Psychological Methods
2006, Vol. 11, No. 3, 271–286

1082-989X/06/$12.00 DOI: 10.1037/1082-989X.11.3.271

Clustering, Seriation, and Subset Extraction of Confusion Data

Michael J. Brusco
Florida State University

Douglas Steinley
University of Missouri—Columbia

The study of confusion data is a well established practice in psychology. Although many types of analytical approaches for confusion data are available, among the most common methods are the extraction of 1 or more subsets of stimuli, the partitioning of the complete stimulus set into distinct groups, and the ordering of the stimulus set. Although standard commercial software packages can sometimes facilitate these types of analyses, they are not guaranteed to produce optimal solutions. The authors present a MATLAB *.m file for preprocessing confusion matrices, which includes fitting of the similarity-choice model. Two additional MATLAB programs are available for optimally clustering stimuli on the basis of confusion data. The authors also developed programs for optimally ordering stimuli and extracting subsets of stimuli using information from confusion matrices. Together, these programs provide several pragmatic alternatives for the applied researcher when analyzing confusion data. Although the programs are described within the context of confusion data, they are also amenable to other types of proximity data.

Keywords: confusion matrix, similarity-choice model, cluster analysis, seriation, subset selection

A confusion matrix, $\mathbf{C} = [c_{ij}]$, is a square $(n \times n)$ matrix with rows that correspond to the n stimuli of an identification experiment and columns associated with the n possible responses for each presented stimulus. The main diagonal of a confusion matrix contains the number (or percentage) of correct identifications for each stimulus. Each off-diagonal element represents the number (or percentage) of times that the response associated with the column was mistakenly called (i.e., confused) by the subject for the stimulus corresponding to the row.

Empirical confusion matrices abound in various areas of psychological inquiry, including visual letter recognition (Fisher, Monty, & Glucksberg, 1969; Townsend, 1971a, 1971b; Van der Heijden, Malhas, & Roovaart, 1984), tactual letter recognition (K. O. Johnson & Phillips, 1981; Loomis, 1974, 1982; Vega-Bermudez, Johnson, & Hsiao, 1991), auditory recognition of letters and/or numbers (Hull, 1973; Morgan, Chambers, & Morton, 1973; Van Son & Pols, 1999), auditory recognition of words (Miralles & Cervera, 1995), recognition of lip-read letters (Manning & Shofner, 1991; Massaro, Cohen, & Gesi, 1993), odor recognition (Youngentob, Markert, Moxell, & Hornung, 1990), taste recognition (Hettinger, Gent, Marks, & Frank, 1999), recognition of texture patterns (Cho, Yang, & Hallett, 2000), morse code recognition (Rothkopf, 1957), and recognition of automotive control signals (Green & Pew, 1978). Although confusion matrices are principally published in experimental psychology journals, the pragmatic value of confusion experiments spans to other branches, such as social psychology (Derryberry, 1991; Lees & Neufeld, 1999), educational psychology (Coldren & Haaf, 2000; Compton, 2003), and human factors (Theise, 1989).

Quantitative modeling of confusion data has assumed many forms. Threshold models make up one class of models for stimulus recognition. One of the most well-known approaches in this category is Townsend's (1971a, 1971b) all-or-none model. This model is based on the theory that subjects either know the presented stimulus with certainty (all) or merely guess from among the set of available responses (none). Choice models are also used to analyze confusion data. One of the earliest models is Clarke's (1957) constant ratio rule, which posits that the probability of a particular response for a given stimulus should be the same regardless of the actual subset of stimuli in the choice set. Although there was some early evidence of the viability of this model (Clarke, 1957; Rich, 1971), more recent studies have generally rejected the constant

Michael J. Brusco, Marketing Department, College of Business, Florida State University; Douglas Steinley, Department of Psychological Sciences, University of Missouri—Columbia.

Additional materials are on the Web at http://dx.doi.org/10.1037/1082-989X.11.3.271.supp.

Douglas Steinley's work was partially supported by Office of Naval Research Grant N00014-06-0106.

Correspondence concerning this article should be addressed to Michael J. Brusco, Marketing Department, College of Business, Florida State University, Tallahassee, FL 32306-1110. E-mail: Mbrusco@cob.fsu.edu

ratio rule as a model of confusion data (Rouder, 2004; Townsend & Landon, 1982).

The Shepard–Luce similarity-choice model (Luce, 1963; Shepard, 1957) is perhaps the most frequently implemented model for confusion data. The model is defined by a symmetric similarity component and a response bias component. The similarity parameters, $0 \leq \eta_{ij} = \eta_{ji} \leq 1$, represent the perceptibility of the stimuli, where more similar stimulus pairs have greater confusability. Response biases are estimated for bias parameters, $0 \leq \beta_j \leq 1$. The probability of response j for presented stimulus i is computed as

$$P_{ij} = \frac{\eta_{ij}\beta_j}{\sum_{r=1}^{n} \eta_{ir}\beta_r}, \quad (1)$$

where the sum in the denominator is across all possible responses. The popularity of the similarity-choice model for analyzing confusion data is based on its efficacy. For example, Rouder (2004) observed that the similarity-choice model is the leading model for letter identification because it has consistently provided better fit than its competitors (Townsend & Landon, 1982) and is invariant to response bias (Townsend & Ashby, 1982).

Statistical comparison of two or more confusion matrices has also been implemented in the literature, and sometimes these comparisons have spanned psychological contexts. For example, Craig (1979); Loomis (1982); and Phillips, Johnson, and Browne (1983) presented comparisons of visual and tactual interletter confusion matrices using correlation measures. Brusco (2004) recently compared a much broader set of visual and tactual matrices using a permutation-based concordance method proposed by Hubert (1987, chap. 5), who demonstrated the technique using classic acoustic recognition confusion matrices originally published by Miller and Nicely (1955).

Structural representations of the stimuli associated with confusion matrices can also be obtained by methods of combinatorial data analysis (Hubert, Arabie, & Meulman, 2001). These methods include cluster analysis, seriation, and subset extraction. Cluster analysis methods, in particular, have often been used to study confusion data. Complete-link hierarchical clustering, which is also known as S. C. Johnson's (1967) *maximum method*, has been an especially popular choice for analyzing confusion data (Gilmore, Hersh, Caramazza, & Griffin, 1979; Hubert, 1973; Hubert & Baker, 1977; Loomis, 1982; Shepard & Arabie, 1979). The complete-link method begins with each stimulus in its own individual cluster. At each level, a pair of clusters is merged, reducing the total number of clusters by one. The criterion for selecting the two clusters to merge is to minimize the maximum dissimilarity element that will be produced by the merger. This process continues until all stimuli are in one cluster. The culmination of the complete-link hierarchical clustering process is a dendrogram (or tree representation) of the mergers at each level.

To illustrate the complete-link hierarchical clustering algorithm, consider the hypothetical 6×6 dissimilarity matrix on the left side of Table 1. The cluster mergers for each stage (denoted by the number of clusters) and the resulting maximum dissimilarity element produced by the merger are shown on the right side of the table. Moving from six clusters to five clusters requires the consideration of joining all possible pairs of individual objects, and the pair (1, 3) is selected because this produces the smallest dissimilarity element of .05. Next, the objects 4 and 6 are joined, reducing the number of clusters to 4. This process continues until all objects are in one cluster.

Although the dendrogram can be a useful visual aid, it is often a partition (or subset of a partition) that is of particular interest. These partitions comprise clusters that each possess a *diameter*, which represents the maximum dissimilarity between pairs of stimuli in that cluster. The partition diameter is the maximum of the cluster diameters. The relationship between complete-link hierarchical clustering and minimum-diameter partitioning is well documented (Baker & Hubert, 1976; Hansen & Delattre, 1978; Hubert, 1974), with Brusco and Cradit (2004) recently providing a discussion regarding the use of diameter-based partitioning methods within the context of confusion matrices. The crux of the relationship is that cutting the hierarchical tree for a complete-link solution does not always produce a minimum-

Table 1
A Small Synthetic Dissimilarity Matrix and the Stages of the Complete-Link Hierarchical Clustering Process

Object	Matrix						Stage	No. of clusters	Clusters of objects	Maximum dissimilarity
	1	2	3	4	5	6				
1	—	.24	.05	.37	.62	.68	1	6	{1} {2} {3} {4} {5} {6}	.00
2	.24	—	.86	.65	.70	.32	2	5	{1, 3} {2} {4} {5} {6}	.05
3	.05	.86	—	.59	.16	.41	3	4	{1, 3} {2} {4, 6} {5}	.12
4	.37	.65	.59	—	.39	.12	4	3	{1, 3, 5} {2} {4, 6}	.62
5	.62	.70	.16	.39	—	.64	5	2	{1, 3, 5} {2, 4, 6}	.65
6	.68	.32	.41	.12	.64	—	6	1	{1, 2, 3, 4, 5, 6}	.86

diameter partition, and the departure from optimality can be severe (Brusco & Cradit, 2004; Hansen & Delattre, 1978). To understand this limitation, we return to the right side of Table 1 and note that the complete-link solution for a three-cluster partition has a partition diameter of .62, which is produced by the object pair (1, 5) in the cluster {1, 3, 5}. This is not the minimum-diameter three-cluster partition. The minimum-diameter three-cluster partition is as follows: {1, 2}, {3, 5}, {4, 6}. These three clusters have diameters of .24, .16, and .12, respectively, and thus the minimum partition diameter is .24.

