the knowledge of interest. For domains in which objective measures of distance are available, PFNETs provide unique representations of underlying structure not obtainable from any other scaling method.

Definitions and Alternatives

In this section we present definitions to provide the proper foundation for the generation procedures and theorems that follow. A PFNET has n nodes, denoted $N_1, N_2, ..., N_n$ (or $N_a, N_b, ...$). A *link* is an association between a pair of nodes which can be either undirected or directed. A *directed link* is called an *arc*, and an *undirected link* is called an *edge*. In this chapter we will deal mainly with networks having undirected links, or edges, but some of the definitions are more general, and a few examples of directed networks will be given to illustrate this generality. In this spirit, links are labeled e_{ij}, for the edge between N_i and N_j (or for the arc from N_i to N_j). N_i and N_j are *end nodes* of the link e_{ij}. The distance from node N_i to N_j (along the link e_{ij}) is the weight w_{ij}, and these weights are often written in matrix form as an $n \times n$ matrix W. The elements of W can be considered as distances between nodes along the direct paths between every pair of nodes. The distances are often considered as dissimilarities, and W is called either the *adjacency* or *weight* matrix. We assume that $w_{ii} = 0$ and $w_{ij} > 0$ for $1 \leq i, j \leq n$ where $i \neq j$. If this matrix is symmetric, then a PFNET derived from it is an undirected network. Typically the distance measures (weights) for each pair of entities (nodes) are found either empirically, from similarity estimates by human subjects, or analytically, using some appropriate measure of set intersection and set union, or some distance metric between entities.

A path from node N_a to node N_e, passing through nodes N_b, N_c, and N_d, is denoted by P_{abcde} (if the intermediate nodes are important) or P_{ae} otherwise. The former presumes the existence of edges e_{ab}, e_{bc}, e_{cd}, and e_{de} (either undirected or with appropriate directions), whereas P_{ae} presumes the existence of some unspecified set of edges (or arcs) connecting N_a and N_e. The weight of a path P is denoted $W(P_{ae})$, and the function $W(P)$ is determined by the r-metric and the weights w_{ij}.

The triangle inequality is incorporated into our generation procedure by means of the q parameter.

Definition 1

A network is q-triangular if and only if all possible triangle inequalities involving paths with $m \leq q$ links are satisfied, using links and weights in the graph and the r-metric chosen. An example is the triangle inequality

$$w_{ae} \leq \left(w_{ab}^{r} + w_{bc}^{r} + \ldots + w_{de}^{r} \right)^{1/r}$$

which is a constraint on the weights of two alternate paths between nodes N_a and N_e. For a graph with n nodes, there can be at most n–1 edges in any path in which there is no *cycle*. Thus the q parameter is at most n–1. Geodetic distances in the network are unchanged if edges which would violate triangle inequalities are omitted.

Definition 2

The (geodetic) network distance d_{ij} between nodes N_i and N_j is computed as a function of all path weights $W(P_{ij})$, for all paths P_{ij} which connect nodes N_i and N_j as

$$D_{ij} = \text{MIN} \left(W(P_{ij1}), W(P_{ij2}), \ldots, W(P_{ijm}) \right)$$

That is, the distance between two nodes is the weight of the smallest path between those nodes, with all path weights calculated using the (same) appropriate r-metric.

The r-metric and the q parameter provide the elements needed to assure that the networks generated from a particular set of proximity data possess the metric properties discussed in Hakimi and Yau (1964), with the following provisions:

1. The distance from a node to itself is assumed to be zero.
2. The data matrix must be symmetric so that the PFNET is undirected; then the distance between any pair of nodes is independent of direction.
3. The triangle inequality is satisfied for all paths having as many as q edges. To assure that no triangle inequalities whatsoever are violated, q can be set to the number of nodes less one.

For situations in which these metric axioms are satisfied, the concept of distance along a path is the same as the weight of that path. Because the r-metric can take on values from one through infinity, and the q parameter can take on values from one through the number of nodes less one, many different PFNETs can be constructed from a given set of proximity data. However, different values of r and q can result in the generation of the same (*isomorphic*) PFNETs. Frequently, important information from a given set of proximity data can be obtained from different PFNETs, constructed using different values of r and q. Thus it is often not essential that particular choices for r and q be made, to the exclusion of other values. Furthermore, it is sometimes desirable (in cognitive modeling, for example) to violate the metric axioms presented above (also in Hakimi & Yau, 1964; and in Tversky & Gati, 1982). The possibility of constructing directed PFNETs from asymmetric proximity data and (independently) of varying the q parameter provide ways of violating these axioms which correspond to observations about human performance (see, for example, Ortony, 1979; Tversky, 1977; and Tversky & Gati, 1982). Modeling traffic flow on one-way streets provides another example in which asymmetric data are relevant.

Definition 3

A *PFNET(r, q)* is a septuple $(N, E, W, LLR, LMR, r, q)$ in which:

N is the set of nodes (concepts), denoted N_i;

E is a square matrix representing names of links in the *complete graph* (i.e., e_{ij} is the name of the link connecting nodes N_i and N_j);

W is the square weight matrix, and its entries are the weights associated with the links in the corresponding positions of the E matrix. The weights on the main diagonal are assumed to be zero, and the remaining weights are assumed to be finite and nonnegative. Thus w_{ij} is the weight of link e_{ij};

LLR, the link-labeling rule, is the procedure used to determine a label for each link, according to some classification scheme;

LMR, the link-membership rule, is the procedure used to determine whether or not each element of the E matrix is added to the PFNET(r, q);

r is the value of the r-metric, and $1 \leq r \leq \infty$;

q is the value of the q parameter, and $q \in \{1, 2, ..., n-1\}$, where n is the number of nodes.

Definition 4

The link-membership rule (LMR) for PFNETs (either directed or undirected) is given by the following procedure:

1. Define a network consisting of all nodes (concepts) N_i, but no links;
2. Order all elements e_{ij} of the E matrix in some nondecreasing order of their associated weights w_{ij};
3. Consider each e_{ij}, and include e_{ij} in the PFNET(r, q), if and only if e_{ij} provides a path from N_i to N_j which has a weight at least as small as the weight of any other path having no more than q links, using the r-metric to compute the weights of multiple-link paths.

This definition is useful primarily in establishing the concepts associated with Pathfinder networks; computationally efficient algorithms for generating PFNETs will be given in the next section. As an example of the LMR, consider the weight matrix:

$$W = \begin{matrix} 0 & 1 & 4 & 5 \\ 2 & 0 & 2 & 4 \\ 1 & 4 & 0 & 1 \\ 5 & 3 & 1 & 0 \end{matrix}$$

for the nodes N_1, N_2, N_3, and N_4. The complete graph is as shown in Figure 1. The arcs are not labeled because we have not yet developed a labeling rule for directed PFNETs. (Labeling edges with some LLR does not affect the edge membership of an undirected PFNET, because the edges are put there by the LMR, which makes no use of edge labels.)

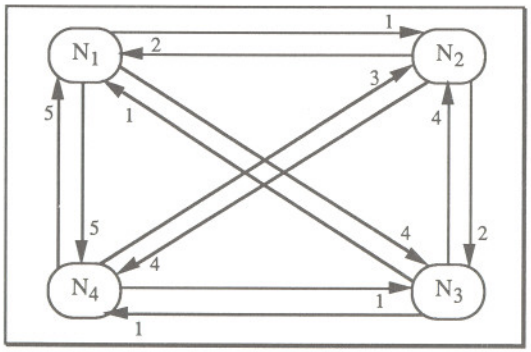


Figure 1. Complete graph for the example.

Let $r = 1$ and $q = 2$. Applying the link membership rule, the PFNET($r = 1$, $q = 2$) shown in Figure 2 is obtained. Note that e_{14} is in the PFNET because its weight ties with the weight of the path P_{124}, even though the arc e_{24} is not itself in the PFNET; if it were in the PFNET, it would violate the triangle inequality for the alternative path P_{234}. The path P_{1234} has less weight, but is not considered because it has three arcs, and for this example we assumed $q = 2$. The PFNET in Figure 2 is two-triangular, since the q parameter is two.

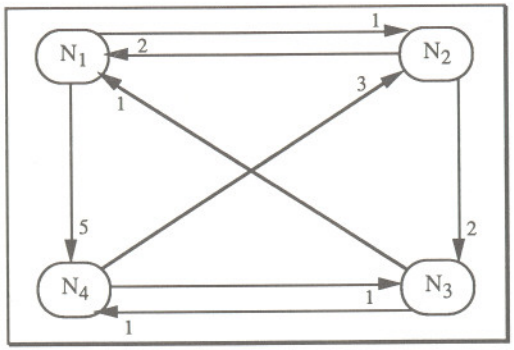


Figure 2. The PFNET($r = 1$, $q = 2$) corresponding to the complete graph of Figure 1.

