LINEAR AND CIRCULAR UNIDIMENSIONAL SCALING FOR SYMMETRIC PROXIMITY MATRICES

Lawrence J. Hubert[1]
Phipps Arabie[2]
Jacqueline J. Meulman[3]

[1]University of Illinois, Champaign, U.S.A.

[2]Faculty of Management, Rutgers University, Newark, U.S.A.

[3]Department of Data Theory, Leiden University, Leiden, The Netherlands

LINEAR AND CIRCULAR UNIDIMENSIONAL SCALING FOR SYMMETRIC PROXIMITY MATRICES

Abstract

The tasks of linear and circular unidimensional scaling can be defined by the attempt to represent the entries in a symmetric proximity matrix through distances between a set of object locations defined either along a linear continuum or around a closed, circular continuum. These two scaling tasks are approached through a least-squares optimization strategy based on a combination of combinatorial search and iterative projection techniques. Extensions are provided for considering multiple linear or circular unidimensional structures, and to the inclusion of several representational alternatives offered by (restricted) additive tree models. Two published data sets are used to illustrate the results obtainable from the optimization method being proposed.

Key words: linear unidimensional scaling, circular unidimensional scaling, least-squares matrix approximation, additive trees, ultrametrics, centroid metrics.

1. Introduction

Explicit discussions of metric unidimensional scaling based on object placement along a linear continuum and the use of a least-squares loss function have appeared in the psychometric literature for three decades (e.g., see Guttman, 1968; de Leeuw & Heiser, 1977; Defays, 1978; Hubert & Arabie, 1986, 1988; Poole, 1991; van Schuur, 1987; Groenen, 1993; Pliner, 1986, 1996). In its simplest formulation, we are given some n × n symmetric matrix, $\mathbf{P} = \{p_{ij}\}$, containing (nonnegative) pairwise proximity values (devoid of missing entries, except along the principal diagonal, which will be ignored in the analyses) among a collection of n objects, $S = \{O_1,...,O_n\}$, and wish to find a set of coordinates, $x_1,...,x_n$, minimizing

$$\sum_{i<j} (p_{ij} - |x_j - x_i|)^2 \ . \qquad (1)$$

It is well-known that the optimization task represented through (1) is inherently combinatorial (e.g., Defays, 1978; de Leeuw & Heiser, 1977), and the crucial element in its solution lies in identifying an appropriate ordering of the n objects along the continuum. Given such an ordering, the actual coordinate estimation can be performed through simple formulae. It is also now well-understood that any direct adaptation of the common gradient-based optimization strategies lead to rather dismal results in identifying those object orderings that will generate the minimal values of (1) (see Hubert & Arabie, 1986, 1988; Pliner, 1996). Modifications are possible, as in the suggestion of Pliner (1996) to use smooth approximations and eliminate the major difficulty caused by the "knife-edge" gradients produced from the absolute values of coordinate differences. But, even with such computational advances, the basic combinatorial nature of the problem remains, and the linear unidimensional scaling task reduces to the search for an appropriate object ordering along the continuum. In theory, we could merely enumerate the values for the loss function in (1) over n!/2 object orderings and pick the smallest to solve the linear unidimensional scaling problem.[1] However, given the enormous size that n!/2 can become for even moderate n, and although some computational savings might be obtained

through partial enumeration strategies such as dynamic programming (e.g., see Hubert & Arabie, 1986), it is obvious that for general use when n is, say, greater than 20, alternative search strategies are needed. These search strategies should be based on good heuristic principles that although not guaranteeing the absolute best solution, are nevertheless capable of producing collections of the better possible solutions with reasonable computational effort.

One of the lesser aims of the present paper, in addition to the more unique development of circular unidimensional scaling introduced below, is to revisit the (linear) unidimensional scaling (LUS) problem but with a slightly more general least-squares loss function of the form

$$\sum_{i<j} (p_{ij} + c - |x_j - x_i|)^2 ,$$

or equivalently, (2)

$$\sum_{i<j} (p_{ij} - \{|x_j - x_i| - c\})^2 ,$$

where c is some constant to be estimated along with the coordinates $x_1,...,x_n$. The optimization task implicit in the use of (2) can be interpreted in either of two ways, as reflected by the equality of the two forms of the loss function given in (2): (a) the interpoint distances among a set of n coordinates along a line, $\{|x_j - x_i|\}$, are being fitted to a constant translation of the originally given proximities, $\{p_{ij} + c\}$; or (b) a generalization from the usual unidimensional model to one of the form $\{|x_j - x_i| - c\}$ is being fitted to the originally given proximities $\{p_{ij}\}$. Although these two interpretations will not affect how we proceed at this point, the second interpretation that includes the additive constant as a part of

---

[1] The division by 2 appears in the number of object orderings requiring enumeration since the left-right reflection for any ordering generates the same value for the loss function in (1). The coordinate values for a complete reversal are constructed from those for the original by starting at the arbitrary coordinate of 0.0 and finding the successive coordinates from left to right by adding in the reverse order the adjacent nonnegative separations identified for the original ordering.

the model will become relevant in how generalizations are framed in Section 4 to the fitting of multiple unidimensional structures to a given proximity matrix. In any case, the presence of the constant c in (2) obviates the need to impose any type of nonnegativity constraints on the proximities originally given (in fact, for consistency of presentation, we routinely begin with proximities standardized to a mean of zero and standard deviation of one). Although this step may seem like a minor modification at first, the presence of negative proximities can produce rather serious difficulties in discussions of linear unidimensional scaling, e.g., they cannot be accommodated in Pliner's (1996) suggestion for a smooth gradient approximation, and their presence invalidates a number of the convenient properties or characterizations that certain optimization heuristics would otherwise possess. In a number of closely related optimization tasks, specialized algorithms have been devised to deal with the possibility of negative proximities, however they might arise (see Heiser, 1989, 1991).

The second and more major purpose of the present paper is to discuss what will be called circular unidimensional scaling (CUS). Here, the objective is to place the n objects around a closed continuum, such that the reconstructed distance between each pair of objects, defined by the minimum length over the two possible paths that join the objects, reflects the originally given proximities as well as possible. Explicitly, and in analogy with the loss function for linear unidimensional scaling (LUS) in (2), we wish to find a set of coordinates, $x_1,\ldots,x_n$, and an $(n+1)^{st}$ value, $x_0 \geq |x_j - x_i|$ for all $1 \leq i \neq j \leq n$, minimizing

$$\sum_{i<j} (p_{ij} + c - \min\{|x_j - x_i|, x_0 - |x_j - x_i|\})^2 ,$$

or equivalently, (3)

$$\sum_{i<j} (p_{ij} - [\min\{|x_j - x_i|, x_0 - |x_j - x_i|\} - c])^2 ,$$

where c is again some constant to be estimated (and which by its placement in the two forms given in (3), generates the two analogous interpretations as discussed for the use of LUS and (2)). The value x_0 represents the total length of the closed continuum, and the expression, $\min\{|x_j - x_i|, x_0 - |x_j - x_i|\}$,

gives the minimum length over the two possible paths joining objects O_i and O_j. In theory, the CUS task could again be solved by complete enumeration of the loss function in (3) over a finite but again typically enormous set for even moderate n. Here, we have all possible (n-1)!/2 distinct object orderings around a closed continuum (see footnote 1 for a comment on the divisor of 2, which applies here as well), and for each such ordering, all the possible inflection patterns for when the directionality of the minimum distance calculation changes for the object pairs. Obviously, for general use and in analogy to LUS, some type of search strategy is called for to generate the better orderings and inflection patterns around a closed continuum but with feasible computational effort.

As indicated by the list of references given earlier, LUS has a long history in the psychometric literature. In contrast, the seemingly obvious and direct extension to CUS as framed in (3) has been ignored, although there is a substantial literature on the related topic of forcing object placements in a two-dimensional Euclidean scaling solution to lie on a circular (or "circumplex") structure (e.g., see Borg & Lingoes, 1979, 1980; de Leeuw & Heiser, 1980; Bentler & Weeks, 1978; Heiser & Meulman, 1983; Lee & Bentler, 1980; Lee, 1984; Cox & Cox, 1991), or on various circumplex models specific to certain kinds of proximities, such as correlations or covariances (e.g., Browne, 1992; Cudek, 1986; Wiggins, Steiger, & Gaelick, 1981). We will approach the CUS task directly, using the loss function in (3), and according to its basic combinatorial character in the selection of an optimization strategy. These former efforts, however, can be interpreted as indirect attacks on the CUS task through the fitting of (transformed) proximities by nonlinear monotonic functions of the minimum distances calculated around a circular structure. For example, in a two-dimensional Euclidean scaling context, the set of induced Euclidean distances between the points on a perfect circular structure are monotonic with respect to the (path) distances around a closed circular structure. Explicitly, using the notation of the loss function in (3), the fitting of Euclidean distances between points lying on a circular structure could be rephrased using the loss function below:

$$\sum_{i<j} (p_{ij} + c - (\pi/x_0)^{-1}\sin[(\pi/x_0)\min\{|x_j - x_i|, x_0 - |x_j - x_i|\}])^2 ,$$

where $(\pi/x_0)^{-1}\sin[(\pi/x_0)\min\{|x_j - x_i|, x_0 - |x_j - x_i|\}]$ is the (two-dimensional) Euclidean distance between the points on the circular structure for objects O_i and O_j. In short, a variety of these kinds of connections exist between various transformed versions of CUS and the fitting of circular structures through constrained multidimensional scaling strategies in two dimensions; but as mentioned, our approach deals with CUS directly through the loss function in (3) without any imposed transformation of the distances around a circular structure. In a very particular sense, the use of (3) may be considered to define a more fundamental task since even though we have used the label of CUS to refer to the optimization task in (3), if a physical model were constructed for the closed continuum and the corresponding object placements, the model could be topologically deformed in any of a variety of ways, e.g., into a rectangle, ellipse, rhombus, or similar figure. What is relevant are distances around the closed continuum, and the actual physical model could take on an infinite number of possible shapes.

The current paper is organized around the two optimization problems posed by LUS and CUS in Sections 2 and 3, respectively. In both cases, there are two main subtasks to be discussed. The first is to obtain an appropriate object ordering along a line or around a closed continuum, and in the case of a closed continuum, the additional relative placement of the objects according to the directionality of minimum distance calculations. We discuss the relevant heuristic strategies of combinatorial search in some detail. Second, once these orderings and relative placements have been identified, the estimation of the additive constant in both loss functions (2) and (3), as well as the identification of the actual coordinates, proceeds through a process of alternating projections onto two closed convex sets. One is defined by the set of all translations of the original proximity values by a constant; the second closed convex set is defined by the collection of all coordinate structures consistent with the given object ordering (and in the case of CUS, the relative object placement that characterizes the

directionality of minimum distance calculations around the closed continuum). In the specific contexts we consider, the process of alternating projections is convergent (see Cheney & Goldstein, 1956) and generates a final solution defined by two points within the two convex sets that are closest to each other.

The emphasis in the present paper is on a single symmetric proximity matrix defined for one finite set of objects to be fitted either by a LUS or CUS model. However, Section 4 will also develop a fairly direct extension to the use of sums of such linear or circular structures that can be fitted to a single proximity matrix through successive residualizations of the given matrix. Thus, in the case of sums of linear unidimensional scalings, we have the task of multidimensional scaling in the city-block metric, and the method we will use of successive residualizations would stand in algorithmic competition with, for example, the simultaneous combinatorial optimization strategy suggested by Hubert, Arabie, and Hesson-Mcinnis (1992), or the majorization approach specialized to the city-block metric developed by Groenen, Mathar, and Heiser (1995). For multiple circular structures, there are apparently no comparable extant methods. Section 4 also suggests several types of tree structures that could be used in conjunction with LUS or CUS, such as ultrametrics, centroid metrics, or additive trees, following the discussion in Hubert and Arabie (1995b) on fitting the latter to a single symmetric proximity matrix through a heuristic optimization strategy based on iterative projection. We will leave to a companion paper the extension of the LUS and CUS models to two-mode pairwise proximity data defined between two disjoint object sets, i.e., to a linear or circular unfolding context, and to the fitting of two-mode proximity data by sums of linear, circular, or various tree structures.

As a final preliminary note on the data that will be used throughout for numerical illustrations, two rather well-known proximity matrices are given in Tables 1(a) and 2. The first gives a 10×10 proximity matrix for the 10 Morse Code symbols that represent the first ten digits: (0: ——; 1: •——; 2: ••—; 3: •••—; 4: ••••-; 5: •••••; 6: -••••; 7: —•••; 8: ——••; 9: ———•). The upper-triangular entries in

Table 1(a) have a dissimilarity interpretation and are defined for each object pair by 2.0 minus the sum of the two proportions for a group of subjects used by Rothkopf (1957) representing "same" judgments to the two symbols when given in the two possible presentation orders. The lower-triangular entries are a transformation of the latter proximities and have mean zero and standard deviation one, and will be the data actually used in the numerical illustrations. Based on previous multidimensional scalings of the complete data set involving all of the Morse code symbols and in which the data of Table 1(a) are embedded (e.g., see Shepard, 1963; Kruskal & Wish, 1978), it might be expected that the symbols for the digits would form a clear linear unidimensional structure that would be interpretable according to a regular progression in the number of dots to dashes. It turns out, as discussed in greater detail below, that a circular model (or actually, a sum of circular structures) is probably more consistent with the patterning of the proximities in Table 1(a) than are representations based on linear unidimensional scalings. Table 1 also includes a second proximity matrix in panel (b) that will be used only in Section 4.1 to illustrate structural representations based on various tree models. The stimuli in Table 1(b) are the eight Morse code symbols containing three dots or dashes, and which represent eight letters: (d: -••; g: —•; k: -•-; o: —; r: •-•; s: •••; u: ••-; w: •—). The entries in Table 1(b) have the same interpretation as those in Table 1(a).

{Tables 1 and 2 here}

The second 15 × 15 proximity matrix in Table 2 is originally from Levelt, van de Geer, and Plomp (1966) and pertains to the perceived similarities of tonal intervals; these data have been discussed by Shepard (1974), Borg and Lingoes (1979, pp. 213-218), Hubert and Arabie (1989), among others, and specifically with respect to the appropriateness of a simple LUS model as a reasonable representation. The 15 stimuli each consist of two simultaneously heard tones having a certain frequency ratio, as indicated in Table 2, but with a fixed mean of 500 cycles per second to eliminate pitch as a source of variation. The stimuli were presented as triads in a balanced

incomplete-block triadic scheme to four subjects who indicated for each triad the most similar pair (given a score of 2 points), the least similar pair (given a score of 0 points), and the third intermediate pair (given a score of 1 point). Summing over the four subjects, each stimulus pair could have an aggregate score ranging from 0 (least similar) to 32 (most similar); the entries in the upper-triangular portion of Table 2 have a dissimilarity interpretation by using 32 minus the Levelt et al. (1966) similarity values (but incorporating the one correction reported by Borg and Lingoes, 1979); the lower-triangular entries are these latter values again standardized to have mean zero and standard deviation one. The most obvious LUS model for these data would be defined by an ordering along a continuum according to frequency ratios, and this substantive interpretation is advocated by Shepard (1974). As we will see in the sections to follow, even though a LUS structure according to frequency ratios does fairly well in representing the Table 2 proximities, a circular variant does slightly better according to a variance-accounted-for measure that will be introduced.