Although many commercial statistical software packages contain a program for complete-link hierarchical clustering, we are not aware of any packages that offer an option for minimum-diameter partitioning. A further complication is that there are often a host of alternative minimum-diameter partitions from which to choose, and the psychological research analyst typically does not have access to programs that facilitate selection from among the alternative optima. One of the principal contributions of this article is to mitigate this software accessibility problem.

We have developed three MATLAB (MathWorks, 2002) programs for the clustering of all stimuli associated with a confusion matrix. These programs are available in the supplemental material appearing on the Web for this article at http://dx.doi.org/10.1037/1082-989X.11.3.271.supp. The first program preprocesses the raw confusion matrix, producing several additional similarity and dissimilarity matrices that can be subjected to further analysis. The second program produces a minimum-diameter partition and is similar in design to the algorithms described by Brusco (2003) and Brusco and Cradit (2004). The program enables the analyst to select a specific dissimilarity matrix for the clustering process and to choose a desired number of clusters. The third program is useful when obtaining clustering solutions for two confusion matrices. This program allows the analyst to select from the set of alternative minimum-diameter solutions for each matrix so as to maximize the agreement between the resulting partitions. Although we could have developed more efficient versions of these programs in Fortran, we opted for a MATLAB implementation because of the user-friendly manner in which data can be manipulated, as well as the recent trend toward MATLAB models in the literature (Bogacz & Cohen, 2004; Brainard, 1997; Brown & Heathcote, 2002; Brusco & Cradit, 2005; Cheng & Gallistel, 2005; Henson & Douglas, 2005; Hubert, Arabie, & Meulman, 2002, 2006; Smith, Morgan, & White, 2005; Steinley, 2003; Walker & Milne, 2005).

The cluster analysis programs require a rigid partitioning of the stimulus set. In some cases, research analysts might prefer a "softer" classification of the data. Options for this type of analysis include hierarchical clustering, multidimensional scaling, and seriation. As noted previously, hierarchical clustering provides a treelike representation of the confusion data and is readily available in commercial software packages. Multidimensional scaling methods provide a spatial representation of the stimuli that reflects psychological distance among those stimuli. Like hierarchical clustering, multidimensional scaling methods are easily accessible and have been frequently applied to confusion matrices (Gilmore et al., 1979; Kikuchi, Yamashita, Sagawa, & Wake 1979; Kruskal, 1964; Townsend, 1971b; Zielman & Heiser, 1996).

Another alternative is to produce a seriation (or ordering) of the stimuli that uncovers structure in the data (Brusco, 2001, 2002; Hubert & Schultz, 1976). A natural extension of seriation is to produce a unidimensional scaling of the stimuli, which has also received attention in the literature (Brusco & Stahl, 2005b; Hubert, Arabie, & Meulman, 1997). Unlike hierarchical clustering and multidimensional scaling, algorithms for producing optimal permutations of stimuli are not available in commercial software packages. This is unfortunate because seriation of stimuli can often uncover interesting relationships among the stimuli. For example, suppose that the rows and columns of the confusion matrix could be permuted such that the elements above the main diagonal were always larger than their mirror image element below the main diagonal. The permutation would represent a perfect dominance relationship among the stimuli in the sense that if stimulus i precedes stimulus j in the permutation, then j is more often a mistaken response for i than i is a mistaken response for presented stimulus j.

To induce a dominance structure, we have developed a stand-alone *.m file that obtains a permutation of the stimuli to maximize the sum of elements above the main diagonal of the reordered matrix. This same matrix permutation problem has important applications in other contexts, such as minimum-feedback arcs in engineering (Lawler, 1964), production structure in economics (Chenery & Watanabe, 1958), paired-comparison ranking (Slater, 1961), and majority rule in social choice (Bowman & Colantoni, 1973).

In some instances, researchers have not sought to seriate, scale, or cluster the complete set of stimuli but instead have attempted to extract one or more stimulus subsets (Brusco & Stahl, 2001; Heiser, 1988). Several of these applications have been conducted within the context of ergonomic design. For example, Green and Pew (1978) and Theise (1989) focused on a goal of identifying a subset of automotive control signals that have minimum confusion among them. In similar applications, Zwaga and Boersema (1983) were concerned with recognition of graphic symbols in a railway station, whereas Moore (1974) attempted to select a viable subset of push buttons for postal sorting equipment. Brusco and Stahl (2001) described other possible applications in this vein, such as reduction of inventories of paint or perfumes on the basis of confusion data or the utilitarian design of cash registers in fast-food restaurants.

The extraction and deployment of subsets of letters has

also been used in a variety of other psychological applications. In a study designed to assess the effect of feedback on arousal, response, and attention, Derryberry (1991) used the letters *A*, *C*, and *F* as priming signals. These priming signals reflected feedback in terms of good (*A*), average (*C*), and poor (*F*) performance on the previous experimental trial. The importance of the low confusability of this subset of stimuli in Townsend's (1971b) study was a crucial factor for discounting possible discriminability effects with respect to the feedback. Lees and Neufeld (1999) selected eight letters (*J*, *L*, *M*, *A*, *Z*, *V*, *P*, and *G*), which served as analog stressors by pairing them with different levels of noise bursts (stress), in a study designed to measure their effect on indices of stress arousal and coping propensity. Lees and Neufeld explicitly noted that this subset was selected because the letters were maximally distinctive (low confusability) in Townsend's confusion matrix. Coldren and Haaf (2000) used three pairs of letter subsets (*E–F*, *Q–O*, and *B–R*) to assess whether infants exhibited a bias for either the presence or absence of a feature in a stimulus. Compton (2003) recently investigated the effects of letter substitutions in a subset of letters selected for a rapid automatized naming task. The importance of the choice-set size (i.e., subset size) on letter identification has also been recently studied by Rouder (2001, 2004). Using the Shepard–Luce choice model to produce a psychological distance, Rouder (2001) found that subjects exhibited better letter processing for larger subset sizes. Rouder (2004) subsequently demonstrated that certain models tend to overestimate a subject's use of the choice set, whereas other models foster underestimation.

Theise (1989) presented an integer programming model for extracting a subset of stimuli from a confusion matrix such that total confusion among pairs of stimuli is minimized. He also presented a bicriterion programming model that incorporated correct identification of subsets into the objective function. Brusco and Stahl (2001) demonstrated that more efficient integer programs, which used fewer binary decision variables to model subset membership decisions and fewer constraints, could be devised for this same problem. Brusco and Stahl also provided an enhanced integer programming formulation that permits analysts to select multiple subsets of stimuli such that the selected subsets maximize the within-subset sums of confusion entries minus the between-subsets sums of confusion entries.

Although the integer programming models developed by Theise (1989) and Brusco and Stahl (2001) facilitate optimal extraction of subsets, they are each inherently limited by the fact that they require mathematical programming software packages that are unfamiliar to most analysts in the field of psychology. To overcome this limitation, we developed a stand-alone MATLAB *.m file that accomplishes the same task. This subset extraction program is available in the Web supplement to this article (http://dx.doi.org/10.1037/1082-989X.11.3.271.supp).

Our goal is to provide analysts in the psychological community with an easy to use set of programs for analyzing their confusion data. These software programs produce optimal solutions to difficult combinatorial data analysis problems that are not handled by commercial statistical packages. In the next section, we describe the MATLAB programs for subset extraction and clustering of confusion data. This is followed by several examples using real confusion matrices from the psychological literature. The article concludes with a discussion and suggestions for future research. An important point in this final section is that the programs we have developed are not restricted to confusion matrices but could also be used for other types of $n \times n$ proximity matrices.

MATLAB Programs for Confusion Matrices

As noted previously, all of our programs for analyzing confusion data are written as MATLAB *.m files. To use these programs, users will need access to the base MATLAB system. Special toolboxes for statistical and optimization programs are not required. We provide general descriptions of how to use our confusion data *.m files in the MATLAB environment. For a thorough coverage of workspace management, file management, and *.m file development, we recommend that interested readers consult the MATLAB user's manual (MathWorks, 2002, chaps. 5–7).

Matrix Preparation

In many applications, some preprocessing of the raw confusion matrix is required prior to the application of a subset extraction or clustering program. We have developed a MATLAB *.m file, prepare.m, which produces six additional matrices based on the raw confusion matrix, $\mathbf{C} = [c_{ij}]$. The raw confusion matrix, *confusion_matrix*, is the only required input for the program, which is invoked by a function call. This is accomplished via the following process, (where ≫ is the command prompt in the MATLAB environment):

```
≫ load confusion_matrix;
≫ [s1, s2, s3, d1, d2, d3] = prepare
(confusion_matrix);
```

The six $n \times n$ output matrices correspond to the variable names in the brackets on the left-hand side of the function call. Like the confusion matrix itself, matrices *s1*, *s2*, and *s3* are *similarity* matrices because larger matrix elements indicate greater similarity between the corresponding pairs of stimuli. However, *s1*, *s2*, and *s3* differ from the confusion matrix because they are symmetric. Matrix *s1* is obtained by computing the arithmetic mean of mirror elements of the

confusion matrix [$(c_{ij} + c_{ji})/2$ for all $i \neq j$]. Although arithmetic averaging or addition of mirror elements is not uncommon in the literature (Cho et al., 2000; Loomis, 1982; Theise, 1989), matrix *s2* is produced using the geometric mean of the mirror elements, $\sqrt{c_{ij}c_{ji}}$, which can be justified on the basis that the confusion matrix elements are frequencies.

