Psychological Methods
2004, Vol. 9, No. 4, 510–523

1082-989X/04/$12.00 DOI: 10.1037/1082-989X.9.4.510

Clustering Binary Data in the Presence of Masking Variables

Michael J. Brusco
Florida State University

A number of important applications require the clustering of binary data sets. Traditional nonhierarchical cluster analysis techniques, such as the popular *K*-means algorithm, can often be successfully applied to these data sets. However, the presence of masking variables in a data set can impede the ability of the *K*-means algorithm to recover the true cluster structure. The author presents a heuristic procedure that selects an appropriate subset from among the set of all candidate clustering variables. Specifically, this procedure attempts to select only those variables that contribute to the definition of true cluster structure while eliminating variables that can hide (or mask) that true structure. Experimental testing of the proposed variable-selection procedure reveals that it is extremely successful at accomplishing this goal.

Cluster analysis is an important method for uncovering structural information in data. The crucial role of this method in psychology is evidenced by its many applications as well as focused efforts to develop better algorithms and careful guidelines for implementation. Monte Carlo studies have been conducted to evaluate clustering objective criteria (Milligan, 1981; Milligan & Cooper, 1986), the effects of error perturbation (Milligan, 1980), the effects of dimensionality and cluster density (Milligan, Soon, & Sokal, 1983), identification of the appropriate number of clusters (Milligan & Cooper, 1985), outlier detection (Cheng & Milligan, 1996), standardization of variables (Milligan & Cooper, 1988), and variable weighting and selection (Brusco & Cradit, 2001; Milligan, 1989). An excellent synopsis of many of these studies is provided by Milligan (1996).

Most in-depth studies in the clustering literature have concentrated on continuous-variable data sets, which are typically generated from multivariate normal distributions with carefully prescribed degrees of cluster overlap on one or more variables (see Milligan, 1985). The performance of algorithms and procedural decisions for other types of data sets is less well-known. One particular category of data sets that is considerably significant corresponds to variables with binary measurements. Binary data can arise in a variety of contexts. Examples include the measurement of the presence or absence of various symptoms for a group of patients (Williams, Barton, White, & Won, 1976) and the measurement of the commission or noncommission of various criminal offenses by a collection of citizens (Cliff, McCormick, Zatkin, Cudeck, & Collins, 1986). In fact, any application in which subjects are asked to provide responses to a set of dichotomous questions will yield a binary data set.

Curry (1976) provided an overview of the statistical properties of binary data, carefully observing that objects measured across a battery of binary variables may be viewed as sets. De Boeck and Rosenberg (1988) subsequently developed a formal classification method for binary data (HICLAS), which capitalizes on the set-theoretic properties of this type of data. More specifically, a Boolean decomposition of a binary matrix is obtained for a prespecified rank. This decomposition requires estimation of object and attribute (or variable) bundle matrices that are obtained by means of an alternating least-squares algorithm. The HICLAS procedure has been deployed in several substantive applications in social and cognitive psychology (Reich, 2000; Rosenberg, 1989; Storms, Van Mechelen, & De Boeck, 1994). Hands and Everitt (1987) compared the performances of traditional hierarchical clustering methods across small binary data sets (up to 200 objects). Among the most salient findings of their study was the propensity for better cluster recovery when the number of clusters was small and the number of clustering variables was large. In their comprehensive reviews of combinatorial data analysis, Arabie and Hubert (1992, p. 175, and 1996, p. 11) provided a number of other relevant references for the clustering of binary data.

More recently, Dimitriadou, Dolničar, and Weingessel (2002) investigated the performances of 15 indices for selecting the appropriate number of clusters in a binary data set. Among the noteworthy aspects of their study were its emphasis on large binary data sets (6,000 objects) and its use of *K*-means clustering (MacQueen, 1967) and competitive learning (Leisch, Weingessel, & Dimitriadou, 1998) as clustering methods. *K*-means is perhaps the most popular

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

partitioning approach and is available in many commercial software packages, such as MINITAB, SPSS, SAS, and SYSTAT. Competitive learning methods fall within the family of neural network models, which have become increasingly popular in the psychometric and classification literature (Ambroise & Govaert, 1996; Balakrishnan, Cooper, Jacob, & Lewis, 1994; Waller, Kaiser, Illian, & Manry, 1998). Neural network models are abstractions of human neural pathways and neurons that enable software programs to learn from information or experience. Good overviews of the history of neural network models and the motivation for their use in classification are provided by Balakrishnan et al. (1994) and Waller et al. (1998).

Dimitriadou et al. (2002) concluded that an index developed by Ratkowsky and Lance (1978) was especially effective for determining the number of clusters when working with binary data. Although they did not emphasize the evaluation of heuristic algorithms, Dimitriadou et al. suggested that both *K*-means and competitive learning were effective methods for recovering true cluster structure in binary data. They also presented results suggesting that these methods outperform latent class clustering approaches for large binary data sets. Throughout the remainder of this article, I emphasize the deployment of the *K*-means algorithm while recognizing that neural network models and other methods provide a viable alternative approach. Comparative analyses of classification criteria and clustering methods for binary data are not provided in this article.

As in Dimitriadou et al.'s (2002) article, the focus in this article is on binary data sets; however, I am concerned with the ability of partitioning algorithms to recover the true cluster structure in the data sets when extraneous or irrelevant variables are present. It is well-recognized in the psychometric literature that analysts often have a large set of variables available for inclusion in a cluster analysis but that only a subset of those variables might be appropriate for uncovering true cluster structure. Throughout the remainder of this article, I use the term *true variables* to refer to those variables that define the true cluster structure. Unfortunately, incorporating the entire set of candidate variables into the cluster analysis is generally ineffective because the inclusion of the irrelevant variables impedes the recovery of the true cluster structure. In other words, the irrelevant variables hide or obfuscate the true structure in the data set. This led Fowlkes and Mallows (1983) to use the term *masking variables* for the irrelevant variables, a term which has been adopted in the psychometric and classification literature.

Masking variables are problematic for applied analyses because they can prevent recovery of true cluster structure and, subsequently, yield erroneous conclusions. Not surprisingly, a significant research effort has been devoted to the selection of variables for a cluster analysis as well as to differential weighting of variables (Bishop, 1995; Brusco & Cradit, 2001; Carmone, Kara, & Maxwell, 1999; DeSarbo, Carroll, Clark, & Green, 1984; De Soete, 1986; Fowlkes, Gnanadesikan, & Kettenring, 1988; Gnanadesikan, Kettenring, & Tsao, 1995; Green, Carmone, & Kim, 1990; Milligan, 1989). Gnanadesikan et al. observed that variable-weighting procedures were frequently outperformed by variable-selection methods with respect to uncovering cluster structure. In addition to superior performance, variable-selection methods have another important advantage over variable-weighting procedures. Specifically, variable-selection procedures exclude masking variables completely, which precludes the need for their measurement in subsequent cluster analyses.

Variable-selection procedures attempt to select the appropriate subset of candidate variables, incorporating true variables in the subset while avoiding masking variables. Exhaustive enumeration of all subsets might be possible for 10 to 15 candidate variables but is computationally infeasible for larger variable sets. A further complication arises because of what Bishop (1995) calls the "monotonicity property" (p. 305) associated with many clustering criteria. This means that if a given subset is modified by adding one or more variables, the resulting clustering criterion cannot possibly improve. The monotonicity property, therefore, makes it difficult to compare subsets of different sizes. For this reason, the accelerated procedures for variable selection described by Bishop (1995, pp. 306–309) are designed for a fixed subset size.

One of the earliest variable-selection procedures for cluster analysis was developed by Fowlkes et al. (1988). The principal limitation of this method is that it requires informal interpretation of graphical information and is, therefore, not conducive to large-scale experimental analyses. Carmone et al. (1999) and Brusco and Cradit (2001) independently developed variable-selection procedures that use Hubert and Arabie's (1985) adjusted Rand index (ARI), which is a measure of agreement between two partitions. These procedures have proven quite successful at eliminating masking variables when the data measurements are continuous. Unfortunately, neither of the methods can be easily adapted for the case of binary data because each relies on single-variable partitions consisting of two or more clusters. For a single binary variable, all of the measured values are 0 or 1 and, therefore, that single variable can only define exactly two clusters. Thus, even though binary data sets are prone to the same type of masking variable problems that can occur with continuous variables, they are not directly amenable to these most recent variable-selection procedures.