2. Linear Unidimensional Scaling (LUS)

As noted in the introduction, the LUS task as framed by the loss function in (2) can be separated into two parts: (a) the identification of an ordering of the n objects along the continuum that, for convenience, we denote by a permutation of the first n integers, $\varphi(\cdot)$, such that $\varphi(i) = j$ if object O_j is placed at position i; and (b) a subsequent estimation of the additive constant and set of coordinates to fit the proximities. We discuss these two subtasks in the reverse order.

2.1. The Estimation of c and $\{|x_j - x_i|\}$ for a Fixed Permutation

Given a fixed permutation, $\varphi(\cdot)$, we denote the set of all $n \times n$ matrices that are additive translations of the off-diagonal entries in the reordered symmetric proximity matrix $\{p_{\varphi(i)\varphi(j)}\}$ by Δ_φ, and let Ξ be the set of all $n \times n$ matrices that represent the interpoint distances between all pairs of n

coordinate locations along a line. Explicitly,

$$\Delta_\varphi \equiv \{\{q_{ij}\} \mid q_{ij} = p_{\varphi(i)\varphi(j)} + c, \text{ for some constant } c,\ i \neq j;\ q_{ii} = 0,\ 1 \leq i \leq n\}\ ;$$

$$\Xi \equiv \{\{r_{ij}\} \mid r_{ij} = |x_j - x_i| \text{ for some collection of n coordinates, } x_1,\ldots,x_n, \text{ such that } x_1 \leq x_2 \leq \ldots \leq x_n,$$

and for which it may be assumed without loss of generality that $x_1 \equiv 0.0\}$.

Alternatively, we may define Ξ through a set of linear inequality/equality constraints as

$$\Xi \equiv \{\{r_{ij}\} \mid 0 \leq r_{ij} \text{ for } 1 \leq i \neq j \leq n;\ 0 = r_{ii} \text{ for } 1 \leq i \leq n;\ r_{ji} = r_{ij} \text{ for } 1 \leq i \neq j \leq n;$$

$$r_{i(i+1)} + \ldots + r_{(j-1)j} = r_{ij} \text{ for } 1 \leq i < j \leq n, \text{ where } 2 \leq j\text{-}i\}\ . \qquad (4)$$

Both Δ_φ and Ξ are closed convex sets (in a Hilbert space), and thus, given any n × n symmetric matrix with a zero main diagonal, its projection onto either Δ_φ or Ξ exists; i.e., there is a (unique) member of Δ_φ or Ξ at a closest (Euclidean) distance to the given matrix (e.g., see Cheney & Goldstein, 1959) (the construction of these projections will be discussed below). Moreover, if a procedure of alternating projections onto Δ_φ and Ξ is carried out (where a given matrix is first projected onto one of the sets, and that result is then projected onto the second, which result is in turn projected back onto the first, and so on), the process is convergent and generates members of Δ_φ and Ξ that are closest to each other (again, this last statement is justified in Cheney and Goldstein, 1959, Theorems 2 and 4).

Given any n × n symmetric matrix with a main diagonal of all zeros, which we denote arbitrarily as $\mathbf{U} = \{u_{ij}\}$, its projection onto Δ_φ may be obtained by a simple formula for the sought constant c. Explicitly, the minimum over c of

$$\sum_{i<j} (p_{\varphi(i)\varphi(j)} + c - u_{ij})^2\ ,$$

is obtained for

$$\hat{c} = (2/n(n-1)) \sum_{i<j} (u_{ij} - p_{\varphi(i)\varphi(j)})\ ,$$

and thus, this last value defines a constant translation of the proximities necessary to generate that member of Δ_φ closest to $\mathbf{U} = \{u_{ij}\}$.

Given any n × n symmetric matrix, again with a main diagonal of all zeros, that we now denote

arbitrarily as $\mathbf{V} = \{v_{ij}\}$ (but which in our applications will generally have the form $\{v_{ij}\} = \{p_{\varphi(i)\varphi(j)} + c\}$ for $i \neq j$ and some constant c), its projection onto Ξ is somewhat more involved and requires minimizing

$$\sum_{i<j} (v_{ij} - r_{ij})^2 ,$$

over r_{ij}, where r_{ij} is subject to the linear inequality/equality constraints in (4). This is a (classic) quadratic programming problem for which a wide variety of optimization techniques have been published. We adopt the iterative projection strategy discussed by Dykstra (1983), Boyle and Dykstra (1986), and Han (1988), among others, and for which fairly detailed presentations and applications are now available in the psychometric literature (e.g., see Hubert & Arabie, 1994, 1995a, 1995b). Briefly summarized, we begin with the matrix $\{v_{ij}\}$, and successively modify it by considering in turn each of the individual constraints given in (4); if a constraint is not satisfied, the matrix we have up to this point is projected onto the set of matrices satisfying the constraint, and that projection becomes the matrix evaluated for satisfaction of the next constraint in the sequence. The complete set of constraints are cycled through repetitively until the whole process converges, but there is one important additional observation on how the constraints are evaluated: each time a constraint is revisited, any changes from the previous time that constraint was considered are "undone" (i.e., the earlier changes are restored to the relevant matrix entries) before the constraint is re-evaluated.

There are four classes of constraints in (4) to be assessed in this iterative process, and we note below what the projection would be if the constraint were <u>not</u> satisfied by the matrix being carried forth to that point in the iterative process:

(A) $0 \leq r_{ij}$ for $1 \leq i \neq j \leq n$:

set the $(i,j)^{th}$ entry equal to 0.

(When violated, the nonnegativity constraint on r_{ij} is enforced by setting the corresponding matrix entry equal to zero.)

(B) $0 = r_{ii}$ for $1 \leq i \leq n$:

set the $(i,i)^{th}$ entry equal to 0.

(When violated, the constraint that the i^{th} diagonal entry r_{ii} is zero is enforced by setting the corresponding matrix entry equal to zero.)

(C) $r_{ji} = r_{ij}$ for $1 \leq i \neq j \leq n$:

set the (i,j) and $(j,i)^{th}$ entries equal to their arithmetic average.

(When violated, the constraint of symmetry for r_{ij} and r_{ji} is enforced by setting the two corresponding matrix entries equal to their average.)

(D) $r_{i(i+1)} + \ldots + r_{(j-1)j} = r_{ij}$ for $1 \leq i < j \leq n$, where $2 \leq j\text{-}i$:

add a value d to the $(i,i+1),\ldots,(j-1,j)^{th}$ entries, and subtract d from the $(i,j)^{th}$ entry, where d is $[1/(j-i+1)]$ times the difference between the current $(i,j)^{th}$ entry and the sum of the current $(i,i+1),\ldots,(j-1,j)^{th}$ entries.

(When violated, the constraint that distances are additive along the continuum [and thus, r_{ij} for $2 \leq j\text{-}i$ should be the sum of the intermediary nonnegative separations $r_{i(i+1)},\ldots,r_{(j-1)j}$] is enforced by subtracting a constant [which could be negative] from the $(i,j)^{th}$ entry and adding this same constant to each of the entries corresponding to the intermediary values; the defined constant, d, produces the equality and the desired projection.)

Example. As an illustration of how the process of estimation proceeds, we use the Levelt et al. (1966) proximity data in their standardized form in the lower-triangular portion of Table 2, and consider the fixed permutation to be the identity, $\varphi(i) = i$, $1 \leq i \leq 15$, representing an object ordering according to frequency ratios. (We might note that although the identity permutation induces an obvious substantive interpretation for the ordering of the stimuli, it is considered here to be chosen a priori and merely to illustrate the results of the estimation process that would be appropriate for any fixed

permutation. As will be seen later, however, the identity permutation is also identified through the search process for determining the best ordering along the continuum.) Beginning with these data, and successively projecting onto the sets Ξ and Δ_{φ}, provides the LUS representation reported in Figure 1, where the first object is given the (arbitrary) coordinate of 0.0, and the additive constant, c, has a final value of 1.239. The convergence criterion set here (and used consistently throughout the examples given in the manuscript) was based on the sum of absolute changes from one iteration to the next in the projections onto both Ξ and Δ_{φ} being less than 10^{-5}; this same convergence criterion was also imposed for each of the individual projections onto Ξ that were themselves obtained through a process of iterative projection.

In addition to a descriptive representation of the LUS structure, such as in Figure 1, a summary measure of "variance accounted for" (VAF) can also be provided. In general, VAF is defined as

$$\text{VAF} = 1 - \{ \sum_{i<j} (p_{\varphi(i)\varphi(j)} + c - |x_j - x_i|)^2 \; / \; \sum_{i<j} (p_{ij} - \overline{p})^2 \} \; , \tag{5}$$

where $\overline{p}$ is the mean of the proximity values (in this example, $\overline{p}$ is zero given the usage of standardized proximities). For the Levelt et al. LUS reported in Figure 1, the VAF based on the fixed identity permutation representing an ordering according to frequency ratios has a value of 66.22%.

{Figure 1 here}

2.2. The Obtaining of Object Orderings Along a Linear Continuum

As noted above, once a permutation $\varphi(\cdot)$ is available, the constant, c, and the coordinates, $x_1,\ldots,x_n$, can be constructed directly using an alternating projection strategy to minimize the loss function in (2). Thus, a crucial element in LUS is generating the permutation to allow this final optimization process to proceed. As a heuristic that can be applied for even rather large n, and which appears to provide a very successful combinatorial search strategy, we will rely on the quadratic assignment task (see Hubert & Schultz, 1976) and the iterative improvement of an arbitrarily given (e.g., random)

permutation. To be explicit, suppose we are given a (random) permutation, $\varphi_0(\cdot)$, and a fixed $n \times n$ (target) matrix, $\mathbf{T} = \{t_{ij}\}$ (that initially is defined by n positions equally-spaced along a line, i.e., $t_{ij} = |i - j|$ for $1 \leq i,j \leq n$); consider the cross-product statistic

$$\Gamma(\varphi_0) = \sum_{i<j} p_{\varphi_0(i)\varphi_0(j)} t_{ij} , \tag{6}$$

using the given proximity matrix $\{p_{ij}\}$. Beginning with $\varphi_0(\cdot)$, we will attempt to create a sequence of permutations with ever increasing cross-product values using a collection of local operations (noted below), i.e., a sequence of permutations, $\varphi_0, \varphi_1, \ldots, \varphi_K$, such that $\Gamma(\varphi_0) < \Gamma(\varphi_1) < \ldots < \Gamma(\varphi_K)$, and no local operation on the final permutation, $\varphi_K(\cdot)$, can increase the cross-product measure. The local operations relied on fall into three general classes: (a) all pairwise interchanges of objects; (b) all insertions of k ($1 \leq k \leq n-1$) consecutive objects between any two existing objects or at the beginning and end of the permutation; (c) all complete order reversals of k ($1 \leq k \leq n-1$) consecutive objects. Thus, the final permutation, $\varphi_K(\cdot)$, using the target initially given, cannot be improved upon through application of any of these local operations.

The outcome of the process just described is a permutation, $\varphi_K(\cdot)$, that is then considered to be fixed in minimizing the loss function in (2). The new target matrix $\mathbf{T} = \{t_{ij}\} = \{|x_j - x_i|\}$ generated in the minimization of (2) can now be used to re-initialize the whole process, beginning with $\varphi_K(\cdot)$, and an attempt made to improve upon the latter through local operations. Eventually, the process terminates, i.e., for a newly constructed target matrix, no improvement is possible using local operations on the permutation used to construct that specific target matrix.

Several observations should be made about the process just suggested for generating reasonable object permutations along the continuum. First, it might be noted that since a cross-product measure is used, this process is invariant under (positive) linear transformations of the original proximities. Thus, without loss of any generality, we may begin with standardized proximities and include (or not) whatever additive constant is estimated in minimizing the loss function in (2). Second, the possibility

should be kept in mind, as in the application of any combinatorially-based heuristic strategy, that different results might occur (i.e., local optima may be observed) depending on different starting permutations, and thus, any analysis should always include a discussion of this possibility.

For the LUS task, such variation appears to be relatively rare for most data sets, given the extensive set of local operations implemented in identifying object permutations for use in the optimization of (2) (this point will be supported by the numerical illustrations). However, in the combinatorial search extensions to be pursued in the context of CUS, somewhat more variation will generally be present, and it is particularly important in this latter context to have a clear sense of the differences that may be possible in the solutions generated from different starting permutations. Finally, it should be remarked that as a computational convenience, cross-product indices have been relied on to identify object orderings, as opposed to considering the least-squares loss function itself and attempting to locate better orderings and target matrix combinations through the direct minimization of that function. A particular anomaly that theoretically could occur— of an increase for the least-squares loss function in the process of using cross-products— has never been observed over the extensive computational experience the authors have had with this strategy (explicitly, the anomaly could occur if a new ordering constructed from a particular target matrix produces a new target matrix with a smaller sum of squared entries than the previous target and this change is not compensated for by a large enough increase in the cross-product index). The much greater computational simplicity in using cross-product indices as opposed to the complete least-squares loss function, as well as the apparent nonoccurrence of the possible anomaly just mentioned, supports the cross-product approach in the optimization paradigm.

2.3. Numerical Illustrations

To illustrate the results of the LUS optimization strategy just described, we use the data of Tables

1(a) and 2. For the standardized Morse code proximities, using 100 different randomly generated starting permutations, only one solution was produced, as given in Figure 2. The VAF for this LUS structure is 58.57%, with the additive constant c estimated as 1.208. It is apparent from the patterning of the Morse code symbols along the continuum that no clear progression in the number of dots to dashes exists as one moves from left to right, contrary to what prior substantive expectations may have been. In fact, the specific ordering we see in Figure 2 is best explained using a CUS structure for these same data given later (as will be noted at that point), i.e., through some type of projection onto a linear continuum passing through a circular structure. (Given that the ten Morse code stimuli representing the digits include pairs of stimuli containing one, two, three, and four dots [or dashes], there are 16 possible distinct orderings along the continuum that would represent perfect progressions in the number of dots to dashes. Explicitly, given the sequence: ——, {•—;—•}, {••—;—••}, {•••—;—•••}, {••••-;-••••}, •••••, each pair within braces could be considered in either order, producing $2^4 = 16$ possible sequences from —— to ••••• that satisfy a progression from five dashes to five dots. Minimizing (2) with respect to each of these 16 fixed permutations produced a best VAF of 34.64%, which is rather different from the 58.57% found for the result provided in Figure 2.)

