Behav Res (2016) 48:487–502
DOI 10.3758/s13428-015-0587-y

Biclustering methods for one-mode asymmetric matrices

Michael J. Brusco[1,5] · Patrick Doreian[2,3] · Douglas Steinley[4]

Published online: 22 April 2015

Abstract An asymmetric one-mode data matrix has rows and columns that correspond to the same set of objects. However, the *roles* of the objects frequently differ for the rows and the columns. For example, in a visual alphabetic confusion matrix from an experimental psychology study, both the rows and columns pertain to letters of the alphabet. Yet the rows correspond to the presented stimulus letter, whereas the columns refer to the letter provided as the response. Other examples abound in psychology, including applications related to interpersonal interactions (friendship, trust, information sharing) in social and developmental psychology, brand switching in consumer psychology, journal citation analysis in any discipline (including quantitative psychology), and free association tasks in any subarea of psychology. When seeking to establish a partition of the objects in such applications, it is overly restrictive to require the partitions of the row and column objects to be identical, or even the numbers of clusters for the row and column objects to be the same. This suggests the need for a biclustering approach that simultaneously establishes separate partitions of the row and column objects. We present and compare several approaches for the biclustering of one-mode matrices using data sets from the empirical literature. A suite of MATLAB m-files for implementing the procedures is provided as a Web supplement with this article.

Keywords Clustering · Biclustering · One-mode asymmetric matrix · Two-mode *KL*-means partitioning · Two-mode blockmodeling · Nonnegative matrix factorization

A common data structure is a $\Re^{N \times N}$ matrix, $\mathbf{X} = [x_{ij}]$, where x_{ij} is some measure of proximity (similarity or dissimilarity) between objects *i* and *j*, for all $1 \leq i \leq N$ and $1 \leq j \leq N$ objects. In some instances, **X** is a symmetric matrix, with $x_{ij} = x_{ji}$ for all $1 \leq i \leq N$ and $1 \leq j \leq N$ objects. For example, when the *N* objects are measured on a collection of metric variables, a nonnegative $N \times N$ symmetric dissimilarity version of **X** can be established on the basis of the pairwise Euclidean distances between pairs of objects. Symmetry is present because the Euclidean distance between objects *i* and *j* is the same as the distance between *j* and *i*. The dissimilarity interpretation is appropriate because larger matrix elements (i.e., distances) imply greater dissimilarity (rather than greater similarity) among the pairs of objects.

Although symmetric matrices are common, there are also many psychological applications for which **X** is an asymmetric matrix. For example, in experimental psychology, when studying confusion among a set of stimulus objects, x_{ij} (for $i \neq j$) is commonly a measure of the number of instances (or the percentage of instances) for which subjects mistakenly responded with stimulus object *j* when stimulus object *i* was actually presented (see Brusco & Steinley, 2006). For any given pair of objects (*i*, *j*), asymmetry can potentially occur because a presented stimulus *i* could be mistaken more frequently for *j* than stimulus *j* would be mistaken for *i*, or vice versa. Likewise, for brand-switching applications in consumer psychology, where x_{ij} (for $i \neq j$) is a measure reflecting the

✉ Michael J. Brusco
mbrusco@fsu.edu

[1] Florida State University, Tallahassee, FL, USA

[2] University of Ljubljana, Ljubljana, Slovenia

[3] University of Pittsburgh, Pittsburgh, PA, USA

[4] University of Missouri, Columbia, MO, USA

[5] Department of Marketing, College of Business, Florida State University, 821 Academic Way, Tallahassee, FL 32306-1110, USA

degree to which the consumers in a sample switch from brand i to brand j, asymmetry occurs because switches from i to j might be more or less frequent than those from j to i. Asymmetry is also apt to be present in matrices associated with social network ties among schoolchildren (in developmental psychological studies—e.g., Anderson, Wasserman, & Crouch, 1999; Parker & Asher, 1993), social groups (in social psychological studies—e.g., Gibbons & Olk, 2003), and organizational members (in studies from industrial/applied psychology—e.g., Totterdell, Wall, Holman, Diamond, & Epitropaki, 2004). Regardless of whether these network examples concern friendship, trust, advice seeking, information sharing, or some other type of relational tie, the potential for asymmetry arises because ties need not be reciprocated: Actor i could identify actor j as a friend, someone they trust, or from whom they seek advice, but actor j might not reciprocate such ties.

Although the rows and columns of the matrix pertain to the same set of objects, the asymmetry in **X** generally stems from the objects having two different roles. In confusion matrices, the role of the row objects is the "presented stimulus," whereas the role of the column objects is the "response to a presented stimulus." Similarly, for brand-switching applications, the role of the row objects is "previously purchased brand" (or "switched from"), with the role of the column objects being "most recently purchased brand" (or "switched to"). In social network situations, the roles of the row and column actors are commonly "tie senders" and "tie receivers," respectively. For journal citation networks, the row journals could be "cited journals" (or "producers"), with the column journals being "citing journals" (or "consumers").

There are a variety of methods for analyzing asymmetric proximity data (a recent review of some of these approaches has been provided by Vicari, 2014). Moreover, regardless of the type of method selected, transformation or decomposition processes are sometimes applied prior to analysis. For example, one popular approach is to decompose **X** into its symmetric ($\mathbf{\Psi}$) and skew-symmetric ($\mathbf{\Lambda}$) components and to pursue analysis of these components independently or jointly (Brusco & Stahl, 2005a; Hubert, 1987, chap. 4; Hubert, Arabie, & Meulman, 2001, chap. 4; Zielman & Heiser, 1996). The elements of the symmetric component are $\psi_{ij} = \psi_{ji} = (x_{ij} + x_{ji})/2$, whereas the elements of the skew-symmetric component are $\lambda_{ij} = (x_{ij} - x_{ji})/2$ for all $1 \leq i, j \leq N$. The decomposition of **X** is then provided by $\mathbf{X} = \mathbf{\Psi} + \mathbf{\Lambda}$.

The methods for analyzing asymmetric data include graphical approaches, seriation, unidimensional scaling, multidimensional scaling, hierarchical clustering, and nonhierarchical clustering. Graphical representation procedures have been described by Constantine and Gower (1978) and Chino (1978). Seriation methods, which seek to develop orderings of the objects associated with an asymmetric matrix, have also been proposed by numerous authors (Baker & Hubert, 1977; Brusco, 2001; Brusco & Stahl, 2005b; DeCani, 1972; Flueck & Korsh, 1974; Hubert, 1976; Hubert et al., 2001, chap. 4; Ranyard, 1976). Multidimensional-scaling approaches for asymmetric matrices have been developed by Harshman, Green, Wind, and Lundy (1982), Okada and Imaizumi (1987), and Zielman and Heiser (1996). Here, however, we restrict attention to methods for clustering or partitioning the objects associated with the one-mode proximity data. Hubert (1973) provided one of the earliest investigations of this topic in his discussion of the application of hierarchical clustering methods to asymmetric data. A more recent review of hierarchical methods was provided by Takeuchi, Saito, and Yadohisa (2007), and Brusco and Steinley (2006) discussed approaches within the framework of partitioning.

The presence of asymmetry in a one-mode proximity matrix presents a challenge when one is seeking to identify a partition of the N objects. One approach is to collapse **X** into a symmetric proximity matrix using some type of transformation procedure, and then to apply traditional one-mode partitioning methods to obtain a solution (see, e.g., Brusco & Steinley, 2006, for a discussion of this approach within the context of confusion data). Unfortunately, this approach discards information associated with the asymmetry. For example, two journals might have very similar roles as producers for the journals that cite them, but markedly different roles as consumers of the journals they cite (see, e.g., Brusco, 2011; Brusco & Doreian, 2015b). A second approach is to sequentially establish partitions based on the row objects and column objects. In the first stage, a symmetric nonnegative matrix of Euclidean distances between the row objects could be established by treating the columns as variables, and a partition of the row objects could then be obtained using hierarchical or nonhierarchical clustering. This process would then be repeated for the column objects by treating the rows as variables. A caveat associated with this approach is that the main diagonal elements are often arbitrary and, therefore, not appropriate for use when computing the distance matrices.

A preferable approach is to adopt a *biclustering* (Madeira & Oliveira, 2004; Prelić et al., 2006; Van Uitert, Meuleman, & Wessels, 2008; Wilderjans, Depril, & Van Mechelen, 2013) perspective that seeks to *simultaneously* establish two distinct partitions of the objects: (i) one based on their role as row objects, and (ii) one based on their role as column objects. The term *biclustering* is used broadly here, and it is noted that alternative terminology, such as *two-mode clustering* or *two-mode partitioning*, is also used in some instances. As with the sequential method, biclustering creates flexibility by allowing different numbers of clusters for the row and column objects. However, unlike the sequential approach, the entire asymmetric proximity matrix is used when establishing the row-object and column-object partitions.

Succinctly, the approach adopted herein is to treat the one-mode asymmetric matrix **X** as though it is two-mode. Strictly, a two-mode matrix has data in which the N row objects are completely distinct from the M column objects. Psychological examples of two-mode matrices include memberships of CEOs on boards of directors and the scores of examinees on test items. A formidable research effort has been devoted to *two-mode* partitioning problems (Brusco & Doreian, 2015a, b; Brusco, Doreian, Lloyd, & Steinley, 2013; Brusco, Doreian, Mrvar, & Steinley, 2013; Brusco & Steinley, 2007, 2009, 2011; Doreian, Batagelj, & Ferligoj, 2004, 2005; Doreian, Lloyd, & Mrvar, 2013; Schepers, Ceulemans, & Van Mechelen, 2008; Schepers & Van Mechelen, 2011; Schepers, Van Mechelen, & Ceulemans, 2011; Van Mechelen, Bock, & DeBoeck, 2004; van Rosmalen, Groenen, Trejos, & Castillo, 2009; Vichi, 2001; Wilderjans, Depril, & Van Mechelen, 2013). An especially important aspect of applying a biclustering approach to a one-mode matrix concerns the handling of the main diagonal. A two-mode matrix has no main diagonal, but a one-mode matrix clearly does. In some situations, having main diagonal elements is nonsensical, as in the case of social network ties (e.g., are people friends with themselves?). However, in brand-switching applications, the main diagonal can reflect the degree of retention of customers. Similarly, in confusion data, the main diagonal exemplifies correct responses to the presented stimulus.