Matrix *s3* is established by the somewhat more sophisticated process of fitting the Shepard–Luce similarity-choice model (Luce, 1963; Shepard, 1957) to the confusion matrix. This is accomplished by using a proportional fitting scheme described by Heiser (1988), which is based on a procedure by Deming and Stephan (1940). For raw confusion matrices where $c_{ij} + c_{ji} = 0$ for any $i \neq j$, the program adds ½ to these cells to avoid division by zero in the estimation process, which is consistent with Heiser's implementation. If the analyst inputs a confusion matrix that has already been converted to confusion percentages, then .001 is added to the cells on the basis of Gilmore et al.'s (1979) recommendation.

The remaining three matrices have a *dissimilarity* interpretation, such that larger matrix elements represent less similarity between stimulus pairs. The first two dissimilarity matrices, *d1* and *d2*, are obtained directly from the similarity matrices *s1* and *s2*, respectively, by subtracting each off-diagonal element from the largest off-diagonal element in the appropriate similarity matrix. Shepard (1957) suggested that the negative of the natural logarithm of the similarity components of the choice model should behave in a manner similar to Euclidean distances (see also, Heiser, 1988). Therefore, the third dissimilarity matrix, *d3*, is obtained by taking $-\ln(s3)$. For all six matrices produced by the prepare.m program, the main diagonal is set to zero prior to completion of the program.

Partitioning Programs

We have developed a MATLAB *.m file, bbdiam.m, that produces a partition of stimuli on the basis of data in the confusion matrix. The primary computational method of bbdiam.m is a branch-and-bound algorithm, which is described in Appendix A. The bbdiam.m program requires only two inputs: (a) an $n \times n$ symmetric dissimilarity matrix (i.e., *d1*, *d2*, or *d3*) and (b) a desired number of clusters for the partition (*num_clusters*). The two outputs of bbdiam.m are the minimum partition diameter, *diameter*, and an $n \times 1$ vector, *partition*, containing the cluster assignment for each stimulus. The implementation of the program requires the following steps:

1. The user selects one of three dissimilarity matrices for analysis, which is defined as matrix *a*, and also defines a desired number of clusters (*num_clusters*). The user subsequently invokes the program bbdiam.m with a function call. For a four-cluster solution based on *d1*, this process is accomplished using the following statements:

```
≫ a = d1;
≫ num_clusters = 4;
≫ [partition, diameter] =
    bbdiam(a, num_clusters);
```

2. A branch-and-bound algorithm similar to the one designed by Brusco and Cradit (2004) is then used to obtain a minimum-diameter partition. The optimal partition is stored in the vector *partition*, whereas the minimum partition diameter is stored in *diameter*.

When processing is complete, the output variables (*partition* and *diameter*) are automatically produced in the MATLAB environment and can be viewed by simply typing the variable name at the command prompt. The analyst can explore different solutions by changing *num_clusters* and assessing the impact on the partition diameter as well as the interpretability of the solution. In addition, optimal partitions for a fixed number of clusters can be compared for different dissimilarity matrices (e.g., *d1* vs. *d2*).

An important aspect of the minimum-diameter criterion is the potential for a large number of alternative optimal partitions (Baker & Hubert, 1976; Brusco & Cradit, 2004, 2005; Guénoche, 1993; Hansen & Jaumard, 1997). When an analyst is comparing partition structures for two or more matrices, alternative optima can be especially problematic. For example, suppose we obtained minimum-diameter partitions for two different dissimilarity matrices related to auditory confusion under different levels of noise. Further, assume that the agreement between these two partitions is low, indicating serious departures in cluster structure between the two matrices. It is quite possible that there are other minimum-diameter partitions for these same two matrices that would exhibit much stronger agreement. For this reason, we have prepared the *.m file, randopt.m, which attempts to find minimum-diameter partitions for two different matrices, such that the agreement between the two partitions is maximized.

The randopt.m program uses Hubert and Arabie's (1985) adjusted Rand index (ARI) as a measure of partition agreement. The formula for computing the ARI between two partitions, π^1 and π^2, is as follows:

$$\text{ARI} = \frac{M(\omega_1 + \omega_2) - [(\omega_1 + \omega_3)(\omega_1 + \omega_4) + (\omega_2 + \omega_3)(\omega_2 + \omega_4)]}{M^2 - [(\omega_1 + \omega_3)(\omega_1 + \omega_4) + (\omega_2 + \omega_3)(\omega_2 + \omega_4)]}, \quad (2)$$

where $M = n(n - 1)/2$, ω_1 is the number of stimulus pairs in the same cluster in π^1 and the same cluster in π^2, ω_2 is the number of stimulus pairs in different clusters for π^1 and in different clusters for π^2, ω_3 is the number of stimulus pairs in the same cluster for π^1 but in different clusters for

π^2, and ω_4 is the number of stimulus pairs in the same cluster in π^2 but different clusters for π^1. An ARI of 1 indicates perfect agreement of the two partitions. A value of 0 indicates only chance agreement. The effectiveness of the ARI for cluster validation has been demonstrated by Milligan and Cooper (1986) and, more recently, by Steinley (2004).

The randopt.m program requires that the analyst obtain minimum-diameter partitions for two dissimilarity matrices, *mata* and *matb*, using bbdiam.m. The optimal partitions for these two matrices are denoted *parta* and *partb*, and the corresponding diameters are stored as *diama* and *diamb*. The randopt.m program reads these data and applies a heuristic algorithm to modify *parta* and *partb* with the objective of maximizing ARI, while not allowing *diama* or *diamb* to increase. A complete cycle of the heuristic tests the effect of moving each stimulus in *parta* from its current cluster to each of the other clusters and then each stimulus in *partb* from its current cluster to each of the other clusters. Each time a stimulus is tested for relocation to another cluster, a check is made to determine if the partition diameter constraint will be exceeded. If the relocation will not cause *diama* or *diamb* to be exceeded, then the effect on ARI is computed. The feasible relocation that produces the largest increase in ARI is implemented at the end of the cycle. New cycles are initiated until there is no feasible relocation of a stimulus that will increase ARI. The implementation of the program requires the following steps:

1. The user obtains optimal partitions, *parta* and *partb* using bbdiam.m for two selected dissimilarity matrices (*mata* and *matb*) and a desired number of clusters. The minimum diameters for these partitions are assigned the variable names *diama* and *diamb*, respectively.

2. The user invokes the program randopt.m with the following function call:

```
≫ [ari,newa,newb] = randopt(mata,matb,parta,
partb,diama,diamb);
```

3. A heuristic algorithm is used to find alternative minimum-diameter partitions with maximum agreement as measured by the ARI. The new partitions are stored in *newa* and *newb*, and the corresponding ARI is stored in *ari*.

Seriation Program

We have developed a MATLAB *.m file, bbdom.m, that will produce a seriation of the n stimuli on the basis of data in the confusion matrix. The only input required for this program is the $n \times n$ confusion matrix (*confusion_matrix*). The bbdom.m program uses a branch-and-bound algorithm, which is described in Appendix B, to produce a reordering of the rows and columns of the confusion matrix so as to maximize the sum of confusion elements above the main diagonal of the reordered matrix. The two primary outputs are the above-diagonal sum of confusion elements for the optimally reordered matrix (*domindex*) and the optimal permutation of the stimuli *permopt*.

The bbdom.m program also produces several secondary outputs that are helpful for comparisons across confusion matrices. One of these outputs is the linearity index (*lindx*), which standardizes the dominance index by dividing *domindex* by the sum of the off-diagonal elements. The remaining outputs, *con* and *incon*, represent, respectively, the number of consistencies and inconsistencies in the optimally reordered matrix. A consistency (inconsistency) occurs when an element above the main diagonal is greater (less) than its mirror element below the main diagonal. The implementation of the bbdom.m program requires the following steps:

1. The user invokes the program bbdom.m with the following function call:

```
≫ [domindex,permopt,lindx,con,incon] =
bbdom(confusion_matrix);
```

2. A branch-and-bound algorithm similar to the one described by Brusco and Stahl (2005a, chaps. 7–8) is then used to find the optimal permutation of stimuli. The optimal permutation is stored in the vector *permopt*, whereas the maximum above-diagonal sum of confusion elements is stored in *domindex*. The linearity index is stored in *lindx*, and *con* and *incon* represent the number of consistencies and inconsistencies, respectively.

When the algorithm terminates, the *reordered matrix* can be obtained using the following command:

```
≫ reordered_matrix = confusion_matrix
(permopt,permopt);
```

The sum above the main diagonal for this reordered matrix is *domindex*.

Subset Extraction Programs

We have developed a MATLAB *.m file, bbsubset.m, that will extract exactly K subsets of size T from a proximity matrix. The primary computational method of bbsubset.m is a branch-and-bound algorithm, which is described in Appendix C. The bbsubset.m program offers considerable flexibility. For example, when applied to *s1* for the case of $K = 1$, bbsubset.m will identify a subset of size T with maximum total confusion within the subset. If a subset of size T with minimum confusion is desired (Green & Pew, 1978; Theise, 1989), the analyst need only multiply *s1* by -1 prior to the function call.