For the Levelt et al. standardized data of Table 2, and 100 different randomly generated starting permutations, only one solution was again produced, which is exactly the same structure given earlier in Figure 1 based on the fixed identity permutation according to frequency ratios. The single permutation identified over the 100 starts was the identity, except for the adjacent interchange of the positions for objects 14 and 15; however, given the identical estimated coordinates for these two stimuli, Figure 2 still applies along with the same VAF and additive constant reported earlier.

{Figure 2 here}

3. Circular Unidimensional Scaling (CUS)

The CUS task as characterized by the loss function in (3) can be considered in two stages as with the presentation of LUS. One subtask is the identification of an ordering of the n objects around a closed continuum that again will be denoted by a permutation of the first n integers, $\varphi(\cdot)$, such that $\varphi(i) = j$ if object O_j is placed at position i; here, position 1 is arbitrarily specified at some point along the closed continuum, and the order of the positions from 1 to n is, for convenience, taken clockwise. In addition to $\varphi(\cdot)$, a set of inflection points must be identified for the n positions to indicate where the minimum distance calculation must change direction around the closed continuum. Explicitly, a set of n-1 integers, $1 \leq k_1 \leq \ldots \leq k_{n-1} \leq n$, is sought, where k_i is associated with position i, $1 \leq i \leq n-1$. For positions $i < j$, the minimum distance is in the clockwise direction when $j \leq k_i$, and in the counterclockwise direction when $j > k_i$ (we note that an integer k_n for position n is unnecessary, and any k_i equal to n merely indicates that for all positions j, for $i < j$, the minimum distance is always in the clockwise direction). The second subtask, once given $\varphi(\cdot)$ and $k_1,\ldots,k_{n-1}$, is the estimation of the set of coordinates and the additive constant c to fit the proximities. We again discuss these two subtasks in the reverse order.

3.1. The Estimation of c and $\min\{|x_j - x_i|, x_0 - |x_j - x_i|\}$ for a Fixed Permutation and Set of Inflection Points

For notational convenience, the set of all n × n matrices that are additive translations of the off-diagonal entries in the reordered proximity matrix, $\{p_{\varphi(i)\varphi(j)}\}$, will again be denoted by Δ_φ (see Section 2.1); the set of all n × n matrices that represent the distances around the closed continuum based on the inflection points $k_1,\ldots,k_{n-1}$ will be more fully denoted by $\Xi(k_1,\ldots,k_{n-1})$ and explicitly defined as follows:

$\Xi(k_1,...,k_{n-1}) \equiv \{\{r_{ij}\} \mid r_{ij} = |x_j - x_i|$ for $i < j \leq k_i$; $r_{ij} = x_0 - |x_j - x_i|$ for $i < j$, $j > k_i$; $r_{ji} = r_{ij}$ for $1 \leq i < j \leq n$; $r_{ii} = 0$ for $1 \leq i \leq n$; for some collection of coordinates, $x_1,...,x_n$, and an $(n+1)^{st}$ value, x_0, where $x_1 \leq ... \leq x_n \leq x_0$; $x_1 \equiv 0.0$; and $|x_j - x_i| \leq x_0 - |x_j - x_i|$ for $i < j \leq k_i$; $|x_j - x_i| \geq x_0 - |x_j - x_i|$ for $i < j$, $j > k_i\}$.

(As noted in this definition, the first position, x_1, is specified without loss of generality to be 0.0; the value, x_0, can either be interpreted as the length of the closed continuum or as a second coordinate value attached to the first position but taken in the clockwise direction.) Alternatively, $\Xi(k_1,...,k_{n-1})$ can be defined through a set of linear inequality/equality constraints as

$\Xi(k_1,...,k_{n-1}) =$

$\{\{r_{ij}\} \mid 0 \leq r_{ij}$ for $1 \leq i \neq j \leq n$; $0 = r_{ii}$ for $1 \leq i \leq n$; $r_{ji} = r_{ij}$ for $1 \leq i \neq j \leq n$; $r_{i(i+1)} + ... + r_{(j-1)j} = r_{ij}$ for $1 \leq i < j \leq k_i$, where $2 \leq j-i$; $r - (r_{i(i+1)} + ... + r_{(j-1)j}) = r_{ij}$ for $1 \leq i < j \leq n$, $j > k_i$, $2 \leq j-i$; and $r = r_{12} + ...+ r_{(n-1)n} + r_{1n}\}$. (7)

Given Δ_φ and $\Xi(k_1,...,k_{n-1})$, the process of alternating projections onto Δ_φ and $\Xi(k_1,...,k_{n-1})$ would proceed exactly as in LUS. The only change to note is a modification in the discussion of Section 2.1 for the (D) class of constraints and what the projection would be if the constraint were <u>not</u> satisfied by the matrix being carried forth to that point in the iterative process. Explicitly, the (D) class should now be characterized as follows:

(D) $r_{i(i+1)} + ... + r_{(j-1)j} = r_{ij}$ for $1 \leq i < j \leq k_i$, where $2 \leq j-i$:

add a value d to the $(i,i+1),...,(j-1,j)^{th}$ entries, and subtract d from the $(i,j)^{th}$ entry, where d is $[1/(j-i+1)]$ times the difference between the current $(i,j)^{th}$ entry and the sum of the current $(i,i+1),...,(j-1,j)^{th}$ entries.

$r - (r_{i(i+1)} + ... + r_{(j-1)j}) = r_{ij}$ for $1 \leq i < j \leq n$, $j > k_i$, $2 \leq j-i$, where $r = r_{12} + ... + r_{(n-1)n} + r_{1n}$:

add a value e to all entries $(1,2),(2,3),...,(n-1,n),(1,n)$ <u>except</u> for $(i,i+1),...,(j-1,j)$ and subtract e from the $(i,j)^{th}$ entry, where e is $[1/(n-(j-i)+1)]$ times the difference between the current $(i,j)^{th}$ entry and

the sum of all entries in (1,2),(2,3),...,(n-1,n),(1,n) except for (i,i+1),...,(j-1,j).

(When violated, the constraint that distances are additive around the closed continuum [and thus, r_{ij} for $2 \leq j\text{-}i$ should be either the sum of the clockwise intermediary nonnegative separations $r_{i(i+1)},\ldots,r_{(j-1)j}$, or the sum of the counterclockwise intermediary separations $r - (r_{i(i+1)} + \ldots + r_{(j-1)j})$, where $r = r_{12} + \ldots + r_{(n-1)n} + r_{1n}$] is enforced by subtracting a constant [which could be negative] from the $(i,j)^{th}$ entry and adding this same constant to each of the entries corresponding to the intermediary separation values; the defined constant, d or e, produces the equality and the desired projection.)

Example. As an illustration of how this estimation proceeds for CUS (and also to give a result to be revisited in Section 3.3 on numerical illustrations), we use the standardized Morse code data in the lower-triangular portion of Table 1; define the fixed permutation to be the identity, $\varphi(i) = i$, $1 \leq i \leq 10$, and the integers, $k_1 = 5$, $k_2 = 6$, $k_3 = 7$, $k_4 = \ldots = k_9 = 10$ (because $k_4 = \ldots = k_9 = 10$ (=n), we note that for all $4 \leq i < j \leq 10$ (=n), the minimum distance calculation will be consistently in the clockwise direction). Successively projecting onto the sets $\Xi(5,6,7,10,10,10,10,10,10)$ and Δ_φ generates the CUS given in Figure 3, where the first position is given the coordinate of 0.0 and, if taken in the clockwise direction, a second coordinate value of $x_0 = 6.22$, denoting the total length of the closed continuum. The additive constant, c, has a final value of 1.686, and the VAF is 72.77%. As is apparent in Figure 3, the identity permutation provides a nicely interpretable ordering of the Morse code symbols around a circular structure involving a regular replacement of dashes by dots moving clockwise until the symbol containing all dots is reached, and then a subsequent replacement of the dots by dashes until the initial symbol containing all dashes is reached.

{Figure 3 here}

3.2. Obtaining Object Orderings and Inflection Points Around a Closed Continuum

Identifying an object ordering around a closed continuum to be used in the minimization of the loss function in (3) follows exactly the same pattern as for LUS. The cross-product statistic in (6) is again maximized but with a different $n \times n$ (target) matrix, $\mathbf{T} = \{t_{ij}\}$, initially defined by n positions equally-spaced around a closed continuum, i.e., $t_{ij} = \min\{|i - j|, n - |i - j|\}$ for $1 \leq i,j \leq n$ (as in LUS, this target is eventually replaced, now by $t_{ij} = \min\{|x_j - x_i|, x_0 - |x_j - x_i|\}$ based on the outcome of the minimization of (3)). Given some best permutation, $\varphi_K(\cdot)$, obtained through the initial target and set of local operations on some randomly given initial permutation, a collection of inflection points, $k_1,...,k_{n-1}$, still must be generated before the optimization of (3) can continue. This latter task will be approached through a heuristic application of an iterative projection strategy of the same general type developed by Hubert and Arabie (1995b) for the fitting of various graph-theoretic structures to a symmetric proximity matrix.

To attempt an identification of $k_1,...,k_{n-1}$ given the permutation $\varphi_K(\cdot)$, we begin with the reordered proximity matrix $\{p_{\varphi_K(i)\varphi_K(j)}\}$, and initialize a process of iterative projection onto the class of constraints given in (7) but with one exception necessitated by the fact that an appropriate set of values for $k_1,...,k_{n-1}$ is not yet known. Explicitly, when considering a pair of positions $i < j$ ($2 \leq j-i$), and the two possible constraints that could be imposed, i.e., either $r_{i(i+1)} + ... + r_{(j-1)j} = r_{ij}$ or $r - (r_{i(i+1)} + ... + r_{(j-1)j}) = r_{ij}$ for $r = r_{12} + ... + r_{(n-1)n} + r_{1n}$, we select according to which left side is smaller, based on the current entries in the matrix being carried forward to this point, and impose that specific constraint. Otherwise, the process continues cyclically through the whole set of constraints in (7) and for each time a constraint is reconsidered, any changes that were made the previous time the constraint was encountered are first "undone".

Because of the procedure of redressing the (immediately) previous changes each time a constraint is reconsidered, the process just described may not converge and could eventually oscillate through a

finite collection of distinct matrices. If such nonconvergence is observed, and previous changes from that point on are not redressed, the process will then converge to a matrix in $\Xi(k_1,...,k_{n-1})$ for some specific values of $k_1,...,k_{n-1}$. A justification for this last assertion of convergence is given by the general results presented in Hubert and Arabie (1995b); also, that source provides empirical evidence that as a heuristic optimization strategy, it is generally better to begin with the procedure of redressing previous changes until an oscillation is observed, rather than immediately starting without the process of redressing previous changes (which would also produce a matrix in $\Xi(k_1,...,k_{n-1})$ for some specific $k_1,...,k_{n-1}$). It should also be noted that although convergence to some matrix in $\Xi(k_1,...,k_{n-1})$ is guaranteed by the strategy just described, and thus to an identified fixed collection of inflection points, $k_1,...,k_{n-1}$, the latter matrix in $\Xi(k_1,...,k_{n-1})$ may now not be optimal for this collection of inflection points in the minimization of (3). Thus, the collection $k_1,...,k_{n-1}$ and the permutation $\varphi_K(\cdot)$ should be used anew in the optimization of (3) to obtain the optimal target matrix $\{\min\{|x_j - x_i|, x_0 - |x_j - x_i|\}\}$.

3.3. Numerical Illustrations

To illustrate the process of CUS presented in the previous two sections, we will again consider the data of Tables 1(a) and 2, beginning with the Levelt et al. standardized proximities in the lower triangular portion of Table 2. From 100 random starting permutations, we found the four (slightly) different circular scalings given in Figure 4, panels (a)-(d). All four scalings are based on the same (identity) permutation (which for convenience is taken in a clockwise direction around the closed continuum) for the stimuli according to frequency ratios, but there are small differences in the estimation of the coordinates arising from the four different patterns of inflection points identified through the heuristic process of iterative projection. Explicitly, panels (b) and (c) allow the minimum distance calculation between positions 1/2 and 12 to be in the counterclockwise direction; for (a) and (b) it is clockwise; for panels (a) and (b) the minimum distance between positions 4 and 14/15 is

clockwise, and for (c) and (d) it is counterclockwise. But generally, the results over the four inflection patterns are very similar and demonstrate minor local optima variations. We give a short summary in Table 3(a) of the VAF, estimated constant c, and number of times a result was observed for each of the four panels in Figure 4.

{Table 3 here}

All four circular scalings given above display a large separation between the 1/2 and 14/15 positions (the notation 1/2 and 14/15 is used to indicate that stimuli 1 and 2 are given the identical coordinate values, and the same is true for 14 and 15). This large separation reflects the LUS structure for these stimuli given in Figure 1 (with a VAF of 66.22%), but for all panels of Figure 4, and in contrast to LUS, it is generally better (i.e., there is an approximate 8% increase in VAF) to allow minimum distance calculations to be counterclockwise for at least some of the extreme frequency ratios. This property underlies the greater adequacy of a CUS structure in accounting for the Levelt et al. data, compared to the best LUS based on the identity permutation (see Figure 1), where the stimuli at the two extremes tend to be more similar to each other than can be allowed in a linear unidimensional representation.

As a somewhat more general observation, we point out that CUS as formulated by (3) does include LUS as a special case when a large enough separation is present between two positions in the circular representation, i.e., all inflection points $k_1,...,k_{n-1}$ are equal to n and the minimum distance calculations are always in a clockwise direction. Consequently, if a LUS were more appropriate than a CUS with some specific proximity matrix, we would hope to see so directly in a CUS representation in which the separation between two positions is so large that all minimum distance calculations are in one (clockwise) direction. Such is not the case for the Levelt et al. proximities, and a (nondegenerate) CUS representation appears to be the more appropriate according to the VAF.

{Figures 4 and 5 and Table 4 here}

However plausible it may be that the world (at least as depicted in the two proximity matrices at hand) is more loopy than linear, we still thought the finding needs more scrutiny. To evaluate the magnitude of change in the VAF that might be expected by just fitting a more general CUS structure to a proximity matrix (that requires the estimation of n nonnegative separations between adjacent positions around a circular structure) in comparison to a LUS structure (that requires the estimation of one fewer, i.e., n-1, nonnegative separations between adjacent positions along a linear continuum), Table 4 presents the results of a small Monte Carlo study. One hundred symmetric proximity matrices were constructed from the existing $(15\times14)/2 = 105$ Levelt et al. proximity values by randomly placing the latter into the upper-triangular portion of a 15×15 matrix (with a zero main diagonal), and then setting the corresponding lower-triangular entries equal to their upper-triangular counterparts (to ensure symmetry). Both a LUS and a CUS structure were fitted to each matrix, generating the stem plots given in Table 4 for the VAF's that were obtained. The mean difference in VAF's for the LUS and CUS structures is 4.1%. Given the absence of any patterning in the proximities because of how they were generated, the 4.1% value represents (for the Levelt et al. 15×15 proximity matrix) what average increase in VAF might be attributed to the presence and estimation of one additional separation value in CUS, compared to LUS. Using this baseline value of 4.1%, the increase in VAF observed in fitting the Levelt et al. data by a CUS versus a LUS model is about twice what we could expect. This difference, along with (a) the identification of the same identity permutation in the best-fitting LUS and CUS models for the Levelt et al. data, (b) an inspection of the residuals (i.e., the consistent over-estimation of proximities between the very extreme frequency ratios for the LUS model), and (c) their generally better estimation in the CUS model, support the greater adequacy of the CUS representation for the Levelt et al. data.