Ideally, psychological researchers would have easy access to biclustering procedures that could be used for one-mode data and that would have the flexibility to include or exclude the main diagonal, depending on the goals of their studies. Unfortunately, such methods are generally unavailable in commercial software packages. Accordingly, in the spirit of other attempts to identify and make accessible the clustering models and methods that are important for psychological applications (Brusco & Steinley, 2006; Köhn, Steinley, & Brusco, 2010; Schepers & Hofmans, 2009; Schepers & Van Mechelen, 2011), we seek to achieve three interrelated objectives in this article: (i) to briefly describe several methods that can be used for biclustering one-mode matrices, (ii) to present a suite of MATLAB programs that implement these biclustering methods, and (iii) to provide two psychologically oriented examples to demonstrate and compare the procedures and programs. The next section presents three different useful methods for biclustering and describes MATLAB m-files for their implementation. These m-files are available in the Web supplement associated with this article. This is followed by a description of a general process for model selection with biclustering based on ideas gleaned from Schepers et al. (2008), Wilderjans, Ceulemans, and Meers (2013), and Brusco and Steinley (2014). Subsequent sections provide illustrations of the software programs for a real-valued asymmetric matrix and a binary asymmetric matrix, respectively. The article concludes with a brief summary.

Biclustering methods

Two-mode KL-means partitioning

Two-mode *KL*-means partitioning (TMKLMP) is a generalization of *K*-means partitioning. Whereas a *K*-means method (see Steinley, 2006, for a review) partitions a single set of objects into K clusters on the basis of minimization of the sum of squared errors, TMKLMP simultaneously establishes distinct partitions of the row and column objects using an analogous criterion. Although similar approaches have been discussed using different names (Brusco & Steinley, 2007; Hartigan, 1972; van Rosmalen et al., 2009), TMKLMP was recently adopted by Brusco and Doreian (2015a, b), such that the "KL" component of the name reflects the fact that the number of row clusters (K) need not be the same as the number of column clusters (L). The goal is to find row-object and column-object partitions that minimize the within-submatrix sum of squared deviations from the within-cluster means. Formally, we denote $P = \{S_1, \ldots, S_K\}$ as a K-cluster partition of the N objects associated with the rows of a data matrix, $\mathbf{X}$, where S_k is the set of objects assigned to cluster k for all $1 \le k \le K$. The standard conditions of a partition having clusters that are nonempty ($S_k \neq \varnothing$, for all $1 \le k \le K$), mutually exclusive ($S_k \cap S_l = \varnothing$ for all $1 \le k \neq l \le K$), and exhaustive ($S_1 \cup \ldots \cup S_k = S$) apply. Similarly, we define $Q = \{T_1, \ldots, T_L\}$ as an L-cluster partition of the N column objects of $\mathbf{X}$, where T_l is the set of objects assigned to cluster l for all $1 \le l \le L$. The standard conditions of a partition hold also for Q.

Denoting Π as the set of all partitions of N objects into K clusters and Ω as the set of all partitions of N objects into L clusters, the optimization problem associated with TMKLMP can be specified as follows:

$$\underset{P \in \Pi, Q \in \Omega}{\text{Minimize}} : f(P, Q) = \sum_{k=1}^{K} \sum_{l=1}^{L} \sum_{i \in S_k} \sum_{j \in T_l} \left(x_{ij} - \bar{x}_{kl}\right)^2, \qquad (1)$$

where

$$\bar{x}_{kl} = \frac{\sum_{i \in S_k} \sum_{j \in T_l} x_{ij}}{|S_k||T_l|}, \qquad (2)$$

and $|S_k|$ $(1 \le k \le K)$ and $|T_l|$ $(1 \le l \le L)$ are the numbers of row objects and column objects in clusters k and l, respectively. Together, the row objects (S_k) in cluster k and the column objects (T_l) in cluster l define a submatrix of $\mathbf{X}$. The value of $\bar{x}_{kl}$ is the mean of the elements in that submatrix, and the degree of homogeneity for the submatrix is a variance type with a measure defined by the sum of the squared deviations of each element from the submatrix mean, $(x_{ij} - \bar{x}_{kl})^2$. Perfect homogeneity of a submatrix is achieved when all elements in the submatrix are the same (e.g., all 0 s or all 1 s, in the case in which **X** is defined by the presence or absence of network ties). From an interpretive standpoint, the row objects (or column objects) in the same cluster can be perceived as being similar, in the sense that they tend to have comparable measures within the submatrices.

Brusco and Doreian (2015b) recently developed an exact algorithm for the TMKLMP that can be successfully applied to problems of size $N = 20$. A number of heuristic algorithms have been designed for TMKLMP, including simulated annealing (Trejos & Castillo, 2000), genetic algorithms (Brusco & Doreian, 2015a; Hansohm, 2002), and variable neighborhood search (Brusco & Steinley, 2007). However, a two-mode generalization of the *K*-means algorithm designed by Baier, Gaul, and Schader (1997) has been shown to be competitive with these more sophisticated procedures (van Rosmalen et al., 2009). Accordingly, we make available the MATLAB m-file tmklmp.m, which implements this procedure. The key inputs to tmklmp.m are **X**, *K*, and *L*. The default setting is for 500 restarts of the algorithm from different initial partitions, which is based on the implementations in previous studies (Brusco & Steinley, 2007; van Rosmalen et al., 2009). The principal outputs are the partitions of the row and column objects. Also, the raw objective function value corresponding to Eq. 1 is reported, as well as a normalized measure representing the total variance accounted for (*vaf*) by the partition, computed as follows:

$$vaf = 1 - \frac{f(P,Q)}{\sum_{i=1}^{N}\sum_{j=1}^{N}\left(x_{ij}-\bar{x}\right)^2}, \tag{3}$$

where

$$\bar{x} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{N} x_{ij}}{N^2} \tag{4}$$

The program tmklmp.m is applicable directly to either two-mode matrices or asymmetric one-mode matrices. However, when applied to the latter data structures, it is important to recognize the inclusion of the main diagonal elements of **X** in the computation of the submatrix means in Eq. 2, as well as in the sum-of-squares objective function in Eq. 1. As we noted previously, a compelling argument for ignoring the main diagonal elements can be made for many applications. Therefore, we also developed an alternative version, tmklmp_nodiag.m, which ignores the main diagonal elements when computing the submatrix means in Eq. 2 and the sum of squares in Eq. 1. The main diagonal is also ignored when computing the overall mean in Eq. 4 and the denominator term in Eq. 3. The inputs and outputs of this program are identical to those for tmklmp.m.

Nonnegative matrix factorization

The second approach for biclustering of one-mode asymmetric matrices has its roots in pioneering work on matrix decomposition (Eckart & Young, 1936; Young & Householder, 1938). More specifically, low-dimensional least-squares approximations of a two-mode matrix can be obtained using singular value decomposition (SVD) or principal component analysis (PCA). Given that the one-mode asymmetric matrix can be treated as two-mode data, these methods are applicable here, as well. However, two limitations are readily apparent: (i) the potential requirement of ignoring the main diagonal and (ii) the possibility of negative elements in the factors, which can hinder interpretability (Lee & Seung, 1999).

As with the SVD and PCA approaches, nonnegative matrix factorization (NMF) seeks a minimum-sum-of-squares, low-dimensionality approximation of **X**. However, in contrast to the eigenvectors obtained by SVD and PCA, which can assume both positive and negative values, these components are required to be strictly nonnegative in NMF. The rationale for this constraint, as described by Lee and Seung (1999), is to preserve the condition of having components be the sum of their parts. The arbitrary sign of the elements of a PCA factorization presents the following difficulty, noted by Lee and Seung (1999, p. 789): "As the eigenfaces are used in linear combinations that generally involve complex cancellations between positive and negative numbers, many individual eigenfaces lack intuitive meaning." Expanding on this point, Fogel, Hawkins, Beecher, Luta, and Young (2013, p. 207) found that NMF "often leads to substantial improvements in the interpretability of the factors," relative to attempts to transform PCA scores of mixed signs into meaningful factors.

To formalize NMF, we denote *D* as the desired dimensionality of the factorization. The problem requires the determination of two nonnegative matrices, **G** and **H**, the product of which will generate a *D*-dimensional least-squares approximation of **X**. Matrix $\mathbf{G} = [g_{id}]$ is an $N \times D$ matrix of nonnegative coefficients for the row objects ($1 \leq i \leq N$) for each of the *D* dimensions ($1 \leq d \leq D$). Similarly, $\mathbf{H} = [h_{dj}]$ is a $D \times N$ matrix of nonnegative coefficients for the column objects ($1 \leq j \leq N$) for each of the *D* dimensions ($1 \leq d \leq D$). The optimization problem for NMF is

$$\text{Minimize}: Z = \|\mathbf{X}-\mathbf{GH}\|^2 = \sum_{i=1}^{N}\sum_{j=1}^{N}\left(x_{ij}-\sum_{d=1}^{D} g_{id}h_{dj}\right)^2, \tag{5}$$

$$\text{Subject to}: \quad g_{id} \geq 0, \quad \text{for } 1 \leq I \leq N \text{ and } 1 \leq d \leq D, \tag{6}$$

$$h_{dj} \geq 0, \quad \text{for } 1 \leq d \leq D \text{ and } 1 \leq j \leq N. \tag{7}$$

Essentially, the optimization problem is to obtain **G** and **H** while minimizing the least-squares loss function in Eq. 5, subject to the nonnegativity constraints in Eqs. 6 and 7. Lee and Seung (2001) described a rescaled gradient descent algorithm for solving the optimization problem stated in Eqs. 5–7 (see also Brusco, 2011, for a description and extension to the case of an asymmetric one-mode matrix with an arbitrary main diagonal). Once the optimization problem has been solved by obtaining **G** and **H**, there are several possibilities for constructing partitions of the row and column objects. Perhaps the simplest approach is to assign each row object i to the cluster d for which g_{id} is maximum, and each column object j to the cluster d for which h_{dj} is maximum. Of course, this approach imposes the restriction that $K = L = D$. An alternative approach, which is employed herein, is to apply K-means partitioning (with, say, 5,000 restarts) to the **G** and **H** matrices in order to obtain K-cluster and L-cluster partitions of the row and column objects, respectively.