Generation Algorithms for PFNETs

A part of our generation procedure, for either directed or undirected PFNETs, requires matrix operations using the weight matrix W. The purpose of these operations is to determine which links providing alternate paths are in the PFNET. For either directed or undirected PFNETs, the matrix operations can be used to determine link membership. These matrix operations find the minimum-weight path(s) having $i \leq q$ links between every pair of nodes, and finally, the minimum-weight path(s) having no more than q edges. The matrix computations are determined partly by the r-metric.

Definition 5

$W^{i+1} = W \otimes W^i$ is computed as follows:

$$w_{jk}^{i+1} = \text{MIN}\left[(w_{jm})^r + (w_{mk}^{\,i})^r\right]^{1/r} \quad \text{for } 1 \leq m \leq n$$

$$\text{where } w_{jm} \geq 0 \text{ and } w_{mk}^{\,i} \geq 0.$$

Definition 6

The minimum-distance matrix for paths not exceeding i links is denoted D^i, and its elements are computed as follows:

$$d_{jk}^{i} = \text{MIN}\left(w_{jk}^{1}, w_{jk}^{2}, \cdots, w_{jk}^{i}\right) \text{for } j \neq k$$

If the weight matrix W is asymmetric, then the corresponding PFNET(r, q) is directed. The generation procedure we have been using for such data does not label the arcs (a suitable labeling rule is under development). Thus the matrix operations described in definitions 5 and 6 provide the basis of a link membership rule for either directed or undirected PFNETs. Data required are (1) a weight (distance between nodes) matrix W, either symmetric or asymmetric, and (2) values for the r-metric and the q parameter. The following

procedure removes links violating triangle inequalities as paths involving more links are considered.

Procedure PATHFINDER PFNET(r, q):

(without link labeling, for either symmetric or asymmetric W matrix)

1. Compute $W^2, D^2, W^3, D^3, \ldots, W^q, D^q$;
2. Comparing elements of D^q and W^1, wherever $d_{ij} = w_{ij}$, then mark e_{ij} as a link in PFNET(r, q).

Actually, W^1 is the weight matrix, and D^1 is identical to it, because it is the distance matrix for paths having one link. W^2 must be computed, however, and provides the minimum cost for paths between each pair of nodes having exactly two links. D^2 provides the lower cost of either one-link or two-link paths for each pair of nodes. Similarly, D^3 provides the lowest cost of paths having one, two, or three links, and finally, D^{n-1} (where n is the number of nodes) would provide the lowest cost of paths having any number of links (without cycles) between every pair of nodes. Thus the second step of the procedure assures that every link e_{ij} in PFNET(r, q) provides a path between nodes N_i and N_j, which has a weight as small as any alternative path having from two to as many as q links. As each W^{i+1} and D^{i+1} are computed from W^i and D^i, links may be removed from the PFNET(r, q), but they cannot be added. Those links which are removed are called *redundant* links, because they do not affect any of the distances between nodes (Schvaneveldt, Dearholt, & Durso, 1988).

This procedure, which is an LMR, can be viewed as a means of uncovering the links responsible for the distance measurements between nodes and omitting all the other links; but if there are multiple paths having the same minimum cost, then links responsible for those ties are included, so that there is no arbitrary aspect to the LMR. As the value of the q parameter increases, links are removed as violations of the triangle inequality are discovered in longer and longer paths by the matrix operations. The distance matrix D^i, for $i < n-1$, may incorporate distances in which links not in the network are utilized as a part of the distance computation between a pair of nodes. This can be so only for cases in which there is a shorter alternative path having more than i links. Therefore, when $q = n-1$, every entry in D^{n-1} is computed using links which are in the network; otherwise, it would mean that some path not in the network is shorter than any path between the two nodes of interest composed of links which are in the network. The procedure guarantees that the weights of the shortest paths having q or fewer links are entered in the D^q matrix at each step.

For situations in which the W matrix must be symmetric (such as the information retrieval problem discussed in the chapter by Fowler and Dearholt (Chapter 12, this volume), or may be assumed to be symmetric, we have developed an LLR which yields important structural information. An LLR based on a node-covering paradigm is included as part of a generation procedure which will be given after a few more definitions.

Definition 7

A set of nodes (of an undirected PFNET) called a *node sublist* is a connected *subgraph* (at a stage of development) of a PFNET, and is denoted by NSL. A family of NSLs partitions the set of nodes, and when an edge joins nodes in two different NSLs, the two NSLs merge to form a single NSL, which consists of all of the nodes in the two original NSLs.

Definition 8

Let $E(w_i)$ denote the equivalence class of edges having the weight value w_i. Edges in a given equivalence class are assumed to have equal saliency.

The purpose of considering the NSLs defined by increasing equivalence class (weight) values is to establish the most salient edges first. The NSLs are merged as successive equivalence classes and are considered until the entire graph is connected. The NSLs also provide clustering information, as discussed later in this chapter in the section entitled Pathfinder Networks and Hierarchical Clustering.

The following definitions, concerned with edge labeling, distinguish three ways of incorporating nodes into a PFNET during construction.

Definition 9

For an undirected PFNET, a *primary* edge is, for some equivalence class of edges $E(w_i)$, the only path joining a node in some NSL to a single-node NSL. That is, a primary edge provides the only path between a node (NSL having node size one) and some other NSL for a particular equivalence class. A primary edge is labeled PRI. Within a network already constructed, an edge e_{ij} is primary if its weight is smaller than the weight of any other edge *incident* to one or both of the nodes N_i or N_j.

Definition 10

For an undirected PFNET, a *secondary* edge is, for some equivalence class $E(w_i)$, an edge joining node sublists which were distinct before $E(w_i)$ is considered, and in which there are either alternate paths to the end nodes, or the node size of both NSLs exceeds one. A secondary edge is labeled SEC. For primary or secondary edges, no alternative path exists in a smaller equivalence class. The primary edges are seen to provide the lowest cost connections between nodes, and the secondary edges either tie for low cost, or provide paths which connect nodes already connected to other nodes in a smaller (prior) equivalence class. For primary and secondary edges, inclusion into the PFNET is accomplished in stages by forming and merging NSLs much as clusters are merged in a hierarchical clustering scheme.

Definition 11

For an undirected PFNET, a *tertiary* edge is, for some equivalence class $E(w_i)$, a link joining nodes within a single NSL, so that alternate paths existed before the class $E(w_i)$ is considered. A tertiary edge is labeled TER.

For undirected PFNETs, the following more detailed procedure is equivalent to the LMR of Definition 4. This procedure also provides an LLR for the labeling of edges as PRI, SEC, or TER. This LLR has proven helpful in identifying categories and subcategories within the data, and will be related to hierarchical clustering discussed in the section entitled Pathfinder Networks and Hierarchical Clustering later in this chapter. Variations of the procedure have been implemented in APL, LISP, Pascal, and C. The concepts (nodes) N, the weight matrix W, and the values for the r-metric and q parameter are assumed to be given, and W is assumed to be symmetrical. This procedure is called *agglomerative* because it begins with each node or object placed in its own cluster, gradually merging these atomic clusters into larger and larger clusters until all nodes are merged into a single cluster or a *connected graph* (Jain & Dubes, 1988).

Procedure PATHFINDER PFNET(r, q)
(with link labeling, for symmetric matrix W)

1. Define a network consisting of all nodes N_i, but no edges;
2. Partition all edges into equivalence classes $E(w_i)$ according to the values of the weights w_i, and arrange these equivalence classes in increasing order $E(w_i)$, $E(w_j)$, ... according to the weight value of each class;
3. The initial node sublists are the individual nodes themselves, with no edges (equivalent to the *weak clustering* of Johnson, 1967);
4. Construct an *incidence table*, in which the nodes are column headings, and the edges, ordered by equivalence class according to weight values, are row headings. All edges in $E(w_1)$, the equivalence class having the smallest weight value, are listed as the first rows of the table;
5. For each edge e_{ij} in $E(w_1)$, place a check mark in columns N_i and N_j;
6. Any column with exactly one check mark in it identifies a primary edge, which is labeled PRI;
7. All other edges in $E(w_1)$ are labeled SEC;
8. The nodes in each NSL are marked with a symbol designating membership in the appropriate NSL;
9. All edges in $E(w_1)$ are added to the PFNET;

Beginning of Loop:

For each equivalence class $E(w_i)$, taken in increasing order of the weight values, do:

10. If there is only a single NSL (i.e., if all nodes are connected), then all remaining edges which have not been entered into the network as primary or secondary edges are considered as candidates for tertiary edges, and go to Step 15;
11. List the edges in $E(w_i)$ as row headings below the edges in $E(w_{i-1})$, and put check marks in the columns of the end nodes for each edge;
12. Any column with only one check mark throughout, for an edge in $E(w_i)$, identifies a new primary edge which is labeled PRI, and is entered into the network;
13. Any edge in $E(w_i)$ which connects two distinct NSLs, each having more than one node (as determined in $E(w_i)$), is labeled SEC and is entered into the network as a secondary edge (each unlabeled edge is a candidate tertiary edge, connecting nodes within some NSL);
14. The NSLs are relabeled to indicate the merging which has occurred as a result of entering new PRI or SEC edges into the network (note that an NSL can never split—the NSL structure is modified only by PRI or SEC edges merging two NSLs into a new, larger NSL which contains all nodes of both parent NSLs from the prior equivalence class);

End of Loop;

15. Compute W^q and D^q;
16. Wherever w_{ij} (from W^1) = d_{ij} (from D^q) and e_{ij} has not been previously labeled, then label e_{ij} as TER and enter it into the network;

End of Procedure.

