PSYCHOMETRIKA
2007
DOI: 10.1007/S11336-007-9019-Y

SELECTION OF VARIABLES IN CLUSTER ANALYSIS: AN EMPIRICAL COMPARISON OF EIGHT PROCEDURES

DOUGLAS STEINLEY

UNIVERSITY OF MISSOURI-COLUMBIA

MICHAEL J. BRUSCO

FLORIDA STATE UNIVERSITY

Eight different variable selection techniques for model-based and non-model-based clustering are evaluated across a wide range of cluster structures. It is shown that several methods have difficulties when non-informative variables (i.e., random noise) are included in the model. Furthermore, the distribution of the random noise greatly impacts the performance of nearly all of the variable selection procedures. Overall, a variable selection technique based on a variance-to-range weighting procedure coupled with the largest decreases in within-cluster sums of squares error performed the best. On the other hand, variable selection methods used in conjunction with finite mixture models performed the worst.

Key words: cluster analysis, variable selection.

1. Introduction

It has long been recognized that not all variables contribute equally to defining cluster structure (DeSarbo, Carroll, Clark, & Green, 1984; De Soete, DeSarbo, & Carroll, 1985; Donoghue, 1990; Fowlkes, Gnanadesikan, & Kettenring, 1988; Gnanadesikan, Kettenring, & Tsao, 1995; Green, Carmone, & Kim, 1990; Milligan, 1989; van Buuren & Heiser, 1989), and the inclusion of variables that do not define cluster structure (coined "masking variables" by Fowlkes & Mallows, 1983) can actually degrade the ability of clustering procedures to effectively recover the true cluster structure (Milligan, 1980; 1989). Recently, there has been a virtual well-spring of procedures attempting to determine the subset of variables that define true cluster structures. These procedures have been developed in both the context of model-based clustering (Dy & Brodley, 2004; Law, Figueiredo, & Jain, 2004; Raftery & Dean, 2006) and non-model-based clustering (Brusco & Cradit, 2001; Carmone, Kara, & Maxwell, 1999; Friedman & Meulman, 2004; Montanari & Lizzani, 2001). Excluding the work of Brusco and Cradit (2001) and Carmone et al. (1999), when new procedures are introduced they are normally demonstrated on a few "choice" data sets and comprehensive comparisons are never provided. Unfortunately, introducing new variable selection procedures in this manner results in an entire collection of techniques where there are no definitive recommendations about when to use which procedure. The purpose of the current study is to provide an extensive comparison of recent variable selection techniques across a wide range of conditions.

2. Description of Methods

To describe the eight methods evaluated in the present study, some common notation is adopted:

Requests for reprints should be sent to Douglas Steinley, Department of Psychological Sciences, University of Missouri-Columbia, 210 McAlester Hall, Columbia, MO 65211, USA. E-mail: steinleyd@missouri.edu

$N :=$ the number of objects, indexed $i = 1, \ldots, N$;
$K :=$ the number of clusters, indexed $k = 1, \ldots, K$;
$C_k :=$ the set of objects in the kth cluster;
$N_k :=$ the number of objects in C_k;
$\mathbf{X} :=$ an $N \times V$ data matrix whose elements, x_{iv}, represent the measurement of object i on variable v;
$\alpha_k :=$ the mixing proportion for the kth cluster (or component), where all $\alpha_k \geq 0$ and $\sum_{k=1}^{K} \alpha_k = 1$;
$\theta_k :=$ the set of parameters for the kth component;
$\Sigma_k :=$ the covariance matrix for the kth cluster;
$\mathbf{z} :=$ the $N \times 1$ vector of cluster membership, which are unknown a priori. This can also be referred to as the partition of $\mathbf{X}$;
$\mathbf{I}_A :=$ the $A \times A$ identity matrix;
$\mathbf{J}_A :=$ an $A \times A$ matrix of ones;
$V :=$ the number of candidate clustering variables, indexed $v = 1, \ldots, V$;
$\sigma^2(v) :=$ the variance of the vth variable;
$r(v) :=$ the range of the vth variable;
$\mathcal{V} :=$ the set of indices for the candidate clustering variables $\{1, 2, \ldots, V\}$;
$V_t, V_m :=$ the number of true clustering and masking variables, respectively, $V_t + V_m = V$;
$x_{v^{(m)}} :=$ the set of observations measured only on the masking variables;
$\mathcal{P} :=$ the power set of $\mathcal{V}$, which contains all $Q = 2^V - 1$ feasible subsets of the elements of $\mathcal{V}$ (the null set is excluded) and let $P \in \mathcal{P}$ denote an arbitrary subset of variables from $\mathcal{P}$;
$\mathcal{P}(s) := \mathcal{P}(s) \subset \mathcal{P}$, such that all feasible subsets of cardinality s in $\mathcal{P}$ are contained in $\mathcal{P}(s)$;
$\pi(P) :=$ a partition of the N objects into K clusters $(C_1, \ldots, C_k, \ldots, C_K)$ obtained based on the variables in P; where C_k denotes the objects assigned to cluster k for $1 \leq k \leq K$.