For the standardized Morse code proximities in Table 1, the use of 100 random starting permutations generated two distinct orderings of the ten symbols around the closed continuum; the

identity permutation occurred 91/100 (which in turn led to four slightly varying CUS structures based on small differences in the pattern of identified inflection points). A second ordering was identified 9/100 and was an obvious local optimum, both according to the original equally-spaced target matrix, given the much lower cross-product statistic it produced, and to the final CUS structures generated, based on VAF. Thus, we limit our discussion below to the four slightly different CUS representations corresponding to the (identity) permutation, assumed to be taken clockwise around the closed continuum. One of these four appeared previously in Figure 3; the others are presented in panels (a)-(c) of Figure 5 and again reflect differences in the estimation of the coordinates stemming from slightly different patterns of inflection points constructed through the heuristic process of iterative projection. Considering the minimum distance calculations between the positions for the digits 3 and 8/9, and 4 and 8/9, in Figure 3 they are both taken clockwise; in panel 5(c) they are both counterclockwise; and in panel 5(b), it is clockwise for 4 and 8/9 and counterclockwise for 3 and 8/9. In panel (d), the positions for the digits 1 and 5 induce the minimum distance calculation counterclockwise, whereas it is clockwise for the three other circular unidimensional scalings. We give a summary in Table 3(b) of the VAF, estimated constant c, and number of times a result was observed for Figure 3 and the three panels of Figure 5.

Although the four circular unidimensional scalings given in Figures 3 and 5(a)-(c) differ slightly according to the directionality of the minimum distance calculation for some of the position pairs, the solutions are all based on the identity permutation and thus have the same clear interpretation mentioned in Section 3.1, i.e., the regular replacement of dashes by dots and conversely when moving clockwise around the closed continuum. In fact, given the circular representation in, say, Figure 3, the best LUS found for these same symbols given in Figure 2 (with a VAF of 58.57%, compared with 72.77% for Figure 3) has at least an intuitively reasonable interpretation. In Figure 3, if we pass a straight line through the antipodes of 4 and 8/9 in the circular representation, the best LUS in Figure 2

would be reflected well by the projections of the symbols onto this line. Thus, Figure 2 is an inferior attempt to represent with a linear continuum what is better done in Figure 3, using distances generated around a closed continuum. (We note that the difference in VAF of 14.2% for the best fitting LUS and CUS structures is more than twice the average difference of 6.3% found for the CUS and LUS VAF distributions reported in Table 4(b); these latter distributions were obtained from the analysis of 100 randomly constructed proximity matrices based on the Morse Code proximities of Table 1(a). The Monte Carlo process was carried out similarly to that using the Levelt et al. proximities generating Table 4(a). The size of the observed VAF increase of 14.2% compared to the average difference of 6.3% in Table 4(b) suggests that the greater adequacy of the CUS model for these data exceeds what could be expected merely because the CUS model is more general and includes one more nonnegative separation value to be estimated.)

4. Representations Through Multiple Unidimensional Structures

The discussion in previous sections has been restricted to the fitting of a single unidimensional structure, either circular or linear, to a symmetric proximity matrix. Given the type of computational approach developed for carrying out this task (and, in particular, because of its lack of dependence on the presence of nonnegative proximities), extensions are very direct to the use of multiple unidimensional structures through a process of successive residualization of the original proximity matrix. For example, the fitting of two CUS structures to a proximity matrix $\{p_{ij}\}$ could be rephrased as the minimization of a least-squares loss function that generalizes (3) to the form

$$\sum_{i<j} (p_{ij} - [\min\{|x_{j1} - x_{i1}|, x_{01} - |x_{j1} - x_{i1}|\} - c_1] - [\min\{|x_{j2} - x_{i2}|, x_{02} - |x_{j2} - x_{i2}|\} - c_2])^2 . \quad (8)$$

The attempt to minimize (8) could proceed with the fitting of a single CUS structure to $\{p_{ij}\}$, $[\min\{|x_{j1} - x_{i1}|, x_{01} - |x_{j1} - x_{i1}|\} - c_1]$, using the computational strategy of Section 3, and once obtained, fitting a second CUS structure, $[\min\{|x_{j2} - x_{i2}|, x_{02} - |x_{j2} - x_{i2}|\} - c_2]$ to the residual matrix,

$\{p_{ij} - [\min\{|x_{j1} - x_{i1}|, x_{01} - |x_{j1} - x_{i1}|\}] - c_1\}$. The process would then cycle by repetitively fitting the residuals from the second circular structure by the first, and the residuals from the first circular structure by the second, until the sequence converges. Analogously, the fitting of two LUS structures to a proximity matrix $\{p_{ij}\}$ would proceed using the generalization of the loss function in (2) to one having the form

$$\sum_{i<j} (p_{ij} - [|x_{j1} - x_{i1}| - c_1] - [|x_{j2} - x_{i2}| - c_2])^2 \quad . \tag{9}$$

In any event, obvious extensions exist for both (8) and (9) to the inclusion of more than two LUS or CUS structures or to some mixture of these two, in the spirit of Carroll and Pruzansky's (1980) hybrid models.

The explicit inclusion of two constants, c_1 and c_2, in (8) and (9) rather than adding these two together and including a single additive constant c, deserves some additional explanation. In the case of fitting a single LUS or CUS structure using the loss functions in (2) and (3), it was noted in the introduction that two interpretations exist for the role of the additive constant c. For example, in LUS we could either consider (a) $\{|x_j - x_i|\}$ to be fitted to the translated proximities $\{p_{ij} + c\}$, or (b) $\{|x_j - x_i| - c\}$ to be fitted to the original proximities $\{p_{ij}\}$, where the constant c becomes part of the actual model. Although these two interpretations do not lead to any algorithmic differences in how we would proceed with minimizing the loss functions in (2) and (3), a consistent use of the second interpretation (b) suggests that we frame extensions to the use of multiple LUS or CUS structures as we did in (8) and (9), where it is explicit that the constants c_1 and c_2 are part of the actual models to be fitted to the (untransformed) proximities $\{p_{ij}\}$. Once c_1 and c_2 are obtained, they could be summed as $c = c_1 + c_2$, and an interpretation made, for example through (9), that we have attempted to fit a transformed set of proximities $\{p_{ij} + c\}$ by the sum $\{|x_{j1} - x_{i1}| + |x_{j2} - x_{i2}|\}$ (and in this latter case, a more usual terminology would be one of a two-dimensional scaling (MDS) based on the city-block distance function). However, such a further interpretation is unnecessary and could lead to at least

some small terminological confusion in further extensions that we might wish to pursue. For instance, if some type of (optimal nonlinear) transformation, say $f(\cdot)$, of the proximities is also sought (e.g., a monotonic function of some form) in addition to fitting multiple LUS or CUS structures, and where p_{ij} in (8) or (9) is replaced by $f(p_{ij})$, and $f(\cdot)$ is to be constructed, the first interpretation would require the use of a "doubly transformed" set of proximities $\{f(p_{ij}) + c\}$ to be fitted by the sum $\{|x_{j1} - x_{i1}| + |x_{j2} - x_{i2}|\}$. In general, it seems best to avoid the need to incorporate the notion of a double transformation in this context, and instead, merely consider the constants c_1 and c_2 to be part of the models being fitted to a transformed set of proximities $\{f(p_{ij})\}$.

Examples. As an illustration of the results obtainable from the process just described using the Morse code data from Table 1(a), Figure 6 gives the best (according to a VAF of 92.45%) two-CUS representation obtained from 100 random starting permutations for each of the circular components in (8). The two CUS structures in Figure 6 have rather clear substantive interpretations: as with Figure 3, the first shows the regular replacement of dots by dashes moving around the closed continuum; the second provides a perfect ordering around the closed continuum according to ratios of dots to dashes or of dashes to dots and where adjacent pairs of stimuli have dashes and dots exchanged one-for-one, i.e., for the adjacent stimuli pairs moving clockwise, we have: 0:5 for {——; •••••}; 1:4 for {•—; -•••}; 2:3 for {••—; —•••}; 2:3 for {•••—; —••}; and 1:4 for {——•; ••••-}. The two additive constants c_1 and c_2 in (8) have values of 1.925 and .886, respectively. (As mentioned, Figure 6 represents the best two-CUS structures obtained for 100 random starting permutations, but as might be expected given the computational results of Section 3, the same type of local optima were observed here as found in the fitting of a single CUS structure, i.e., several local optima were generated from small differences in the estimation of inflection points but with the identical object orderings around the closed continua; in this case, the result of Figure 6 with a VAF of 92.45% was observed 9/100; a

second with a VAF of 92.43% at 45/100; and a third with a VAF of 92.37% at 23/100.)

As a second illustration, Figure 7 gives the best two-LUS representation for the Morse code data, having a VAF of 90.49%, and additive constants c_1 and c_2 of 1.534 and .973, respectively. This particular representation was the single result observed over all 100 random starting permutations for each of the linear components of (9). By observing the pattern of monotonic change in the coordinates as one progresses from digit 0 to 9 and then back to 0, Figure 7 reflects the same type of circular arrangement of the symbols present in the single CUS representation given in Figures 3 and 5(a)-(c) (and also in the first CUS structure of Figure 6), but now constructed through the use of a two-LUS representation.

{Figures 6 and 7 here}

4.1. Other Representational Structures for a Symmetric Proximity Matrix

In addition to the multiple LUS and CUS models that might be fitted to a given proximity matrix, a variety of other extant structural representations for a symmetric proximity matrix could also be used in some combination. One of the more extensively studied classes of such models is based on the length of the path that joins two objects in a tree structure (for a recent comprehensive review, see De Soete and Carroll, 1996; also, see Barthélemy and Guénoche, 1991). These tree models, either by themselves or in conjunction with LUS or CUS structures, can provide a wide array of possible structural representations to consider for any given proximity matrix. Here we will only give a very brief summary of the defining properties for these tree structures and a few illustrations using the data from Table 1(b) on the eight Morse code stimuli that contain three symbols. For these data, tree structures seem to work particularly well (and somewhat in contrast to the Morse code data for the ten digits used thus far, where CUS structures appear to provide the preferred representations).

To introduce the task of fitting various tree structures to a symmetric proximity matrix, we

consider the least-squares loss function

$$\sum_{i<j} (p_{ij} - \{d_{ij} - c\})^2 , \qquad (10)$$

to be minimized over $\{d_{ij}\}$, where c is an additive constant that can be chosen sufficiently large to ensure that $d_{ij} \geq 0$ for $1 \leq i,j \leq n$, and d_{ij} is subject to an additional set of restrictions that define the particular form of the tree representation being sought. Three classes of possible restrictions are considered below:

(a) $\mathbf{D} = \{d_{ij}\}$ is an ultrametric, where over all distinct i, j, and k, $d_{ij} \leq \max\{d_{ik}, d_{jk}\}$ (or equivalently, among the three terms, d_{ij}, d_{ik}, and d_{jk}, the largest two are equal). An ultrametric is representable using a tree structure such as Figure 8(a) for the Morse code data of Table 1(b), where d_{ij} defines the length of the path that joins objects O_i and O_j (all horizontal distances in the representations we give for trees are assumed to have length zero, and are used for graphical purposes only). There is also a location on the tree, indicated by a small box in Figure 8(a), for the "root" that is equidistant from all of the n objects; the necessary existence of such an equidistant location distinguishes an ultrametric from an additive tree metric, characterized below in (c).

(b) $\mathbf{D} = \{d_{ij}\}$ is a centroid metric, where for all distinct i, j, k, and l, $d_{ij} + d_{kl} = d_{ik} + d_{jl} = d_{il} + d_{jk}$ (or equivalently, there exists a set of coordinates, $y_1,\ldots,y_n$, such that $y_i \geq 0$ for $1 \leq i \leq n$ and $d_{ij} = y_i + y_j$ for $1 \leq i \neq j \leq n$). A centroid metric is representable using a very restricted tree structure (sometimes called a "star" or "bush" pattern), such as Figure 8(b) where there is a single internal node to which all objects are attached.

(c) $\mathbf{D} = \{d_{ij}\}$ is an additive tree metric where over all distinct i, j, k, and l, $d_{ij} + d_{kl} \leq \max\{d_{ik} + d_{jl}, d_{il} + d_{jk}\}$ (or equivalently, among the three sums, $d_{ij} + d_{kl}$, $d_{ik} + d_{jl}$, and $d_{il} + d_{jk}$, the largest two are equal). An additive tree metric is representable using a tree structure such as Figure 8(c), where d_{ij} again defines the length of the path that joins the two objects O_i and O_j. In contrast to an ultrametric, no location on the tree needs to exist that is equidistant from all n objects.

Several points should be made concerning the minimization of (10) over one of the classes of constraints presented in (a), (b), or (c):

(i) The choice of the constant c in (10) required for enforcing the nonnegativity constraints on d_{ij} (i.e., $0 \leq d_{ij}$ for $1 \leq i,j \leq n$) can be effected by a sufficiently large positive number greater than or equal to some value c_0; the use of any constant $c \geq c_0$ would suffice since the difference $c - c_0$ would be absorbed into the representation matrix **D**. (In obtaining the tree structures given in Figure 8(a)-(c), for example, and starting from the standardized proximities in the lower-triangular portion of Table 1(b), a value for c of 2.0 was used to ensure the nonnegativity of the entries in **D**.)

(ii) The fitting of a centroid metric using (10) can be done through the use of closed form expressions (e.g., see Carroll, 1976), i.e.,

$$x_i = (1/(n-2)) \sum_{(j \neq i)} (p_{ij} + c) - (1/(n-1)(n-2)) \sum_{i<j} (p_{ij} + c) ,$$

where c is chosen sufficiently large to ensure $y_i \geq 0$ for $1 \leq i \leq n$.

(iii) The identification of ultrametrics or additive trees to minimize (10) can be approached using a heuristic iterative projection strategy very similar in general form to that used in CUS for identifying appropriate inflection points in the minimum distance calculations around the closed continuum. This computational approach for ultrametrics and additive trees is developed extensively in Hubert and Arabie (1995b) and will not be reviewed here, but was used to obtain the representations given in Figures 8(a) and (c). The ultrametric in Figure 8(a) was the best according to its VAF of 79.41%, observed over 100 random starts (and identified 80/100); the best additive tree in Figures 8(c) with a VAF of 90.05% was generated 98/100; the centroid metric in Figure 8(c) was obtained through the closed form expressions given earlier and has a VAF of 14.22%.