The strategy used for tertiary edges (steps 15 and 16 of the procedure) could have been used to determine link membership for the entire PFNET (as in the earlier procedure for either directed or undirected PFNETs), but the disadvantage of this approach is that the procedure to label edges is less clear. Labeling edges after generation of the PFNET requires the use of a shortest-path algorithm. On the other hand, the procedure described above requires the matrix operations anyway, and thus appears cumbersome. From the procedure, it can easily be determined that every PFNET is connected—when the last NSLs are merged, the graph must be connected.

An example of the construction and labeling of an undirected PFNET is given now, with the symmetric weight matrix:

$$W = \begin{matrix} 0 & 1 & 9 & 5 & 9 & 7 \\ 1 & 0 & 1 & 9 & 9 & 9 \\ 9 & 1 & 0 & 2 & 9 & 9 \\ 5 & 9 & 2 & 0 & 1 & 9 \\ 9 & 9 & 9 & 1 & 0 & 1 \\ 7 & 9 & 9 & 9 & 1 & 0 \end{matrix}$$

For computational simplicity in the example, suppose that $r = 1$ and $q = 2$. Considering only edges in the upper triangular matrix because of symmetry, the equivalence classes are:

$E(1)$: e_{12}, e_{23}, e_{45}, e_{56}
$E(2)$: e_{34}
$E(5)$: e_{14}
$E(7)$: e_{16}
$E(9)$: e_{13}, e_{15}, e_{24}, e_{25}, e_{26}, e_{35}, e_{36}, e_{46}

The incidence table for $E(1)$ is shown in Table 1.

Table 1. The incidence table for the first equivalence class $E(1)$.

		NSL 1			*NSL 2*			
Equivalence Class	*Edge*	N_1	N_2	N_3	N_4	N_5	N_6	*Edge Label*
$E(1)$	e_{12}	√	√					PRI
$E(1)$	e_{23}		√	√				PRI
$E(1)$	e_{45}				√	√		PRI
$E(1)$	e_{56}					√	√	PRI

Because the columns under N_1, N_3, N_4, and N_6 have only single check marks, the corresponding edges are primary. There are no secondary edges in $E(1)$ for this example. The next step is to label the nodes in the incidence table according to membership in an NSL. For this example, there are two NSLs after $E(1)$ is considered. The PFNET, after Step 9, consists of the edges shown in Figure 3.

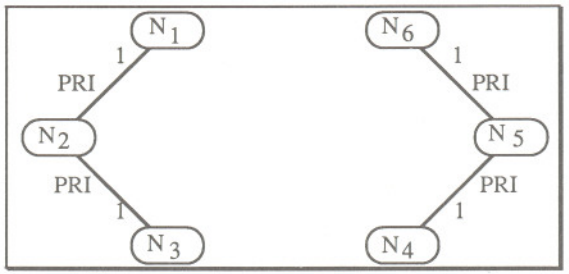


Figure 3. The PFNET after $E(1)$ is considered.

The equivalence class $E(2)$ is considered next, and its (only) edge e_{34} is appended to the edge-node table as the last row. For convenience, the NSL identifiers are placed above the node identifiers. The incidence table is now as shown in Table 2.

Table 2. The incidence table for the first two equivalence classes.

		NSL 1						
Equivalence Class	*Edge*	N_1	N_2	N_3	N_4	N_5	N_6	*Edge Label*
$E(1)$	e_{12}	√	√					PRI
	e_{23}		√	√				PRI
	e_{45}				√	√		PRI
	e_{56}					√	√	PRI
$E(2)$	e_{34}			√	√			SEC

Since e_{34} connects nodes in two different NSLs, each having node size greater than one, it is a secondary edge, and it is entered in the PFNET. Because it joins the only two NSLs, none of the candidate edges in the remaining equivalence classes can be either primary or secondary edges. Thus we consider $E(5)$, $E(7)$, and $E(9)$ together (reentering the top of the loop at Step 10 in the procedure), because there is now only one NSL. We next compute $W^2 = W \otimes W$ as previously described, and

$$W^2 = \begin{matrix} - & 10 & 2 & 10 & 6 & 10 \\ 10 & - & 10 & 3 & 10 & 8 \\ 2 & 10 & - & 10 & 3 & 10 \\ 10 & 3 & 10 & - & 10 & 2 \\ 6 & 10 & 3 & 10 & - & 10 \\ 10 & 8 & 10 & 2 & 10 & - \end{matrix}$$

For $r = 1$, the weight of a path is the sum of the weights of the edges along the path. The entries on the main diagonal of W are not relevant, because they consist of two-edge paths only in a degenerate sense: either (1) no edges are traversed from the indicated node, or (2) an edge is traversed, but then immediately retraced. Similarly, two-edge paths

involving a zero-valued entry in W are omitted from consideration in off-diagonal elements of W because such paths really consist only of a single edge. Next, D^2 is found by taking the smaller value in each position of W and W^2 and noting from which matrix this smaller distance came.

$$D^2 = \begin{matrix} - & 1 & 2 & 5 & 6 & 7 \\ 1 & - & 1 & 3 & 9 & 8 \\ 2 & 1 & - & 2 & 3 & 9 \\ 5 & 3 & 2 & - & 1 & 2 \\ 6 & 9 & 3 & 1 & - & 1 \\ 7 & 8 & 9 & 2 & 1 & - \end{matrix}$$

Comparing elements of D^2 to elements of W, the entries which are equal are in the locations corresponding to e_{12}, e_{14}, e_{16}, e_{23}, e_{25}, e_{34}, e_{36}, e_{45}, and e_{56}. The edges in this list which have not yet been labeled are labeled TER and are added to the PFNET, shown in final form in Figure 4.

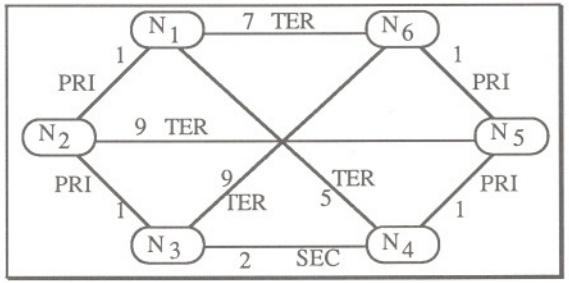


Figure 4. The labeled PFNET($r = 1$, $q = 2$) for the example.

Since no NSLs can be merged, and all edges have been considered, there can be no more edges added, and the procedure terminates.

This network illustrates two aspects of the triangle inequality: No link violates the triangle inequality considering paths of two links ($q = 2$), and some links do violate a triangle inequality considering paths of three or more links. An example of the latter is e_{14}, in which $w_{14} = 5$. An alternate (three-link) path is P_{1234}, which has a weight of 4. Similarly, $w_{16} = 7$ and the path P_{123456} with weight of 6 illustrates the violation of a triangle inequality involving a five-link path.

Fundamental Properties

It is appropriate to begin by showing that the two generation procedures just described add precisely the same edges to the PFNET as the LMR given in Definition 4, for the situation in which the W matrix is symmetrical (the case in which both procedures apply).

Theorem 1

For a given r, q, and symmetric W, either of the PATHFINDER PFNET(r, q) procedures given in the preceding section results in the same edges in the PFNET as does the

LMR given in Definition 4. (Recall that one of the procedures is intended for either asymmetric or symmetric weight matrices, and the other is only for symmetric matrices. An asymmetric weight matrix yields directed networks, and a symmetric weight matrix yields undirected networks, having labels on the edges if the latter procedure is used.)