One objective of this article is to experimentally assess the ability of the *K*-means partitioning procedure to recover true cluster structure in binary data sets. A second, more important, contribution is the development and testing of a variable-selection procedure for binary data sets. In the next section of this article, I describe the *K*-means algorithm and provide an interpretation of the *K*-means criterion within the

context of a similarity index for binary clusters. I also present the variable-selection algorithm for binary data sets. In subsequent sections I provide the results of experiments designed to evaluate the performance of the *K*-means algorithm both with and without the prior implementation of the variable-selection procedure. I conclude the article with a brief summary and suggestions for future research.

Algorithms

Anderberg (1973) and Späth (1980, chap. 2) provide excellent coverage of many possible indices that can be developed for binary data. The development of the relevant objective criterion for clustering binary data that is used herein requires the following notation:

- N = the number of objects (subjects, observations, etc.) to be clustered, indexed $i = 1, \ldots, N$;
- K = the number of clusters, indexed $k = 1, \ldots, K$;
- P = the number of candidate clustering variables (dimensions), indexed $p = 1, \ldots, P$;
- R = $\{p = 1, \ldots, P\}$; the complete set of all candidate clustering variables;
- $\mathbf{X}$ = an $N \times P$ binary matrix with elements x_{ip} representing the measurement for object i on variable p;
- S_i = $\{p \in R | x_{ip} = 1\}$; the set of variables for which the measured value for object i is 1. In other words, S_i is the subset, for object i ($i = 1, \ldots, N$), of those elements in R for which $x_{ip} = 1$;
- $\bar{S}_i$ = $\{p \in R | x_{ip} = 0\}$; the complement of S_i, this represents the set of variables for which the measured value for object i is 0, that is, the subset, for object i ($i = 1, \ldots, N$), of those elements in R for which $x_{ip} = 0$;
- π = $\{C_1, C_2, \ldots, C_K\}$ a feasible partition of the N objects into K clusters, where C_k represents the set of objects that are assigned to cluster k ($k = 1, \ldots, K$). I also define $N_k = |C_k|$ as the cardinality of C_k, which represents the number of objects in cluster k ($k = 1, \ldots, K$);
- Π_K = the set of all feasible partitions of the N objects into K clusters.

Restle (1959) and Curry (1976) have described a metric for binary data that is based on the set-theoretic properties of such data. With set notation, the metric can be represented as follows:

$$d_{ij} = d_{ji} = \left|(S_i \cap \bar{S}_j) \cup (\bar{S}_i \cap S_j)\right| \quad \text{for } 1 \le i < j \le N. \quad (1)$$

This metric represents, for any pair of objects i and j, the number of variables for which the objects have different measurements. Because of the binary property of the data, the metric can also be represented as the sum of squared deviations between pairs of row vectors of $\mathbf{X}$:

$$d_{ij} = d_{ji} = \sum_{p \in R} (x_{ip} - x_{jp})^2 \quad \text{for } 1 \le i < j \le N. \quad (2)$$

To illustrate the relationship between Equations 1 and 2, consider the 2×8 data matrix, $\mathbf{X}$:

$$\mathbf{X} = \begin{bmatrix} 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \end{bmatrix}.$$

Row 1 of this matrix contains ones in Columns 1, 3, 4, and 7, which yields $S_1 = \{1, 3, 4, 7\}$ and $\bar{S}_1 = \{2, 5, 6, 8\}$. Similarly, Row 2 has ones in Columns 1, 4, 5, 7, and 8, resulting in $S_2 = \{1, 4, 5, 7, 8\}$ and $\bar{S}_2 = \{2, 3, 6\}$. The element common to both S_1 and $\bar{S}_2$ is $S_1 \cap \bar{S}_2 = \{3\}$, whereas the elements common to both S_2 and $\bar{S}_1$ are $\bar{S}_1 \cap S_2 = \{5, 8\}$. Joining these two subsets results in $(S_1 \cap \bar{S}_2) \cup (\bar{S}_1 \cap S_2) = \{3, 5, 8\}$, and these elements correspond to the columns of $\mathbf{X}$ for which Rows 1 and 2 have different values. The metric for binary data expressed by Equation 1 is the number of elements in the subset, $d_{12} = |\{3, 5, 8\}| = 3$. With Equation 2, the same value for the metric is obtained by means of $d_{12} = (1 - 1)^2 + (0 - 0)^2 + (1 - 0)^2 + (1 - 1)^2 + (0 - 1)^2 + (0 - 0)^2 + (1 - 1)^2 + (0 - 1)^2 = 3$.

I define $\mathbf{D} = \{d_{ij}\}$ as the $N \times N$ matrix of distances computed using Equations 1 or 2, also noting that $d_{ii} = 0$ for $i = 1, \ldots, N$. Hubert, Arabie, and Meulman (2001, chap. 3) have proposed a wide assortment of criteria for obtaining a K-cluster partition of $\mathbf{D}$. Perhaps the most popular among their collection is the within-cluster sum of squares, or K-means criterion:

$$\min_{\pi \in \Pi_K} : Z = \sum_{k=1}^{K} \frac{1}{N_k} \sum_{(i<j) \in C_k} d_{ij}. \quad (3)$$

Although Equation 3 can generally be used for any $N \times N$ nonnegative matrix of dissimilarities (i.e., a large matrix element indicates less similarity between the corresponding object pair), when applied to $\mathbf{D}$, the criterion is equivalent to minimizing the within-cluster sum of squares.

With the exception of a few special cases, partitioning problems of the type posed by Equation 3 generally fall into the class of nondeterministic polynomial-time (NP-hard) optimization problems (Day, 1996). For this class of problems, there are no available algorithms that are guaranteed to produce globally optimal solutions with computational effort that is a polynomial function of problem size. From a pragmatic standpoint, this means that algorithms designed to provide guaranteed globally optimal solutions are not computationally feasible for NP-hard optimization problems of practical size. The cpu times for such algorithms, which include dynamic programming (Hubert et al., 2001; Jensen, 1969) and branch-and-bound methods (Brusco, 2003; Koontz, Narendra, & Fukunaga, 1975), tend to grow exponentially as problem size increases. Although dynamic programming and branch-and-bound programming can pro-

vide globally optimal solutions for small data sets, they are not widely available in commercial software products because of the severe limitations on the sizes of problems they can feasibly handle.

Heuristic procedures for Equation 3 have been devised by Forgy (1965), MacQueen (1967), Hartigan and Wong (1979), and many others. Despite the fact that these *K*-means procedures do not guarantee globally optimal solutions, they are quite scalable and can be applied to data sets with thousands of objects. It is well known that *K*-means algorithms are sensitive to the initial partition of objects that is provided as input to the algorithm (see Steinley, 2003, for a recent discussion of the severity of this problem). To avoid realization of a possibly poor solution from a single implementation of a *K*-means algorithm, it is common practice to run replications of the algorithm. For each replication, the algorithm is supplied with a different (often randomly generated) initial partition. Larger data sets tend to require more replications to ensure that optimal (or, at least, near-optimal) solutions are identified. Fortunately, on current microcomputer platforms, it is often possible to run thousands of replications for these larger data sets in a modest amount of time.

Dimitriadou et al. (2002) reported considerable effectiveness for a *K*-means algorithm used to cluster binary data. The *K*-means algorithm deployed in this article is consistent with Dimitriadou et al.'s implementation, and each replication begins with the random selection of *K* rows from **X**. These rows serve as the initial seed values (or initial cluster centroids) for the algorithm. Each object is subsequently assigned to its nearest seed, and cluster centroids are recomputed. Reassignment and recomputation of the centroids occurs, in an iterative manner, until no change in cluster memberships is observed. In an effort to mitigate problems associated with local minima, Dimitriadou et al. reported using 100 replications of the *K*-means algorithm in their study. To provide a reasonable assessment of true cluster structure recovery for large binary data sets, it was necessary to use many more replications. Therefore, 10,000 replications of the *K*-means algorithm were used for each test problem in this study.