As can be seen in both the ultrametric and additive tree metric in Figures 8(a) and (c), respectively, the dominant structure for the Morse code symbols that are defined by the eight possible patterns of three dots or dashes appears to be a partition into four classes; each contains two of the

symbols that are identical in their first two constituents, but in the third there is an interchange of a dot for a dash, i.e., {-••; -•-}, {•-•; •––}, {––•; –––}, and {•••; ••-}. At least among these eight symbols, the major confusions underlying the proximity data are driven by a lack of salience for the final constituent. The centroid metric, which essentially provides a measure of aggregate proximity for each stimulus, suggests some greater degree of dissimilarity for the two symbols containing only dots or dashes versus all others, and some greater degree of similarity to all others for the two symbols -•• and -•-, but this effect is rather weak (e.g., a VAF of 14.22% for the centroid representation by itself). In short, the dominant structure reflected in Figures 8(a) and (c) is the partitioning of the eight symbols into classes containing stimuli differing only in their last constituents.

{Figure 8 here}

(iv) Any additive tree metric can be decomposed (although not uniquely) into an ultrametric and a centroid metric (see, e.g., Carroll, 1976; Carroll, Clark, & DeSarbo, 1984, Appendix II) (this result assumes the use of a sufficiently large constant c to ensure nonnegativity for both the ultrametric and centroid metric components). Thus, the fitting of an additive tree metric to a given proximity matrix can be reinterpreted as the fitting of a sum of two structures— a centroid metric and an ultrametric; alternatively, we can view an additive tree as merely an ultrametric augmented by a second centroid metric structure.

Similarly, a centroid metric could be used in conjunction with either a LUS or a CUS, and fitted to a given proximity matrix through a process of residualization analogous to fitting two-LUS or two-CUS structures. In particular, a LUS augmented by a centroid metric would rely on the loss function

$$\sum_{i<j} (p_{ij} - [(y_i + y_j) - c_1] - [|x_j - x_i| - c_2])^2 , \qquad (11)$$

where c_1 is chosen sufficiently large to ensure $y_i \geq 0$ for $1 \leq i \leq n$. Moreover, once both components are generated from the minimization of (11), they can be represented jointly using a restricted additive tree of the form given in Figure 9(a) for the Morse code data of Table 1(b), where there are <u>at most</u>

two internal line segments (i.e., line segments not attached to an object) intersecting any internal node of the tree (see Sattath & Tversky, 1977, pp. 339-340, for a brief discussion of additive trees having this restricted structure). Figure 9(a) has a VAF of 73.51%, with c_2 equal to 1.454 and c_1 set equal to 2.0 to ensure the non-negativity of y_i, $1 \leq i \leq n$, and was the sole result observed over 100 random starting permutations for the linear component (and irrespective of beginning the residualization process using (11) with the centroid metric or the linear structure). Although not performing as well according to its VAF of 73.51%, compared to that for the ultrametric or additive tree in Figures 8(a) and (c), the clustering structure used to interpret Figures 8(a) and (c) is very dramatic in Figure 9(a), i.e., three of the two symbol classes share a common node and the forth is associated with two internal nodes separated only by .01.

There are obviously many possible combinations of LUS and CUS structures, as well as those defined by ultrametrics, additive trees, and centroid metrics, that could be fitted to any given symmetric proximity matrix. We will provide only one such final combination in Figure 9(b), by considering an LUS model along with an ultrametric, and minimizing the loss function

$$\sum_{i<j} (p_{ij} - [|x_j - x_i| - c_1] - [d_{ij} - c_2])^2 , \qquad (12)$$

where $\{d_{ij}\}$ satisfies the ultrametric restrictions, and c_2 is chosen to ensure the nonnegativity of d_{ij}, $1 \leq i,j \leq n$. Figure 9(b) gives the best result obtained in minimizing (12) with a VAF of 93.78% (the highest of all representations considered thus far), c_1 having the value of .689, and c_2 set equal to 2.0. As is apparent in Figure 9(b), the ultrametric component displays the familiar partition into the four classes, having stimuli within a class varying only in their third constituents; the linear component displays rather well the elusive (at least up until now) ordering from left to right of the eight stimuli according to the number of dots to dashes, i.e., beginning with —, there are three consecutive stimuli with one dot, -•-, —•, •—; three consecutive stimuli with two dots, -••, •-•, ••-, and ending with •••.

{Figure 9 here}

5. Concluding Comments

For a symmetric proximity matrix, the various representations emphasized in this paper (i.e., structures based on LUS and CUS and more briefly in Section 4.1, those obtainable through ultrametrics, centroid metrics, or additive trees) have one salient common feature: all are definable through some specific topological structure but with an additional imposed parameterization. Once estimated, the values attached to the parameters are then used to reconstruct the input proximities and provide the specific graphic representation depicted (as was done in Figures 1 to 9). Although not pursued in the present paper, the topological structures by themselves can be used to fit a matrix to a given set of proximities, so as to satisfy only the order constraints mandated by the specific topological structure under consideration. Moreover, the reconstructed matrix (or possibly the sum of such matrices if multiple topological structures are used) may be sufficient to explain the pattern present among the original proximities, and without the use of any intermediary parameterization. In the specific case of LUS, and the underlying topological structure defined by an ordering of the n objects along a line, Hubert and Arabie (1994) developed in some detail the fitting of a given proximity matrix by sums of anti-Robinson matrices, which represent only the order properties mandated by a specific object placement along a continuum. There is much more that could be done, however, in relying solely on order constraints implied by the various topologies that underlie CUS and tree structures; these generalizations are currently under active study by the present authors.

References

Barthélemy, J.-P., & Guénouche, A. (1991). Trees and proximity representations. Chichester: Wiley.

Bentler, P. M., & Weeks, D. G. (1978). Restricted multidimensional scaling models. Journal of Mathematical Psychology, 17, 138-151.

Borg, I., & Lingoes, J. C. (1979). Multidimensional scaling with side constraints on the distances. In J. C. Lingoes, E. E. Roskam, & I. Borg (Eds.), Geometric representations of relational data (pp. 753-790). Ann Arbor, MI: Mathesis Press.

Borg, I., & Lingoes, J. C. (1980). A model and algorithm for multidimensional scaling with external constraints on the distances. Psychometrika, 45, 25-38.

Boyle, J. P., & Dykstra, R. L. (1986). A method for finding projections onto the intersection of convex sets in Hilbert space. In R. L. Dykstra, T. Robertson, & F. T. Wright (Eds.), Advances in order restricted statistical inference (volume 17, Lecture Notes in Statistics) (pp. 28-47). Berlin: Springer-Verlag.

Browne, M. W. (1992). Circumplex models for correlation matrices. Psychometrika, 57, 469-497.

Carroll, J. D. (1976). Spatial, non-spatial, and hybrid models for scaling. Psychometrika, 41, 439-463.

Carroll, J. D., Clark, L. A., & DeSarbo, W. S. (1984). The representation of three-way proximity data by single and multiple tree structure models. Journal of Classification, 1, 25-74.

Carroll, J. D., & Pruzansky, S. (1980). Discrete and hybrid scaling models. In E. D. Lantermann & H. Feger (Eds.), Similarity and choice (pp. 108-139). Bern: Hans Huber.

Cheney, W., & Goldstein, A. (1959). Proximity maps for convex sets. Proceedings of the American Mathematical Society, 10, 448-450.

Cox, T. F., & Cox, M. A. A. (1991). Multidimensional scaling on a sphere. Communications in

Statistics- Theory and Methods, 20, 2943-2953.

Cudek, R. (1986). A note on structural models for the circumplex. Psychometrika, 51, 143-147.

de Leeuw, J., & Heiser, W. (1977). Convergence of correction matrix algorithms for multidimensional scaling. In J. C. Lingoes, E. E. Roskam, & I. Borg (Eds.), Geometric representations of relational data (pp. 735-752). Ann Arbor, MI: Mathesis Press.

de Leeuw, J., & Heiser, W. (1980). Multidimensional scaling with restrictions on the configuration. In P. R. Krishnaiah (Ed.), Multivariate analysis, Vol. 5 (pp. 501-522). Amsterdam: North-Holland.

Defays, D. (1978). A short note on a method of seriation. British Journal of Mathematical and Statistical Psychology, 31, 49-53.

De Soete, G., & Carroll, J. D. (1996). Tree and other network models for representing proximity data. In P. Arabie, L. J. Hubert, & G. De Soete (Eds.), Clustering and classification (pp. 169-209). River Edge, NJ: World Scientific Publishers.

Dykstra, R. L. (1983). An algorithm for restricted least squares regression. Journal of the American Statistical Association, 78, 837-842.

Groenen, P. J. F. (1993). The majorization approach to multidimensional scaling: Some problems and extensions. Leiden: DSWO Press.

Groenen, P. J. F., Mathar, R., & Heiser, W. J. (1995). The majorization approach to multidimensional scaling for Minkowski distances. Journal of Classification, 12, 3-19.

Guttman, L. (1968). A general nonmetric technique for finding the smallest coordinate space for a configuration of points. Psychometrika, 33, 469-506.

Han, S. P. (1988). A successive projection method. Mathematical Programming, 40, 1-14.

Heiser, W. J. (1989). The city-block model for three-way multidimensional scaling. In R. Coppi & S. Bolasco (Eds.), Multiway data analysis (pp. 395-404). Amsterdam: North-Holland.

Heiser, W. J. (1991). A generalized majorization method for least squares multidimensional scaling of pseudodistances that may be negative. Psychometrika, 56, 7-27.

Heiser, W. J., & Meulman, J. (1983). Constrained multidimensional scaling, including confirmation. Applied Psychological Measurement, 7, 381-404.

Hubert, L. J., & Arabie, P. (1986). Unidimensional scaling and combinatorial optimization. In J. de Leeuw, W. Heiser, J. Meulman, & F. Critchley (Eds.), Multidimensional data analysis (pp. 181-196). Leiden, The Netherlands: DSWO Press.

Hubert, L. J., & Arabie, P. (1988). Relying on necessary conditions for optimization: Unidimensional scaling and some extensions. In H. H. Bock (Ed.), Classification and related methods of data analysis (pp. 463-472). Amsterdam: North-Holland.

Hubert, L. J., & Arabie, P. (1989). Combinatorial data analysis: Confirmatory comparisons between sets of matrices. Applied Stochastic Models and Data Analysis, 5, 273-325.

Hubert, L. J., & Arabie, P. (1994). The analysis of proximity matrices through sums of matrices having (anti-)Robinson forms. British Journal of Mathematical and Statistical Psychology, 47, 1-40.

Hubert, L. J., & Arabie, P. (1995a). The approximation of two-mode proximity matrices by sums of order-constrained matrices. Psychometrika, 60, 573-605.

Hubert, L. J., & Arabie, P. (1995b). Iterative projection strategies for the least-squares fitting of tree structures to proximity data. British Journal of Mathematical and Statistical Psychology, 48, 281-317.

Hubert, L. J., Arabie, P., & Hesson-Mcinnis, M. (1992). Multidimensional scaling in the city-block metric: A combinatorial approach. Journal of Classification, 9, 211-236.

Hubert, L. J., & Schultz, J. V. (1976). Quadratic assignment as a general data analysis strategy. British Journal of Mathematical and Statistical Psychology, 29, 190-241.

Kruskal, J. B., & Wish, M. (1978). Multidimensional scaling. Newbury Park, CA: Sage.

Lee, S. Y., & Bentler, P. M. (1980). Functional relations in multidimensional scaling. British Journal of Mathematical and Statistical Psychology, 33, 142-150.

Lee, S. Y. (1984). Multidimensional scaling models with inequality and equality constraints. Communications in Statistics--Simulations and Computations, 15, 127-140.

Levelt, W. J. M., van de Geer, J. P., & Plomp, R. (1966). Triadic comparisons of musical intervals. British Journal of Mathematical and Statistical Psychology, 19, 193-179.

Pliner, V. (1986). The problem of multidimensional metric scaling. Automation and Remote Control, 47, 560-567.

Pliner, V. (1996). Metric unidimensional scaling and global optimization. Journal of Classification, 13, xxx-xxx.

Poole, K. T. (1990). Least squares metric, unidimensional scaling of multivariate linear models. Psychometrika, 55, 123-149.

Rothkopf, E. Z. (1957). A measure of stimulus similarity and errors in some paired-associate learning tasks. Journal of Experimental Psychology, 53, 94-101.

Sattath, S., & Tversky, A. (1977). Additive similarity trees. Psychometrika, 42, 319-345.

Shepard, R. N. (1963). Analysis of proximities as a technique for the study of information processing in man. Human Factors, 5, 33-48.

Shepard, R. N. (1974). Representation of structure in similarity data: Problems and prospects. Psychometrika, 39, 373-421.

van Schuur, W. H. (1987). Constraint in European party activists' sympathy scores for interest groups. European Journal of Political Research, 15, 347-362.

Wiggins, J. S., Steiger, J. H., & Gaelick, L. (1981). Evaluating circumplexity in personality data. Multivariate Behavioral Research, 16, 263-289.

TABLE 1

(a) A Proximity Matrix for the Ten Morse Code Symbols Representing the First Ten Digits (Data from Rothkopf, 1957); (b) A Proximity Matrix for the Eight Morse Code Symbols Containing Three Dots or Dashes. The Upper-triangular Entries are the Original Dissimilarities; the Lower-triangular Entries are Standardized to a Mean of Zero and a Standard Deviation of One.

(a)

Symbol	0	1	2	3	4	5	6	7	8	9
0: -----	x	.75	1.69	1.87	1.76	1.77	1.59	1.26	.86	.95
1: •----	-1.79	x	.82	1.54	1.85	1.72	1.51	1.50	1.45	1.63
2: ••---	.61	-1.61	x	1.25	1.47	1.33	1.66	1.57	1.83	1.81
3: •••--	1.07	.22	-.52	x	.89	1.32	1.53	1.74	1.85	1.86
4: ••••-	.78	1.01	.04	-1.44	x	1.41	1.64	1.81	1.90	1.90
5 •••••	.81	.68	-.31	-.34	-.11	x	.70	1.56	1.84	1.64
6: -••••	.35	.15	.53	.20	.48	-1.92	x	.70	1.38	1.70
7: --•••	-.49	.12	.55	.73	.91	.27	-1.92	x	.83	1.22
8: ---••	-1.51	-.01	.96	1.01	1.14	.99	-.19	-1.59	x	.41
9: ----•	-1.28	.45	.91	1.04	1.14	.48	.63	-.59	-2.66	x

(b)

Symbol	d	g	k	o	r	s	u	w
d: -••	x	1.22	.46	1.85	1.32	1.41	1.51	1.51
g: --•	-.57	x	1.42	.42	1.53	1.90	1.74	1.43
k: -•-	-2.55	-.05	x	1.43	1.57	1.85	1.33	1.38
o: ---	1.08	-2.66	-.02	x	1.78	1.91	1.83	1.49
r: •-•	-.31	.24	.35	.89	x	1.66	1.37	.77
s: •••	-.07	1.21	1.08	1.23	.58	x	.93	1.76
u: ••-	.19	.79	-.28	1.02	-.18	-1.32	x	1.47
w: •--	.19	-.02	-.15	.14	-1.74	.84	.08	x

TABLE 2

A Proximity Matrix for the Fifteen Tonal Stimuli Defined by the Given Frequency Ratios (Data from Levelt, van de Geer, & Plomp, 1966). The Upper-triangular Entries are the Original Dissimilarities; the Lower-triangular Entries are Standardized to a Mean of Zero and a Standard Deviation of One.