Proof

We begin the proof by considering the procedure for symmetric W. The edges labeled PRI and SEC, at the time of labeling and entry into the PFNET, represent the only path, or equal and minimum-cost paths, between the nodes involved. Because edges are considered by equivalence classes of increasing weight, no edge to be added later can decrease the cost of the direct path afforded by a single primary or secondary edge. A tertiary edge provides a minimum-cost path alternative to multiple-edge paths of PRI and SEC edges. The matrix operations in Step 15 of the generation procedure for symmetric weight matrices find all edges which provide lowest-cost direct paths, considering all paths having up to q edges. These matrix operations identify all edges in the PFNET(r, q), whether or not the weight matrix W is symmetric, and independently of the edge labeling rule. If the edge labels are not needed, then these matrix operations alone identify all edges providing minimum-cost links between nodes, considering multiple-link paths of q or fewer edges. Thus the Pathfinder procedure used for asymmetric data, and the procedure for symmetric data, both result in the same LMR as given in Definition 4. □

By definition of the LMR, and as described in the procedure, edges providing alternate, equal-cost paths are all entered in the PFNET. This feature eliminates a random or arbitrary choice of edges to represent a particular association, and results in a unique network, given a particular weight matrix W, and values for r and q.

A network of particular interest is PFNET($r = \infty, q = n–1$). This PFNET always has the minimum number of edges, because the path-length metric is simply the value of the maximum weight along the path (this is called the *dominant* metric), and no edge in the PFNET can result in the violation of a triangle inequality for any path. The maximum number of edges in a path without a cycle, for a graph having n nodes, is $n–1$. The edges in an undirected PFNET($r = \infty, q = n–1$) are all labeled PRI or SEC, because (1) PRI and SEC edges are added to the PFNET based on node-covering properties and are independent of values of r and q, and (2) since TER edges always provide an alternate path to PRI or SEC edges established in an earlier equivalence class, the path length with $r = \infty$ is always less through the PRI or SEC edges. Thus tertiary edges never appear in an undirected PFNET($r = \infty, q = n–1$). For a given symmetric weight matrix W, if PFNET(r, q) is generated, then PFNET($r = \infty, q = n–1$) can always be found simply by deleting all edges labeled TER in the PFNET(r, q).

Definition 12

A minimum-cost spanning *tree* (mintree) of an undirected PFNET is a connected graph having no cycles in which (1) there is a path between any pair of nodes, and (2) the sum of the weights associated with the edges is a minimum.

This definition is similar to that given in Even (1979, Chapter 2). A PFNET can have several mintrees; a simple example is a PFNET in which each off-diagonal weight in W is one. Then all potential edges are in the first (and only) equivalence class, all edges are secondary edges, and all edges are entered in the PFNET. Clearly there are several mintrees for such a PFNET (which is the complete graph), although this is an extreme case.

Definition 13

The *minimum-cost network* (MCN) of an undirected PFNET(r, q) is the union of all mintrees of the PFNET.

Theorem 2

For a given symmetric W, r, and q, the MCN of PFNET(r, q) is PFNET(∞, $n–1$).

Proof

Every primary edge is in every mintree, because a primary edge provides the lowest-cost access to at least one of the end nodes responsible for the edge being labeled PRI. Each secondary edge is in some mintree, because a secondary edge provides either (1) a unique, lowest-cost path between two NSLs for which the node size is greater than one, or (2) an equally lowest-cost alternative path between two NSLs, in which the alternatives are provided by (secondary) edges in the same equivalence class. In the first case, the secondary edge is in every mintree; in the second case, alternative mintrees correspond to the secondary edges providing alternative paths in the same equivalence class. A tertiary edge cannot be in any mintree of PFNET(r, q), because such an edge provides an alternative path (covering the same end nodes) to one already existing. Thus the tertiary edges always form cycles, and there is always an alternative path to a tertiary edge which is composed of primary and secondary edges established in earlier equivalence classes. Therefore the PFNET(∞, $n–1$) is independent of the r-metric and the q parameter, and includes all primary and secondary edges (as does the MCN). □

The significance of this result is that the PFNET which has fewest edges, the MCN, is unique, provides all minimum-cost paths between nodes, has no violations of the triangle inequality, and is the sparsest network which possesses these properties. For cognitive modeling, the scaling of data, and for clustering, these properties of economy are significant indeed.

Properties of Inclusion

The networks generated in Procedure PATHFINDER PFNET(r, q) possess properties of inclusion, or nesting, as values of the r-metric and q parameter change.

Definition 14

A PFNET1 is *included in* (or is a *spanning subgraph* of) PFNET2 if and only if:

(1) PFNET1 and PFNET2 have the same set of nodes, and

(2) every link in PFNET1 is also in PFNET2.

The first inclusion property to be discussed is related to the r-metric used in computing path lengths within a PFNET.

Theorem 3

For a weight matrix W, PFNET(r_2, q) is a spanning subgraph of PFNET(r_1, q) if and only if $r_1 \leq r_2$.

Proof

Since edge labeling is not a consideration, we will show inclusion by using the Pathfinder procedure that uses only the matrix operations. First suppose that $r_1 \leq r_2$, and that e_{ij} is a link in PFNET(r_2, q), then

$$w_{ij}(r_2) = d_{ij}^{q}(r_2).$$

We must show that e_{ij} is also in PFNET(r_1, q). The reason that e_{ij} is in PFNET(r_2, q) is that alternative paths having more than one link are at least as costly; these path lengths are computed in W^2, W^3, ..., W^q. The following inequality is helpful:

$$\left(\sum_{i=1}^{m}(w_i)^{r_1}\right)^{\frac{1}{r_1}} \geq \left(\sum_{i=1}^{m}(w_i)^{r_2}\right)^{\frac{1}{r_2}} \text{ if and only if } r_1 \leq r_2.$$

This inequality is well-known in mathematics, and a proof and references are given in Schvaneveldt, Dearholt, and Durso (1988). Thus a given path that is longer than w_{ij} using $r = r_2$ is at least as long using $r = r_1$. Therefore the matrix operations which identify links by computing these alternative path lengths cannot result in a shorter alternative path, and e_{ij} is in PFNET(r_1, q) also. But suppose that the minimum-length paths connecting N_i and N_j in PFNET(r_1, q) and PFNET(r_2, q) are different, because of a single link e_{ij} which is in PFNET(r_1, q) but is not in PFNET(r_2, q). Therefore we know that

$$w_{ij} \leq \left(w_1^{r_1} + w_2^{r_1} + \ldots + w_m^{r_1}\right)^{\frac{1}{r_1}}$$

and

$$w_{ij} > \left(w_1^{r_2} + w_2^{r_2} + \ldots + w_m^{r_2}\right)^{\frac{1}{r_2}}$$

Note that, since $r_1 \leq r_2$, the converse is not possible (because of the previous inequality); that is, a link e_{ij} cannot be in PFNET(r_2, q) without also being in PFNET(r_1, q).

Now suppose that if e_{ij} is in PFNET(r_2, q), then e_{ij} is also in PFNET(r_1, q). Then any path that exists in PFNET(r_2, q) also exists in PFNET(r_1, q). We must show that $r_1 \leq r_2$. Therefore

$$w_{ij} \leq \left(w_1^{r_2} + w_2^{r_2} + \ldots + w_m^{r_2}\right)^{\frac{1}{r_2}}$$

implies

$$w_{ij} \leq \left(w_1^{r_1} + w_2^{r_1} + \ldots + w_m^{r_1}\right)^{\frac{1}{r_1}}$$

over the same path in each network, by hypothesis.

Therefore

$$\left(w_1^{r_2} + w_2^{r_2} + \ldots + w_m^{r_2}\right)^{\frac{1}{r_2}} \leq \left(w_1^{r_1} + w_2^{r_1} + \ldots + w_m^{r_1}\right)^{\frac{1}{r_1}}$$

which is true provided $r_1 \leq r_2$, as given in the inequality. □

An observation for the symmetric case, in which the edge-labeling procedure is used, is that PFNET(r_1, q) and PFNET(r_2, q) share the same MCN, which is PFNET(∞, n–1). Thus PFNET(r_1, q) and PFNET(r_2, q) have the same primary and secondary edges, and may differ only in the tertiary edges, as discussed previously.

The second inclusion property concerns the q parameter, in a manner analogous to the r-metric.

Theorem 4

For a weight matrix W, PFNET(r, q_2) is a spanning subgraph of PFNET(r, q_1) if and only if $q_1 \leq q_2$.

Proof

It is sufficient to observe that, as W^{i+1} is computed from W^i using the given r-metric, links to the resulting PFNET cannot be added, but may be eliminated. This is true because W^i and D^i determine the links satisfying the triangle inequality considering paths of i or fewer links. A potential link e_{jk} not satisfying such a triangle inequality results in an entry

$$d_{jk}^{i} \leq w_{jk}$$

at step i, so that e_{jk} will not be in PFNET(r, $q = i$). □

The inclusion relations can be illustrated by means of a graph in two-dimensional space. As shown in Figure 5, we will use the r-metric for the abscissa axis and the q parameter for the ordinate axis. Then the r and q values used for a particular PFNET are plotted at the intersection of the two lines. The inclusion relations for other PFNETs computed from the same data, but in which different values of r and q are used, are the rectangular areas shown with hatch marks. The PFNET represented by the point (r, q) is a spanning subgraph of every PFNET represented by the r and q values below and to the left, since inclusion is transitive. Similarly, the PFNET represented by the point (r, q) includes every PFNET represented by the r and q values above and to the right. In practice, the PFNET having fewest links (the MCN) is generated with r and q values substantially smaller than the theoretical values of $r = \infty$ and $q = n$–1, which are the values guaranteed to produce the MCN.