We developed a MATLAB m-file implementation of the NMF algorithm, nmf.m. The nmf.m program is similar to the module that is available in the MATLAB system, yet it is differentiated by distinct parameter settings for the descent algorithm and, more importantly, by the inclusion of a K-means clustering subroutine to cluster the rows and columns on the basis of **G** and **H**, respectively. The inputs to the program are **X**, K, L, and D. The outputs of the nmf.m program are the percentage reduction of error (*PRE*) provided by the D-dimensional factorization, the partition of the row objects, and the partition of the column objects. As was the case for TMKLMP, a second version of the NMF program, nmf_nodiag.m, was prepared that ignores the main diagonal elements in the estimation process. The inputs and outputs of this program are identical to those of nmf.m.

Two-mode blockmodeling

Although the previously described methods can be applied to binary data matrices (i.e., $x_{ij} \in \{0, 1\}$ for all i and j), some analysts might prefer a method that is explicitly designed for such data. Two-mode blockmodeling methods are especially well-suited for binary data (Brusco & Steinley, 2007, 2011; Brusco, Doreian, Lloyd, & Steinley, 2013; Brusco, Doreian, Mrvar, & Steinley, 2013; Doreian et al., 2004, 2005). These methods seek partitions of the row and column objects such that the ideal submatrices formed by the row and column clusters are either *complete* (having all 1 s) or *null* (having all 0 s) to the greatest extent possible. Accordingly, the typical objective is to obtain partitions minimizing the number of inconsistencies with this ideal structure, as measured by a count of the total number of violations. More formally, the objective of the two-mode blockmodeling problem (TMBP) that we consider is to minimize the total of the sum of 1 s in submatrices that are mostly 0 s plus the sum of 0 s in submatrices that are mostly 1 s. The optimization problem is:

$$\underset{(P \in \Pi, Q \in \Omega)}{\text{Minimize}} : g(P, Q) = \sum_{k=1}^{K} \sum_{l=1}^{L} \min\{\eta_{kl}, \rho_{kl}\}, \tag{8}$$

where

$$\eta_{kl} = \sum_{i \in S_k} \sum_{j \in T_l} x_{ij}, \quad \forall 1 \le k \le K \, and \, 1 \le l \le L, \tag{9}$$

and

$$\rho_{kl} = \sum_{i \in S_k} \sum_{j \in T_l} \left(1 - x_{ij}\right) \quad \forall 1 \le k \le K \, and \, 1 \le l \le L. \tag{10}$$

Brusco, Doreian, Lloyd, and Steinley (2013) designed an exact algorithm for TMBP that is scalable for applications in which $N \le 20$. Heuristic methods for TMBP include tabu search (Brusco & Steinley, 2011) and variable neighborhood search (Brusco, Doreian, Mrvar, & Steinley, 2013). However, recent computational results reported by Brusco, Doreian, Mrvar, and Steinley have shown that a two-mode relocation heuristic designed by Doreian et al. (2004, 2005) often produces solutions that are as good as those obtained by these more sophisticated procedures. For this reason, we prepared a MATLAB m-file, tmbp.m, implementing the relocation heuristic procedure. The key inputs to tmbp.m are **X**, K, and L. The default setting is for 5,000 restarts of the algorithm from different initial partitions. The principal outputs are the objective function value (Eq. 8) and the partitions of the row and column objects.

Model selection

All three of the methods described in the previous section require the selection of a model. In the case of TMKLMP and TMBP, model selection is generally limited to the selection of K and L. We recommend running these algorithms for all possible different combinations of K and L obtained from the intervals $K_1 \le K \le K_2$ and $L_1 \le L \le L_2$. The total number of clusters for any given combination is $\xi = K + L$, which represents the level of complexity of the model. To measure the improvement in the objective function with respect to an increase in complexity, we employ the convex hull (*CHull*) approach, which has been widely adopted in the context of multimode clustering (Ceulemans & Van Mechelen, 2005; Schepers et al., 2008; Schepers & Van Mechelen, 2011; Wilderjans, Ceulemans, & Meers, 2013).

To illustrate the *CHull* approach for TMKLMP, we consider the *vaf* as the objective criterion of interest. The process begins with a deviance plot of the *vaf* values obtained from all combinations of K and L. The vertical axis of this plot is *vaf*, and the horizontal axis is the total number of clusters ($\xi = K + L$). The second step retains only those solutions falling on the upper boundary of the convex hull and places them in an ordered list based on complexity. We use B to denote the total number of solutions falling on the boundary, *vaf*(b), as the *vaf* for the solution b ($1 \leq b \leq B$), and $\xi(b)$ to denote the complexity of solution b ($1 \leq b \leq B$). The upper boundary is used because the goal is to maximize *vaf*. For a minimization objective function, the lower boundary would be used. The third step is to select one of the B solutions on the basis of visual inspection of the deviance plot and/or the use of measures based on the slopes of segments of the convex hull. A visually based selection of a solution from the boundary is made via a search for an "elbow" in the plot, in a manner similar to the use of a scree plot in factor analysis. Ceulemans and Van Mechelen (2005) have provided two slope-based measures to augment visual inspections. The difference measure, *Diff*CH, is used to choose the solution b that maximizes

$$\frac{|vaf(b)-vaf(b-1)|}{|\xi(b)-\xi(b-1)|}-\frac{|vaf(b+1)-vaf(b)|}{|\xi(b+1)-\xi(b)|}. \tag{11}$$

Similarly, the ratio measure, *Ratio*CH, leads to a choice of the solution b maximizing

$$\frac{[|vaf(b)-vaf(b-1)|/|\xi(b)-\xi(b-1)|]}{|vaf(b+1)-vaf(b)|/|\xi(b+1)-\xi(b)|}. \tag{12}$$

Although the absolute value signs in Eqs. 11 and 12 are not required in the case of a maximization objective such as *vaf*, they are included to avoid any sign confusion that might arise in a minimization context, such as $g(P, Q)$ in Eq. 8. Ceulemans and Van Mechelen (2005) found that *Ratio*CH outperformed *Diff*CH in their simulation study. Nevertheless, some caution regarding the use of *Ratio*CH is advisable, because it can be extremely sensitive to *very small changes* in the criterion function. To illustrate this, suppose that the *vaf*(b) values for four possible solutions $B - 3$, $B - 2$, $B - 1$, and B are .6, .9, .92, and .921, respectively. Assuming that $\xi(b) = b$ for all $1 \leq b \leq B$, solution $B - 2$ has *Diff*CH = (.9 – .6) – (.92 – .9) = .28. In contrast, solution $B - 1$ has *Diff*CH = (.92 – .9) – (.921 – .92) = .019. Solution $B - 2$ would be preferred to $B - 1$ on the basis of *Diff*CH. However, $B - 1$ is preferred to $B - 2$ on the basis of *Ratio*CH, because (.92 – .9)/(.921 – .92) = 20 exceeds (.9 – .6)/(.92 – .9) = 15. Arguably, the $B - 2$ solution is the proper choice. Yet the very small change when moving from $B - 1$ to B makes $B - 1$ preferred according to the *Ratio*CH measure. For this reason, we recommend consideration of both measures, along with visual inspection to select a model. Examples of doing this are provided in our empirical examples.

In addition to K and L, NMF also requires the selection of the dimensionality of the factorization. We recommend decoupling the selection of D from the selection of K and L. More specifically, a choice for D is made first on the basis of the analysis of *PRE* at different values of D. We recommend choosing D by using visual inspection of the plot of *PRE* measures at different values of D together with the *Diff*CH and *Ratio*CH measures. Once D is selected, the numbers of clusters (K and L) for NMF can be chosen in a variety of ways. One approach would be to obtain a set of K-means clustering solutions for $\mathbf{G}$ using different Ks and to select K on the basis of any of the host of different indices (see, e.g., Steinley & Brusco, 2011). This process would then be repeated for $\mathbf{H}$ to choose the best value for L. An alternative approach, adopted here, is to select K and L on the basis of the results obtained using TMKLMP (TMBP could also be used). Although this approach does make NMF dependent on TMKLMP, it has the advantage of enabling the joint (or simultaneous) selection of K and L.

Example 1

Lipread consonant data

The first example comes from a study of confusion among $N = 21$ lipread consonants (Manning & Shofner, 1991, p. 596). The elements (x_{ij}) of the data matrix correspond to the proportions of responses given as letter j when letter i was the presented stimulus. All elements of the lipread consonant data are nonnegative real-valued numbers. Accordingly, two-mode *KL*-means clustering and nonnegative matrix factorization can be applied for analyzing these data. Since their original publication, these lipread consonant data have been reanalyzed using clustering or seriation methods (Brusco & Stahl, 2005b; Brusco & Steinley, 2010). However, the latter studies analyzed the data subsequent to transforming the asymmetric confusion proportions to a symmetric matrix. In contrast, our analysis herein preserves the asymmetry in these data.

The main diagonal of the confusion matrix represents the proportions of correct responses for the corresponding stimulus objects. For most consonants the proportions are, by far, the largest numbers in the rows. At the extreme, the letters {y} and {w} have main diagonal entries of .974 and .968, respectively. However, a few of the less frequently used letters have high confusability with more popular letters, and accordingly, their main diagonal entries are much smaller. Examples include the letters {x} and {z}, which have main diagonal entries of .091 and .151, respectively. Given that the focus of this type of study is usually on the patterns of confusion, as well as on the artificial inflation in variation that is likely to be incurred from large diagonal elements, we emphasize that we used tmklmp_nodiag.m and nmf_nodiag.m for our analyses. However, our presentation of the results will culminate with a brief discussion of the effects of including the main diagonal in the analysis, via the use of tmklmp.m.