Variable-Selection Heuristic for Binary Data Sets

The description of the proposed variable-selection procedure for binary data sets (VSBD) uses the following notation in addition to the terms defined previously:

- φ = the proportion of objects from the full data set that is randomly selected to create the sampled data set;
- M = the number of objects ($M = \varphi N$) in the sampled data set;
- L = a set of M objects that is randomly selected from the collection of all N objects;
- Q = the set of clustering variables ($Q \subseteq R$) that has been selected for inclusion in the cluster analysis;
- V = the number of selected variables ($V = |Q|$);
- V_I = an initial choice for the number of selected variables (i.e., the starting value for V in the VSBD algorithm);
- $\mathbf{A}$ = an $M \times V$ binary matrix that is extracted from $\mathbf{X}$ using the objects from L and the variable indices from Q;
- δ = a parameter ($0 < \delta < 1$) that defines a stopping rule for the variable-selection process.

A concise presentation of the VSBD algorithm is displayed in the Appendix. The algorithm begins in Step 0 with the initialization of parameters. For binary data sets with $N = 500$ or fewer objects, I recommend applying the VSBD algorithm using all of the objects, and this is accomplished by setting $\varphi = 1$, which results in $M = N$. However, this setting is both computationally inefficient and unnecessary for larger binary data sets. For binary data sets with $N \geq 2{,}000$, I found that $\varphi = .10$ was sufficient to enable the VSBD algorithm to extract the proper subset of clustering variables. For binary data sets with $500 < N < 2{,}000$ objects, it would seem pragmatic to use $.2 \leq \varphi \leq .3$. My decision to initialize the number of selected variables at $V = V_I = 4$ was based on two factors. First, it is unlikely that researchers would wish to consider anything less than four clustering variables for most binary data sets. As an illustration, consider the fact that any set of three binary variables could provide a perfect separation of eight clusters. Second, any value larger than $V_I = 4$ would tend to result in a prohibitive number of combinations for large values of P.

In Step 1 of the algorithm, M objects are randomly selected from all N possible objects. Initially, all objects are considered as "unselected." An integer is randomly generated on the interval [1, N], and the object corresponding to that integer is placed in set L. This process is repeated until M unique objects are contained in L. The remainder of the VSBD algorithm operates using only the M objects in set L, not all N objects.

Steps 2 and 3 are the "engine" of VSBD. In Step 2, the algorithm attempts to find the best subset of V_I variables by explicit evaluation of all possible subsets. This is extremely crucial because of the need to provide an exceptionally good small subset to the less computationally intensive, iterative selection process in Step 3. For example, if $P = 10$, then all subsets of $V_I = 4$ of the 10 variables are tested in Steps 2a and 2b. The subsets, Q', are generated and tested sequentially: {1, 2, 3, 4}, {1, 2, 3, 5}, {1, 2, 3, 6}, . . . , {6, 8, 9, 10}, {7, 8, 9, 10}. After evaluation of each of these subsets, it might be observed that the subset $Q = \{2, 5, 7, 8\}$ yields the minimum value of Equation 3, and this subset is subsequently passed to Step 3 for possible augmentation.

Clearly, the feasibility of Step 2 is dictated by the values of P and V_I. From a practical standpoint, as the values of

these parameters increase, complete enumeration of all combinations in Step 2 becomes less plausible. In those situations, a possible strategy would be to replace complete enumeration of all possible subsets with a branch-and-bound algorithm (Bishop, 1995; Narendra & Fukunaga, 1977). Branch-and-bound is a partial enumeration scheme that would guarantee identification of an optimal subset without complete evaluation of all possible subsets. Although more efficient than complete enumeration, branch-and-bound algorithms can also become computationally infeasible for a large P and V_I, at which point an exchange algorithm that randomly moves variables in and out of the set Q is recommended.

The best subset of $V = V_I$ variables from Step 2 is passed to Step 3, when additional variables can be added one at a time. Each unselected variable ($p \in R \backslash Q$) is considered for inclusion in Q, and the candidate variable, p', yielding the minimum value for Equation 3, is retained. For example, if $Q = \{2, 5, 7, 8\}$ was passed to Step 3, the algorithm would evaluate the effect of adding each of the unselected variables to Q so as to increase the number of selected variables to five. This would require testing the following subsets: {2, 5, 7, 8, 1}, {2, 5, 7, 8, 3}, {2, 5, 7, 8, 4}, . . . , {2, 5, 7, 8, 10}. Assuming that {2, 5, 7, 8, 9} yielded the minimum value of Equation 3, then a judgment regarding the inclusion of the variable $p' = 9$ is made in Step 4.

The decision to include the candidate variable in Step 4 is based on the increase in the within-cluster sum of squares that is realized from its inclusion. If a variable with a random distribution of 50% zeros and 50% ones were measured with respect to the clustering solution, the expected cluster means for that variable are .5 and thus each object would contribute $(1 - .5)^2$ or $(0 - .5)^2$ to the total within-cluster sum of squared errors. Thus, the expected contribution would be .25 for each of the M objects, resulting in a total contribution of $M/4$. For the clustering variable to be worthy of inclusion, however, a somewhat smaller contribution to the within-cluster sum of squares is desired. Therefore, the parameter δ ($0 < \delta < 1$) is multiplied by $M/4$ to set the threshold. I observed considerable success with $\delta = .5$ throughout the experiments. Larger (smaller) values of δ would provide looser (tighter) restrictions on variable inclusion. The algorithm terminates at Step 4 if no candidate variable meets the constraint on the increase in the within-cluster sum of squares or at Step 5 if all candidate variables have been selected for inclusion.

Motivation for the VSBD Algorithm

The VSBD algorithm is predicated on the principle of identifying an excellent core subset of clustering variables and, subsequently, on attempting to augment that core subset. For this reason, the identification of the core subset in Step 2 of the algorithm is the most computationally intensive, yet most crucial, component of the process. Through complete evaluation of all possible subsets of $V_I = 4$ variables, Step 2 identifies a core subset of four variables that defines a sound cluster structure. Additional variables can only be added in Steps 3 and 4 if they work well in combination with this core subset of variables. If a variable can be added with only a modest increase in Equation 3, then the core subset is augmented accordingly. This is typically the case for variables that define true structures. Masking variables, on the other hand, tend not to work well with the core subset of variables and often grossly inflate the value of Equation 3. For this reason, masking variables tend not to be added to the core subset during execution of the VSBD algorithm.

A Numerical Illustration

Although VSBD was primarily designed for larger binary data sets, I provide an illustration for a very small hypothetical data set where $P = 10$ binary variables are measured for $N = 20$ subjects. As shown in Table 1, the variable set R consists of $P_T = 5$ true variables that define a perfect five-cluster structure as well as $P_M = 5$ masking variables. The fourth column of Table 1 provides the cluster memberships for the structure associated with the true variables. For example, Cluster 1 consists of 6 subjects each sharing the same measurements on the five true variables: $x_{i1} = x_{i3} = x_{i5} = 1$ and $x_{i2} = x_{i4} = 0$, for $i = 1, \ldots, 6$. Clusters 2, 3, 4, and 5 have 5, 4, 3, and 2 subjects, respectively.

I ran 100,000 replications of the K-means algorithm for the binary data set in Table 1 using $K = 5$ clusters and all $P = 10$ clustering variables. The algorithm yielded a within-cluster sum of squares value of $Z = 20.167$, which corresponded to the cluster memberships shown in the fifth column of Table 1. The disparity between the fourth and fifth columns of Table 1 clearly reveals the potential problems that can arise from the presence of masking variables. Although Cluster 3 was perfectly recovered by the K-means algorithm, Clusters 1 and 4 have been mixed together, as have Clusters 2 and 5.