When applied to *s1* for the case of $K > 1$, bbsubset.m will extract K subsets of size T so as to maximize the within-subset sums minus the between-subsets sums, which is the criterion selected by Brusco and Stahl (2001) and is consistent with the subset extraction goals outlined by Heiser (1988). Heiser observed that the extraction of multiple pairs

(i.e., $T = 2$) of stimuli is an especially important problem in experimental contexts, with applications to foreknowledge of position in single-item recognition (Van der Heijden, Schreuder, & Wolters, 1985), as well as single-item recognition with multiple-item visual arrays (Van der Heijden, 1986). The underlying premise in both of these applications is that misrecognition of a stimulus should principally be associated with the other member of its pair and that confusion with a member from another pair should be rare and likely result from random guessing.

In its current form, bbsubset.m is restricted to selection of subsets of the same size (i.e., T is the same for all clusters). For $K = 1$, which has been used in a number of applications (Green & Pew, 1978; Lees & Neufeld, 1999; Theise, 1989), this is obviously not a relevant issue. For $K \geq 2$, we have only come across instances where pairs of stimuli (i.e., $T = 2$ for all subsets) were selected (Coldren & Haaf, 2000; Heiser, 1988). Our program provides a more general subset extraction model by allowing $T > 2$. Although bbsubset.m could be modified to allow for multiple subsets of different sizes, there appears to be no current need for this extension on the basis of applications observed in the literature. The implementation of the program requires the following steps:

1. The user selects one of the similarity matrices for analysis, which is defined as matrix *a*, and also specifies a number of subsets, *num_subsets*, and the subset size, *sub_size*. The user subsequently invokes the program bbdiam.m with a function call. For a four-subset solution consisting of two stimuli per subset and based on *s1*, the MATLAB commands are as follows:

```
≫ a = s1;
≫ num_subsets = 4;
≫ sub_size = 2;
≫ [subset, index] = bbsubset(a, num_
subsets, sub_size);
```

2. A branch-and-bound algorithm is used to extract the subsets to maximize the subset selection criterion. The optimal subset is stored in the vector *subset*, whereas the maximum criterion value is stored in *index*.

MATLAB provides an accommodating environment for viewing the results of the selected subsets obtained from bbsubset.m. The following commands provide a convenient display of the confusion submatrix (*submatrix*) for the extracted subset(s):

```
≫ subset = subset(1:num_subsets.*
sub_size);
≫ submatrix = confuse(subset,subset);
```

For the case of *num_subsets* = 1, *submatrix* contains the within-subset confusion entries for the selected subset. For *num_subsets* > 1, *submatrix* will contains a series of $T \times T$ submatrices of within-subset confusion along the main diagonal, and all elements outside these submatrices represent between-subsets confusion.

Demonstration for Partitioning

There are a variety of examples from the literature that address structural comparison of confusion matrices that were obtained under different experimental conditions (Brusco, 2004; Cho et al., 2000; Loomis, 1982; Miller & Nicely, 1955; Morgan et al., 1973; Phillips et al., 1983; Townsend, 1971a, 1971b). To illustrate the MATLAB partitioning programs for confusion matrices, we use two confusion matrices published by Vega-Bermudez et al. (1991, p. 537). The first matrix corresponds to tactual confusion among uppercase letters of the alphabet under conditions of "active touch," whereas the second matrix corresponds to confusions for the same stimulus set under "passive touch." In the active touch experiment, subjects were able to reach out and stroke the embossed stimulus letter. In contrast, for passive touch, the subject's arm was immobilized and the embossed stimulus letter was pressed against the subject's index finger. Our goal is to investigate the confusion structures for active and passive touch in tactual letter recognition.

We used prepare.m to produce the similarity and dissimilarity matrices for the active touch and passive touch confusion matrices. We subsequently applied bbdiam.m to the Shepard distances (matrix *d3*) for these two matrices for $2 \leq K \leq\leq 10$ clusters. Optimal partitions were obtained in less than 1 CPU s for all values of K for both matrices (2.2 GHz, Pentium 4 PC, 1 gigabyte RAM). Table 2 reports the optimal partition diameters for both matrices for each value of K. The table also contains the percentage reduction in the diameter that was realized from increasing the number of clusters, which Hansen and Delattre (1978) recommended as a guideline for selecting the appropriate number of clusters.

Table 2 shows that there was very little improvement in partition diameter for either active touch or passive touch on the range of $2 \leq K \leq 5$ clusters; however, some significant improvements occurred thereafter. We selected the 9-cluster partition for active touch because there was a large improvement (9.41%) when moving from 8 to 9 clusters, but a much smaller improvement (4.68%) when moving from 9 to 10 clusters. Similarly, we adopted the 8-cluster solution for passive touch because of the large improvement (11.03%) when moving from 7 to 8 clusters and the small improvement when moving from 8 to 9 clusters (2.35%). After selecting these solutions, we applied the randopt.m program to produce minimum-diameter partitions with maximal agreement as measured by the ARI. The resulting minimum-diameter partitions for active touch and passive touch are displayed in Table 3.

The agreement between the active touch and passive touch partitions, as measured by ARI, was .73. On the basis of Steinley's (2004) experimental analysis of the index, this would be characterized as fairly strong agreement. The

Table 2
Optimal Partition Diameters ($2 \leq K \leq 10$) for the Shepard's Pseudo-Distance Matrices Obtained for the Active and Passive Touch Confusion Matrices Published by Vera-Bermudez et al. (1991, p. 537)

	Active touch		Passive touch	
Number of clusters (K)	Optimal diameter	% diameter reduction	Optimal diameter	% diameter reduction
2	5.6668		5.8792	
3	5.5078	2.81	5.7148	2.80
4	5.4886	0.35	5.5567	2.77
5	5.2431	4.47	5.2459	5.59
6	4.3578	16.89	5.0433	3.86
7	4.0015	8.18	4.2525	15.68
8	3.9877	0.34	3.7835	11.03
9	3.6125	9.41	3.6944	2.35
10	3.4434	4.68	3.6139	2.18

partitions in Table 3 are ordered to facilitate comparison of the similarities and differences. The first four clusters of the two partitions are identical. These include two of the larger clusters, {*B*, *C*, *D*, *G*, *O*, *Q*} and {*F*, *P*, *T*, *Y*}, and two singleton clusters, {*I*} and {*L*}. The fifth clusters for the two partitions are very similar, differing only in that {*Z*} is included in the fifth cluster for active touch, but not passive touch. The sixth cluster for active touch, {*H*, *M*, *W*}, is a subset of the sixth cluster for passive touch, {*H*, *M*, *R*, *U*, *W*}.

The comparison of minimum-diameter partitions also helps to uncover some important differences in confusion structure for active touch and passive touch. For example, consider Clusters 5 and 7. If, for the active touch partition, the letter *Z* could be moved from Cluster 5 to Cluster 7, then Cluster 5 would be the same for the two partitions and Cluster 7 would also exhibit greater agreement. Unfortunately, this was not possible and a closer inspection of the confusion matrix uncovers the reason. For active touch, the letter *J* was never offered as a mistaken response for stimulus *Z* and vice versa. Therefore, these two letters are dissimilar in the active touch data and could not be placed in the same cluster. For passive touch, *J* was offered as a mistaken response for stimulus *Z* 5 times, and *Z* as a mistaken response for stimulus *J* 6 times. Thus, there was much greater confusability between the two letters in the passive touch study, and therefore, the two letters could be placed in the same cluster.

Table 3
Minimum-Diameter Partitions for the Shepard's Pseudo-Distance Matrices Obtained for the Active Touch and Passive Touch Confusion Matrices Published by Vera-Bermudez et al. (1991, p. 537)

Cluster number (k)	Cluster memberships	
	Active touch	Passive touch
1	{*B, C, D, G, O, Q*}	{*B, C, D, G, O, Q*}
2	{*F, P, T, Y*}	{*F, P, T, Y*}
3	{*I*}	{*I*}
4	{*L*}	{*L*}
5	{*E, K, N, X, Z*}	{*E, K, N, X*}
6	{*H, M, W*}	{*H, M, R, U, W*}
7	{*J*}	{*J, S, Z*}
8	{*A, R, S*}	{*A, V*}
9	{*U, V*}	

On the whole, the results of the partitioning analysis confirm the findings of Vega-Bermudez et al. (1991) that there is a strong similarity in the confusion structure for active and passive touch in tactual letter recognition. However, the partitioning algorithm also uncovers some salient differences in confusion that might be worthy of further investigation. For example, why are the letters *J* and *Z* never confused in active touch but frequently confused in passive touch?

Demonstration for Seriation

We applied bbdom.m to the active touch and passive touch confusion matrices published by Vega-Bermudez et al. (1991, p. 537). The optimal permutations for these two matrices are as follows:

Active touch: *H P T A L V Y J Z F S X C G B Q N I M K E R D U O W*

Passive touch: *P T Y J L Z X F I S E C G B A R H N M Q D U O V W K*

The optimal above-diagonal sum for the active touch matrix is 1,268, and 1,046 CPU s were required to obtain the optimal permutation. The optimal permutation for the passive touch matrix was obtained in 188 s, and the optimal

above-diagonal sum is 1,528. The linearity indices for active touch and passive touch are .72 and .73, respectively. The optimally reordered active touch confusion matrix has 165 consistencies and 44 inconsistencies, whereas the corresponding figures for the optimally reordered passive touch matrix are 167 and 36. The rank correlation coefficient between the two optimal permutations is .64 ($p < .001$).