Results

We began our analysis by applying tmklmp_nodiag.m to the confusion matrix for all combinations of $2 \leq K \leq 9$ and $2 \leq L \leq 9$. The *vaf* for each combination is displayed in Table 1, and the deviance plot of the *vaf* values is provided in Fig. 1. An inspection of the upper boundary of the convex hull reveals three levels of complexity ($\xi = K + L = 5$, $\xi = K + L = 7$, and $\xi = K + L = 9$) as not producing a solution on the boundary. Moreover, visual inspection of the plot shows the sharpest elbow occurring at $\xi = K + L = 8$. The selection of $\xi = 8$ is supported strongly also by its *Diff*CH measure of .0743, which is far larger than the corresponding measures for the other values of ξ. The $\xi = 8$ solution on the upper boundary has the second largest *Ratio*CH measure (2.65) and is appreciably larger than the measures at most other values of ξ. The lone exception is $\xi = 16$ (*Ratio*CH = 4.29). However, its large measure arises because of the issue noted in the Model Selection section. In this instance, the very small increase in *vaf* when moving from $\xi = 16$ to $\xi = 17$ inflates the *Ratio*CH measure for $\xi = 16$. Considering this caveat with respect to the *Ratio*CH measure, along with the visual inspection of Fig. 1 and the *Diff*CH measure, $\xi = 8$ is a more appropriate level of model complexity. The particular solution on the upper boundary at $\xi = 8$ corresponds to $K = L = 4$, and this solution was selected for interpretation. The computation time for tmklmp_nodiag.m for $K = L = 4$ was approximately 3.3 s on a 2.2-GHz Pentium 4 PC, which is a circa 2000–2002 hardware platform. The best-found *vaf* was identified on 97 % of the 500 restarts of the algorithm, which is a high attraction rate suggesting (but not guaranteeing) that the global optimum has been found.

Table 1 Two-mode *KL*-means partitioning results for the Manning and Shofner (1991, p. 596) lipread consonant confusion data: Variance accounted for at different combinations of *K* and *L*

	$L = 2$	$L = 3$	$L = 4$	$L = 5$	$L = 6$	$L = 7$	$L = 8$	$L = 9$
$K = 2$	.2797	.3124	.3445	.3530	.3547	.3593	.3601	.3608
$K = 3$	.3206	.5401	.5722	.6028	.6113	.6157	.6189	.6207
$K = 4$	.3275	.5778	.7787	.8108	.8417	.8557	.8640	.8685
$K = 5$	.3326	.5854	.8002	.8687	.9008	.9234	.9320	.9381
$K = 6$	.3345	.5920	.8069	.8753	.9069	.9390	.9478	.9543
$K = 7$	.3363	.5937	.8108	.8791	.9136	.9447	.9521	.9603
$K = 8$	.3372	.5956	.8122	.8799	.9148	.9467	.9539	.9617
$K = 9$	.3378	.5967	.8136	.8808	.9187	.9480	.9558	.9627

Next, we applied nmf_nodiag.m to the lipread consonant confusion matrix using $1 \leq D \leq 6$. A plot of the *PRE* values is displayed in Fig. 2. Visual inspection of the plot reveals a sharp elbow at $D = 3$, with a lesser elbow at $D = 4$. On the basis of the *Diff*CH measure, $D = 3$ (*Diff*CH = 0.17) would be preferred to $D = 4$ (*Diff*CH = 0.08). However, on the basis of the *Ratio*CH measure, $D = 4$ (*Ratio*CH = 4.83) is preferred to $D = 3$ (*Ratio*CH = 2.70). Once again, we recommend caution when using the *Ratio*CH measure because of its sensitivity to very small changes on the right. In this instance, the *PRE* = 95.63 % at $D = 4$, so there is very little room for further improvement by increasing the dimensionality to $D = 5$. The move to $D = 5$ only improves the *PRE* to 97.77 %. This small improvement drives the inflated *Ratio*CH measure for $D = 4$. Considering the visual inspection of Fig. 2, together with the slope measures overall, $D = 3$ is more reasonable for the dimensionality. We applied *K*-means clustering, independently, to the **G** (using $K = 4$) and **H** (using $L = 4$) matrices obtained by the factorization using $D = 3$. The selection of $K = L = 4$ was made for these *K*-means analyses, to facilitate a comparison with the TMKLMP results. The total computation time for nmf_nodiag.m using the settings of $D = 3$, $K = L = 4$, and 500 restarts of the *K*-means algorithm to obtain the row and column partitions was 21.9 s on the 2.2-GHz Pentium 4 PC.

The $K = L = 4$ partitions obtained using tmklmp_nodiag.m and nmf_nodiag.m ($D = 3$) were identical. This solution is displayed in Fig. 3. Two observations regarding the solution are immediately observable: (i) There is considerable symmetry in the partitions—that is, the partition of the consonants as stimuli is strikingly similar to the partition of the same consonants as responses—and (ii) the partitions each consist of three small clusters and one large cluster. The consonants {b, p} form both a row (stimulus) cluster and column (response) cluster. These two letters were highly confused with one another but were not confused strongly with any of the other consonants. The consonant "p" was frequently (.440) a mistaken response for the stimulus "b," and similarly, "b" was frequently (.424) a mistaken response for the stimulus "p." The cluster {c, d, t, z} also emerges in both the row and column partitions, because of the generally modest levels of (symmetric) confusion among these four letters. Unlike "b" and "p," which exhibited high symmetry in their confusion, the consonants "s" and "x" show strong asymmetry in confusion. The consonant "s" was frequently (.582) a mistaken response for stimulus "x," but "x" was seldom (.074) a mistaken response for stimulus "s." For this reason, {s, x} emerges as a cluster in the row (stimulus) objects, and {s} is a singleton cluster in the column (response) objects.

One final analysis of the lipread consonant data was conducted by applying tmklmp.m to the confusion matrix. This program includes the main diagonal elements (i.e., the proportions of correct responses for each stimulus letter) in the computation of the submatrix means and the sum of squared

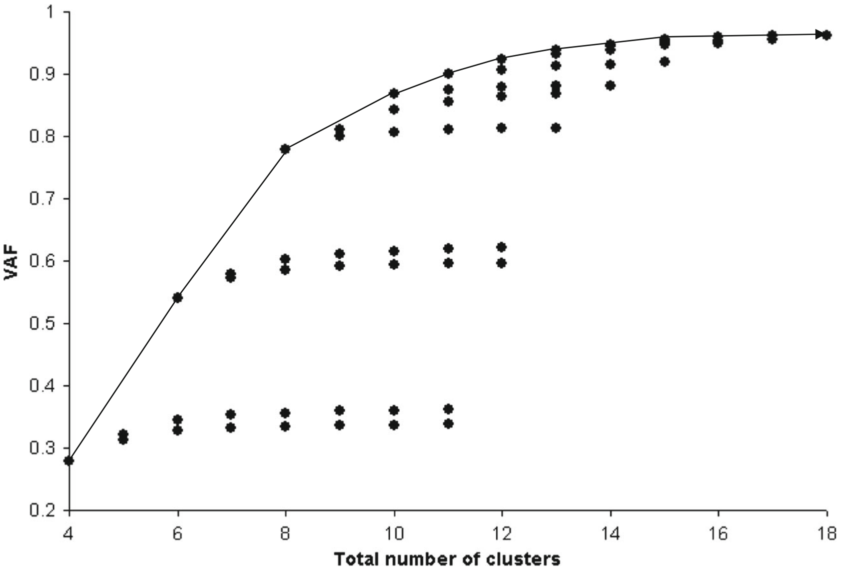

Fig. 1 Convex hull for the two-mode *KL*-means partitioning (TMKLMP) application to the lipread consonant data

deviations. For comparative purposes, we selected $K = 4$ and $L = 4$. The tmklmp.m algorithm produced a four-cluster partition of the stimulus (row) letters, whereby the three letters {f}, {w}, and {y} were each a singleton cluster and all other letters were placed in the fourth cluster. The partition of response (column) letters was the same as the row partition. The explanation for the extraction of the singleton clusters is that those three letters have the largest main diagonal elements and, therefore, their isolation reduces the total variation. Clearly, the tmklmp.m solution is far less useful and interesting than the tmklmp_nodiag.m solution. This highlights the importance of having available software that permits the exclusion of the main diagonal.

Example 2

Friendship ties among third-grade students

The second asymmetric one-mode matrix comes from a study of friendship ties and peer group acceptance among elementary schoolchildren (Parker & Asher, 1993). The particular classroom used in this example corresponded to 22 third-grade students (14 boys and eight girls). The data were collected using the roster method: The schoolchildren were provided with a list of their classmates and asked to identify their "very best friend," three best friends, and as many other friends as they liked. Anderson et al. (1999) analyzed the

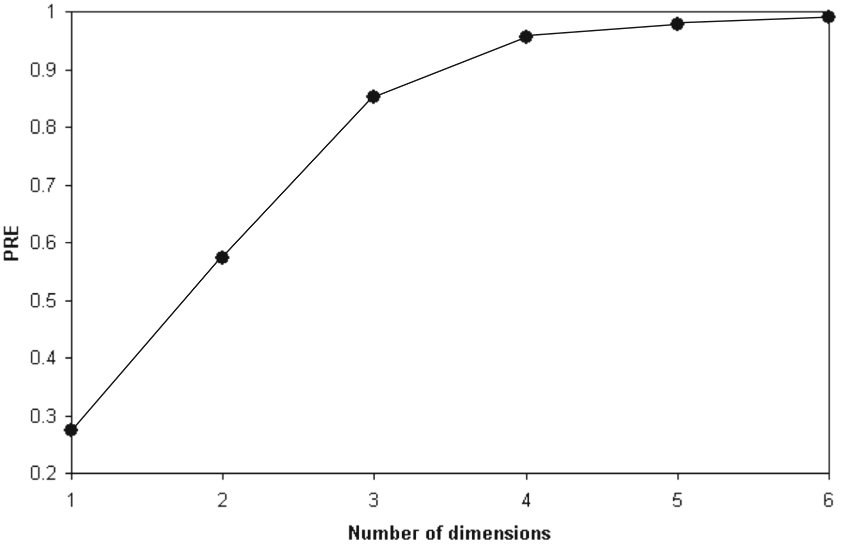

Fig. 2 Convex hull for the nonnegative matrix factorization (NMF) application to the lipread consonant data. The plot represents the proportions of reductions in error (*PRE*) as a function of the number of dimensions in the factorization (*D*)