Frequency ratio	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1: 15:16	X	0	3	13	18	17	24	23	26	20	25	25	24	23	23
2: 11:12	-2.30	X	0	10	15	22	24	22	25	21	21	22	29	18	18
3: 8:9	-1.87	-2.30	X	4	9	19	18	18	19	20	17	25	23	26	26
4: 5:6	-.43	-.87	-1.73	X	4	10	7	19	12	7	18	22	23	24	22
5: 4:5	.28	-.15	-1.01	-1.73	X	7	8	14	11	12	14	15	18	24	25
6: 3:4	.14	.86	.43	-.87	-1.30	X	5	11	15	8	18	22	17	20	20
7: 5:7	1.14	1.14	.28	-1.30	-1.15	-1.58	X	10	10	19	16	13	23	23	22
8: 2:3	1.00	.86	.28	.43	-.29	-.72	-.87	X	5	7	19	14	18	22	2
9: 5:8	1.43	1.29	.43	-.58	-.72	-.15	-.87	-1.58	X	8	5	14	20	22	14
10: 3:5	.57	.71	.57	.14	-.58	-1.15	.43	-1.30	-1.15	X	2	14	19	15	14
11: 4:7	1.29	.71	.14	.28	-.29	.28	.00	.43	-1.58	-2.01	X	6	10	12	18
12: 8:15	1.29	.86	1.29	.86	-.15	.86	-.43	-.29	-.29	-.29	-1.44	X	6	19	19
13: 1:2	1.14	1.86	1.00	1.00	.28	.14	1.00	.28	.57	.43	-.87	-1.44	X	3	7
14: 4:9	1.00	.28	1.43	1.14	1.14	.57	1.00	.86	.86	-.15	-.58	.43	-1.87	X	2
15: 2:5	1.00	.28	1.43	.86	1.29	.57	.86	.00	-.29	-.29	.28	.43	-1.30	-2.01	X

TABLE 3

(a) Summary Statistics for Fitting a Single CUS Structure to the Levelt et al. Data of Table 2 Based on 100 Random Starting Permutations; the Four CUS Structures Identified are Represented Graphically in Figure 4(a)-(d). (b) Summary Statistics for Fitting a Single CUS Structure to the Morse Code Proximities in Table 1 Based on 100 Random Starting Permutations (Excluding the Obvious Local Optimum Obtained 9/100); the Four CUS Structures Identified are Represented Graphically in Figures 3 and 5(a)-(c).

(a)

panel	VAF	c	times observed
4(a)	74.32%	1.486	23/100
4(b)	74.23%	1.504	11/100
4(c)	74.13%	1.524	55/100
4(d)	74.11%	1.505	11/100

(b)

panel	VAF	c	times observed
Figure 3	72.77%	1.686	44/91
5(a)	72.70%	1.693	25/91
5(b)	72.55%	1.679	18/91
5(c)	71.99%	1.655	4/91

TABLE 4

Stem Plots for the VAF Measure Obtained When Fitting LUS and CUS Structures to the Randomly Constructed Proximity Matrices Based on (a) the Levelt et al. Proximities of Table 2, and (b) the Morse Code Proximities in Table 1(a)

(a)

Linear

1 4
1 6677
1 8888899999
2 0000000111111111
2 2222233333333333
2 44444555555555555
2 66666777777777
2 8888888899999
3 000111
3 3
3 55

Circular

1 89
2 0001111
2 2223333
2 444444555555
2 66666667777777777
2 8888888888899999999
3 00001111111
3 2222222333
3 444455555
3 667
3 888

(b)

Linear

2 0
2
2 44555
2 6666777
2 889999999999
3 00000111
3 222222222333333333
3 44444455555555
3 66777777777777
3 889999
4 00111
4 22223
4 455
4
4 88

Circular

2 667
2 9
3 111
3 2233
3 44444445555
3 66666666677777
3 888888999999
4 00111111111
4 222222233333
4 4444444455555
4 66677
4 8889
5 0011
5
5 45
5 6

Figure Captions

Figure 1. A LUS for the Levelt et al. data of Table 2 based on the fixed identity permutation. The first stimulus is given an arbitrary coordinate of 0.0. Also, the frequency ratios are indicated above each stimulus. (VAF = 66.22%)

Figure 2. The best LUS identified for the Morse code data of Table 1(a). The first symbol is given an arbitrary coordinate of 0.0. (VAF = 58.57%)

Figure 3. A CUS for the Morse code data of Table 1(a) based on the fixed identity permutation and the inflection integers, $k_1 = 5$, $k_2 = 6$, $k_3 = 7$, $k_4 = \ldots = k_9 = 10$. The first symbol is given an arbitrary coordinate of 0.0, and the total length, $x_0 = 6.22$. (VAF = 72.77%)

Figure 4. The four CUS structures observed for the Levelt et al. data of Table 2. The first position is given an arbitrary coordinate value of 0.0, and the total length of the closed continuum is stated in parentheses beneath the 0.0 coordinate. (VAF's = 74.32%, 74.23%, 74.13%, and 74.11%)

Figure 5. The three CUS structures observed for the Morse code data of Table 1(a); a fourth is given in Figure 3. The first position is given an arbitrary coordinate value of 0.0, and the total length of the closed continuum is stated in parentheses beneath the 0.0 coordinate. (VAF's = 72.70%, 72.55%, and 71.99%)

Figure 6. The best two-CUS representation observed for the Morse code data of Table 1(a). The first position in each closed continuum is given an arbitrary coordinate value of 0.0, and the total length of each closed continuum is stated in parentheses beneath the 0.0 coordinate. (VAF = 92.45%)

Figure 7. The best two-LUS representation observed for the Morse code data of Table 1(a). The first positions are given arbitrary coordinate values of 0.0. (VAF = 90.49%)

Figure 8. The best ultrametric (a) (VAF = 79.41%) and additive tree metric (c) (VAF = 90.05%) observed for the Morse code data of Table 1(b); (b) (VAF = 14.22%) is the centroid metric generated from the closed form expressions. An additive constant of 2.0 was added to the

standardized proximities in Table 1(b) to ensure non-negativity of all the lengths attached to line segments in the given tree representations. All horizontal lengths in (a) and (c) are assumed to be zero and are for graphical purposes only.

Figure 9. The best centroid metric/LUS combination (a) (VAF = 73.51%) and best ultrametric/LUS combination (b) (VAF = 93.78%) observed for the Morse code data of Table 1(b). In (b), all horizontal lengths are assumed to be zero and are for graphical purposes only.

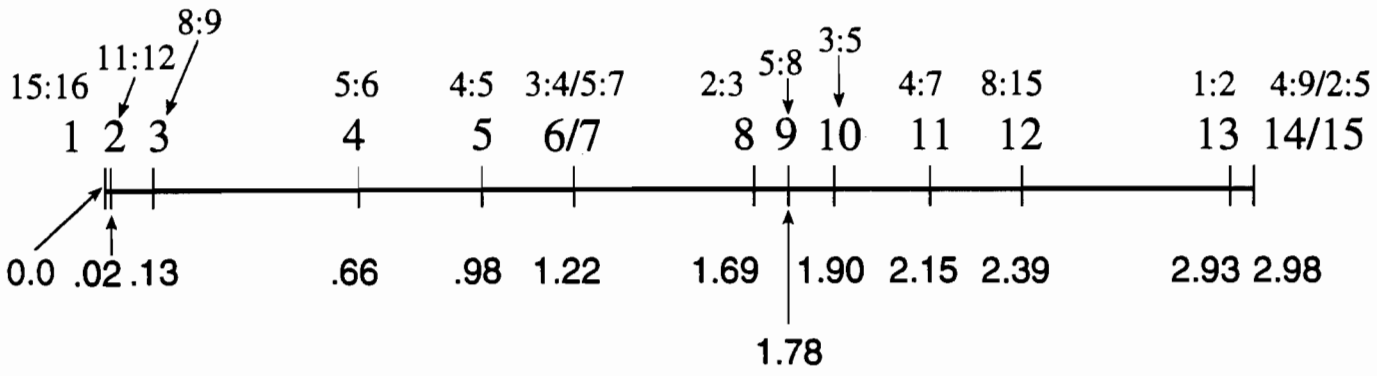

15:16
11:12
8:9
5:6
4:5
3:4/5:7
2:3
5:8
3:5
4:7
8:15
1:2
4:9/2:5
1 2 3 4 5 6/7 8 9 10 11 12 13 14/15
0.0 .02 .13 .66 .98 1.22 1.69 1.90 2.15 2.39 2.93 2.98
1.78

Figure 1

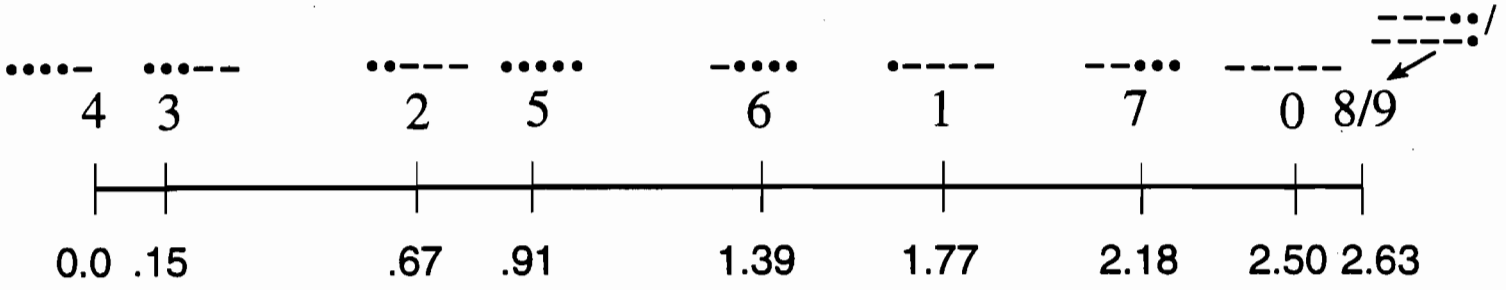


Figure 2

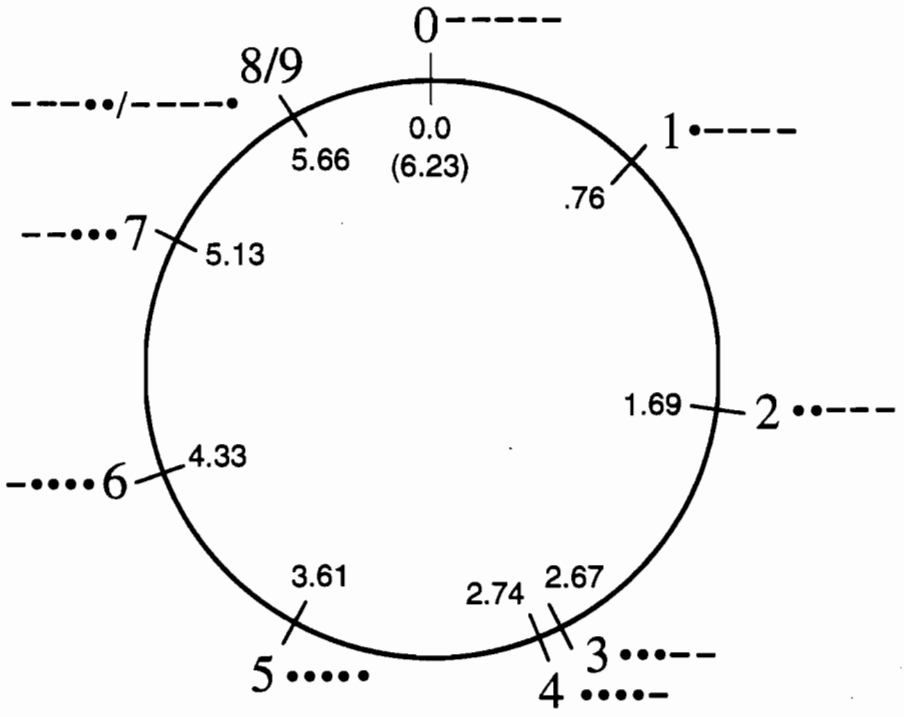
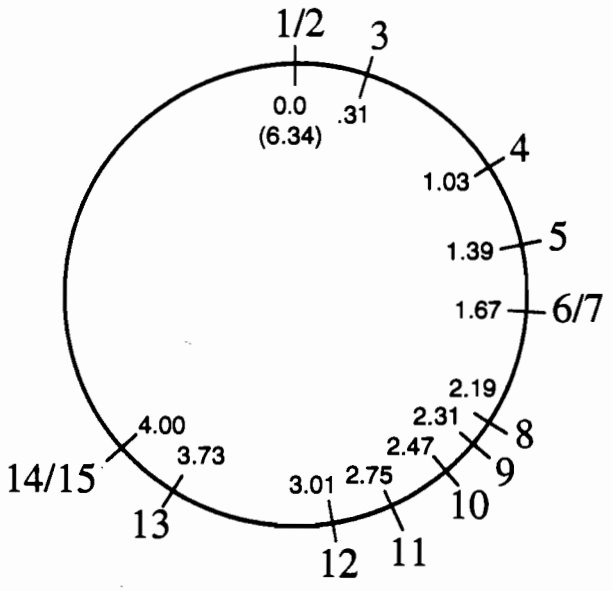
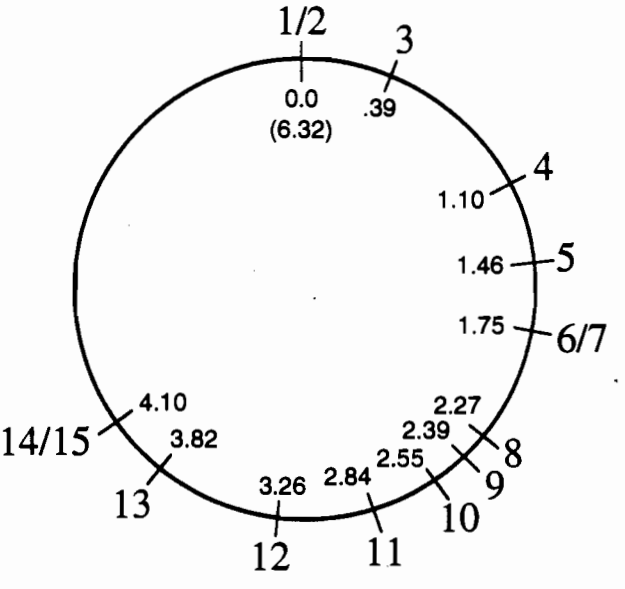