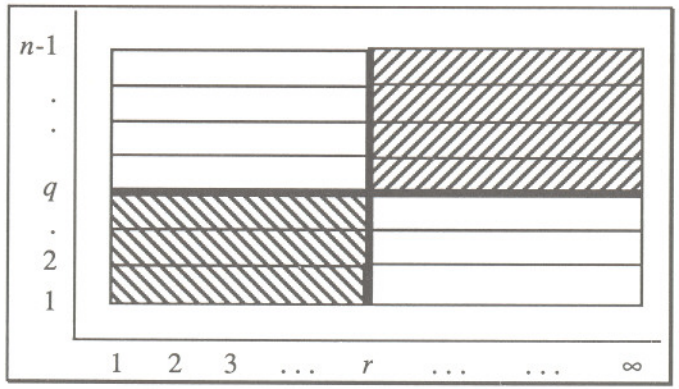


Figure 5. Illustration of the inclusion relations.

Data Transformations and Network Structure

An important issue in data scaling and modeling concerns the effects upon network structure of various types of transformations upon the proximity data. Uncertainty associated with the meaning of data values should be accommodated by appropriate options in the scaling procedure.

Definition 15

Let T be a transformation to be applied to the elements w_{ij} of the weight matrix W, and denote the transformed weight values as $T(w_{ij}) = t_{ij}$. The transformed weight matrix is denoted Wt, and the resulting network is denoted by PFNETt.

The first result is that a multiplicative transformation preserves link structure for both directed and undirected PFNETs, and edge labels are also preserved for undirected, labeled PFNETs.

Theorem 5

Given nodes N_i and a weight matrix W, resulting in PFNET(r, q), and given the transformation,

$$\text{T: } t_{ij} = bw_{ij} \text{ where } b > 0,$$

then e_{ij} is in PFNET(r, q) if and only if e_{ij} is in PFNET$t(r, q)$. If e_{ij} is in both networks, and if the networks are undirected and labeled, then it has the same label in both networks.

Proof

We first consider the undirected, labeled case. The transformation T can be viewed as transforming the equivalence classes $E(w_i)$ into equivalence classes $E(bw_i)$. The transformed (corresponding) equivalence classes are hence considered in the same order as the original equivalence classes. Thus the construction of the incidence tables is identical for both PFNET(r, q) and PFNET$t(r, q)$ so the primary and secondary edge structure of the two networks is isomorphic.

For the tertiary edges, the matrix operations can be expressed by the observation that e_{ij} is added to PFNET$t(r, q)$ if and only if

$$bw_{ij} \le \left(\sum_{k=1}^{K} (bw_k)^r \right)^{1/r} = b \left(\sum_{k=1}^{K} (w_k)^r \right)^{1/r}$$

where $K \le q$, for any path connecting N_i and N_j. Both sides can be divided by b to obtain the comparison

$$w_{ij} \le \left(\sum_{k=1}^{K} (w_k)^r \right)^{1/r}$$

which is the inequality used to determine whether or not e_{ij} is added to PFNET(r, q). Thus e_{ij} is either added to both networks, with the same label in each, or it is not added to either. □

That arc structure is preserved for the directed PFNETs under multiplicative transformations is evident by referring to the matrix operations invoked for tertiary edges, in the undirected case just discussed. The multiplier b factors out, and the decisions for all arcs are the same, with or without the multiplier.

The other type of transformation we consider is a monotonic transformation T.
T: $t_{ij} = f(w_{ij})$ so that,

If $w_{ab} = w_{xy}$, then $t_{ab} = t_{xy}$, and

If $w_{ab} > w_{xy}$, then $t_{ab} > t_{xy}$.

Our second result is that, provided $r = \infty$, a monotonic transformation preserves link structure for both directed and undirected PFNETs, and preserves edge labels for undirected, labeled PFNETs.

Theorem 6

Given nodes N_i and a weight matrix W, and given a monotonic transformation,

$$T\text{: } t_{ij} = f(w_{ij}),$$

then e_{ij} is in PFNET$(r = \infty, q)$ if and only if e_{ij} is in PFNET$t(r = \infty, q)$. If e_{ij} is in both networks, and if the networks are undirected and labeled, then it has the same label in both networks.

Proof

We first consider the undirected, labeled case. The transformation T can be viewed as transforming the equivalence classes $E(w_i)$ into equivalence classes $E(f(w_i))$. Furthermore, the order of the transformed equivalence classes remains the same for the generation procedure. Thus the incidence tables are the same for each network, so the primary and secondary edges are also the same for each network. With $r = \infty$, the tertiary edges in the

PFNET are determined by the matrix operations described earlier in the section entitled Generation Algorithms for PFNETs, and the maximum weight on an edge in a path determines the cost of that path. But the transformed maximum weight is the same as the maximum of the transformed weights, since monotonic transformations preserve order. Thus the same tertiary edges are in the PFNET in either the transformed version or the original version. Therefore e_{ij} is added to both networks, with the same label in each, or it is not added to either.

To see that arc structure is preserved for the directed case, we note that a monotonic transformation preserves order. Thus the same decisions are made regarding arc membership on either the original data or the transformed data, provided $r = \infty$. □

In considering properties of data, ordinal data are presumed to be data in which the experimenter is confident that the data values are, at the minimum, in proper order. Further claims on properties of the data, such as the meaning of intervals or of ratios, however, may not be warranted. For these cases, the PFNETs used should be constrained to those in which $r = \infty$. Ratio-scale data are data in which each data value is presumed to be within a multiplicative constant of the "correct" value. For these types of data, any values of r and q can be used in generating a PFNET, for a given data matrix, and the resulting PFNET is independent of such multipliers.

Pathfinder Networks and Hierarchical Clustering

For approximately two decades now, hierarchical clustering has been an important tool in the analysis of proximity data. For certain types of data, it is very appropriate and leads to meaningful clusters, or more precisely, to meaningful families of clusters. The purpose of this section is to explicate the relationships between hierarchical clustering and Pathfinder networks. We will show that Pathfinder networks can provide the same information that is available in the minimum method of Johnson (1967), also called the single-linkage (or single-link) method (Ling, 1972; Sokal & Sneath, 1963). Furthermore, we will show that hierarchical clustering cannot provide the structural information available in Pathfinder networks.

For hierarchical clustering schemes (abbreviated HCS for convenience), distance is regarded as being the primary feature of interest. For Pathfinder networks, however, both structural properties and distance are regarded as important. As shown in the preceding section, the structure of a Pathfinder network is invariant under multiplicative transformations of the weight matrix; and for $r = \infty$, the structure is invariant under monotonic transformations of the weight matrix. Pathfinder networks maintain the original (most salient) proximities (associations) of the data by preserving all minimum-cost paths, thus supporting the representation of these associations explicitly in the network (as shown in Theorem 2, each Pathfinder network contains all mintrees). Thus geodetic paths (minimum distance paths between each pair of nodes) are preserved in Pathfinder networks, and the computation of the distances is a part of the generation procedure (Step 15). But why, if the scaling of data is the objective, should organization be of any interest or consequence when distances are all that is required for clustering? The answer lies in the origin of Pathfinder networks in modeling the organization of human semantic memory and in other applications of these networks which have been explored recently.

We will use *object* (in the same way as Johnson, 1967) to mean either a single entity or node, or a cluster of entities or nodes; the context will be sufficient to establish the precise meaning if it is important. A complication for HCS and Pathfinder is the question of

measuring the distance between two objects. Should this distance be (1) the smallest distance between some object in one cluster and an object in the other cluster, as in the minimum method; (2) the largest distance between an object in one cluster and an object in the other cluster, as in the maximum or complete-link method; (3) the median distance of the between-cluster distances between objects; or (4) some average distance, perhaps weighted, of the distances between pairs of objects in which one object is in each cluster? Choices (1), (2), and (3) require only ordinal properties of data, while (4) requires ratio properties. The latter is not often assumed for subjective data, but for problems in which objective data are available, then ratio assumptions are frequently appropriate. Because the computations are simple, the minimum method and the maximum method, in the terminology of Johnson (1967), have been examined in great detail for many applications of psychological proximity data.

An assumption made by Johnson (1967, p. 249) is that the off-diagonal distances in the original matrix are all positive and *distinct* (the distances on the main diagonal are all assumed to be zero). This assumption of distinctness of the off-diagonal distances is motivated by his assumption that the data are strictly ordinal, in the sense that the subjects or researchers can differentiate between each pair of entities. Nevertheless, we have found this assumption to be unwarranted—real data we have collected have frequently exhibited multiple entries having the same values. Furthermore, data representing large systems are quite likely to have large numbers of ties. Thus attempts to model proximity data must include the general case in which data can exhibit ties. As illustrated in several of the examples in this chapter, ties in the data are easily modeled in Pathfinder by including them as links in the network (provided they offer paths having smallest weights between node sublists).