The optimal permutations for the two matrices do exhibit some evident structural similarities. For example, 7 of the first 10 letters in both permutations are *F*, *J*, *L*, *P*, *T*, *Y*, and *Z*. The sequence of these 7 letters in the two permutations are also similar. For the active touch permutation, the order of the 7 letters is *P–T–L–Y–J–Z–F*. The order is the same for passive touch, with the exception that the letter *L* is moved two places to the right. The ends of the optimal permutations for the two matrices are also comparable. Four of the last 6 letters in both permutations are *D*, *U*, *O*, and *W*, and these letters are in the same order in both permutations.

It is also insightful to link the seriation results to the partitioning results discussed in the previous section. For example, consider the cluster {*B*, *C*, *D*, *G*, *O*, *Q*}, which is Cluster 1 for both the active touch and passive touch data. The ordering of these letters is *C–G–B–Q–D–O* in both the optimal permutation for the active touch confusion matrix and the optimal permutation for the passive touch matrix. This same property holds for the cluster {*F*, *P*, *T*, *Y*}, which is Cluster 2 in both partitions in Table 3. The order of these four letters in both optimal permutations is *P–T–Y–F*.

Like the partitioning results in the previous section, the optimal seriation of letters supports the theory that the confusion structures for active touch and passive touch have some degree of agreement. However, once again, we observe some marked differences in the two permutations. For example, the letters *E*, *K*, *N*, and *X* appear in Cluster 5 of both the active touch and passive touch partitions. However, the order of these stimuli in the active touch partition is *X–N–K–E*, whereas the order for passive touch is *X–E–N–K*. Once again, a closer inspection of the confusion matrix reveals why this occurs. For passive touch, stimulus *K* is a mistaken response for stimulus *E* 17 times, and *E* is a mistaken response for *K* 16 times, which is consistent with *E* being to the left of *K* in the permutation. For active touch, however, stimulus *K* is a mistaken response for stimulus *E* 10 times, and *E* is a mistaken response for *K* 18 times, which creates strong pressure to place *K* to the left of *E* in the permutation.

Whereas the partitioning analysis provides information for comparison of active and passive touch with respect to the magnitude of confusion, the dominance analysis provides additional comparative information regarding symmetry. Under passive touch, the confusion between *E* and *K* was nearly symmetric; however, in the active touch experiment, *E* was a mistaken response for *K* far more often than *K* was a mistaken response for *E*.

Demonstration for Subset Extraction

We applied bbsubset.m to each of the similarity matrices (arithmetic average, geometric average, and similarity choice) associated with the active touch and passive touch confusion matrices published by Vega-Bermudez et al. (1991, p. 537). Following the example outlined by Heiser (1988), the goal was to select $K = 4$ pairs ($T = 2$) of letters with high similarity within subsets and low similarity between subsets. The key questions of interest are the following: (a) Are the subsets extracted for active touch consistent with those extracted for passive touch, and (b) how consistent are the subsets across the different similarity matrices? The optimal subsets and measure of fit indices for each similarity matrix are displayed in Table 4.

The optimal subsets for the similarity matrices based on arithmetic and geometric means were extracted in less than 1 CPU s, whereas those based on the Shepard–Luce similarity-choice model required 5 s and 17 s for active touch and passive touch, respectively. For active touch, the optimal pairs of letters extracted from the arithmetic and geo-

Table 4
Optimal Subset Extraction for the Active Touch and Passive Touch Similarity Matrices Published by Vera-Bermudez et al. (1991, p. 537), Assuming That $K = 4$ Subsets of Size $T = 2$ Are Desired

Confusion data	Similarity matrix	Optimal measure of fit, $h^*(S_1, \ldots, S_4)$	Optimal subsets (letter pairs)			
			Pair 1	Pair 2	Pair 3	Pair 4
Active touch	Arithmetic	105.0000	(*F, P*)	(*M, W*)	(*O, Q*)	(*X, Z*)
	Geometric	93.9630	(*F, P*)	(*M, W*)	(*O, Q*)	(*X, Z*)
	SCM	1.6844	(*B, S*)	(*M, W*)	(*V, Y*)	(*X, Z*)
Passive touch	Arithmetic	135.5000	(*F, P*)	(*M, W*)	(*O, Q*)	(*X, Z*)
	Geometric	103.6500	(*M, W*)	(*O, Q*)	(*V, Y*)	(*X, Z*)
	SCM	1.4006	(*F, P*)	(*N, R*)	(*O, Q*)	(*X, Z*)

Note. SCM = similarity-choice model.

metric similarity matrices were identical ({*F*, *P*}, {*M*, *W*}, {*O*, *Q*}, and {*X*, *Z*}). For the similarity-choice matrix, {*F*, *P*} and {*O*, *Q*} were replaced with {*B*, *S*} and {*V*, *Y*}.

For passive touch, the optimal pairs of letters extracted from the arithmetic matrix were identical to those extracted for the active touch arithmetic matrix. Three of the four optimal pairs of letters extracted from the geometric matrix for passive touch were identical to those extracted for the active touch geometric matrix; however, {*V*, *Y*} replaced {*F*, *P*} as the fourth pair. For passive touch, the optimal pairs of letters extracted from the similarity-choice matrix differed violently from those extracted for the active touch similarity-choice matrix, with {*X*, *Z*} as the only consistent pair between the optimal subsets.

Across the six solutions reported in Table 4, the pair {*X*, *Z*} is selected 6 times, the pairs {*M*, *W*} and {*O*, *Q*} are each selected 5 times, and the pair {*F*, *P*} is selected 4 times. No other pair is selected more than twice. In addition, {*F*, *P*}, {*M*, *W*}, {*O*, *Q*}, and {*X*, *Z*} composed the optimal collection of subsets for three of the six solutions. Therefore, we selected these pairs as our consensus subset. The confusion matrix subsets corresponding to these pairs are displayed in Tables 5 and 6 for active touch and passive touch, respectively. The within-subset elements are encased within the boxes along the main diagonal, whereas the between-subsets elements correspond to all remaining values. We note that the within-subset confusion elements for the four subsets are large, whereas the between-subsets confusion elements are much smaller. This is indicative of considerable similarity within subsets and separation between subsets.

Like the partitioning and seriation results, the optimal subset extraction of letter pairs reveals considerable similarity between the active touch and passive touch data. The four optimal pairs of letters for these two data sources are identical when arithmetic averaging is used to produce the similarity matrices. When geometric averaging is used, three of the four pairs are the same. There was, however, a somewhat profound difference in the optimal letter pairs for active touch and passive touch when the similarity-choice model was used to produce the similarity matrix.

Discussion

Summary of Methods

The analysis of confusion matrices is a well-recognized problem in the psychological literature. Unfortunately, some of the most important methodological tools for modeling confusion data are not widely available. We have attempted to address this gap by providing a set of MATLAB programs for confusion data. We selected MATLAB because of its flexible interface, as well as to build on the recent trend of MATLAB software for combinatorial data analysis. The first program, prepare.m, preprocesses a confusion matrix by producing six new matrices derived from the raw confusion data. The preprocessing ranges from simple arithmetic and geometric averaging of mirror elements to the creation of a similarity matrix based on the popular Shepard–Luce similarity-choice model. After running prepare.m, the analyst is left with three additional similarity matrices and three dissimilarity matrices for possible analysis. The modification of the program for variations of these matrices should be straightforward.

We have also created two MATLAB programs, bbdiam.m and randopt.m, which can be used for diameter-based partitioning of the dissimilarity matrices generated by prepare.m. Using bbdiam.m, a quantitative analyst can obtain optimal partitions for dissimilarity matrices based on geometric or arithmetic averaging, as well as a psychological distance measure based on the Shepard–Luce choice model. These analyses can be completed for various numbers of clusters, and confidence can be gained by observing similar clustering patterns

Table 5
The Submatrix for a Four-Group, Two-Member Subset Extraction Solution for the Active Touch Confusion Matrix Published by Vega-Bermudez et al. (1991, p. 537)

Letter	*F*	*P*	*M*	*W*	*O*	*Q*	*X*	*Z*
F	—	**18**	0	0	0	0	3	0
P	**29**	—	0	1	0	1	0	0
M	2	0	—	**37**	0	0	0	0
W	2	0	**19**	—	0	1	0	0
O	0	0	0	0	—	**5**	0	0
Q	0	1	1	0	**64**	—	0	0
X	0	0	3	3	0	1	—	**16**
Z	5	0	0	2	0	0	**48**	—

Note. Values in bold are the within-group terms, whereas the remaining entries are between-groups terms.