I applied the VSBD algorithm to the binary data set using initial values of $V_I = 3$ and $\delta = .5$. Although a value of $V_I = 4$ (or possibly larger) might be more appropriate for larger data sets, the smaller value of $V_I = 3$ was selected for this example because it enables a more descriptive illustration. Because of the small number of subjects in this hypothetical example, I used all $M = N = 20$ subjects in the implementation, which precludes the need for sampling in Step 1. Step 2 of the algorithm yielded $Q = \{1, 2, 3\}$ as the best subset of three variables, with $Z_B = 0$. The first iteration of Step 3 resulted in the selection of variable $p' = 4$, with $Z_{B2} = 0$. This variable passed the test in Step 4 and was appended to Q, resulting in $Q = \{1, 2, 3, 4\}$. In a similar manner, Variable 5 was appended to Q in the next iteration, with Z_B remaining at zero. When considering the inclusion of a sixth variable in Q, candidate Variables 6, 7, 8, 9, and

Table 1
Data for a Numerical Demonstration of Variable Selection for Binary Data Sets

Subject	True variables	Masking variables	True cluster membership	*K*-means membership
1	1 0 1 0 1	1 1 0 0 1	1	4
2	1 0 1 0 1	0 0 1 1 0	1	1
3	1 0 1 0 1	1 1 0 0 1	1	4
4	1 0 1 0 1	0 0 0 1 0	1	1
5	1 0 1 0 1	0 0 1 0 1	1	1
6	1 0 1 0 1	1 0 0 1 0	1	1
7	0 1 1 0 0	0 1 1 1 0	2	2
8	0 1 1 0 0	1 1 0 1 0	2	2
9	0 1 1 0 0	1 0 1 0 1	2	5
10	0 1 1 0 0	0 1 0 1 1	2	2
11	0 1 1 0 0	1 0 1 0 0	2	5
12	0 0 1 1 1	0 1 1 1 0	3	3
13	0 0 1 1 1	0 0 0 0 0	3	3
14	0 0 1 1 1	1 0 0 0 1	3	3
15	0 0 1 1 1	0 1 0 1 1	3	3
16	1 1 0 0 1	0 0 1 1 0	4	1
17	1 1 0 0 1	1 1 0 0 1	4	4
18	1 1 0 0 1	0 1 1 1 1	4	1
19	0 1 0 1 0	0 0 0 1 0	5	2
20	0 1 0 1 0	1 1 0 1 0	5	2

Note. Together, the second and third columns form a 20 × 10 data matrix. The fourth column contains the cluster number for each subject and is based on the second column. The fifth column shows the cluster number for each subject based on a five-cluster *K*-means solution for the 20 × 10 data set.

10 would increase the within-cluster sum of squares by 4.62, 4.70, 3.95, 4.37, and 4.37, respectively. Thus, $Z_{B2} = 3.95$ for this iteration of Step 3. Because $Z_{B2} = 3.95 > Z_B + \delta(M/4) = 2.5$, the algorithm terminates at Step 4 with $Q = \{1, 2, 3, 4, 5\}$. Thus, VSBD accomplished its goal of selecting the five true variables while omitting the five masking variables. A *K*-means implementation using only the five true variables provides perfect recovery of the true structure.

Experimental Analyses

Experimental Design

I designed an experiment to investigate the performance of the *K*-means algorithm with respect to recovery of true cluster structure for binary data sets with masking variables. Particularly interesting was the performance of the *K*-means algorithm when applied to the full data set (with masking variables included) versus application of the *K*-means algorithm using only those variables selected by VSBD. The experiment for completing this evaluation was developed to incorporate a broad range of data conditions, and its design was similar to those used in previous Monte Carlo comparisons (Brusco & Cradit, 2001; Carmone et al., 1999; Milligan, 1989; Steinley, 2003).

A full factorial design with six factors and three levels for each factor was used. The six factors were (a) the number of objects, *N;* (b) the number of clusters, *K;* (c) the number of true clustering variables in the data set, P_T; (d) cluster density, *DENS;* (e) level of data perturbation, *PERT;* and (f) the number of masking variables in the data set, P_M. The levels of each factor are displayed in Table 2. The levels of $N = 2{,}000$, 3,000, and 4,000 were selected so as to be considered large, yet still enable a large number of problems to be analyzed in a reasonable amount of time. The *K*-means and VSBD algorithms can easily be applied to problems with 10,000 or more objects. The levels of $K = 4$, 6, and 8 are comparable with, though slightly broader than, the $K = 4$, 5, and 6 levels used by Dimitriadou et al. (2002) in their binary clustering experiments. The levels of $P_T = 4$, 6, and 8 are consistent with those used in previous variable-selection studies (Brusco & Cradit, 2001; Milligan, 1989).

For each object a P_T-dimensional vector was randomly selected from among *K* candidates so as to define the true cluster structure for that object. The vectors used for each combination of P_T and *K* are shown in Table 3. Although there is some arbitrariness with respect to the development of these vectors, for each combination of P_T and *K,* the vectors were carefully selected to clearly distinguish between clusters.

The cluster density factor for the experimental study, which corresponds to the relative sizes of the clusters, was operationalized using probability distributions for choosing among the *K* candidates. For Level 1 of the density factor,

Table 2
Experimental Factors and Levels

Factor	Level 1	Level 2	Level 3
Number of objects (N)	2,000	3,000	4,000
Number of clusters (K)	4	6	8
Number of true variables (P_T)	4	6	8
Density of clusters	Equal	[.375, .375, .125, .125] [.25, .25, .25, .083, .083, .083] [.1875, .1875, .1875, .1875, .0625, .0625, .0625, .0625]	[.5, .25, .15, .1] [.3, .25, .2, .125, .075, .05] [.25, .2, .175, .125, .075, .065, .06, .05]
Level of data perturbation (%)	None	2	4
Number of masking variables (P_T)	0	4	8

Note. Probabilities for cluster membership under Levels 2 and 3 of the density of clusters factor are displayed in brackets for 4, 6, and 8 clusters. There are $3^5 = 243$ test problems associated with each cell.

each of the P_T-dimensional vectors has an equal probability ($1/K$) of being selected. Thus, for this level of the density factor, K clusters of approximately the same size would be produced. For the second and third levels, the probability of assignment to one of the first $K/2$ clusters is 75%, thus producing clusters of markedly different sizes. The distinction between Levels 2 and 3 of the density factor corresponds to the probability distributions within the first $K/2$ and second $K/2$ clusters. For example, for $K = 4$ clusters, Levels 2 and 3 of the density factor both have a 75% probability of assigning objects to one of the first $K/2 = 2$ clusters. For Level 2, this probability is equally divided between Clusters 1 and 2 (.375 and .375 for both clusters), whereas for Level 3 the probability of cluster assignment is more heavily skewed toward Cluster 1 (.5 for Cluster 1 and .25 for Cluster 2).