	B	P	C	D	T	Z	S	F	G	H	J	K	L	M	N	Q	R	V	W	X	Y
B	**.473**	.440	.001	.003	.002	.002	.001	.008	.001	.001	.001	.000	.010	.029	.003	.001	.002	.007	.002	.001	.000
P	.424	**.425**	.002	.006	.005	.001	.003	.009	.005	.002	.000	.002	.005	.074	.015	.000	.000	.014	.002	.000	.000
C	.006	.010	**.256**	.286	.180	.143	.007	.014	.020	.001	.002	.002	.014	.022	.007	.000	.001	.002	.005	.000	.000
D	.002	.007	.127	**.381**	.204	.099	.003	.014	.071	.003	.003	.000	.011	.025	.006	.003	.002	.012	.002	.000	.000
T	.002	.005	.141	.325	**.183**	.094	.009	.004	.100	.003	.012	.013	.012	.031	.019	.005	.002	.013	.001	.002	.000
Z	.011	.009	.245	.275	.171	**.151**	.005	.007	.024	.000	.005	.001	.014	.015	.010	.002	.004	.014	.002	.000	.000
S	.000	.000	.003	.000	.001	.000	**.655**	.007	.001	.135	.000	.012	.018	.006	.075	.000	.004	.000	.001	.074	.004
X	.000	.001	.002	.002	.001	.000	.582	.005	.003	.117	.007	.005	.013	.013	.127	.000	.001	.003	.002	**.091**	.003
F	.000	.000	.002	.001	.002	.000	.000	**.946**	.002	.000	.000	.000	.008	.009	.003	.002	.000	.006	.003	.000	.000
G	.000	.001	.001	.005	.006	.004	.001	.003	**.905**	.025	.020	.000	.005	.006	.003	.005	.002	.000	.000	.000	.000
H	.002	.000	.000	.001	.002	.000	.024	.003	.010	**.904**	.050	.000	.006	.007	.004	.002	.000	.001	.050	.010	.000
J	.000	.000	.000	.004	.003	.000	.001	.019	.042	.007	**.879**	.000	.008	.020	.000	.001	.004	.000	.002	.000	.001
K	.002	.003	.001	.005	.006	.004	.007	.046	.002	.012	.007	**.370**	.172	.244	.070	.000	.007	.004	.002	.004	.007
L	.002	.002	.003	.002	.000	.001	.034	.023	.001	.030	.007	.018	**.578**	.069	.187	.000	.011	.008	.002	.009	.001
M	.003	.005	.000	.002	.005	.000	.002	.010	.001	.001	.003	.000	.011	**.910**	.016	.001	.001	.006	.009	.001	.000
N	.004	.006	.011	.012	.004	.002	.021	.032	.015	.108	.016	.046	.134	.081	**.421**	.001	.012	.017	.000	.017	.000
Q	.003	.007	.007	.004	.004	.000	.001	.003	.013	.001	.008	.004	.022	.009	.014	**.676**	.025	.029	.034	.000	.001
R	.003	.005	.001	.001	.001	.000	.000	.005	.002	.006	.000	.000	.019	.004	.006	.025	**.884**	.000	.000	.003	.001
V	.007	.005	.002	.004	.002	.004	.001	.044	.001	.000	.003	.000	.006	.004	.000	.000	.003	**.910**	.001	.001	.001
W	.000	.001	.001	.001	.002	.001	.000	.000	.001	.001	.000	.000	.005	.000	.003	.001	.002	.000	**.968**	.000	.000
Y	.000	.001	.000	.000	.000	.000	.000	.001	.000	.000	.002	.000	.004	.002	.003	.000	.000	.001	.001	.000	**.974**

Fig. 3 The ($K = 4$, $L = 4$) bipartition for the lipread consonant confusion study by Manning and Shofner (1991, p. 596), which was obtained using both two-mode *KL*-means partitioning and nonnegative matrix factorization ($D = 3$). The values in bold are the main diagonal elements, which are ignored in the computation of the variance accounted for (*vaf*)

resulting data using p* models. For simplicity, they ignored the information regarding the degree or strength of the friendship ties. Our analysis herein focuses on the binary matrix corresponding to the friendship ties as published in Anderson et al. (1999, p. 42), where the rows and columns of the data matrix correspond to the senders and receivers of friendship ties, respectively. A value of x_{ij} = 1 indicates that student i identified student j as one of his or her friends, whereas x_{ij} = 0 indicates the lack of a friendship tie. The main diagonal of the friendship matrix is arbitrary: Students did not identify themselves as friends. Accordingly, methods ignoring the main diagonal are appropriate.

Results

Given the binary nature of the friendship ties matrix, all three methods (TMKLMP, TMBP, and NMF) can be applied. These data were analyzed using tmklmp_nodiag.m, tmbp_nodiag.m, and nmf_nodiag.m. The tmklmp_nodiag.m program was implemented for all combinations of $2 \leq K \leq 5$ and $2 \leq L \leq 5$. The *vaf* results are reported in Table 2, and the deviance plot is displayed in Fig. 4. Visual inspection of the deviance plot reveals an elbow at $\xi = 7$, which is supported by having the maximum values of both *Diff*CH (0.0219) and *Ratio*CH (1.81) achieved at $\xi = 7$. The solution on the upper boundary of the convex hull for $\xi = 7$ corresponds to having $K = 4$ and $L = 3$. The computation time for tmklmp_nodiag.m using $K = 4$ and $L = 3$ was approximately 3.9 s on the 2.2-GHz Pentium 4 PC. Once again, the attraction rate was high, such that the the best-found *vaf* was identified on 97 % of the 500 restarts of the algorithm.

Next, we applied the tmbp_nodiag.m program for all combinations of $2 \leq K \leq 5$ and $2 \leq L \leq 5$. The $g(P, Q)$ results and the numbers of equally well-fitting partitions (shown in parentheses) are reported in Table 3, and the deviance plot is displayed in Fig. 5. Unlike the *vaf* measure associated with TMKLMP, which we seek to maximize, the goal is to minimize g(P, Q). Therefore, when examining the deviance plot, the *lower* boundary of the convex hull is the measure of interest. Visual inspection of the deviance plot in Fig. 5 reveals elbows at both $\xi = 6$ and $\xi = 7$. There were two equally well-fitting partitions for both the $\xi = 6$ and 7 levels of model complexity, and both levels also produced a *Diff*CH measure

Table 2 Two-mode *KL*-means partitioning results for the Parker and Asher (1993) third-grade classroom friendship data: Variance accounted for at different combinations of *K* and *L*

	$L = 2$	$L = 3$	$L = 4$	$L = 5$
$K = 2$	.2684	.3054	.3213	.3300
$K = 3$	.3527	.4194	.4426	.4535
$K = 4$	.3899	.4681	.4933	.5060
$K = 5$	.4119	.4908	.5218	.5487

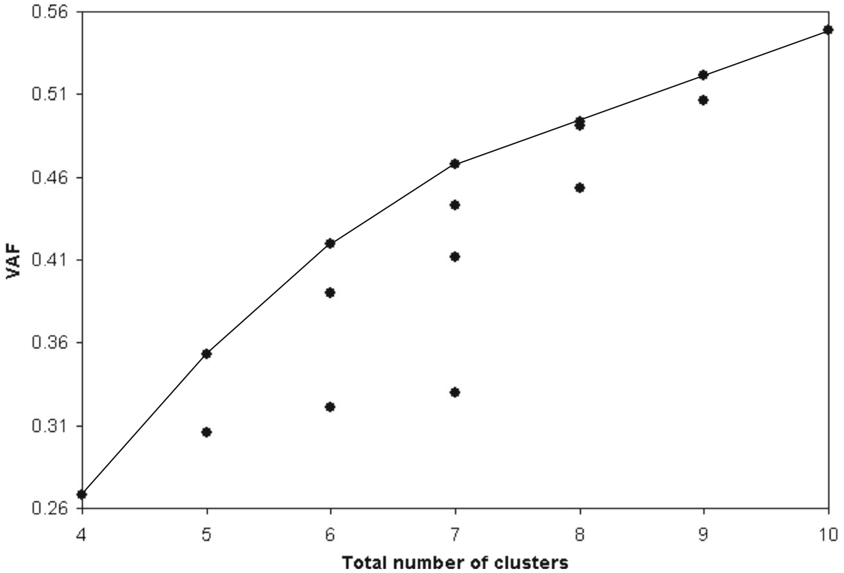

Fig. 4 Convex hull for the TMKLMP application to the third-grade friendship data

of 2.0. Although the $\xi = 6$ solution is more parsimonious, given the competitiveness of $\xi = 6$ and $\xi = 7$, we opted for the $\xi = 7$ solution, to provide consistency with the selected model for TMKLMP. The solution on the lower boundary of the convex hull for $\xi = 7$ corresponds to $K = 4$ and $L = 3$. The computation time for tmbp_nodiag.m using $K = 4$ and $L = 3$ was approximately 2 min on the 2.2-GHz Pentium 4 PC. The best-found *vaf* was identified on only 0.1 % of the 5,000 restarts of the algorithm, which is an appreciably lower attraction rate than what was achieved by tmklmp_nodiag.m.

Finally, we applied nmf_nodiag.m using $1 \leq D \leq 5$. A plot of the *PRE* values is displayed in Fig. 6. Visual inspection of the plot reveals a clear and sharp elbow at $D = 2$—indeed, the only elbow. Moreover, $D = 2$ is preferred to all other dimensionalities for the factorization based on both the *Diff*CH (*Diff*CH = 0.1458) and *Ratio*CH (*Ratio*CH = 2.94) measures. Considering the visual inspection of Fig. 6 and the slope measures, we selected $D = 2$ as the dimensionality and obtained a ($K = 4$, $L = 3$) partition for comparison with the TMKLMP and TMBP results. As we noted previously, the selection of K and L for NMF was based on the TMKLMP solution. The computation time for nmf_nodiag.m using $D = 2$, $K = 4$, and $L = 3$ was approximately 17.8 s on the 2.2-GHz Pentium 4 PC.

The partition obtained by tmklmp_nodiag.m is displayed in Fig. 7. This partition is identical to one of the two equally well-fitting partitions produced by tmbp_nodiag.m (the other equally well-fitting partition differed only by the relocation of one student in the partition of columns). The eight girls form one of the three clusters for the column objects (friendship tie receivers), and the 14 boys are split into two clusters of approximately equal size. The row clusters (friendship tie senders) are a bit more complex: There is one cluster of seven boys, one cluster of four girls, one singleton cluster consisting of one girl, and one cluster consisting of seven boys and three girls.