Table 6
The Submatrix for a Four-Group, Two-Member Subset Extraction Solution for the Passive Touch Confusion Matrix Published by Vega-Bermudez et al. (1991, p. 537)

Letter	*F*	*P*	*M*	*W*	*O*	*Q*	*X*	*Z*
F	—	**18**	1	1	0	0	1	1
P	**46**	—	0	2	0	1	1	2
M	0	1	—	**59**	0	2	0	0
W	0	4	**17**	—	0	0	0	2
O	0	2	0	0	—	**4**	0	0
Q	0	2	0	0	**94**	—	0	0
X	3	0	2	5	0	1	—	**22**
Z	4	1	1	1	0	0	**52**	—

Note. Values in bold are the within-group terms, whereas the remaining entries are between-groups terms.

for different dissimilarity matrices. If the relationship between the partitions for two dissimilarity matrices is antagonistic, the analyst can use the program randopt.m to identify partitions that improve the agreement as measured by the ARI, while maintaining minimum-diameters for both data sources.

We have prepared a seriation program, bbdom.m, which can be used to obtain a permutation of the stimuli that maximizes the sum of the confusion elements above the main diagonal of the reordered matrix. This procedure is less restrictive than the rigid partitioning process imposed by bbdiam.m and can help to uncover asymmetry in the data. We have also developed a program, bbsubset.m, that can be applied to one of the derived similarity matrices. The program is designed to accommodate the broader problem of extracting K subsets of size T so as to maximize within-subset confusion minus between-subsets confusion. For the case of $K = 1$, the problem reduces to the identification of a single subset with maximum within-subset confusion (there is no between-subsets confusion because there is only one subset). If minimum within-subset confusion is desired, the analyst need only multiply all elements of the similarity matrix by -1 prior to running bbsubset.m.

Limitations and Extensions

Although the branch-and-bound algorithms were successfully applied to some of the larger confusion matrices published in the literature, the algorithms can be sensitive to matrix characteristics. The bbdiam.m and randopt.m programs should be reliable for matrices of 50×50 or larger, particularly when there are fewer than 10 clusters. The bbdom.m and bbsubset.m programs are a bit more sensitive to the matrix size and structure and could conceivably require hours of CPU time for some matrices of size 30×30 or larger. Although there are exceptions (Hull, 1973; Rothkopf, 1957), most confusion matrices published in the psychological literature are 26×26 or smaller and can be accommodated by all of our MATLAB programs.

Another important issue associated with our programs is the uniqueness of the solutions provided by the algorithms. There are frequently multiple (often many) minimum-diameter partitions, and in such cases, the final partition obtained by bbdiam.m is not unique. The randopt.m program facilitates tie breaking among minimum-diameter partitions; however, it is sensitive to the initial input partitions and does not necessarily produce optimal partition agreement. The potential for alternative optima also exists for the seriation (bbdom.m) and subset extraction (bbsubset.m) programs; however, the uniqueness problem is much less severe for these applications.

We have limited our demonstrations and discussion to the analysis of confusion matrices because of the prevalence and criticality of these data in the psychological literature. However, the clustering programs are amenable to any $n \times n$ symmetric dissimilarity matrix, perhaps produced by a paired-comparison or sorting experiment. The bbdom.m program could be applied to a variety of asymmetric matrices, such as paired-comparison or journal cross-citation matrices. In a similar vein, there is no reason that bbsubset.m could not be applied to any $n \times n$ symmetric similarity matrix. Thus, it should be clear that the methods described herein have portability beyond the confusion data context.

References

Baker, F. B., & Hubert, L. J. (1976). A graph-theoretic approach to goodness of fit in complete-link hierarchical clustering. *Journal of the American Statistical Association, 71,* 870–878.

Bogacz, R., & Cohen, J. D. (2004). Parameterization of connectionist models. *Behavior Research Methods, Instruments, & Computers, 36,* 732–741.

Bowman, V. J., & Colantoni, C. S. (1973). Majority rule under transitivity constraints. *Management Science, 19,* 1029–1041.

Brainard, D. H. (1997). Psychophysics software for use with MATLAB. *Spatial Vision, 10,* 433–436.

Brown, S., & Heathcote, A. (2002). On the use of nonparametric regression in assessing parametric regression models. *Journal of Mathematical Psychology, 46,* 716–730.

Brusco, M. J. (2001). Seriation of asymmetric matrices using integer linear programming. *British Journal of Mathematical and Statistical Psychology, 54,* 367–375.

Brusco, M. J. (2002). Identifying a reordering of rows and columns for multiple proximity matrices using multiobjective programming. *Journal of Mathematical Psychology, 46,* 731–745.

Brusco, M. J. (2003). An enhanced branch-and-bound algorithm for a partitioning problem. *British Journal of Mathematical and Statistical Psychology, 56,* 83–92.

Brusco, M. J. (2004). On the concordance among empirical confusion matrices for visual and tactual letter recognition. *Perception & Psychophysics, 66,* 392–397.

Brusco, M. J., & Cradit, J. D. (2004). Graph coloring, minimum-diameter partitioning, and the analysis of confusion matrices. *Journal of Mathematical Psychology, 48,* 301–309.

Brusco, M. J., & Cradit, J. D. (2005). Bicriterion methods for partitioning dissimilarity matrices. *British Journal of Mathematical and Statistical Psychology, 58,* 319–332.

Brusco, M. J., & Stahl, S. (2001). Compact integer-programming models for extracting subsets of stimuli from confusion matrices. *Psychometrika, 66,* 405–420.

Brusco, M. J., & Stahl, S. (2005a). *Branch-and-bound applications in combinatorial data analysis.* New York: Springer.

Brusco, M. J., & Stahl, S. (2005b). Optimal least squares unidimensional scaling: Improved branch-and-bound procedures and a comparison to dynamic programming. *Psychometrika, 70,* 253–270.

Chenery, H. R., & Watanabe, T. (1958). International comparisons of the structure of production. *Econometrica, 26,* 487–521.

Cheng, K., & Gallistel, C. R. (2005). Shape parameters explain data from spatial transformations: Comment on Pearce et al. (2004) and Tommasi & Polli (2004). *Journal of Experimental Psychology: Animal Behavior Processes, 31,* 254–259.

Cho, R. Y., Yang, V., & Hallett, P. E. (2000). Reliability and dimensionality of judgments of visually textured materials. *Perception & Psychophysics, 62,* 735–752.

Clarke, F. R. (1957). Constant-ratio rule for confusion matrices in speech communication. *Journal of the Acoustical Society of America, 29,* 715–720.

Coldren, J. T., & Haaf, R. (2000). Asymmetries in infants' attention to the presence or absence of features. *Journal of Genetic Psychology, 161,* 420–434.

Compton, D. L. (2003). The influence of item composition on RAN letter performance in first-grade children. *Journal of Special Education, 37,* 81–94.

Craig, J. C. (1979). A confusion matrix for tactually presented letters. *Perception & Psychophysics, 26,* 409–411.

DeCani, J. S. (1972). A branch and bound algorithm for maximum likelihood paired comparison ranking by linear programming. *Biometrika, 59,* 131–135.

Deming, W. E., & Stephan, F. F. (1940). On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *Annals of Mathematical Statistics, 11,* 427–444.

Derryberry, D. (1991). The immediate effects of positive and negative feedback signals. *Journal of Personality and Social Psychology, 61,* 267–278.

Fisher, D. F., Monty, R. A., & Glucksberg, S. (1969). Visual confusion matrices: Fact or artifact? *Journal of Psychology, 71,* 111–125.

Flueck, J. A., & Korsh, J. F. (1974). A branch search algorithm for maximum likelihood paired comparison ranking. *Biometrika, 61,* 621–626.

Gilmore, G. C., Hersh, H., Caramazza, A., & Griffin, J. (1979). Multidimensional letter similarity derived from recognition errors. *Perception & Psychophysics, 25,* 425–431.

Green, P., & Pew, R. W. (1978). Evaluating pictographic symbols: An automotive application. *Human Factors, 20,* 103–114.

Guénoche, A. (1993). Enumération des partitions de diamètre minimum [Enumeration of the partitions of minimum diameter]. *Discrete Mathematics, 111,* 277–287.

Hansen, P., & Delattre, M. (1978). Complete-link cluster analysis by graph coloring. *Journal of the American Statistical Association, 73,* 397–403.

Hansen, P., & Jaumard, B. (1997). Cluster analysis and mathematical programming. *Mathematical Programming, 79,* 191–215.

Heiser, W. J. (1988). Selecting a stimulus set with prescribed structure from empirical confusion frequencies. *British Journal of Mathematical and Statistical Psychology, 41,* 37–51.

Henson, R., & Douglas, J. (2005). Test construction for cognitive diagnosis. *Applied Psychological Measurement, 29,* 262–277.

Hettinger, T. P., Gent, J. F., Marks, L. E., & Frank, M. E. (1999). A confusion matrix for the study of taste perception. *Perception & Psychophysics, 61,* 1510–1521.

Hubert, L. (1973). Min and max hierarchical clustering using asymmetric proximity measures. *Psychometrika, 38,* 63–72.

Hubert, L. J. (1974). Some applications of graph theory to clustering. *Psychometrika, 39,* 283–309.

Hubert, L. J. (1987). *Assignment methods in combinatorial data analysis.* New York: Marcel Dekker.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification, 2,* 193–218.

Hubert, L., Arabie, P., & Meulman, J. (1997). Linear and circular unidimensional scaling for symmetric proximity matrices. *British Journal of Mathematical and Statistical Psychology, 50,* 253–284.