Level 1 of the data perturbation factor (0%) provides data sets with a single, "error-free" true structure, whereas Levels 2 and 3 were carefully selected to yield some degradation of the true structure in the data sets. Data perturbation was conducted by randomly selecting 2% (or 4%) of the NP_T data elements (x_{ip}) defining true cluster structure and setting $x_{ip} = 1 - x_{ip}$. This recoding of binary measurements for randomly selected objects was implemented to produce

Table 3
Binary Variable Vectors Defining True Cluster Structure

K	P_T = 4	P_T = 6	P_T = 8
4	1 0 0 1	1 0 0 1 1 0	1 0 0 1 1 0 1 0
	1 1 1 0	1 1 1 0 0 0	1 1 1 0 0 0 1 0
	0 0 1 1	0 0 1 1 0 0	0 0 1 1 0 0 0 0
	0 1 0 1	0 1 0 1 0 1	0 1 0 1 0 1 1 0
6	1 0 0 1	1 0 0 0 1 1	1 0 0 0 1 1 0 1
	1 1 1 1	1 1 0 1 1 0	1 1 0 1 1 0 1 0
	1 0 1 0	1 1 1 0 0 0	1 1 1 0 0 0 0 1
	0 1 0 1	0 1 0 0 0 1	0 1 0 0 0 1 1 1
	0 0 0 1	0 1 1 1 1 0	0 1 1 1 1 0 1 1
	0 1 1 0	0 0 0 1 1 0	0 0 0 1 1 0 0 1
8	1 0 1 1	1 0 0 1 1 1	1 0 0 1 1 1 0 1
	1 0 0 0	1 0 1 0 0 0	1 0 1 0 0 0 1 1
	1 1 1 0	1 1 1 1 1 1	1 1 1 1 1 1 0 0
	1 1 0 1	1 1 0 0 0 1	1 1 0 0 0 1 0 1
	0 1 0 1	0 1 0 0 1 0	0 1 0 0 1 0 0 1
	0 1 0 0	0 1 1 0 0 1	0 1 1 0 0 1 0 1
	0 0 1 1	0 0 1 1 1 0	0 0 1 1 1 0 1 0
	0 0 0 1	0 0 1 0 0 1	0 0 1 0 0 1 0 1

noise in the data set, thus denigrating the true structure. The factor levels of 2% and 4% were carefully chosen on the basis of experimentation that suggested greater levels of perturbation could diminish the cluster structure too severely.

The measurement for each object on each masking variable was randomly generated on the basis of a uniform distribution. The factor levels for P_M are especially important. The level of $P_M = 0$ provides a good baseline because it is expected that *K*-means would perform well in the absence of masking variables, and it is desired that VSBD would select all of the P_T true variables for inclusion when no masking variables are present. The selected levels also provide several combinations where the number of masking variables equals or exceeds the number of true variables, with the most extreme condition being $P_T = 4$ and $P_M = 8$. This condition presents a formidable challenge for VSBD because it must select a small subset of true variables from a large set of variables that consists of mostly random noise. Without the successful prior implementation of VSBD, it is hypothesized that the *K*-means algorithm would have considerable difficulty recovering the true cluster structure for data sets where the number of masking variables (nearly) equals or exceeds the number of true variables.

The experimental design resulted in $3^6 = 729$ data sets. For each data set, I ran 10,000 replications of the *K*-means algorithm and stored the partition corresponding to the minimum value of Equation 3. I then used Hubert and Arabie's (1985) ARI to measure cluster recovery as the agreement between the partition obtained by the algorithm and the true cluster memberships for the objects. The formula for computing the ARI between two partitions, π_1 and π_2, is as follows:

$$\text{ARI} = \frac{H(\tau_1 + \tau_2) - [(\tau_1 + \tau_3)(\tau_1 + \tau_4) + (\tau_2 + \tau_3)(\tau_2 + \tau_4)]}{H^2 - [(\tau_1 + \tau_3)(\tau_1 + \tau_4) + (\tau_2 + \tau_3)(\tau_2 + \tau_4)]}, \quad (4)$$

where $H = N(N - 1)/2$, τ_1 is the number of object pairs that are in same cluster for both π_1 and π_2, τ_2 is the number of object pairs that are in different clusters for both π_1 and π_2, τ_3 is the number of object pairs that are in the same cluster in π_1 but different clusters for π_2, and τ_4 is the number of object pairs that are in the same cluster in π_2 but different clusters for π_1. The ARI yields a value of 1 for perfect agreement, whereas values near 0 indicate near-chance agreement. The index has been identified as the most effective external criterion for cluster validation (Milligan & Cooper, 1986). An excellent overview of the basic properties and applications of the ARI is provided by Steinley (2004).

The entire process of problem generation, partitioning, and measurement of cluster recovery was repeated for each data set, with the exception that VSBD was implemented to select clustering variables prior to the execution of the *K*-means algorithm. All algorithms were written in Fortran and were implemented on a 2.2-GHz Pentium IV PC with 1 GB of random-access memory.

Experimental Results

Computation times for the *K*-means algorithm, when using all of the clustering variables, were comparable with those associated with implementation of the VSBD algorithm followed by the *K*-means procedure. This important result is attributable to the fact that the time required to implement the VSBD algorithm is frequently offset, at least to a large extent, by savings in cpu time for the *K*-means algorithm. This occurs because the elimination of masking variables by VSBD enables the *K*-means algorithm to converge much more rapidly.

The experimental results are summarized in Tables 4 and 5. Table 4 presents an analysis of variance (ANOVA; main effects and all two-way interactions) associated with ARI as the dependent variable. For the ANOVA, the clustering method was included as a seventh factor (*METH*) with two levels: (a) *K*-means using all variables and (b) VSBD followed by *K*-means using only those variables selected by VSBD. Because of the potential for violation of the normality and constant error variance assumptions, I evaluated logarithmic and square root transformations of the dependent variable. These transformations did not produce any major differences relative to the analysis of the raw ARIs. Therefore, all reported results in Tables 4 and 5 correspond to raw values of the dependent variable.

All main effects, with the exception of the number of objects (*N*), were significant. The size of an effect, $\hat{\eta}^2$, is measured as the proportion of the corrected total sum of squares (sum of squares for all the effects and the error term but not the intercept) that is attributed to the effect. The number of true variables in the data set, P_T ($\hat{\eta}^2 = .2549$), and the level of perturbation in the data set, *PERT* ($\hat{\eta}^2 = .1567$), yielded the largest effect sizes. The interaction of these two factors ($P_T \times PERT$) also resulted in the largest effect size ($\hat{\eta}^2 = .0565$) among the two-way interactions. The second largest effect size ($\hat{\eta}^2 = .0416$) among the two-way interactions was associated with the number of masking variables in the data set, P_M, and the clustering method used (*METH*). The $P_M \times METH$ interaction is perhaps the most crucial result in Table 4 because it reflects the importance of using VSBD prior to applying the *K*-means algorithm. For the level of $P_M = 0$, there was no difference between using all variables and using only those selected by VSBD (i.e., the *METH* factor has no effect). However, for the levels $P_M = 4$ and $P_M = 8$, using the VSBD algorithm to select variables prior to running the *K*-means procedure resulted in much better recovery than

Table 4
Analysis of Variance (ANOVA) and Effect Sizes for the Adjusted Rand Index (ARI)

Source	*SS*	*df*	M^2	*F*	*p*	Effect size ($\hat{\eta}^2$)
Corrected model	13.3274	85	0.1568	111.9408	.0000	
Intercept	1233.0122	1	1233.0122	880296.7378	.0000	
No. of objects (*N*)	0.0003	2	0.0002	0.1173	.8893	0.0000
No. of clusters (*K*)	1.4272	2	0.7136	509.4696	.0000	0.0936
No. of true variables (P_T)	3.8864	2	1.9432	1387.3399	.0000	0.2549
Density (*DENS*)	0.1538	2	0.0769	54.9004	.0000	0.0101
Perturbation (*PERT*)	2.3892	2	1.1946	852.8861	.0000	0.1567
No. of masking variables (P_M)	0.6514	2	0.3257	232.5139	.0000	0.0427
Method (*METH*)	1.2685	1	1.2685	905.6540	.0000	0.0832
$N \times K$	0.0026	4	0.0006	0.4585	.7662	0.0002
$N \times P_T$	0.0015	4	0.0004	0.2704	.8971	0.0001
$N \times DENS$	0.0005	4	0.0001	0.0936	.9845	0.0000
$N \times PERT$	0.0041	4	0.0010	0.7315	.5705	0.0003
$N \times P_M$	0.0003	4	0.0001	0.0531	.9947	0.0000
$N \times METH$	0.0014	2	0.0007	0.4956	.6093	0.0001
$K \times P_T$	0.2261	4	0.0565	40.3481	.0000	0.0148
$K \times DENS$	0.0684	4	0.0171	12.2048	.0000	0.0045
$K \times PERT$	0.1216	4	0.0304	21.6998	.0000	0.0080
$K \times P_M$	0.2190	4	0.0548	39.0952	.0000	0.0144
$K \times METH$	0.4143	2	0.2071	147.8820	.0000	0.0272
$P_T \times DENS$	0.0812	4	0.0203	14.4994	.0000	0.0053
$P_T \times PERT$	0.8610	4	0.2152	153.6670	.0000	0.0565
$P_T \times P_M$	0.2303	4	0.0576	41.1126	.0000	0.0151
$P_T \times METH$	0.4766	2	0.2383	170.1447	.0000	0.0313
$DENS \times PERT$	0.0063	4	0.0016	1.1327	.3394	0.0004
$DENS \times P_M$	0.0702	4	0.0176	12.5354	.0000	.0046
$DENS \times METH$	0.1247	2	0.0624	44.5310	.0000	0.0082
$PERT \times P_M$	0.0016	4	0.0004	0.2841	.8884	0.0001
$PERT \times METH$	0.0038	2	0.0019	1.3449	.2609	0.0002
$P_M \times METH$	0.6350	2	0.3175	226.6591	.0000	0.0416
Error	1.9217	1372	0.0014			
Total	1248.2613	1458				
Corrected total	15.2491	1457				