We will now review the agglomerative procedure described in Johnson (1967, p. 248) for generating an HCS using the minimum method, although Johnson does not, in that article, begin from a general weight or *adjacency matrix*; instead, he shows only a distance matrix which could have been reduced from any one of a large number of different weight matrices by application of the HCS procedure. He assumes, however, that a weight matrix is available at the beginning of the procedure. In the following statement of Johnson's procedure, the first part of each step is taken verbatim from his paper (1967); the parenthetical comments concluding each step relate the terminology he used to that used for Pathfinder and also include additional comments intended to be helpful.

Johnson's HCS Procedure:

1. Clustering C_0, with value 0, is the *weak clustering*. (It is called the weak clustering because each object (node) is considered to be an individual cluster. The value is simply the distance to other nodes, and corresponds to the equivalence classes discussed in the section Generation Algorithms for PFNETs.)
2. Assume we are given the clustering C_{j-1} with the similarity function d, defined for all objects or clusters in C_{j-1}. Let α_j be a minimal nonzero entry in the matrix. Merge the pair of objects and/or clusters with distance α_j to create C_j, of value α_j. (This step is ambivalent, because "a minimal nonzero entry" seems to imply that there could be ties in the data, but Johnson specifically excludes ties in his discussion of clustering. The merging of two clusters can be considered equivalent to the joining of two NSLs with an edge in the graphical paradigm of Pathfinder. Johnson's similarity function corresponds to the distance matrix of the Pathfinder procedure.)

3. We create a new similarity function for C_j in the following manner: If x and y are clustered in C_j and not in C_{j-1} (i.e., $d(x, y) = \alpha_j$), we define the distance from the cluster $[x, y]$ to any third object or cluster, z, by $d([x, y], z) = \min [d(x, z), d(y, z)]$.

   If x and y are objects and/or clusters in C_{j-1} not clustered in C_j, $d(x, y)$ remains the same. We obtain a new similarity function d for C_j in this way. (The distance between two clusters is defined to be the minimum distance between any pair of elements in which one element is in each cluster. That is, if x and y are two objects—an object can be either a single element or a cluster—at level C_{j-1}, and if $d(x,y) = \alpha_j$ (so that x and y become clustered in C_j), and if z is any other object or cluster at level C_{j-1}, then $d(x,z) = d(y,z)$. Johnson (1967, p. 245) gives a brief proof of this statement, and it is equivalent to the use of infinity for the value of the r-metric, which also implies the use of the *ultrametric inequality* of Johnson.)

4. We now repeat steps 2 and 3 until we finally obtain the *strong clustering*—we are then finished. (The strong clustering is one in which all elements (nodes) are in the same cluster. In the paradigm of Pathfinder, the nodes and edges then form a connected graph.)

Because HCS were not designed to incorporate subtle aspects of structure, it is not surprising that they have limitations in structural matters. Some of these are:

1. If ties in the proximity data are allowed, then the mapping of weight matrices to HCS is not one-to-one; that is, several different weight matrices can yield the same clustering in an HCS (this is also true of PFNETs, but to a lesser extent as will be shown later).
2. Nonhierarchical structural relationships cannot be represented. There are two ways these can occur; first, equal weights can lead to cycles, in the graphical paradigm, and second, clusters can be overlapping, as described in Shepard and Arabie (1979). We will address only the first nonhierarchical situation, as Pathfinder has not yet been sufficiently developed to apply it to the latter.
3. Information regarding the pair of objects or nodes responsible for establishing the distance between two clusters (that is, which pair has the most "salient" relationship) is lost once the clustering is established.

Ties in proximity data used in HCS have been regarded as a problem for many years. Furthermore, there are significant differences in the single-link (minimum) and complete-link (maximum) methods in dealing with ties. The single-link method has a continuity property (Jain & Dubes, 1988, Section 3.2.6) which insures that adding or subtracting small amounts to tied values results in dendograms which merge smoothly into the same dendogram as the added amounts tend to zero, no matter how the ties are broken. The complete-link method, however, does not possess this property, and can yield different dendograms depending upon how the ties are resolved. As mentioned previously, the ultrametric inequality is equivalent to the use of $r = \infty$ in PFNETs; the weight of a path between nodes (objects) not directly linked is the maximum of any of the weights on links making up that path, as computed by the Minkowski metric as r approached infinity. By

modifying Step 2 of Johnson's procedure, we obtain a version of his minimum method which models ties in the proximity data. This modified step is:

2′. Assume we are given the clustering C_{j-1} with the similarity function d, defined for all objects or clusters in C_{j-1}. Let α_j be a minimal nonzero entry in the matrix. Merge *all pairs of objects and/or clusters* (formerly read: the pair of objects and/or clusters) with distance α_j to create C_j of value α_j.

We will use UHCS to denote the HCS with this step substituted in Johnson's minimum method, because a unique dendogram is generated even if there are ties in the weight matrix (the alternative is to break ties arbitrarily). The next example will illustrate this modified method, and will also show that the mapping from the weight matrix to the UHCS is not one-to-one.

$$W_1 = \begin{matrix} 0 & 1 & 1 & 5 \\ 1 & 0 & 4 & 2 \\ 1 & 4 & 0 & 5 \\ 5 & 2 & 5 & 0 \end{matrix}$$

The UHCS using the minimum method for this matrix is shown in Figure 6, and the PFNET($r = \infty, q = n-1$) for the matrix is shown in Figure 7.

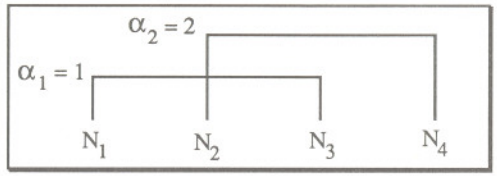


Figure 6. The UHCS for matrices W_1 and W_2.

The matrix W_2, shown below, has the same UHCS as does W_1; however, the structural aspects relating to specific associations are not a factor in obtaining the UHCS.

$$W_2 = \begin{matrix} 0 & 1 & 4 & 5 \\ 1 & 0 & 1 & 2 \\ 4 & 1 & 0 & 5 \\ 5 & 2 & 5 & 0 \end{matrix}$$

The PFNET($r = \infty, q = n-1$), however, is different for W_2, as shown in Figure 8.

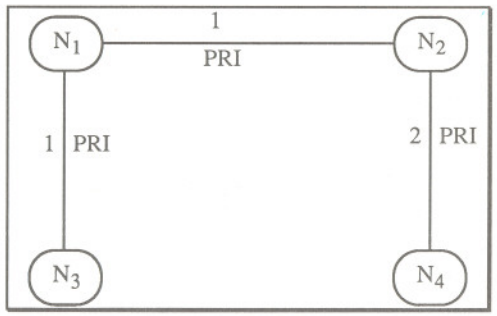


Figure 7. PFNET($r = \infty, q = n-1$) for W_1.

A bit of combinatorics shows that there are, in fact, 12 different matrices (and 12 different PFNETs) which have the same UHCS as W_1 or W_2 (nodes 1, 2, and 3 can have any one edge missing of the three edges that are possible, or all three present, and the link between this cluster and node 4 can be between node 4 and any one of the other three nodes). If the weight matrix is viewed as a representation of the complete graph, then any change in a weight or weights sufficient to change any

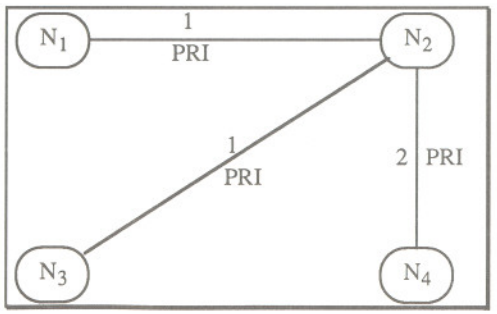


Figure 8. PFNET($r = \infty, q = n-1$) for W_2.

mintree will modify the structure of PFNET(∞, n–1), but may not necessarily modify the UHCS. Therefore the primary and secondary edges, considered together, define the same clusters via the node sublists as the UHCS. The labels used in PFNETs (primary, secondary, and tertiary) thus distinguish between edges in ways not possible in HCS or UHCS. Again, if distance is the only concern, then one needs only the UHCS; but if navigating through a data structure by following links is important, or if computing cluster metrics using structural information is necessary, then the structural differences can be important. HCS or UHCS are hierarchical and unambiguous only in the preservation of distances between entities.