Hubert, L., Arabie, P., & Meulman, J. (2001). *Combinatorial data analysis: Optimization by dynamic programming.* Philadelphia: Society for Industrial and Applied Mathematics.

Hubert, L., Arabie, P., & Meulman, J. (2002). Linear unidimen-

sional scaling in the L_2 norm: Basic optimization methods using MATLAB. *Journal of Classification, 19,* 303–328.

Hubert, L., Arabie, P., & Meulman, J. (2006). *The structural representation of proximity matrices with MATLAB.* Philadelphia: Society for Industrial and Applied Mathematics.

Hubert, L. J., & Baker, F. B. (1977). The comparison and fitting of given classification schemes. *Journal of Mathematical Psychology, 16,* 233–253.

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

Hull, A. J. (1973). A letter-digit matrix of auditory confusions. *British Journal of Psychology, 64,* 579–585.

Johnson, K. O., & Phillips, J. R. (1981). Tactile spatial resolution. I. Two-point discrimination, gap detection, grating resolution, and letter recognition. *Journal of Neurophysiology, 46,* 1177–1191.

Johnson, S. C. (1967). Hierarchical clustering schemes. *Psychometrika, 32,* 241–254.

Kikuchi, T., Yamashita, Y., Sagawa, K., & Wake, T. (1979). An analysis of tactile letter confusions. *Perception & Psychophysics, 26,* 295–301.

Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness-of-fit to a nonmetric hypothesis. *Psychometrika, 29,* 1–28.

Lawler, E. L. (1964). A comment on minimum feedback arc sets. *IEEE Transactions on Circuit Theory, 11,* 296–297.

Lees, M. C., & Neufeld, R. W. J. (1999). Decision-theoretic aspects of stress arousal and coping personality. *Journal of Personality and Social Psychology, 77,* 185–208.

Loomis, J. M. (1974). Tactile letter recognition under different modes of stimulus presentation. *Perception & Psychophysics, 16,* 401–408.

Loomis, J. M. (1982). Analysis of tactile and visual confusion matrices. *Perception & Psychophysics, 31,* 41–52.

Luce, R. D. (1963). Detection and recognition. In R. D. Luce, R. Bush, & E. Galanter (Eds.), *Handbook of mathematical psychology* (Vol. 1, pp. 103–189). New York: Wiley.

Manning, S. K., & Shofner, E. (1991). Similarity ratings and confusability of lipread consonants compared with similarity ratings of auditory and orthographic stimuli. *American Journal of Psychology, 104,* 587–604.

Massaro, D. W., Cohen, M. M., & Gesi, A. T. (1993). Long-term training, transfer, and retention in learning to lipread. *Perception & Psychophysics, 53,* 549–562.

MathWorks. (2002). *Using MATLAB* (Version 6). Natick, MA: Author.

Miller, G. A., & Nicely, P. E. (1955). Analysis of perceptual confusions among some English consonants. *Journal of the Acoustical Society of America, 27,* 338–352.

Milligan, G. W., & Cooper, M. C. (1986). A study of the comparability of external criteria for hierarchical cluster analysis. *Multivariate Behavioral Research, 21,* 441–458.

Miralles, J. L., & Cervera, T. (1995). Voice intelligibility in patients who have undergone laryngectomies. *Journal of Speech and Hearing Research, 38,* 564–571.

Moore, T. G. (1974). Tactile and kinaesthetic aspects of push-buttons. *Applied Ergonomics, 5,* 66–71.

Morgan, B. J. T., Chambers, S. M., & Morton, J. (1973). Acoustic confusion of digits in memory and recognition. *Perception & Psychophysics, 14,* 375–383.

Phillips, J. R., Johnson, K. O., & Browne, H. M. (1983). A comparison of visual and two modes of tactual letter resolution. *Perception & Psychophysics, 34,* 243–249.

Rich, R. (1971). Constant ratio rule for confusion matrices from short-term memory experiments. *British Journal of Psychology, 62,* 27–35.

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

Rouder, J. N. (2001). Absolute identification with simple and complex stimuli. *Psychological Science, 12,* 318–322.

Rouder, J. N. (2004). Modeling the effects of choice-set size on the processing of letters and words. *Psychological Review, 111,* 80–93.

Shepard, R. N. (1957). Stimulus and response generalization: A stochastic model relating generalization to distance in psychological space. *Psychometrika, 22,* 325–345.

Shepard, R. N., & Arabie, P. (1979). Additive clustering: Representation of similarities as combinations of discrete overlapping properties. *Psychological Review, 86,* 87–123.

Slater, P. (1961). Inconsistencies in a schedule of paired comparisons. *Biometrika, 48,* 303–312.

Smith, J. L., Morgan, C. L., & White, P. H. (2005). Investigating a measure of computer technology domain identification: A tool for understanding gender differences and stereotypes. *Educational and Psychological Measurement, 65,* 336–355.

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

Steinley, D. (2004). Properties of the Hubert–Arabie adjusted Rand index. *Psychological Methods, 9,* 386–396.

Theise, E. S. (1989). Finding a subset of stimulus-response pairs with minimum total confusion: A binary integer programming approach. *Human Factors, 31,* 291–305.

Townsend, J. T. (1971a). Alphabetic confusion: A test of models for individuals. *Perception & Psychophysics, 9,* 449–454.

Townsend, J. T. (1971b). Theoretical analysis of an alphabet confusion matrix. *Perception & Psychophysics, 9,* 40–50.

Townsend, J. T., & Ashby, F. G. (1982). Experimental test of contemporary mathematical models of visual letter recognition. *Journal of Experimental Psychology: Human Perception and Performance, 8,* 836–864.

Townsend, J. T., & Landon, D. E. (1982). An experimental and theoretical investigation of the constant-ratio rule and other models of visual letter confusion. *Journal of Mathematical Psychology, 25,* 119–162.

Van der Heijden, A. H. C. (1986). On a selection in vision. *Psychological Research, 48,* 211–220.

Van der Heijden, A. H. C., Malhas, M. S., & Roovaart, B. P. (1984). An empirical interletter confusion matrix for continuous-line capitals. *Perception & Psychophysics, 35,* 85–88.

Van der Heijden, A. H. C., Schreuder, R., & Wolters, G. (1985). Enhancing single-item recognition accuracy by cueing spatial locations in vision. *Quarterly Journal of Experimental Psychology: Human Experimental Psychology, 37*(A), 427–434.

Van Son, R. J. J. H., & Pols, L. C. W. (1999). Perisegmental speech improves consonant and vowel identification. *Speech Communication, 29,* 1–22.

Vega-Bermudez, F., Johnson, K. O., & Hsiao, S. S. (1991). Human tactile pattern recognition: Active versus passive touch, velocity effects, and patterns of confusion. *Journal of Neurophysiology, 65,* 531–546.

Walker, I., & Milne, S. (2005). Exploring function estimators as an alternative to regression in psychology. *Behavior Research Methods, 37,* 23–36.

Youngentob, S. L., Markert, L. M., Moxell, M. M., & Hornung, D. E. (1990). A method for establishing a five odorant identification confusion matrix task in rats. *Physiology and Behavior, 47,* 1053–1059.

Zielman, B., & Heiser, W. J. (1996). Models for asymmetric proximities. *British Journal of Mathematical and Statistical Psychology, 49,* 127–146.

Zwaga, H. J., & Boersema, T. (1983). Evaluation of a set of graphic symbols. *Applied Ergonomics, 14,* 43–54.

Appendix A

Branch-and-Bound Algorithm for Minimum-Diameter Partitioning

We define $N = \{1, 2, \ldots, n\}$ as the set of indices for the n stimuli and $\mathbf{A} = [a_{ij}]$ as an $n \times n$ dissimilarity matrix obtained from the raw confusion data. The goal is to find a minimum-diameter, K-cluster partition, $\pi_K = \{S_1, S_2, \ldots, S_K\}$ of the n stimuli, where S_k is the subset of stimuli assigned to cluster k ($1 \leq k \leq K$). A mathematical statement of the minimum-diameter partitioning problem is

$$\min : f(\pi_K) = \max_{k=1,\ldots,K} \left(\max_{(i<j)\in S_k} (a_{ij}) \right) \tag{A1}$$

subject to

$$S_k \neq \emptyset \text{ for } 1 \leq k \leq K; \tag{A2}$$

$$S_l \cap S_k = \emptyset \text{ for } 1 \leq l < k \leq K; \tag{A3}$$

and

$$S_1 \cup S_2 \cup \ldots \cup S_K = N. \tag{A4}$$

The objective function, Equation A1, represents the partition diameter. The constraints in Equations A2, A3, and A4 require the clusters to be nonempty, mutually exclusive, and exhaustive, respectively.

The branch-and-bound solution procedure is adopted from Brusco and Stahl (2005a, chaps. 2–3). We denote p as a pointer that marks position in the search tree, τ as the number of empty clusters, and n_k as the number of stimuli in cluster k, for $1 \leq k \leq K$. The vector $\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_p]$ contains the cluster memberships of the first p stimuli ($1 \leq j \leq p$) and represents a partial solution in the search process. The vector $\lambda^* = [\lambda_1, \lambda_2, \ldots, \lambda_n]$ is the incumbent (best found) complete partition of the stimulus set. The partition diameters for λ and λ^* are denoted $f(\lambda)$ and $f(\lambda^*)$, respectively. The steps of the algorithm are as follows:

Step 0. Obtain λ^* and $f(\lambda^*)$ using multiple restarts of an exchange algorithm. Set $p = 0$, $\tau = K$, $\lambda_j = 0$ for $1 \leq j \leq n$, and $n_k = 0$ for $1 \leq k \leq K$.