Note. Main effects and two-way interactions ANOVA with ARI as the dependent variable.

did applying *K*-means using all variables (i.e., the *METH* factor had a strong effect at these two levels of P_M).

Table 5 presents the average ARI, as well as the percentage of data sets for which perfect recovery was achieved, for each factor level of the experimental study. To provide a measure of the magnitude of the effect of using variable selection, a paired *t* statistic is reported for each pair of means. All *t* statistics are statistically significant ($p < .01$) for every cell except $P_M = 0$, where the results were identical regardless of whether or not VSBD was used. The principal information from the *t*-value column, however, is the comparison of the relative magnitude of the statistics within each factor. For example, the *t* statistics corresponding to different levels of *N* are quite similar, suggesting that VSBD provides a roughly similar benefit regardless of the number of objects. By contrast, the *t* statistics for different levels of *K* are markedly different. The VSBD algorithm provides increasingly greater benefit for larger values of *K*.

The results indicate that VSBD frequently provided significant improvement in cluster recovery when used to select variables for a *K*-means clustering of binary data sets. In the absence of the VSBD algorithm, the *K*-means algorithm was able to provide perfect recovery for 22.22% of the data sets and resulted in an average ARI of .8901. When used in conjunction with VSBD, the *K*-means algorithm yielded perfect recovery for 33.33% of the data sets and produced an average ARI of .9491. Closer inspection of the results for different levels of data perturbation are especially revealing. Using VSBD prior to the *K*-means algorithm enabled perfect recovery for each of the 243 (100%) error-free data sets, whereas perfect recovery was not realized for 81 (33.33%) of these data sets when VSBD was not used. This shows that VSBD was extremely successful at accomplishing its goal of selecting only those variables that define true cluster structure.

The number of objects in the data set had very little effect

Table 5
Experimental Results Pertaining to Cluster Recovery

Factor and level	Mean adjusted Rand index			% of perfect recoveries	
	All	VSBD	*t*	All	VSBD
No. of objects (*N*)					
2,000	.8887	.9496	10.55	22.22	33.33
3,000	.8904	.9502	10.27	21.81	33.33
4,000	.8912	.9475	10.03	22.63	33.33
No. of clusters (*K*)					
4	.9504	.9691	3.89	32.10	33.33
6	.8872	.9442	10.64	23.46	33.33
8	.8328	.9340	17.68	11.11	33.33
No. of true variables (P_T)					
4	.7929	.9026	14.75	17.28	33.33
6	.9244	.9639	10.12	23.46	33.33
8	.9530	.9808	8.01	25.93	33.33
Density of clusters					
Equal	.9002	.9502	9.25	23.46	33.33
Skewed 1	.9073	.9496	9.57	23.46	33.33
Skewed 2	.8628	.9476	12.52	19.75	33.33
Level of data perturbation (%)					
0	.9365	1.0000	10.11	66.67	100.00
2	.8931	.9499	9.94	0.00	0.00
4	.8408	.8974	11.02	0.00	0.00
No. of masking variables (P_M)					
0	.9495	.9495		33.33	33.33
4	.8621	.9489	14.18	16.87	33.33
8	.8587	.9489	14.31	16.46	33.33
Overall	.8901	.9491	17.83	22.22	33.33

Note. The columns labeled *All* correspond to the use of all variables in the *K*-means analysis, whereas the columns labeled *VSBD* correspond to application of *K*-means after using the variable selection for binary data sets procedure.

on the recovery performance of the *K*-means algorithm, regardless of whether or not VSBD was used to select variables prior to the cluster analysis. I explored the possibility that the range of values of *N* might be too narrow to observe significant differences by replacing the $N = 2{,}000$ factor level results with those for the $N = 200$ level (this yields a 20:1 ratio for the high-to-low levels of *N*). This replacement did not produce any consequential changes in the observed findings, as the number of objects in the data set remained an insignificant factor in the ANOVA results.

Recovery performance generally worsened as the number of clusters increased; however the decrease in recovery was much more pronounced in the absence of VSBD. The average ARIs for the *K*-means algorithm when VSBD was not applied were .9504 and .8328 for $K = 4$ and $K = 8$ clusters, respectively. When the VSBD algorithm was used for variable selection, the average ARI for the *K*-means algorithm dipped only slightly from .9691 at $K = 4$ to .9340 at $K = 8$.

Recovery performance improved as the number of true variables in the data set increased. This finding, which has been observed in previous variable-selection studies (Brusco & Cradit, 2001; Milligan, 1989) is not surprising because each additional true variable further strengthens the cluster structure. The number of masking variables in the data set had a marked effect on the recovery performance of the *K*-means algorithm when VSBD was not used. As expected, *K*-means performed very well in the absence of masking variables, but recovery performance dropped significantly when four or more masking variables were present. When VSBD was used in conjunction with the *K*-means algorithm, recovery performance was very consistent across all levels of the masking variable factor.

Further support for the ability of VSBD to select variables that define true cluster structure, while discarding those that do not, was observed by examining the variable sets obtained by VSBD. For each of the 243 test problems without any masking variables, the VSBD algorithm appropriately selected all clustering variables. For each of the 486 test problems with masking variables, the VSBD algorithm also appropriately selected all of the true clustering variables in the data set. Furthermore, for these same 486 test problems, VSBD properly omitted all masking variables in the data set in 484 cases. For the remaining two data sets, VSBD mistakenly included one masking variable in the selected set.

However, in both these cases, eight true variables were selected and only one masking variable was selected, so the detriment to cluster recovery was very mild.

In sum, VSBD perfectly identified the true set of variables for 727 of the 729 test problems. Thus, the ability of the algorithm to select the proper set of variables appears to be robust to all of the factors considered in this study. Although ARIs of 1 were not achieved for any data sets with levels of data perturbation of 2% or 4% when using VSBD prior to the *K*-means algorithm, these findings should not be misinterpreted as failure of the VSBD algorithm. Instead, these results are almost solely attributable to the failure of the *K*-means algorithm to correctly assign all of the objects to their true cluster memberships because of the noise in the data, despite the fact that VSBD had selected the proper set of variables.

Follow-Up Experiment: Multiple True Cluster Structures

Our experimental study used binary data sets with a single true structure consisting of K clusters as measured on P_T variables. This was necessary to provide an unambiguous assessment of the ability of VSBD to aid in recovering that structure. However, as observed by Brusco and Cradit (2001), it is possible to have multiple true cluster structures in a data set. In these situations, I believe that, like the ARI-based procedure described by Brusco and Cradit, the VSBD algorithm tends to target one of the true cluster structures. This supposition is based on the fact that VSBD, like Brusco and Cradit's variable selection heuristic for *K*-means clustering (VS-KM) algorithm, attempts to find an initial core of variables (Step 2) and then augments those variables with additional ones that work well in conjunction with the core (Step 3).