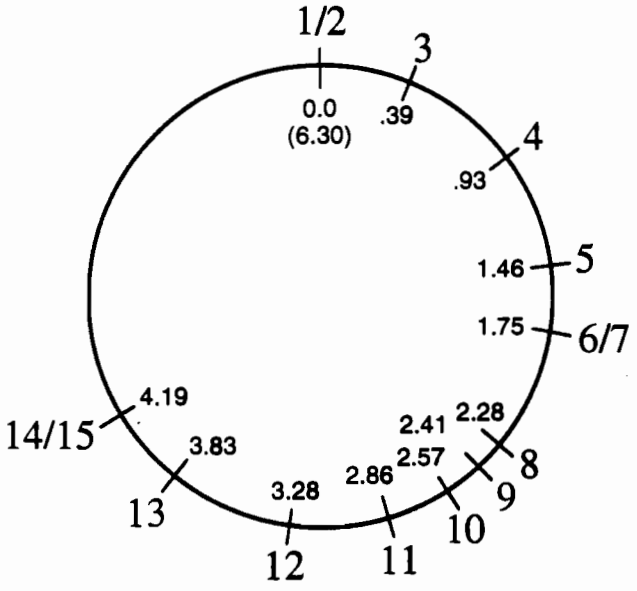
Figure 3



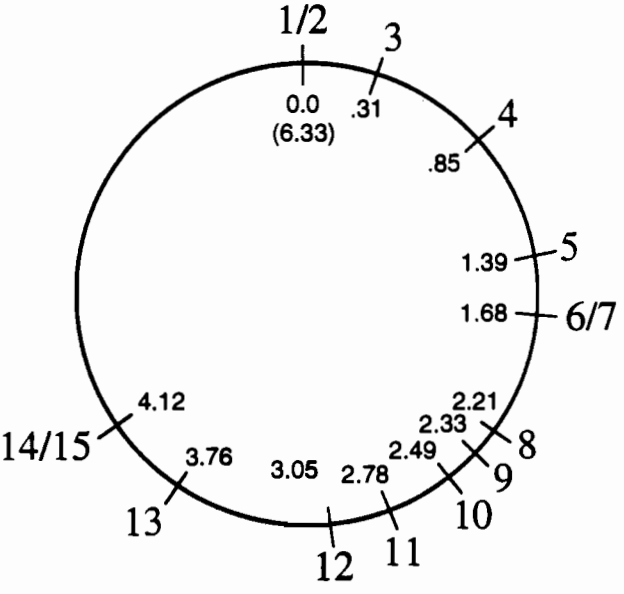
(a)



(b)



(c)



(d)

Figure 4

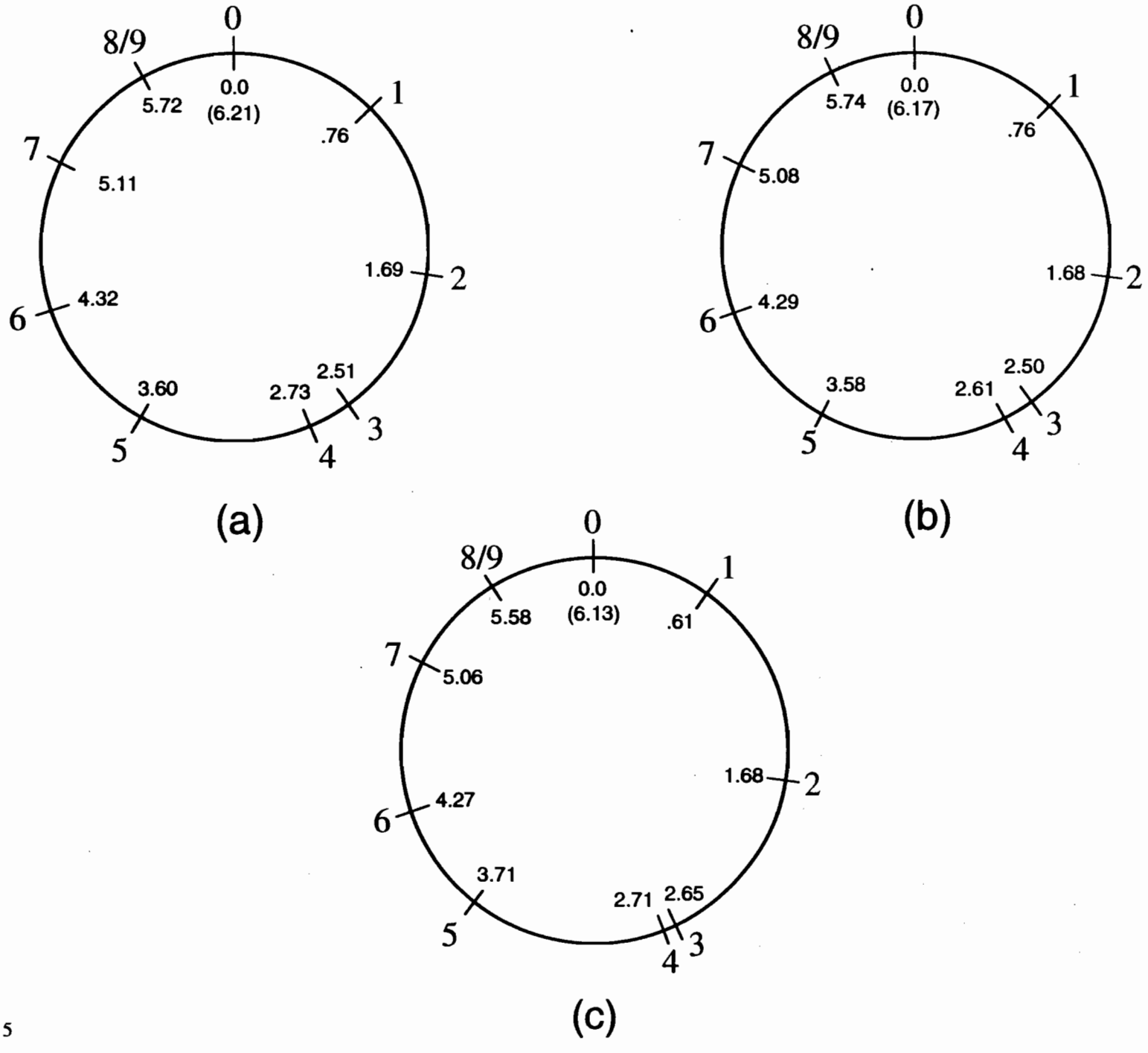
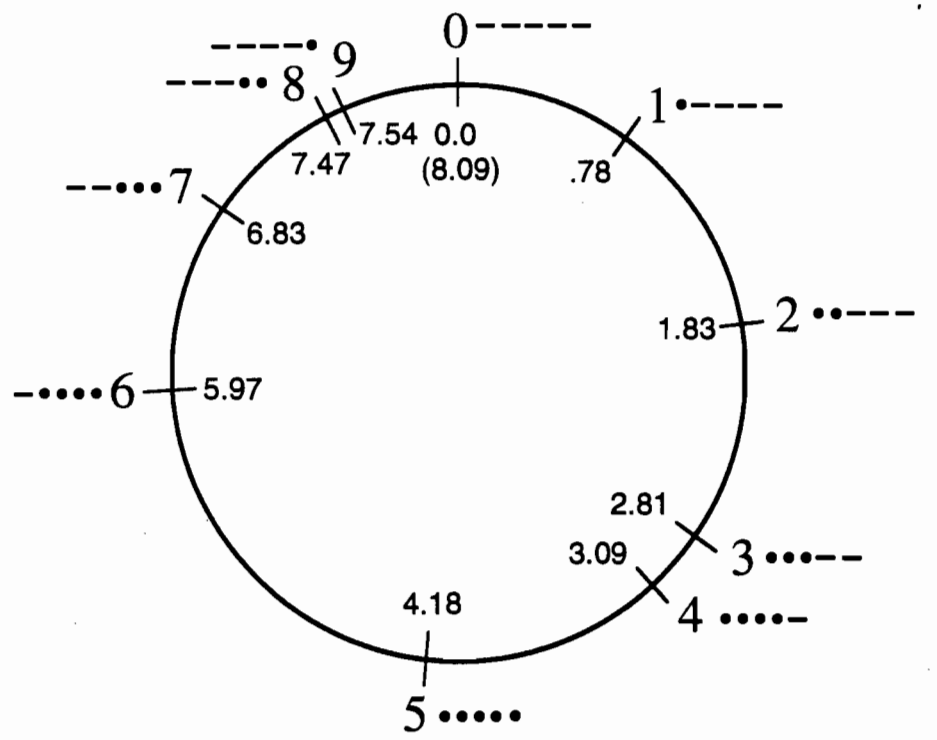

0
8/9
0.0
(6.21)
5.72
1
.76
7
5.11
1.69
2
6
4.32
3.60
2.73
2.51
3
5
4
(a)
0
8/9
0.0
(6.17)
5.74
1
.76
7
5.08
1.68
2
6
4.29
3.58
2.61
2.50
3
5
4
(b)
0
8/9
0.0
(6.13)
5.58
1
.61
7
5.06
1.68
2
6
4.27
3.71
2.71
2.65
3
5
4
(c)

Figure 5


0 -----
1 •----
2 ••---
3 •••--
4 ••••-
5 •••••
6 -••••
7 --•••
8 ---••
9 ----•
0.0
(8.09)
.78
1.83
2.81
3.09
4.18
5.97
6.83
7.47
7.54

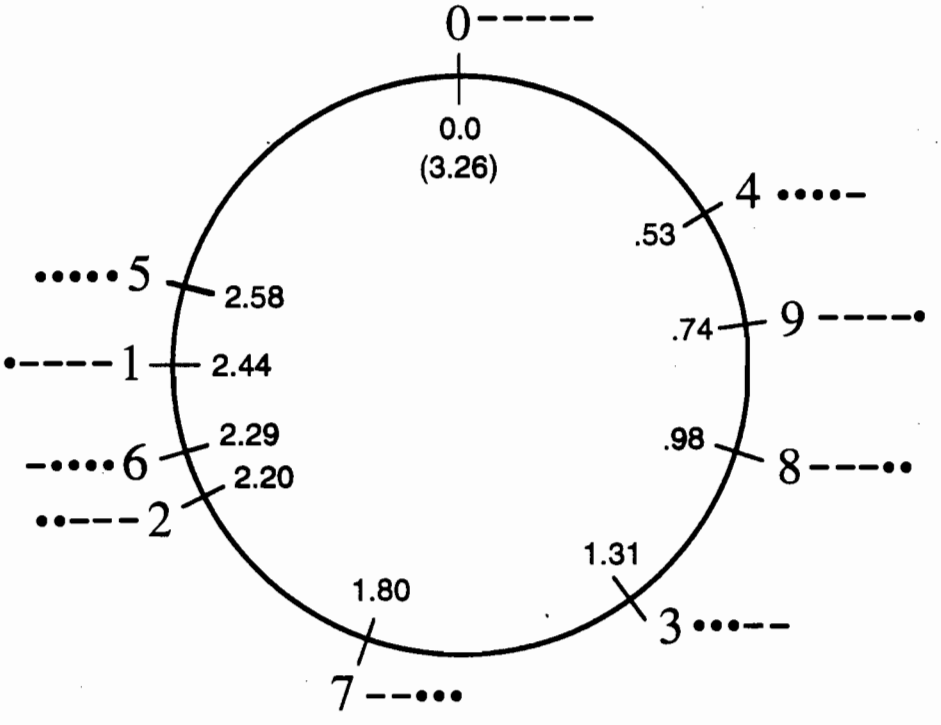
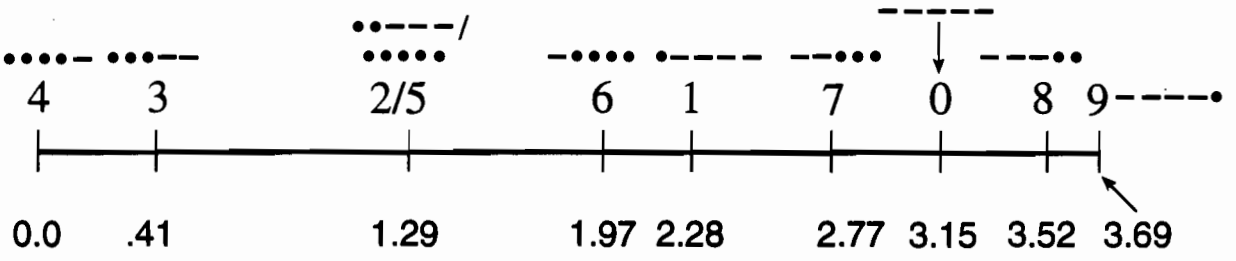
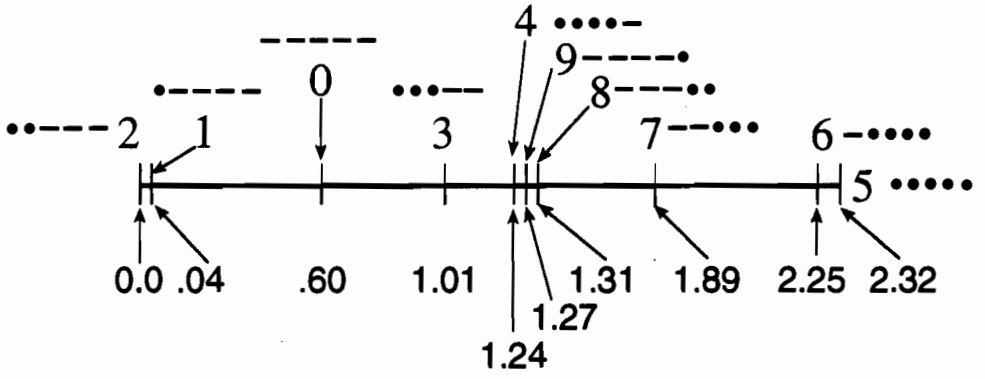

0 -----
4 ••••-
5 •••••
1 •----
6 -••••
2 ••---
3 •••--
7 --•••
8 ---••
9 ----•
0.0
(3.26)
.53
.74
.98
1.31
1.80
2.20
2.29
2.44
2.58