While different weight matrices can also yield the same PFNETs, structure (the nodes linked because of salient relationships) depends only upon the minimum-distance paths, which are always preserved in PFNETs. Stated another way, if a given weight is sufficiently large, the corresponding edge will not be in the PFNET. For example, in the matrix W_2 above, suppose that the distance w_{13} (which has a value of 4 in this example) is allowed to vary. Provided that w_{13} has a value greater than 1, then the structure of PFNET($r = \infty$, $q = 3$) remains the same as in Figure 8, since no mintree is changed.

There are other graphical approaches to hierarchical clustering. An example is the family of *threshold graphs*, undirected graphs in which link membership is determined by selecting successively larger thresholds, beginning with the smallest weight value in the weight matrix. For a given threshold, each link having a weight less than or equal to that threshold is in the threshold graph. As the threshold is increased in value, a new graph is generated and the new graph typically has more links (it never has fewer links). Ultimately, the threshold is made large enough to include sufficient links to make the graph connected. This occurs precisely when the threshold value equals the value of α_i which would yield the strong clustering defined in Johnson's procedure. A *proximity graph* is a threshold graph with the links labeled with their corresponding weights. Threshold and proximity graphs are discussed in Jain and Dubes (1988). These graphs simply provide an alternative perspective on the formation of the clusters using the minimum method. A disadvantage, however, is that they have only the context supplied by the global threshold on weight values. In contrast, a PFNET has a more local context provided by the triangle inequality, in which a link with a certain weight value may be included in one part of the network, while another link having the same weight value may not be included in another part of the network. Thus a connected threshold graph can have more links than PFNET ($r = \infty$, $q = n$–1), but cannot have fewer links. However, we will show that the same family of clusters is obtained from the PFNETs.

Before presenting two theorems, we will discuss the relationships between the minimum method UHCS and Pathfinder networks with respect to points 2 and 3 above (modeling ties through cycles in the graphical paradigm, and maintaining the information regarding the pair of entities responsible for establishing the distance between two clusters), in terms of both an example and the procedures involved for the generation of the clustering information and the Pathfinder networks.

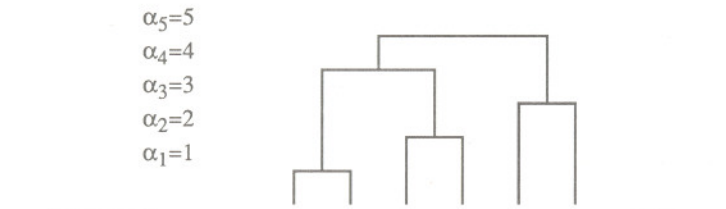
We will argue that (1) the information available in a minimum method UHCS is also available in PFNET($r = \infty, q = n$–1), and therefore (2) is also available in any PFNET (r, q) by the inclusion theorems, although some computation may be required, and finally (3) that the converses of (1) and (2) do not hold.

The aspects in which the minimum method of UHCS and Pathfinder networks are equivalent will be discussed in terms of the following example:

$$W_3 = \begin{matrix} 0 & 1 & 4 & 5 & 6 & 7 \\ 1 & 0 & 4 & 7 & 8 & 6 \\ 4 & 4 & 0 & 2 & 6 & 8 \\ 5 & 7 & 2 & 0 & 5 & 7 \\ 6 & 8 & 6 & 5 & 0 & 3 \\ 7 & 6 & 8 & 7 & 3 & 0 \end{matrix}$$

The incidence table is constructed below, and the membership in node sublists (NSLs) is indicated by the tree growing from the top of the incidence table:

Table 3. The incidence table for the example.



Equivalence Class	*Edge*	N_1	N_2	N_3	N_4	N_5	N_6	*Edge Label*
$E(1)$	e_{12}	√	√					PRI
$E(2)$	e_{34}			√	√			PRI
$E(3)$	e_{56}					√	√	PRI
$E(4)$	e_{13}	√		√				SEC
	e_{23}		√	√				SEC
$E(5)$	e_{14}	√			√			no edge
	e_{45}				√	√		SEC

There are two observations which can be made on the basis of this example. First, the α_i levels of Johnson (1967) correspond to the equivalence classes $E(w_i)$, which in turn correspond to weights of the Pathfinder algorithm. Second, the clusters formed by the minimum method UHCS are the same as the NSLs of the Pathfinder procedure for symmetric data. A proof showing that the latter claim is general will be offered in Theorem 7.

In the third step of Johnson's HCS procedure, a *similarity function* is updated after each α_j is considered. Usually given in matrix form, this function provides the distances between objects and has dimension equal to the number of objects after merging at each new value of α_j. There is a simple procedure for computing similar distances between nodes in the agglomerative Pathfinder generation procedure illustrated above. This procedure derives a square matrix which we will denote *DM*, and is as follows:

1. Begin with the weight matrix W;
2. Using node membership within NSLs, eliminate from consideration all w_{ij} in W in which N_i and N_j are in the same NSL (this results in an effective distance of zero between nodes within the same NSL);
3. For each pair NSL_a and NSL_b, select the minimum w_{mn} in which $N_m \in NSL_a$, and $N_n \in NSL_b$ to represent the distance between that pair of NSLs (If some NSL has at least two nodes, the dimension of the distance matrix is decreased because some weights are discarded.);
4. The resulting matrix is the distance matrix DM between NSLs.

The results of this reduction of the W matrix to a new distance matrix is a matrix having the same number of rows and columns as the number of NSLs, in which the distance between each pair of NSLs is the minimum of all the weights on edges connecting the pair of NSLs. Like an object in Johnson's paper, an NSL need not have more than one node.

The distance matrix computed in this procedure differs from the matrix D^i computed in Step 15 of the agglomerative procedure, or in Step 1 of the matrix method, because the latter has not examined all paths until $q = n-1$. The former provides the distances between entities considering all paths, and the latter provides the distances between entities in which only paths having i or fewer links are considered. If Johnson had retained intracluster distances within his computations for each new similarity function (Step 3), thus maintaining the same dimension as the original weight function, the matrix resulting after the strong clustering is obtained would be the same as the distance matrix D^{n-1} obtained in the Pathfinder procedure (using r = ∞, of course).

We will now derive the distances between each pair of entities from the PFNET(r = ∞, $q = 5$) for the above example, and then show that the distances are the same for the minimum method UHCS. Indeed, because of the similarities of the procedures, these distances must always be the same.

Addressing the distance question in terms of the previous example, the UHCS distance matrix and the PFNET distance matrix D^5 (defined and discussed in the section entitled Generation Algorithms for PFNETs) both reduce to:

$$D^5 = W_{3r} = \begin{matrix} 0 & 1 & 4 & 4 & 5 & 5 \\ 1 & 0 & 4 & 4 & 5 & 5 \\ 4 & 4 & 0 & 2 & 5 & 5 \\ 4 & 4 & 2 & 0 & 5 & 5 \\ 5 & 5 & 5 & 5 & 0 & 3 \\ 5 & 5 & 5 & 5 & 3 & 0 \end{matrix}$$

The PFNET($r = \infty, q = n-1$) for W_3 is shown in Figure 9. Structurally, it can be seen that the relationships in the PFNET are not hierarchical, in the sense that there is a cycle containing nodes 1, 2, and 3; that is, the NSL formed in E_1, consisting of joining nodes 1 and 2 with an edge having weight 1, is joined with node 3 via both nodes 1 and 2 with edges having weights of 4. Thus nodes 1, 2, and 3 form what is known as a *clique* in graph theory (after $E(4)$ is considered), because each is directly linked to the other. This sort of structural information is not readily available using UHCS. Ties in weight values are evident in Figure 9, in which the edges e_{13} and e_{23} are each in different mintrees, and connect two node sublists with minimum (and equal) costs. Some of the more recent work in clustering has recognized the importance of having a procedure which does not arbitrarily break ties (Shepard & Arabie, 1979, p. 93).

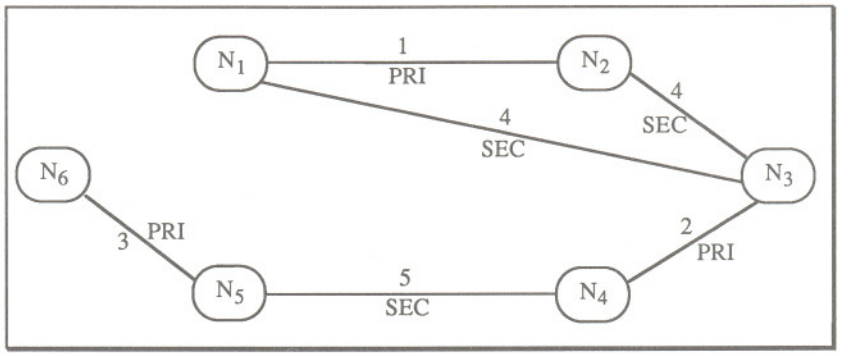


Figure 9. PFNET($r = \infty, q = n-1$) for W_3.