To evaluate the performance of VSBD in the presence of multiple true cluster structures, a small follow-up study was done. Specifically, data sets with two independent six-cluster structures were produced, with each structure defined by $P_T = 6$ true variables. Cluster density was equal for all data sets. A total of eight data sets was generated by varying three factors at two levels each. The levels of the first factor, the number of objects, were $N = 3{,}000$ and $N = 6{,}000$. The second factor, the number of masking variables, was examined at levels of $P_M = 0$ and $P_M = 6$. Based on these first two factors, the largest data sets in the experiment consisted of 6,000 objects measured on 18 binary variables (6 variables for the first true cluster structure, 6 variables for the second true cluster structure, and 6 masking variables). The levels of the third factor, data perturbation, were 0% and 3%.

The results for the eight test problems in the follow-up experiment are provided in Table 6. When using all of the clustering variables, the *K*-means algorithm always failed to recover either of the true cluster structures. It seemed that the algorithm tended toward one of the true structures but could not completely recover that structure because of the effects of the other true structure in the data set. This is evidenced by the fact that, for each of the eight test problems, the ARI for one of the two structures was approximately .55 to .60, whereas the ARI for the other true structure was approximately .18.

When the VSBD algorithm was used to select a subset of variables prior to implementation of the *K*-means algorithm, perfect (or near-perfect) recovery was always achieved for one of the clusters. This finding is attributable to the iterative nature of subset generation by means of the VSBD algorithm. After a particularly good subset is identified at Step 2, the algorithm avoids inclusion of variables from other structures because of the detriment to the within-cluster sum of squares. If the research analyst is interested in identifying a second true cluster structure in a data set, the VSBD/*K*-means implementation can be repeated after imposing constraints to preclude the same subset of variables from being selected again.

Table 6
Experimental Results for Data Sets with Two True Cluster Structures

Experimental factors			Cluster recovery − Adjusted Rand index			
			K-means only		VSBD + *K*-means	
N	P_M	Data perturbation	Structure 1	Structure 2	Structure 1	Structure 2
3,000	0	0%	.1788	.5898	1.0000	.0000
3,000	0	3%	.1855	.5517	−0.0001	.9215
3,000	6	0%	.1788	.5898	1.0000	.0001
3,000	6	3%	.1848	.5510	−0.0001	.9215
6,000	0	0%	.5905	.1825	1.0000	−.0003
6,000	0	3%	.5519	.1862	−0.0003	.9163
6,000	6	0%	.5905	.1825	1.0000	−.0003
6,000	6	3%	.5509	.1842	−0.0003	.9163

Note. For each of the eight test problems, the adjusted Rand index was computed between observed partitions and each true structure. Two observed partitions were created for each test problem, one using *K*-means only and one using variable selection for binary data sets (VSBD) followed by *K*-means.

Discussion

Summary of Findings

The results reported in this article generally support the premise that *K*-means is an effective method for clustering binary data sets, which is consistent with the results recently reported by Dimitriadou et al. (2002). However, the presence of masking variables in a binary data set can significantly diminish the ability of the *K*-means algorithm to recover the true structure in that data set. I have presented a straightforward algorithm, VSBD, which can be used to select an appropriate subset of variables while omitting the masking variables. By implementing VSBD prior to execution of the *K*-means algorithm, significant improvement in true cluster recovery can be realized. In this experimental study, when all variables were included in the cluster analysis, the *K*-means algorithm provided perfect recovery for only 50% (81 of 162) of the error-free data sets with four or eight masking variables. However, when using only those variables selected by VSBD, perfect recovery was realized for all 162 of these data sets.

Limitations and Extensions

I have provided general guidelines for parameter settings and implementation decisions for the VSBD algorithm; however, these might need to be modified within the context of particular applications. For example, it might be necessary to reduce the value of V_I from 4, perhaps to 3 or even 2, if the number of candidate variables, *P,* is very large. Alternatively, the value of V_I could be set at 4 (or possibly larger) if the complete enumeration of all combinations of four variables were replaced with a branch-and-bound (Narendra & Fukunaga, 1977) or variable-exchange algorithm. Larger or smaller values of φ and δ can also be selected for the algorithm. For example, if there are numerous candidate variables and the quantitative analyst wishes to retain only a very small number of variables for the clustering, the value of δ can be reduced to set a higher standard for inclusion of variables. Finally, I have observed that more or fewer replications of the *K*-means algorithm can be used in Steps 2 and 3 of the VSBD heuristic, as warranted by precision and resource constraints. However, for large binary data sets, I strongly recommend at least several hundred replications in Step 2 and several thousand in Step 3 to avoid poor local minima (see Steinley, 2003).

Milligan (1996, pp. 342–343) has outlined a number of important decisions for an applied cluster analysis: (a) selection of clustering elements; (b) selection of clustering variables; (c) variable standardization; (d) measure of association; (e) clustering method; (f) number of clusters; and (g) interpretation, testing, and replication. Although I have primarily focused on only one piece of this puzzle, I believe that selection of the clustering variables is one of the most challenging and important steps for binary data sets. Variable standardization is really a nonissue for binary data sets because all variables are 0/1 measures. As described earlier in the article, the set-theoretic properties of binary data sets suggest natural measures of association that can be directly mapped to the *K*-means criterion of minimizing the within-cluster sum of squares. Because of its ability to handle large data sets, the *K*-means algorithm is therefore a natural choice for the clustering methodology.

This is not to suggest that *K*-means is the only plausible option for clustering binary data sets. As noted previously, there are a variety of possible indices for the classification of binary data (Anderberg, 1973; Späth, 1980, Chapter 2). The VSBD algorithm is general enough to be implemented with many of these indices; however, the stopping rule in Step 4 would have to be appropriately modified for the selected index. Finally, it must be observed that adaptive learning procedures based on neural network models have received significant attention in the classification literature (Balakrishnan et al., 1994; Bishop, 1995; Leisch et al., 1998; Waller et al., 1998). These methods might also prove to be effective for clustering binary data sets in the presence of masking variables, particularly when asymmetry of the binary information invalidates the use of the within-cluster sum of squares measure. Such circumstances can occur, for example, when two objects matching ones on a variable is much more important than two objects matching zeros on the same variable.

The determination of the appropriate number of clusters is an important decision for binary data sets, and the results provided by Dimitriadou et al. (2002) can help facilitate this decision. On the basis of their results, it seems that an integration of VSBD with the Ratkowsky and Lance (1978) criterion might provide an excellent starting point for the interrelated decisions of variable selection and determination of the number of clusters. Another important direction for future research involves the deployment of VSBD for real-world binary data sets. Such implementations can facilitate examination of interpretation, testing, and replication issues, which were not applicable for the synthetic data sets analyzed in this article.

References

Ambroise, C., & Govaert, G. (1996). Constrained clustering and Kohonen self-organizing maps. *Journal of Classification, 13,* 299–313.

Anderberg, M. R. (1973). *Cluster analysis for applications.* New York: Academic Press.

Arabie, P., & Hubert, L. J. (1992). Combinatorial data analysis. *Annual Review of Psychology, 43,* 169–203.

Arabie, P., & Hubert, L. J. (1996). An overview of combinatorial data analysis. In P. Arabie, L. J. Hubert, & G. De Soete (Eds.), *Clustering and classification* (pp. 5–63). River Edge, NJ: World Scientific Publishing.

Balakrishnan, P. V., Cooper, M. C., Jacob, V. S., & Lewis, P. A.

(1994). A study of the classification capabilities of neural networks using unsupervised learning: A comparison with K-means clustering. *Psychometrika, 59,* 505–525.

Bishop, C. M. (1995). *Neural networks for pattern recognition.* New York: Oxford University Press.

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., & Cradit, J. D. (2001). A variable-selection heuristic for *k*-means clustering. *Psychometrika, 66,* 249–270.

Carmone, F. J., Kara, A., & Maxwell, S. (1999). HINoV: A new model to improve market segmentation by identifying noisy variables. *Journal of Marketing Research, 36,* 501–509.

Cheng, R., & Milligan, G. W. (1996). K-means clustering with influence detection. *Educational and Psychological Measurement, 56,* 833–838.