The friendship tie-sending cluster of four girls exhibits a strong linkage to the tie-receiving cluster consisting of all eight girls, because the resulting submatrix contains mostly 1 s. However, the same cluster of girls exhibits no friendship ties to any of the 14 boys, as is shown by the two null submatrices. Similarly, the friendship tie-sending cluster consisting only of boys (b2, b5, b11, b8, b20, b16, and b19) identifies most of the other boys in the class but very few of the girls. However, the boys in this tie-sending cluster are more strongly linked to the first cluster of tie-receiving boys (b2, b5, b11, b8, b13, and b21) than to the second. The largest friendship tie-sending cluster, consisting of boys and girls (b1, b3, b4, b7, b10, b13, b21, g14, g6, and g22) also has strong linkages to the first cluster of tie-receiving boys (b2, b5, b11, b8, b13, and b21), but not to the other two tie-receiving clusters. Accordingly, the first cluster of tie-receiving boys might be characterized as the "popular boys," since they are much

Table 3 Two-mode blockmodeling (TMBP) results for the Parker and Asher (1993) third-grade classroom friendship data: Numbers of inconsistencies relative to the ideal block structure for different combinations of K and L

	$L = 2$	$L = 3$	$L = 4$	$L = 5$
$K = 2$	104 (858)	103 (4)	103 (113)	103 (238)
$K = 3$	95 (1)	83 (2)	80 (4)	80 (271)
$K = 4$	89 (1)	77 (2)	74 (9)	72 (52)
$K = 5$	89 (212)	74 (107)	69 (6)	66 (38)

The cells of the table contain the values of the TMBP objective function, $g(P, Q)$, and, in parentheses, the numbers of equally well-fitting partitions that produce that objective value

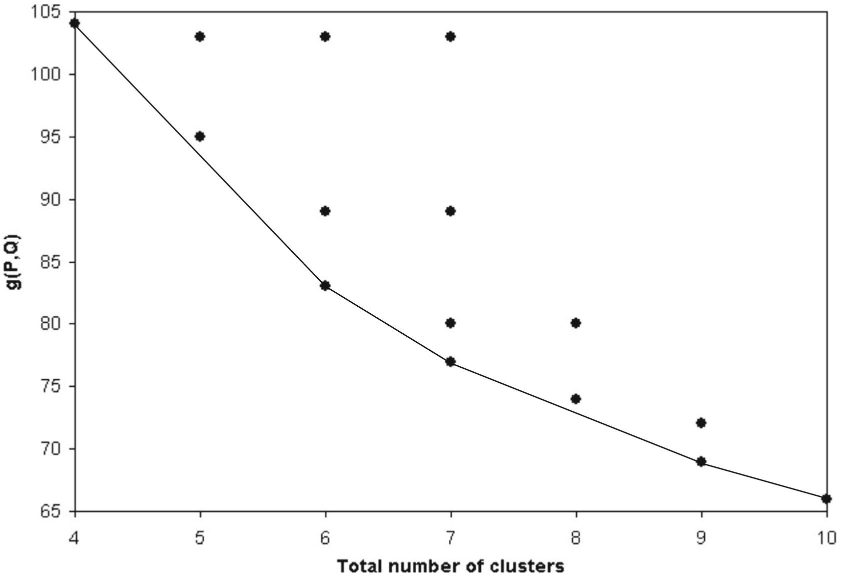

Fig. 5 Convex hull for the two-mode blockmodeling (TMBP) application to the third-grade friendship data

more heavily linked to the two largest clusters of tie senders than to the other cluster of tie-receiving boys. Finally, we note that g12 emerges as a singleton cluster of tie senders, because she is unique in her identification of almost everyone in the class as friends.

Although interpretable, the partition obtained from using nmf_nodiag.m, shown in Fig. 8, exhibited some marked differences from the TMKLMP and TMBP partition in Fig. 7. The tie-receiving cluster of girls remains intact, as does the tie-sending cluster of four girls. However, the sending and receiving clusters of boys are carved up somewhat differently. Moreover, g12 is folded into a tie-sending cluster with two other girls, despite being appreciably different from those two girls with respect to the pattern of ties. The net result is that the submatrices associated with the NMF solution in Fig. 8 are appreciably less homogeneous than those in the TMKLMP/TMBP solution in Fig. 7: Whereas the number of inconsistencies [$g(P, Q)$] in Fig. 7 is 77, there are 104 in Fig. 8. This substantial difference in submatrix homogeneity raises concerns regarding the effectiveness of NMF in this application.

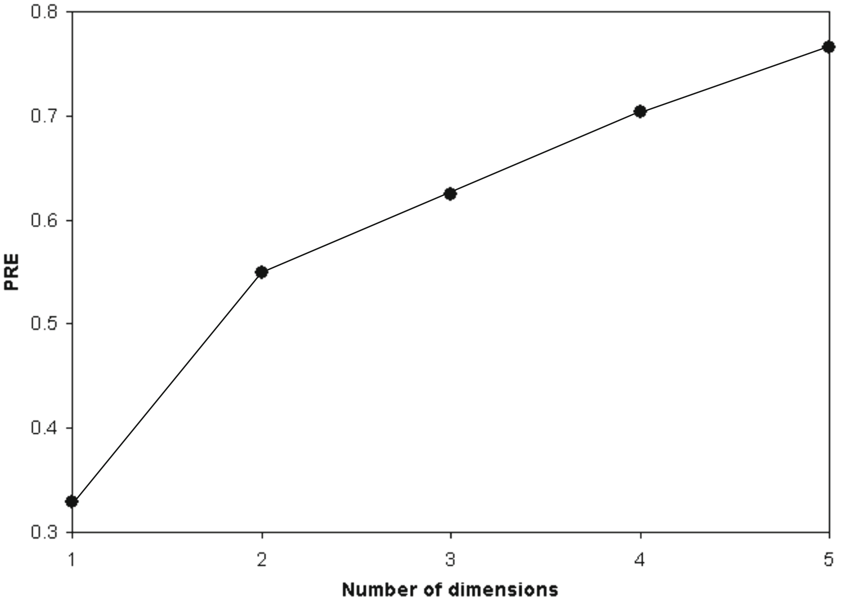

Fig. 6 Convex hull for the NMF application to the third-grade friendship data. The plot represents the proportions of reductions in error (*PRE*) as a function of the number of dimensions in the factorization (*D*)

		b	b	b	b	b	b	b	b	b	b	b	b	b	b	g	g	g	g	g	g	g	g
		2	5	11	8	13	21	1	3	4	7	20	10	16	19	15	12	17	18	14	9	6	22
b	2	xx	1	1	0	1	1	1	1	0	1	1	0	1	1	0	0	1	0	0	0	0	1
b	5	1	xx	1	1	1	1	1	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0
b	11	1	1	xx	1	1	1	0	1	1	1	0	1	0	1	0	0	0	0	0	0	0	0
b	8	1	1	1	xx	1	1	1	1	1	0	0	1	1	1	0	0	0	0	1	0	0	0
b	20	1	1	1	1	1	1	1	1	0	0	xx	1	1	1	0	0	0	0	0	0	1	0
b	16	1	1	0	1	1	1	1	0	0	0	1	1	xx	1	0	0	0	0	0	0	0	0
b	19	1	1	1	1	1	1	1	1	1	0	1	1	1	xx	0	0	0	0	1	0	0	0
g	12	1	1	1	1	0	1	1	1	1	0	0	1	0	1	1	xx	1	1	1	1	1	1
b	1	1	1	0	0	0	0	xx	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
b	3	1	1	0	0	0	1	0	xx	0	0	0	0	0	0	0	0	0	0	0	0	0	0
b	4	0	0	1	1	1	1	0	0	xx	0	0	0	0	1	0	0	0	0	0	0	0	0
b	7	0	1	1	1	1	1	0	1	0	xx	1	0	0	0	0	0	1	1	0	0	0	0
b	10	1	1	1	0	0	1	0	0	0	0	0	xx	0	1	0	0	0	0	1	0	0	0
b	13	1	0	0	1	xx	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0
b	21	1	1	1	1	1	xx	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
g	14	0	1	1	0	1	1	1	0	0	0	0	0	0	1	0	0	1	1	xx	0	1	0
g	6	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	xx	0
g	22	1	1	1	0	0	1	0	0	0	0	0	1	0	0	1	1	1	0	0	1	0	xx
g	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	xx	1	1	1	1	1	1	1
g	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	xx	1	1	1	0	1
g	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	xx	1	0	1	0
g	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	xx	1	1

Fig. 7 The ($K = 4$, $L = 3$) biclustering solution obtained using TMKLMP and TMBP for the friendship ties among 22 third-grade students (see Anderson et al., 1999; Parker & Asher, 1993). The students are labeled using the numbering scheme used in the source, with "b" and "g" being used to label boys and girls, respectively. The solid lines distinguish the row and column clusters

		b	b	b	b	b	b	b	b	b	b	b	b	b	b	g	g	g	g	g	g	g	g
		2	3	5	11	8	10	19	13	21	1	4	7	20	16	15	12	17	18	14	9	6	22
b	2	0	1	1	1	0	0	1	1	1	1	0	1	1	1	0	0	1	0	0	0	0	1
b	5	1	1	0	1	1	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0
b	11	1	1	1	0	1	1	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0
b	8	1	1	1	1	0	1	1	1	1	1	1	0	0	1	0	0	0	0	1	0	0	0
b	20	1	1	1	1	1	1	1	1	1	1	0	0	0	1	0	0	0	0	0	0	1	0
b	16	1	0	1	0	1	1	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0
b	19	1	1	1	1	1	1	0	1	1	1	1	0	1	1	0	0	0	0	1	0	0	0
b	21	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
b	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
b	3	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
b	4	0	0	0	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0
b	7	0	1	1	1	1	0	0	1	1	0	0	0	1	0	0	0	1	1	0	0	0	0
b	10	1	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0
b	13	1	0	0	0	1	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	0	0
g	6	0	0	0	0	1	0	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0
g	12	1	1	1	1	1	1	1	0	1	1	1	0	0	0	1	0	1	1	1	1	1	1
g	14	0	0	1	1	0	0	1	1	1	1	0	0	0	0	0	0	1	1	0	0	1	0
g	22	1	0	1	1	0	1	0	0	1	0	0	0	0	0	1	1	1	0	0	1	0	0
g	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
g	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	1
g	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0
g	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	1	1