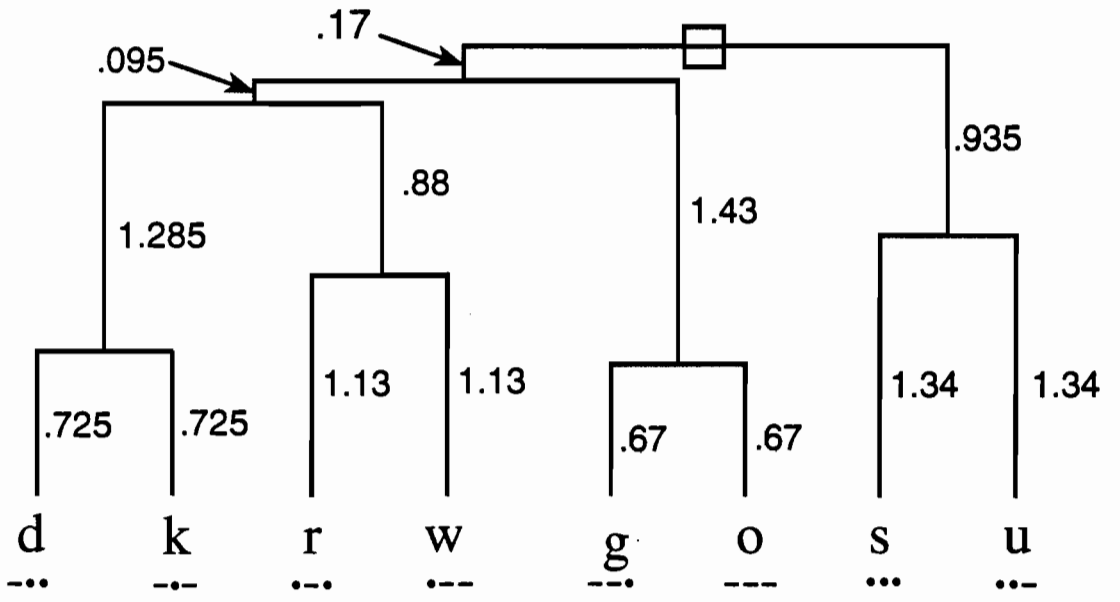
Figure 6



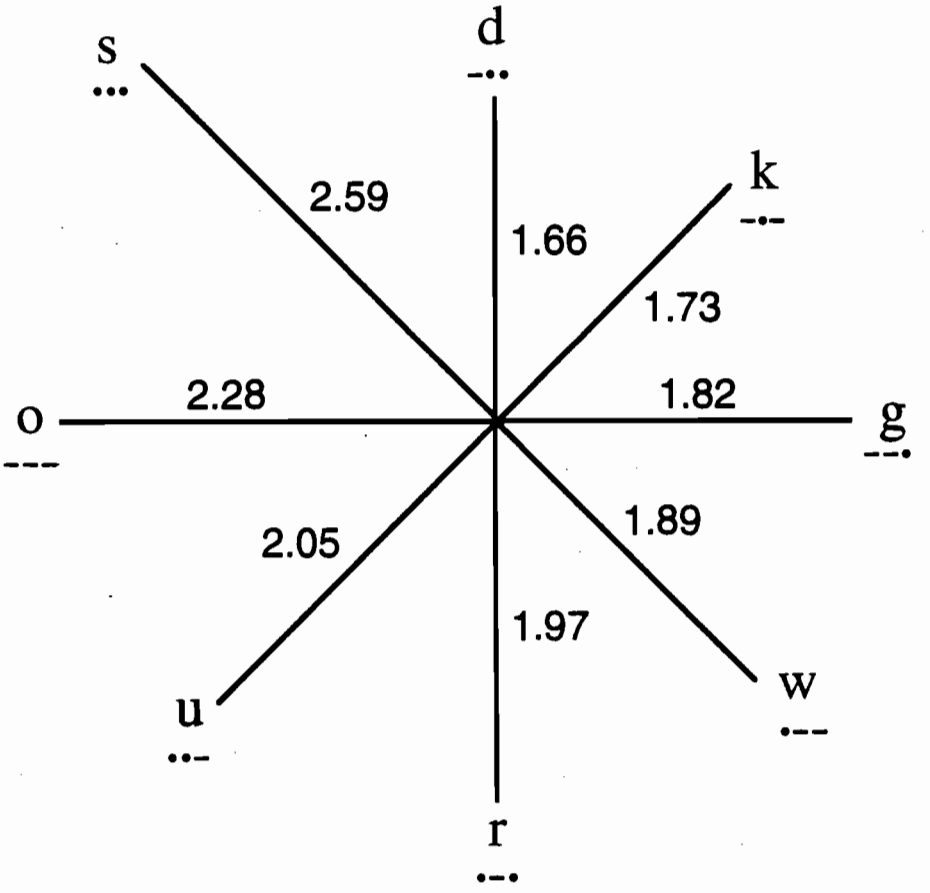


digit	coordinates	
0	3.15	0.60
1	2.28	0.04
2	1.29	0.0
3	0.41	1.01
4	0.0	1.24
5	1.29	2.32
6	1.97	2.25
7	2.77	1.89
8	3.52	1.31
9	3.69	1.27

Figure 7

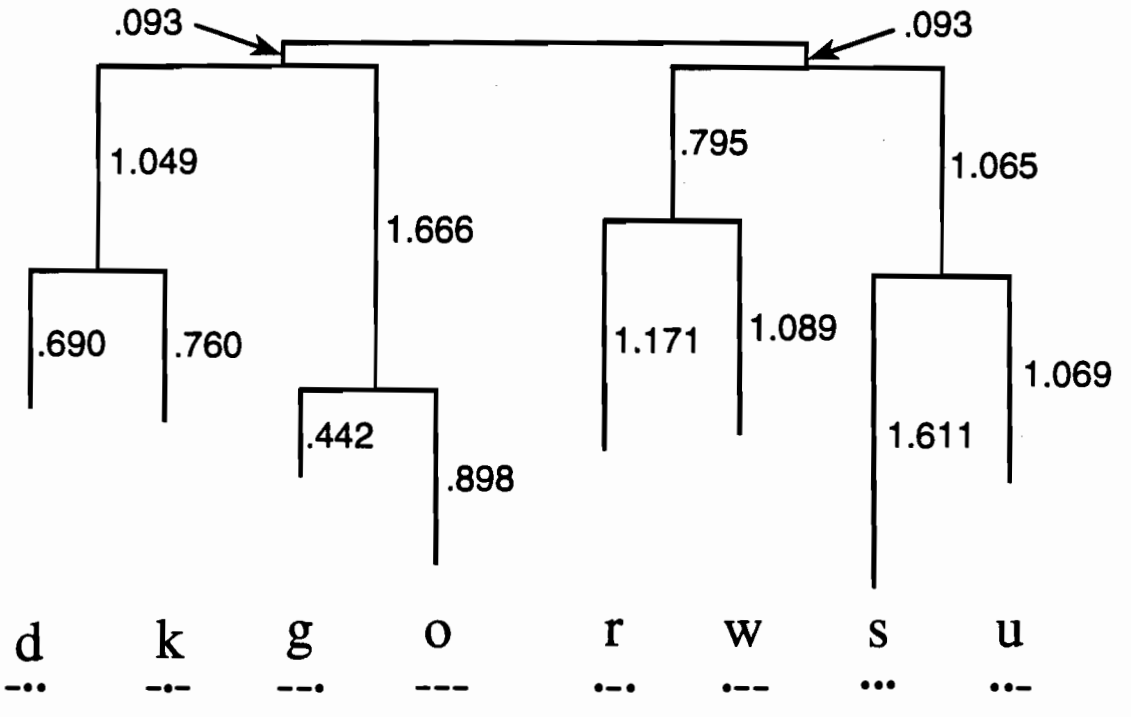

.17
.095
.935
.88
1.285
1.43
1.13
1.13
1.34
1.34
.725
.725
.67
.67
d
k
r
w
g
o
s
u

(a)


s
d
k
2.59
1.66
1.73
o
2.28
1.82
g
1.89
2.05
1.97
u
w
r

(b)

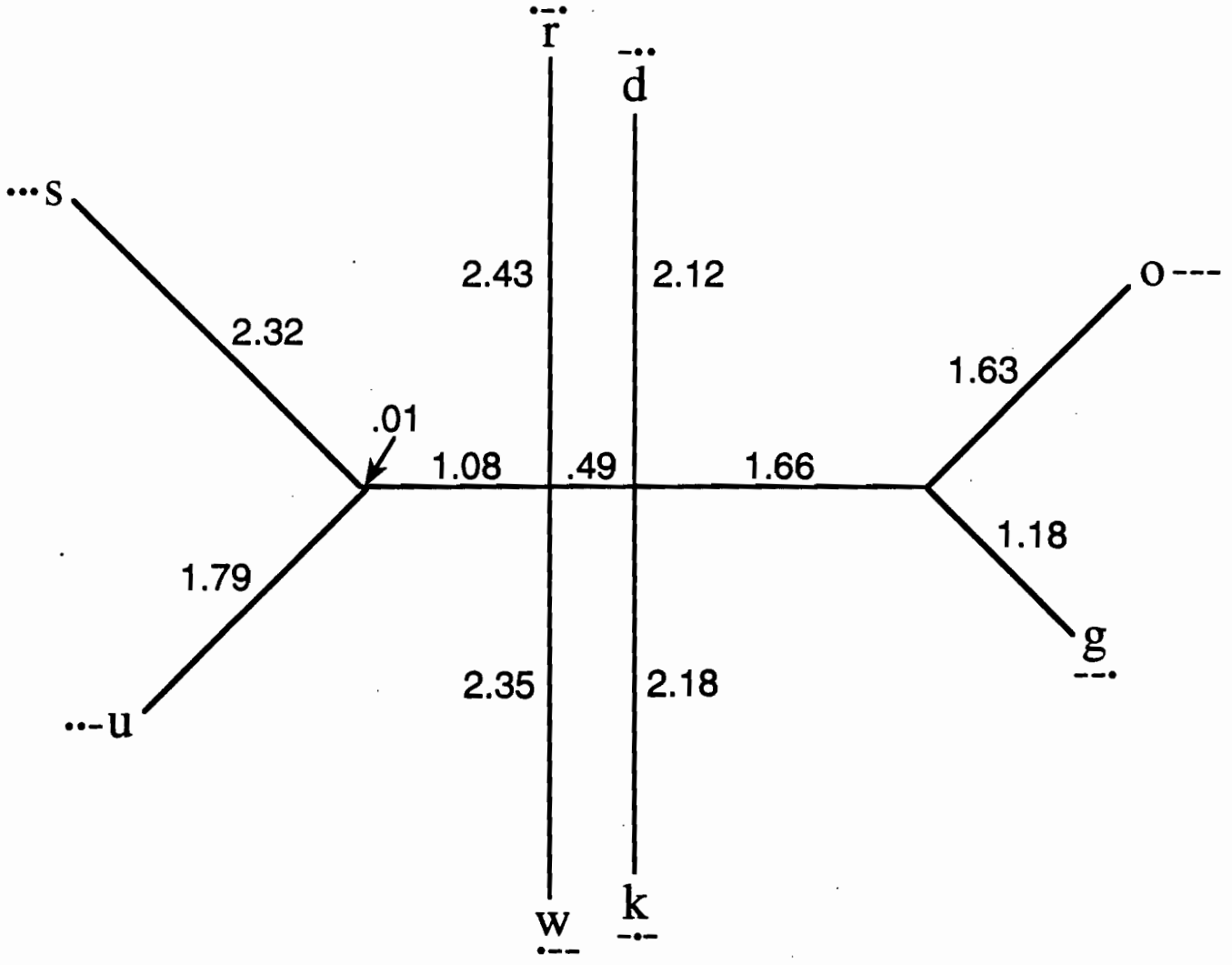
Figure 8


.093
.093
1.049
1.666
.795
1.065
.690
.760
1.171
1.089
1.069
.442
.898
1.611
d
k
g
o
r
w
s
u
-••
-•-
--•

•-•
•--
•••
••-

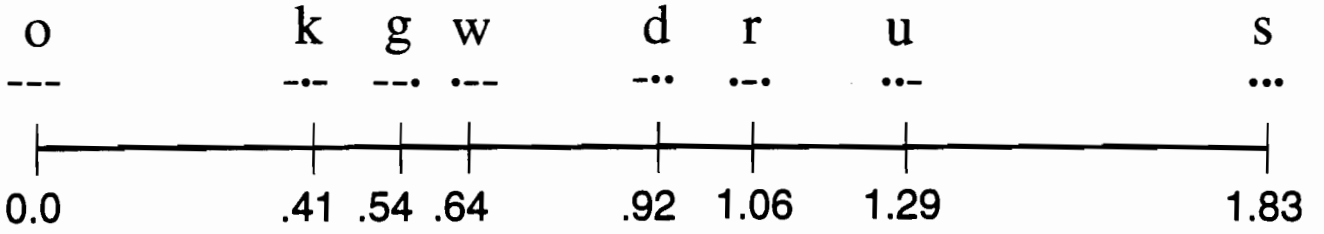
(c)

Figure 8

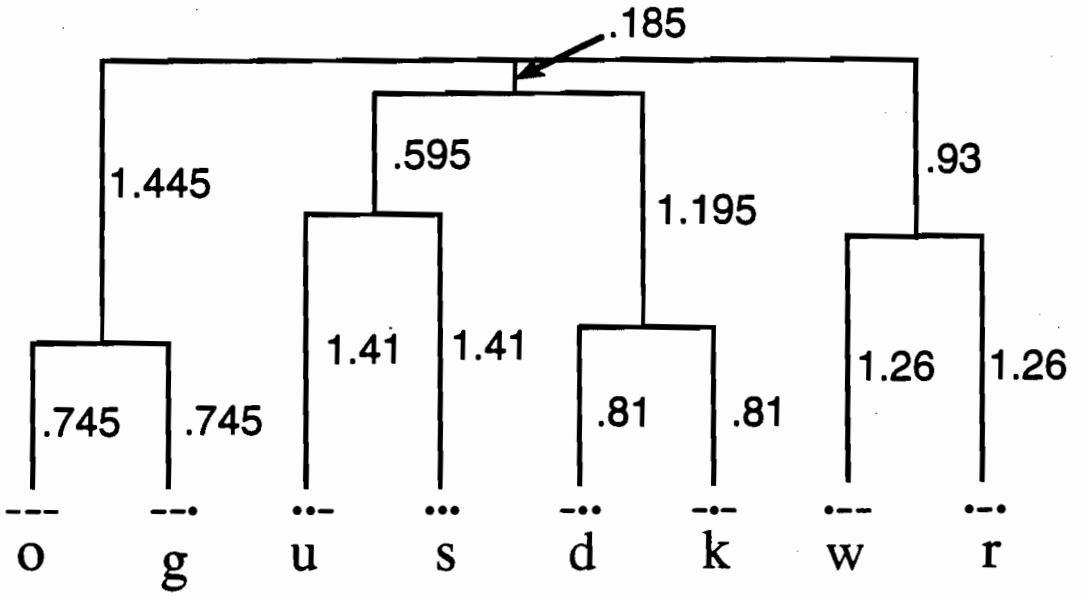


(a)

Figure 9


o
k
g
w
d
r
u
s

-•-
--•
•--
-••
•-•
•••
•••
0.0
.41
.54
.64
.92
1.06
1.29
1.83


.185
1.445
.595
1.195
.93
1.41
1.41
1.26
1.26
.745
.745
.81
.81

--•
•••
•••
-••
-•-
•--
•-•
o
g
u
s
d
k
w
r

(b)

Figure 9