Cliff, N., McCormick, D. J., Zatkin, J. L., Cudeck, R. A., & Collins, L. M. (1986). BINCLUS: Nonhierarchical clustering of binary data. *Multivariate Behavioral Research, 21,* 201–227.

Curry, D. J. (1976). Some statistical considerations in clustering with binary data. *Multivariate Behavioral Research, 11,* 175–188.

Day, W. H. E. (1996). Complexity theory: An introduction for practitioners of classification. In P. Arabie, L. J. Hubert, & G. De Soete (Eds.), *Clustering and classification* (pp. 199–233). River Edge, NJ: World Scientific Publishing.

De Boeck, P., & Rosenberg, S. (1988). Hierarchical classes: Model and data analysis. *Psychometrika, 53,* 361–381.

DeSarbo, W. S., Carroll, J. D., Clark, L. A., & Green, P. E. (1984). Synthesized clustering: A method for amalgamating different clustering bases with different weighting of variables. *Psychometrika, 49,* 57–78.

De Soete, G. (1986). Optimal variable weighting for ultrametric and additive tree clustering. *Quality and Quantity, 20,* 169–180.

Dimitriadou, E., Dolničar, S., & Weingessel, A. (2002). An examination of indices for determining the number of clusters in binary data sets. *Psychometrika, 67,* 137–160.

Forgy, E. W. (1965). Cluster analyses of multivariate data: Efficiency versus interpretability of classifications. *Biometrics, 21,* 768.

Fowlkes, E. B., Gnanadesikan, R., & Kettenring, J. R. (1988). Variable selection in clustering. *Journal of Classification, 5,* 205–228.

Fowlkes, E. B., & Mallows, C. L. (1983). A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association, 78,* 553–584.

Gnanadesikan, R., Kettenring, J. R., & Tsao, S. L. (1995). Weighting and selection of variables for cluster analysis. *Journal of Classification, 12,* 113–136.

Green, P. E., Carmone, F. J., & Kim, J. (1990). A preliminary study of optimal variable weighting in *K*-means clustering. *Journal of Classification, 7,* 271–285.

Hands, S., & Everitt, B. (1987). A Monte Carlo study of the recovery of cluster structure in binary data by hierarchical clustering techniques. *Multivariate Behavioral Research, 22,* 235–243.

Hartigan, J. A., & Wong, M. A. (1979). Algorithm AS136: A *k*-means clustering program. *Applied Statistics, 28,* 100–128.

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

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

Jensen, R. E. (1969). A dynamic programming algorithm for cluster analysis. *Operations Research, 17,* 1034–1057.

Koontz, W. L. G., Narendra, P. M., & Fukunaga, K. (1975). A branch and bound clustering algorithm. *IEEE Transaction on Computers, C-24,* 908–915.

Leisch, F., Weingessel, A., & Dimitriadou, E. (1998). Competitive learning for binary valued data. In L. Niklasson, M. Bodén, & T. Ziemke (Eds.), *Proceedings of the 8th International Conference on Artificial Neural Networks (ICANN 98)* (pp. 779–784). Berlin: Springer-Verlag.

MacQueen, J. B. (1967). Some methods for classification and analysis of multivariate observations. In L. M. Le Cam & J. Neyman (Eds.), *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability* (Vol. 1, pp. 231–297), Berkeley: University of California Press.

Milligan, G. W. (1980). An examination of the effect of six types of error perturbation on fifteen clustering algorithms. *Psychometrika, 45,* 325–342.

Milligan, G. W. (1981). A Monte Carlo study of thirty internal criterion measures for cluster analysis. *Psychometrika, 46,* 187–199.

Milligan, G. W. (1985). An algorithm for generating artificial test clusters. *Psychometrika, 50,* 123–127.

Milligan, G. W. (1989). A validation study of a variable-weighting algorithm for cluster analysis. *Journal of Classification, 6,* 53–71.

Milligan, G. W. (1996). Clustering validation: Results and implications for applied analyses. In P. Arabie, L. J. Hubert, & G. De Soete (Eds.), *Clustering and classification* (pp. 341–375). River Edge, NJ: World Scientific Publishing.

Milligan, G. W., & Cooper, M. C. (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika, 50,* 159–179.

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.

Milligan, G. W., & Cooper, M. C. (1988). A study of the standardization of variables in cluster analysis. *Journal of Classification, 5,* 181–204.

Milligan, G. W., Soon, S. C., & Sokal, L. M. (1983). The effect of cluster size, dimensionality, and the number of clusters on the recovery of true cluster structure. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 5,* 40–47.

Narendra, P. M., & Fukunaga, K. (1977). A branch and bound algorithm for feature subset selection. *IEEE Transactions on Computers, 26,* 917–922.

Ratkowsky, D. A., & Lance, G. N. (1978). A criterion for determining the number of groups in a classification. *Australian Computer Journal, 10,* 115–117.

Reich, W. A. (2000). Identity structure, narrative accounts, and commitment to a volunteer role. *Journal of Psychology, 134,* 422–434.

Restle, F. (1959). A metric and an ordering on sets. *Psychometrika, 24,* 207–220.

Rosenberg, S. (1989). A study of personality in literary autobiography: An analysis of Thomas Wolfe's *Look Homeward, Angel. Journal of Personality and Social Psychology, 56,* 416–430.

Späth, H. (1980). *Cluster analysis algorithms for data reduction and classification of objects.* New York: Wiley.

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.*

Storms, G., Van Mechelen, I., & De Boeck, P. (1994). Structural analysis of the intension and extension of semantic concepts. *European Journal of Cognitive Psychology, 6,* 43–75.

Waller, N. G., Kaiser, H. A., Illian, J. B., & Manry, M. (1998). A comparison of the classification capabilities of the 1-dimensional Kohonen neural network with two partitioning and three hierarchical cluster analysis algorithms. *Psychometrika, 63,* 5–22.

Williams, G. W., Barton, G. M., White, A. A., & Won, H. (1976). Cluster analysis applied to symptom ratings of psychiatric patients: An evaluation of its predictive ability. *British Journal of Psychiatry, 129,* 178–185.

Appendix

The Steps of the Variable Selection for Binary Data Algorithm

Step 0: Initialize $M = \varphi N$, $Q = \varnothing$, $V = V_I$, K, $Z_B = \infty$, and $Z_{B2} = \infty$.

Step 1: Obtain L by random sampling from the collection of all N objects.

Step 2: For each of the $\binom{P}{V} = \frac{P!}{V!(P-V)!}$ possible subsets of size V that can be formed from the P variables, create the $M \times V$ matrix, $\mathbf{A}$, using the objects from L and corresponding possible subset of variables, which will be denoted as Q', and perform Steps 2a and b.

Step 2a: Perform 500 replications of the *K*-means algorithm on $\mathbf{A}$, and let Z_{min} represent the minimum value of Equation 3 obtained across all replications.

Step 2b: If $Z_{min} < Z_B$, then set $Z_B = Z_{min}$ and store the current subset as the best subset found by setting $Q = Q'$.

Step 3: For each $p \in R \backslash Q$ (i.e., the unselected variables, which correspond to those variables in R but not in Q) create the $M \times (V + 1)$ matrix, $\mathbf{A}$, using the objects from L and variables from $Q \cup \{p\}$ and perform Steps 3a and b.

Step 3a: Perform 5,000 replications of the *K*-means algorithm on $\mathbf{A}$, and let Z_{min} represent the minimum value of Equation 3 obtained across all replications.

Step 3b: If $Z_{min} < Z_{B2}$, then set $Z_{B2} = Z_{min}$ and store the current subset, $Q \cup \{p'\}$, as the best subset found.

Step 4: If $Z_{B2} > Z_B + \delta(M/4)$, then stop. Otherwise, set $Z_B = Z_{B2}$ and $Q = Q \cup \{p'\}$.

Step 5: If $Q = R$, then stop. Otherwise, return to Step 3.

Received September 5, 2003
Revision received June 24, 2004
Accepted July 6, 2004 ■