Step 1. Set $p = p + 1$, $k = 1$, $n_k = n_k + 1$, $\lambda_p = k$. If $n_k = 1$, then set $\tau = \tau - 1$.

Step 2. If $n - p < \tau$, go to Step 6.

Step 3. If $[a_{ip} \geq f(\lambda^*) \mid \lambda_i = k]$, for any $1 \leq i \leq p - 1$, then go to Step 6.

Step 4. If $\min_k [\max_{q=1,\ldots,p} (a_{jq} | \lambda_q = k)] \geq f(\lambda^*)$, for any $p + 1 \leq j \leq n$, then go to Step 6.

Step 5. If $p \neq n$, then go to Step 1. Otherwise, set $\lambda^* = \lambda$ and store $f(\lambda^*)$.

Step 6. If $k = K$ or $n_k = 1$, then go to Step 8.

Step 7. Set $n_k = n_k - 1$. Set $k = k + 1$ and $n_k = n_k + 1$. If $n_k = 1$, then set $\tau = \tau - 1$. Set $\lambda_p = k$ and return to Step 2.

Step 8. Set $\lambda_p = 0$. Set $n_k = n_k - 1$ and $p = p - 1$. If $n_k = 0$, then set $\tau = \tau + 1$. If $p = 0$, then return the incumbent solution, which is an optimal solution, and Stop; otherwise, set $k = \lambda_p$ and return to Step 6.

Appendix B

Branch-and-Bound Algorithm for Seriation

Let $\mathbf{C} = [c_{ij}]$ denote the $n \times n$ confusion matrix, where c_{ij} = the number (or percentage) of responses for stimulus j when stimulus i is presented. Define Γ as the set of all $n!$ feasible permutations of the n stimuli, and let γ represent an arbitrary permutation from that set. Further, let $\gamma(k)$ indicate the stimulus in position k of permutation γ. The matrix permutation problem can now be concisely stated as

$$\max_{\gamma \in \Gamma} : g(\gamma) = \sum_{k<l} c_{\gamma(k)\gamma(l)}. \tag{B1}$$

The problem posed by Equation B1 requires finding a permutation, γ^*, that maximizes the sum of the elements above the main diagonal, $g(\gamma^*)$. DeCani (1972) and Flueck and Korsh (1974) proposed branch-and-bound algorithms for obtaining optimal solutions to the matrix triangularization problem. Their methods are based on a general branch-and-bound paradigm that can be used for a variety of matrix permutation problems (Brusco & Stahl, 2005a, chaps. 7–11). We adopt our branch-and-bound solution procedure from Brusco and Stahl (2005a, chaps. 7–8). We denote p as a pointer that marks the current position in the partial sequence of stimuli. In addition, we obtain a lower bound, g_{LB}, for $g(\gamma^*)$ using a pairwise interchange heuristic. The steps of the algorithm are as follows:

Step 0. Obtain the lower bound, g_{LB} using a pairwise interchange heuristic solution, γ_B. Set $p = 1$, $\gamma(p) = 1$, $\gamma(k) = 0$ for $k = 2, \ldots, n$.

Step 1. Set $p = p + 1$.

Step 2. Set $\gamma(p) = \gamma(p) + 1$.

Step 3. If $\gamma(p) = \gamma(k)$ for any $1 \leq k \leq p - 1$, then go to Step 2.

Step 4. If $p = 1$ and $\gamma(p) > n$, then return γ_B as γ^* and Stop.

Step 5. If $p > 1$ and $\gamma(p) > n$, then go to Step 11.

Step 6. If $p = n - 1$, then let $\gamma(n) = j \mid j \neq \gamma(k)$ for any $1 \leq k \leq n - 1$, and go to Step 7. Otherwise, go to Step 8.

Step 7. Compute $g(\gamma)$. If $g(\gamma) > g_{LB}$, then set $g_{LB} = g(\gamma)$ and $\gamma_B = \gamma$. Go to Step 2.

Step 8. Perform adjacency test: If $c_{\gamma(p-1)\gamma(p)} \geq c_{\gamma(p)\gamma(p-1)}$, then go to Step 9; otherwise, go to Step 2.

Step 9. Perform insertion test: If $\sum_{r=q+1}^{p} c_{\gamma(q)\gamma(r)} < \sum_{r=q}^{p-1} c_{\gamma(p)\gamma(r)}$ for any $1 \leq q \leq p - 2$, then go to Step 2; otherwise go to Step 10.

Step 10. Perform bound test: If $\sum_{k=1}^{p-1} \sum_{l=k+1}^{p} c_{\gamma(k)\gamma(l)} + \sum_{(i<j) \notin R_p} \max(c_{ij}, c_{ji}) > g_{LB}$, where R_p = the set of stimuli assigned to the first p positions, then go to Step 1. Otherwise, go to Step 2.

Step 11. Backtrack in the stimulus sequence by setting $\gamma(p) = 0$ and $p = p - 1$. Go to Step 2.

Appendix C

Branch-and-Bound Algorithm for Subset Extraction

We define $\mathbf{B} = [b_{ij}]$ as an $n \times n$ similarity matrix obtained from the raw confusion data. The goal is to extract K subsets $\{S_1, S_2, \ldots, S_K\}$ of size T from a complete set of n stimuli, so as to maximize the sum of the within-subset similarities minus the sum of the between-subsets similarities. A mathematical statement of the subset extraction problem is

$$\max : h(S_1, S_2, \ldots, S_K) = \sum_{k=1}^{K} \sum_{(i<j) \in S_k} b_{ij} - \left(\sum_{k=1}^{K-1} \sum_{i \in S_k} \sum_{l=k+1}^{K} \sum_{j \in S_l} b_{ij} \right) \tag{C1}$$

subject to

$$S_l \cap S_k = \emptyset \text{ for } 1 \leq l < k \leq K; \tag{C2}$$

and

$$|S_k| = T \text{ for } 1 \leq k \leq K. \tag{C3}$$

Equation C1 represents the measure of fit for the subset extraction problem, which corresponds to the sum of the within-cluster dissimilarities minus the sum of the between-clusters dissimilarities. The constraint in Equation C2 guarantees that a stimulus is not selected for more than one subset, and the constraint in Equation C3 requires the subset size to be T for all subsets. Modification of

(*Appendixes continue*)

the model for different sizes for different subsets could be accomplished; however, we have not found any subset extraction problems in the literature where this was necessary.

We use a modification of the branch-and-bound algorithm described in Appendix B to produce optimal solutions to the subset extraction problem. We obtain a lower bound, h_{LB}, for $h(\gamma^*)$ using a pairwise interchange heuristic. We also make use of the rank ordering (descending) of the pairwise similarity elements in **B**. For any pointer position in the search tree, p, there is a particular number of within-subset similarity terms that have been collected, y_1, and a particular number that remain to be collected, y_2. A best-case bound for those terms that remain to be collected can be obtained by scanning the ranked list of similarities and summing the first y_1 terms. We call this sum $\theta(p)$. In a similar manner, summing the last y_2 terms in the ranked array provides a best-case bound for the between-subsets contribution, and we denote this sum as $\rho(p)$. The steps of the algorithm are as follows:

Step 0. Obtain the lower bound, h_{LB}, using a pairwise interchange heuristic solution, γ_B. Set $p = 1$, $\gamma(p) = 1$, $\gamma(k) = 0$ for $k = 2 \ldots n$. Define $\upsilon[T(k - 1)+t] = k$ for $1 \leq k \leq K$ and $1 \leq t \leq T$.

Step 1. Set $p = p + 1$.

Step 2. Set $\gamma(p) = \gamma(p) + 1$.

Step 3. If $\gamma(p) = \gamma(k)$ for any $1 \leq k \leq p - 1$, then go to Step 2.

Step 4. If $\upsilon(p) = \upsilon(p - 1)$ and $\gamma(p) < \gamma(p - 1)$, then go to Step 2.

Step 5. If $\upsilon(p) \neq \upsilon(p - 1)$ and $\gamma(p) < \gamma(p - T)$, then go to Step 2.

Step 6. If $p = 1$ and $\gamma(p) > n$, then return γ_B as γ^* and Stop.

Step 7. If $p > 1$ and $\gamma(p) > n$, then go to Step 11.

Step 8. If $p = KT$, then go to Step 10. Otherwise, go to Step 9.

Step 9. Compute $h(\gamma)$. If $h(\gamma) > h_{LB}$, then set $h_{LB} = h(\gamma)$ and $\gamma_B = \gamma$. Go to Step 2.

Step 10. Perform bound test by computing Equation C1 for the current partial solution and adding the contribution $\theta(p)$ and subtracting $\rho(p)$. If this quantity exceeds h_{LB}, then go to Step 1. Otherwise, go to Step 2.

Step 11. Backtrack in the stimulus sequence by setting $\gamma(p) = 0$ and $p = p - 1$. Go to Step 2.

Received July 25, 2005
Revision received December 5, 2005
Accepted April 17, 2006 ■