We will now show that the distance matrices for a PFNET($r = \infty, q = n-1$) and the UHCS derived from a common weight matrix W are identical, and the clusters formed are the same at each step in the agglomerative methods.

Theorem 7

Given a symmetric weight matrix W having zero-valued entries on the diagonal and positive values off the diagonal, the clusters formed using the UHCS procedure and the NSLs formed using the agglomerative Pathfinder procedure are isomorphic at each step. Furthermore, the interobject distances obtained from UHCS and PFNET($r = \infty, q = n-1$) are also identical at each step.

Proof

The proof is by induction over equivalence classes for the Pathfinder network and over the α_j levels of the UHCS. Both agglomerative methods begin with the weak clustering, in which each node or object is also a cluster. In the graphical paradigm, the nodes are not linked at this stage; in the clustering scheme, no clusters have been merged. Clearly the distance matrices are now identical to the W matrix, and the clusters are the same for each paradigm. For the induction step, assume that the clustering formed at level α_{j-1}, denoted by C_{j-1}, has the same clusters as the NSLs formed after the equivalence class $E(w_{j-1})$ is considered, using the agglomerative Pathfinder method described in the section entitled Fundamental Properties. We also assume that the interobject distances resulting from both paradigms are identical at this stage.

Now consider the clustering level α_j, which corresponds to a weight value of w_j in the W matrix. In the UHCS procedure, all weights having this value are considered, and objects having this weight value between them are merged, provided that they are not already in the same cluster. In the Pathfinder procedure, the equivalence class of weights having this value is considered, and new NSLs are formed based upon precisely the same criteria as for the UHCS. Therefore the objects in the next clustering, C_j, are isomorphic to the NSLs in the Pathfinder paradigm after $E(w_j)$ is considered.

It remains to show that the interobject distances of the UHCS and of the PFNET($r = \infty$, $q = n-1$) are the same after the merging of clusters which occurs at the clustering level α_j and the merging of NSLs when the equivalence class $E(w_j)$ is considered. Since the level

α_j corresponds to the weight w_j in the ordering of distances given by the original weight matrix W, clusters separated by this distance are then merged in the UHCS, and NSLs separated by links having this weight are also merged in Pathfinder. For the UHCS, suppose that clusters C_a and C_b are merged in this step; then each entity in C_a is at a distance of α_j from each entity in C_b, as in Step 3 of the procedure of Johnson. For PFNET($r = \infty$, $q = n$–1), there are NSL_a and NSL_b corresponding to C_a and C_b (by the inductive hypothesis), and the weight w_j is equivalent to the level α_j. Therefore each node in NSL_a is at a distance of w_j from each node in NSL_b (because the use of $r = \infty$ is equivalent to the use of the ultrametric inequality discussed in Johnson, 1967, p. 245). That is, the weight of a path is equal to the largest weight on any link in the path for $r = \infty$. In UHCS and in Pathfinder, previously determined distances (from smaller-valued equivalence classes or α levels) remain unchanged. Distances larger than w_j may be modified in either UHCS or Pathfinder at this step, because the smallest weight connecting two objects is used to represent the distance between those objects; but they must be modified the same in either paradigm because the weights are the same, and the interobject distances are the same at the preceding step (inductive hypothesis). Therefore the interobject distances obtained in each method are equal. □

Given a UHCS, using the minimum method, can a PFNET(r, q) be derived? In general, the answer is no; the distance matrix obtained from application of the UHCS algorithm contains little information about the structure. For example, consider the information available about two clusters in both UHCS and in Pathfinder. Once the merging of the two clusters has occurred, all entities from one cluster take on the same distance from all the entities in the other cluster. At this point, the only ways to construct networks from the distance matrix are (1) to link each node in one cluster with each node in the other cluster; or (2) to provide some arbitrary link(s) between clusters having the appropriate weight(s). One of these might, of course, correspond to the PFNET(r, q), but the correspondence would surely be incidental in terms of the link structure. In general, the UHCS distance matrix cannot yield enough information to construct a PFNET(r, q) which would be computed from the distance matrix.

Theorem 8

Given W_1 and W_2, suppose that the PFNET($r = \infty$, $q = n$–1) computed from W_1 is isomorphic to the PFNET($r = \infty$, $q = n$–1) computed from W_2. Then the UHCS(W_1) computed from W_1 is also isomorphic to the UHCS(W_2) computed from W_2.

Proof:

From Theorem 7, the clusters and distances of PFNET($r = \infty$, $q = n$–1) computed from W_1 are isomorphic to the clusters and distances computed from UHCS(W_1). Similarly, the clusters and distances of PFNET($r = \infty$, $q = n$–1) computed from W_2 are isomorphic to the clusters and the distances computed from UHCS(W_2). Therefore, the clusters and distances of UHCS(W_1) are isomorphic to those of UHCS(W_2), by transitivity, since the PFNETs are isomorphic. □

The converse of Theorem 8 is not true, however, because a unique PFNET cannot be constructed from a UHCS, as discussed in the examples using W_1 and W_2 and illustrated in Figures 6, 7, and 8 early in this section.

Given a PFNET(r, q), obtaining the minimum method UHCS is simple. If the incidence (edge-node) table has been constructed, then this already contains the UHCS in the merging of the node sublists. If the matrix method was used to determine edge membership, with labeling done after generation via a shortest-path algorithm, then the UHCS can be computed as follows.

Procedure UHCS given PFNET(r, q):

1. Disregard (remove) all tertiary edges in PFNET(r, q);
2. Compute the distance matrix D^{n-1} using $r = \infty$ over the primary and secondary edges in the PFNET. This can be done using the matrix operations described in the section entitled Generation Algorithms for PFNETs by constructing a weight matrix W^* in which the weights associated with the primary and secondary edges are entered in the appropriate locations, and other off-diagonal entries in W^* are filled with some value greater than any weight associated with a primary or a secondary edge;
3. D^{n-1} is closely related to the similarity function of Johnson, as discussed previously, and provides sufficient information to construct the tree representation of the UHCS if that is desired. Although D^{n-1} is equivalent to the UHCS, it is sometimes desirable to construct a tree to represent the UHCS, as is done in Johnson (1967, p. 243).

As a simple example, suppose that the distance matrix D^5, obtained for the PFNET in Figure 9, has been computed as described above, and the UHCS is desired for this matrix. To construct the tree shown at the top of Table 3 from the matrix D^5, simply apply the UHCS procedure. The significance of this is that the same clusters are obtained whether or not the distance matrix is obtained from a Pathfinder network. Computationally, it is desirable to compute clusters directly, using one procedure or the other. Some of our Pathfinder programs print the NSLs after each equivalence class is considered, so that the UHCS is obtained as the network is generated.

These results (that the minimum method of UHCS is available from any Pathfinder network) are consistent with the intuitive observations of many of those who have worked with Pathfinder; clustering of similar entities is usually obvious from a drawing or display of the network. The importance of graphical display and some of the power of the additional structural information available in Pathfinder beyond that available in UHCS are illustrated in Esposito (Chapter 6, this volume; and 1988). For example, the basic-level categories of Rosch, Mervis, Gray, Johnson, and Boyes-Braem (1976) have a distinctive representation in Pathfinder networks, with the category name in the center of a "wheel," and the exemplars of the category at the end of the "spokes" of the wheel. Thus the *degree* of the node representing the category name (compared with the number of exemplars) is an appropriate measure to investigate whether or not a category can be considered a basic-level category. The degree of a node is also important in a measure proposed by Collins and Loftus (1975), in which the similarity of two concepts depends upon the number of edges shared by those concepts. Edge labels depend only upon ordinal properties of the data (see Theorem 6), so they offer an appealing approach for further measures to investigate. They are, of course, useful in identifying mintrees and deriving the minimum method UHCS, as shown in this chapter.

Conclusions

We have presented a class of graphs, PFNETs, whose structure is determined algorithmically by proximity data and two parameters, the r-metric and q parameter. The proximity data can either be obtained empirically, or in some domains, determined by objective measurements.

It was shown that each Pathfinder network contains all mintrees and is unique, given a particular weight matrix and particular values for the r-metric and the q parameter. Inclusion properties as r and q vary were demonstrated, so that systematic families of PFNETs are generated as r and q vary. PFNET structure becomes sparser (has fewer links) as either r or q increases.

Structure-preserving properties of the PFNETs under monotonic and multiplicative transformations were proven, so that any PFNET can be used with ratio data, and PFNET($r = \infty$, q) can be appropriately used with ordinal data.

Last, it was shown that the information in a minimum method hierarchical clustering scheme is also available in every PFNET, but that there is not sufficient information in a minimum method HCS to construct a unique PFNET. Pathfinder networks retain information concerning the entities responsible for establishing minimum paths (salient associations), unlike HCS, and thus make structural distinctions unavailable in HCS.