Fig. 8 The ($K = 4$, $L = 3$, $D = 2$) biclustering solution obtained using NMF for the friendship ties among 22 third-grade students (see Anderson et al., 1999; Parker & Asher, 1993). The students are labeled using the numbering scheme used in the source, with "b" and "g" used to label boys and girls, respectively. The solid lines distinguish the row and column clusters

Implementation

Acquiring the software

The software programs associated with this article are accessible in the folder ASYM, which can be downloaded from the following website: http://myweb.fsu.edu/mbrusco. Six primary m-file scripts are included in the folder: (i) tmklmp.m, (ii) tmklmp_nodiag.m, (iii) tmbp.m, (iv) tmbp_nodiag.m, (v) nmf.m, and (vi) nmf_nodiag.m. In addition, two m-script subroutines are called by the NMF programs, which are named hkmeans.m and ssquares.m, respectively. There are also two data files, lipread.prn and 3rdGrade.prn, which contain the input matrices for Examples 1 and 2, respectively. Succinct nontechnical pseudocodes for TMKLMP, NMF, and TMBP are provided in Fig. 9. More rigorous descriptions are provided by Brusco and Doreian (2015a), Brusco (2011), and Doreian et al. (2004) for the respective algorithms.

A document in the ASYM folder also contains the source code for all eight m-file scripts. Moreover, for each of the six primary m-scripts, there are descriptions of how to perform the function calls in MATLAB, as well as the primary inputs and outputs of the program. The parameter settings that can be adjusted by users familiar with MATLAB are also identified. Finally, the Word document contains small numerical examples illustrating some of the procedures. As we noted previously, these examples were solved using a 2.2-GHz Pentium 4 PC (circa 2000–2002), and accordingly, the displayed computation times for the examples should be considered conservative.

Choosing among the methods

Two questions concerning the properties of the available asymmetric one-mode data can be used to guide the selection of which method to choose: (i) Should the main diagonal be considered? and (ii) Are the elements of the network matrix binary or nonnegative real-valued? In our experience, the most common answer to question (i) is “no.” Consider, for example, asymmetric matrices that stem from social network analyses, in which it is generally illogical for someone to identify him- or herself as a friend, someone they trust, someone from whom they seek advice, and so forth. Accordingly, in these circumstances, the “nodiag” versions of the programs

```
TMKLMP
For i = 1 to 500 (500 is the default number of restarts)
         1. Randomly construct an initial partitions of row ($K$ clusters) and column ($L$ clusters) objects
         2a. Compute submatrix means and variances
         2b. Assign row objects to row cluster associated with smallest distance from submatrix means
         2c. Correct for any empty row clusters
         3a. Compute submatrix means and variances
         3b. Assign column objects to column cluster associated with smallest distance from submatrix means
         3c. Correct for any empty column clusters
         4. If no row or column objects change membership in steps 2 and 3, then stop; otherwise repeat steps 2 and 3.
Next i
Return partitions associated with the maximum vaf across all 500 restarts.
NMF
For i = 1 to 20 (20 is the default number of restarts)
         1. Randomly populate the G and H matrices
         While error exceeds tolerance (.0000001) and iterations are less than the limit (5000)
            2. Update G and H, normalize G and H with respect to G, increment iterations
            3. Compute GH and error
         End while
Next i
Return G and H yielding minimum error across all 20 restarts.
Establish a $K$-cluster partition of the row objects by applying $K$-means to G.
Establish an $L$-cluster partition of the column objects by applying $K$-means to H.
TMBP
For i = 1 to 5000 (5000 is the default number of restarts)
         1. Randomly construct an initial partitions of row ($K$ clusters) and column ($L$ clusters) objects
         2. Compute the objective criterion index (number of inconsistencies)
         3. Evaluate all possible relocations of objects from their current cluster to each of the other clusters.
         4. Each time a relocation is identified that reduces the number of inconsistencies, the change is implemented
         5. The algorithm terminates when there is no relocation of a row or column object that will reduce the number of inconsistencies
Next i
Return partitions associated with the minimum number of inconsistencies across all 5000 restarts.
```

Fig. 9 Nontechnical pseudocode for TMKLMP, NMF, and TMBP

would seem more appropriate. In other situations the decision might be less clear. For example, in a confusion matrix or brand-switching matrix, the main diagonals represent correct responses and successive purchases of the same brand repeatedly. These measures do have a logical interpretation; however, they often tend to be rather large and could be retained. Therefore, if the goal of the study were to analyze patterns of confusion (or switching), then it still might be preferable to use the "nodiag" programs to avert undue influence from the large diagonal terms. This issue also dovetails with the decision to possibly normalize the input matrix in some manner, which is a thorny matter that is beyond the scope of this article.

If the input matrix consists more generally of nonnegative real-valued integers, then the choice of method is restricted to the TMKLMP and NMF programs, because TMBP is limited to binary data. Our experience is that both TMKLMP and NMF perform well in most instances for nonnegative real-valued data. However, whereas both methods require the specification of the number of row and column clusters (i.e., K and L), NMF has the additional requirement of selecting the dimensionality (D) of the factorization. Moreover, we recommend the use of TMKLMP to select K and L for the NMF program, and therefore, the former procedure would have to be used anyway. Therefore, a reasonable strategy in the case of a nonnegative real-valued asymmetric matrix might be to run TMKLMP first, identify appropriate values for K and L based on *CHull*, and store the solution corresponding to those K and L values. Subsequently, NMF can be applied using the same K and L, but evaluating different values of D. Upon selection of the appropriate D, the NMF solution could be compared directly to the TMKLMP solution. This basic approach was applied gainfully in the case of the lipread consonant confusion data (Example 1), for which NMF produced a readily interpretable partition comporting well with the TMKLMP solution.

In the case of a binary input matrix, the computational evidence provided herein, as well as in other sources (Brusco, Doreian, Lloyd, & Steinley, 2013; Brusco, Doreian, Mrvar, & Steinley, 2013; Brusco & Steinley, 2007, 2011; Doreian et al., 2004, 2005), reveals the efficacy of TMBP. Accordingly, our general recommendation is that TMBP should typically be evaluated in the case of binary data, despite the fact that its attraction to the best-found objective criterion value across multiple restarts is commonly less than that in TMKLMP. However, the computational evidence reported by Brusco and Steinley (2007) and Brusco and Doreian (2015a) also revealed good performance for TMKLMP for binary data, as well as for nonnegative real-valued data. Although NMF can also be applied to either binary or nonnegative real-valued input matrices, our experience is that it tends to produce more easily interpretable results in the latter case. When applied to the binary friendship social network data among third graders, the NMF solution was less interpretable and, therefore, we do not recommend it highly for binary data.

To summarize, TMKLMP appears to produce good results for both nonnegative real-valued and binary matrices. Nevertheless, it would seem prudent to augment a TMKLMP analysis with either NMF or TMBP in the cases of nonnegative real-valued or binary matrices, respectively. In the examples provided herein, the TMKLMP and NMF methods both produced interpretable results for the lipread consonant confusion data, whereas TMKLMP and TMBP both produced interpretable results for the binary social network data.

Conclusions

Summary

Examples of one-mode asymmetric proximity data abound in the psychological sciences. Examples include data obtained from free association tasks, stimulus recognition tasks, brand selection, and social network analyses. When approaching the problem of partitioning objects in these applications, we suggest that it is generally advisable to adopt a biclustering perspective. More specifically, when the data are arranged in a one-mode asymmetric matrix, two distinct partitions of the objects must be obtained: one partition based on their role in the context of the rows of the matrix, and one partition based on their role in the context of the columns of the matrix. Effective methods for simultaneously establishing these partitions are not readily available in most commercial software packages. Accordingly, our goals were (i) to present three alternative methods for biclustering one-mode asymmetric matrices, (ii) to make available a suite of MATLAB m-files that implement these methods, and (iii) to demonstrate these methods and software using psychologically oriented examples in the literature. Furthermore, the methods are applicable to many areas of behavioral research.

The MATLAB m-file software programs included in the Web supplement associated with this article fall into three categories of methods: (i) two-mode *KL*-means partitioning (TMKLMP), (ii) nonnegative matrix factorization (NMF), and (iii) two-mode blockmodel partitioning (TMBP). Within each category, two m-file programs are provided, differentiated by the inclusion or exclusion of the main diagonal in the analysis. For example, in the case of TMKLMP, the program tmklmp.m includes the main diagonal in the analysis, whereas tmklmp_nodiag.m ignores the main diagonal. The former program can also be used more generally for any two-mode matrix.

Limitations and extensions

The limitations of the methods presented herein can be characterized along three dimensions: (i) scalability, (ii) suboptimality, and (iii) model selection. Regarding scalability, it is important to recognize that we have made the programs available as MATLAB m-files. Although MATLAB is a user-friendly environment, m-files are not compiled, and therefore they run much slower than comparable codes written in Fortran or C. It also appears that the TMKLMP programs are appreciably more efficient than the TMBP and NMF programs. Nevertheless, most of the m-files should scale for object set sizes of $N \leq 200$. Larger matrices can also be tackled; however, it might be necessary to scale back on the number of restarts.

All of the m-files use heuristic procedures to produce solutions to their respective optimization problems. For this reason, a globally optimal solution is not guaranteed. However, as we noted previously, computational results reported in the literature suggest that each of the methods performs well and produces solutions competitive with those obtained by more sophisticated metaheuristics, such as simulated annealing, tabu search, genetic algorithms, and variable neighborhood search. Despite this evidence, we deemed it useful to allow the TMKLMP and TMBP programs to count the numbers of restarts for which the best-found solution was obtained. If the best-found solution was obtained only once or twice out of 500 or 5,000 restarts, it is quite possible that the global optimum was not located. Greater confidence (although no guarantee) of global optimality would be afforded by a larger number of discoveries of the best-found objective function across multiple restarts. Our limited analyses suggest that TMKLMP has a greater attraction to the best-found solution across multiple restarts than does TMBP. A related issue is the agreement among different locally optimal partitions. A capability for the measurement of agreement has not been integrated into the programs at the present time, because it is not entirely clear how users should best interpret such information.

Perhaps the most challenging aspect of all of the procedures described in this article is model selection. For TMKL MP and TMBP, model selection requires the choices of K and L and the decision to include or exclude the main diagonal. The NMF method also requires these decisions, in addition to a choice of D. Furthermore, for real-valued $\mathbf{X}$ matrices, transformations of the asymmetric proximity matrix prior to application of the method must be a concern. For example, Brusco and Doreian (2015b) considered applications to journal citation and brand-switching matrices in which a transformation was employed to adjust for scale differences prior to the implementation of TMKLMP.

References

Anderson, C. J., Wasserman, S., & Crouch, B. (1999). A *p** primer: Logit models for social networks. *Social Networks, 21,* 37–66.

Baier, D., Gaul, W., & Schader, M. (1997). Two-mode overlapping clustering with applications in simultaneous benefit segmentation and market structuring. In R. Klar & O. Opitz (Eds.), *Classification and knowledge organization* (pp. 557–566). Heidelberg: Springer.

Baker, F. B., & Hubert, L. J. (1977). Applications of combinatorial programming to data analysis: Seriation using asymmetric proximity measures. *British Journal of Mathematical and Statistical Psychology, 30,* 154–164.

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

Brusco, M. (2011). Analysis of two-mode network data using nonnegative matrix factorization. *Social Networks, 33,* 201–210.

Brusco, M., & Doreian, P. (2015a). A real-coded genetic algorithm for the two-mode *KL*-means partitioning problem with application to homogeneity blockmodeling. *Social Networks, 41,* 26–35.

Brusco, M. J., & Doreian, P. (2015b). An exact algorithm for the two-mode *KL*-means partitioning problem. *Journal of Classification*.

Brusco, M., Doreian, P., Lloyd, P., & Steinley, D. (2013a). A variable neighborhood search method for a two-mode blockmodeling problem in social network analysis. *Network Science, 1,* 191–212.

Brusco, M., Doreian, P., Mrvar, A., & Steinley, D. (2013b). An exact algorithm for blockmodeling of two-mode network data. *Journal of Mathematical Sociology, 37,* 61–84.

Brusco, M. J., & Stahl, S. (2005a). Bicriterion seriation methods for skew-symmetric matrices. *British Journal of Mathematical and Statistical Psychology, 58,* 333–343.

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

Brusco, M. J., & Steinley, D. (2006). Clustering, seriation, and subset extraction of confusion data. *Psychological Methods, 11,* 271–286.

Brusco, M., & Steinley, D. (2007). A variable neighborhood search method for generalized blockmodeling of two-mode binary matrices. *Journal of Mathematical Psychology, 51,* 325–338.

Brusco, M. J., & Steinley, D. (2009). Integer programs for one- and two-mode blockmodeling based on prespecified image matrices for structural and regular equivalence. *Journal of Mathematical Psychology, 53,* 577–585. doi:10.1016/j.jmp.2009.08.003

Brusco, M., & Steinley, D. (2010). *K*-balance partitioning: An exact method with application to generalized structural balance and other psychological contexts. *Psychological Methods, 15,* 145–157. doi:10.1037/a0017738

Brusco, M., & Steinley, D. (2011). A tabu search heuristic for deterministic two-mode blockmodeling of binary network matrices. *Psychometrika, 76,* 612–633.

Brusco, M. J., & Steinley, D. (2014). Model selection for minimum diameter partitioning. *British Journal of Mathematical and Statistical Psychology, 67,* 471–495.

Ceulemans, E., & Van Mechelen, I. (2005). Hierarchical classes models for three-way three-mode binary data: Interrelations and model selection. *Psychometrika, 70,* 461–480. doi:10.1007/s11336-003-1067-3

Chino, N. (1978). A graphical technique in representing asymmetric relationships between *n* objects. *Behaviormetrika, 5,* 23–40.

Constantine, A. G., & Gower, J. C. (1978). Graphical representation of asymmetric matrices. *Applied Statistics, 27,* 297–304.

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

Doreian, P., Batagelj, V., & Ferligoj, A. (2004). Generalized blockmodeling of two-mode network data. *Social Networks, 26,* 29–53.

Doreian, P., Batagelj, V., & Ferligoj, A. (2005). *Generalized blockmodeling*. Cambridge: Cambridge University Press.

Doreian, P., Lloyd, P., & Mrvar, A. (2013). Partitioning large signed two-mode networks: Problems and prospects. *Social Networks, 35,* 1–21.

Eckart, C., & Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika, 1,* 211–218.

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

Fogel, P., Hawkins, D. M., Beecher, C., Luta, G., & Young, S. S. (2013). A tale of two matrix factorizations. *American Statistician, 67,* 207–218.

Gibbons, D., & Olk, P. M. (2003). Individual and structural origins of friendship and social position among professionals. *Journal of Personality and Social Psychology, 84,* 340–351.

Hansohm, J. (2002). Two-mode clustering with genetic algorithms. In W. Gaul & G. Ritter (Eds.), *Classification, automation and new media* (pp. 87–93). Berlin: Springer.

Harshman, R. A., Green, P. E., Wind, Y., & Lundy, M. E. (1982). A model for the analysis of asymmetric data in marketing research. *Marketing Science, 1,* 205–242.

Hartigan, J. (1972). Direct clustering of a data matrix. *Journal of the American Statistical Association, 67,* 123–129.

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

Hubert, L. J. (1976). Seriation using asymmetric proximity measures. *British Journal of Mathematical and Statistical Psychology, 29,* 32–52.

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

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

Köhn, H.-F., Steinley, D., & Brusco, M. J. (2010). The *p*-median model as a tool for clustering psychological data. *Psychological Methods, 15,* 87–95. doi:10.1037/a0018535

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature, 401,* 788–791.

Lee, D. D., & Seung, H. S. (2001). Algorithms for nonnegative matrix factorization. In T. L. Keen, T. K. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems 13* (pp. 556–562). Cambridge: MIT Press.

Madeira, S. C., & Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis: A survey. *IEEE Transactions in Computational Biology and Bioinformatics, 1,* 24–45.

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.

Okada, A., & Imaizumi, T. (1987). Nonmetric multidimensional scaling of asymmetric proximities. *Behaviormetrika, 21,* 81–96.

Parker, J. G., & Asher, S. R. (1993). Friendship and friendship quality in middle childhood: Links with peer group acceptance and feelings of loneliness and social dissatisfaction. *Developmental Psychology, 29,* 611–621.

Prelić, A., Blueler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W., . . . Zitzler, E. (2006). A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, *22*, 1122–1129.

Ranyard, R. H. (1976). An algorithm for maximum likelihood ranking and Slater's *i* from paired comparisons. *British Journal of Mathematical and Statistical Psychology, 29,* 242–248.

Schepers, J., Ceulemans, E., & Van Mechelen, I. (2008). Selection among multi-mode partitioning models of different complexities. *Journal of Classification, 25,* 67–85.

Schepers, J., & Hofmans, J. (2009). TwoMP: A MATLAB graphical user interface for two-mode partitioning. *Behavior Research Methods, 41,* 507–514. doi:10.3758/BRM.41.2.507

Schepers, J., & Van Mechelen, I. (2011). A two-mode clustering method to capture the nature of the dominant interaction pattern in large profile data matrices. *Psychological Methods, 16,* 361–371. doi:10.1037/a0024446

Schepers, J., Van Mechelen, I., & Ceulemans, E. (2011). The real-valued model of hierarchical classes. *Journal of Classification, 28,* 363–389. doi:10.1007/s00357-011-9089-5

Steinley, D. (2006). *K*-means clustering: A half-century synthesis. *British Journal of Mathematical and Statistical Psychology, 59,* 1–34.

Steinley, D., & Brusco, M. J. (2011). Choosing the number of clusters in K-means clustering. *Psychological Methods, 16,* 271–285.

Takeuchi, A., Saito, T., & Yadohisa, H. (2007). Asymmetric agglomerative hierarchical clustering algorithms and their evaluations. *Journal of Classification, 24,* 123–143.

Totterdell, P., Wall, T., Holman, D., Diamond, H., & Epitropaki, O. (2004). Affect networks: A structural analysis of the relationship between work ties and job-related affect. *Journal of Applied Psychology, 89,* 854–867. doi:10.1037/0021-901.89.5.854

Trejos, J., & Castillo, W. (2000). Simulated annealing optimization for two-mode partitioning. In W. Gaul & R. Decker (Eds.), *Classification and information at the turn of the millennium* (pp. 135–142). Heidelberg: Springer.

Van Mechelen, I., Bock, H. H., & DeBoeck, P. (2004). Two-mode clustering methods: A structured overview. *Statistical Methods in Medical Research, 13,* 363–394.

van Rosmalen, J., Groenen, P. J. F., Trejos, J., & Castillo, W. (2009). Optimization strategies for two-mode partitioning. *Journal of Classification, 26,* 155–181.

Van Uitert, M., Meuleman, W., & Wessels, L. (2008). Biclustering sparse binary genomic data. *Journal of Computational Biology, 15,* 1329–1345.

Vicari, D. (2014). Classification of asymmetric proximity data. *Journal of Classification, 31,* 386–420.

Vichi, M. (2001). Double *K*-means clustering for simultaneous classification of objects and variables. In S. Borra, R. Rocchi, & M. Schader (Eds.), *Advances in classification and data analysis—Studies in classification, data analysis and knowledge organization* (pp. 43–52). Heidelberg: Springer.

Wilderjans, T. F., Ceulemans, E., & Meers, K. (2013a). CHull: A generic convex-hull-based model selection method. *Behavior Research Methods, 45,* 1–15. doi:10.3758/s13428-012-0238-5

Wilderjans, T. F., Depril, D., & Van Mechelen, I. (2013b). Additive biclustering: A comparison of one new and two existing ALS algorithms. *Journal of Classification, 30,* 56–74. doi:10.1007/s00357-013-9120-0

Young, G., & Householder, A. S. (1938). Discussion of a set of points in terms of their mutual distances. *Psychometrika, 3,* 19–22.

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