2.1. *Method 1: Feature Saliency* (Law et al., 2004)

Law et al. (2004) estimate feature saliency (e.g., variable weighting in the cluster analysis literature) by embedding the estimation within the EM algorithm procedure commonly used for estimating finite mixture models (see McLachlan & Peel, 2000). Let the population density,

$$f = \sum_{k=1}^{K} \alpha_k f_k, \tag{1}$$

be a mixture of K components, f_k, where f_k are assumed to be of the same parametric family. Law et al. (2004) make the common assumption that all components are Gaussian. Then, let $p(k|x) = \alpha_k f_k(x) / \sum \alpha_k f_k(x)$ represent the posterior probability that object x belongs to cluster k. The mixture model can be formulated as

$$f(x) = \sum \alpha_k f_k(x, \theta_k), \tag{2}$$

where θ_k are relevant parameters for the distribution f_k. Given $x_1, \ldots, x_N$, the maximum likelihood estimates (MLEs) for α_k and θ_k satisfy

$$\sum_{i=1}^{N} p(k|x_i) \mathrm{d}/\mathrm{d}\theta_k \{f_k(x_i, \theta_k)\} = 0, \tag{3}$$

and

$$\alpha_k = \sum_{i=1}^{N} p(k|x_i)/N. \tag{4}$$

This estimation proceeds in steps. First, given $p(k|x_i)$, estimate θ_k (weighting x_i by its probability of belonging to the kth cluster). Second, given the new θ_k, estimate $p(k|x_i)$. Repeat this process until the estimates do not change (or only change by a minimal, preset amount). The EM algorithm (Dempster, Laird, & Rubin, 1977) is the standard method used to fit these models (Bartholomew & Knott, 1999; McLachlan & Basford, 1988; McLachlan & Peel, 2000).

Law et al. (2004) assume that the variables are conditionally independent given the component (e.g., the within-component covariance matrices are chosen to be diagonal), allowing the density to be rewritten as

$$f(x) = \sum_{k=1}^{K} \alpha_k \prod_{v=1}^{V} f(x_v|\theta_{kv}), \tag{5}$$

where $f(\cdot|\theta_{kv})$ is the density function of the vth variable in the kth component. Law et al. (2004) denote the vth variable as irrelevant if its distribution follows a common density, denoted by $q(x_v|\lambda_v)$, across classes (i.e., the distribution is independent of class label). Furthermore, a binary parameter for each feature, ϕ_v, is introduced such that $\phi_v = 1$ if the vth feature contributes to the cluster structure and $\phi_v = 0$ otherwise, allowing (5) to be rewritten as

$$f(x) = \sum_{k=1}^{K} \alpha_k \prod_{v}^{V} [f(x_v|\theta_{kv})]^{\phi_v} [q(x_v|\lambda_v)]^{1-\phi_v}. \tag{6}$$

See Law et al. (2004) for an extensive discussion on how to use the EM algorithm to estimate all of the unknown parameters in (6).

2.2. *Method 2: Model Selection* (Raftery & Dean, 2006)

As in Law et al. (2004), Raftery and Dean (2006) attempt to determine which variables contribute to the overall cluster structure by utilizing the power of finite mixture models. However, unlike Law et al. (2004), the variables are chosen via model comparison instead of having their relevance estimated as part of the EM algorithm. Raftery and Dean (2006) also assume the components are drawn from a multivariate normal mixture, $f(\cdot|\theta_k) = MVN(\cdot|\mu_k, \Sigma_k)$; however, Raftery and Dean implement the unique covariance matrix decomposition given by Banfield and Raftery (1993),

$$\Sigma_k = \lambda_k D_k A_k D_k, \tag{7}$$

where λ_k is the largest eigenvalue of Σ_k, D_k is the matrix of eigenvectors of Σ_k, and A_k is a diagonal matrix with scaled eigenvalues as entries. The three parameters control the volume, orientation, and shape of the kth cluster, respectively. For details on fitting this variant of the finite mixture model, see Banfield and Raftery (1993). Raftery and Dean (2006) divide the data set $\mathbf{X}$ into three distinct parts:

$\mathcal{V}^{(1)}$: the set of already selected clustering variables
$\mathcal{V}^{(2)}$: the variable(s) being considered for inclusion into or exclusion from the set of clustering variables
$\mathcal{V}^{(3)}$: the remaining variables

The decision for inclusion or exclusion of $\mathcal{V}^{(2)}$ from the set of clustering variables is then formulated as comparing two models:

$$\begin{aligned} M^{(1)}: &\quad p(X|\mathbf{z}) = p\big(\mathcal{V}^{(3)}|\mathcal{V}^{(2)}, \mathcal{V}^{(1)}\big)\, p\big(\mathcal{V}^{(2)}|\mathcal{V}^{(1)}\big)\, p\big(\mathcal{V}^{(1)}|\mathbf{z}\big), \\ M^{(2)}: &\quad p(X|\mathbf{z}) = p\big(\mathcal{V}^{(3)}|\mathcal{V}^{(2)}, \mathcal{V}^{(1)}\big)\, p\big(\mathcal{V}^{(2)}, \mathcal{V}^{(1)}|\mathbf{z}\big). \end{aligned} \tag{8}$$

Model $M^{(1)}$ implies that $\mathcal{V}^{(2)}$ provides no additional information about the clustering of the variables, while $M^{(2)}$ implies that $\mathcal{V}^{(2)}$ provides additional information about cluster membership above and beyond $\mathcal{V}^{(1)}$. The two models are compared via an approximation to the Bayes factor (see Raftery & Dean, 2006, for details). The variable selection algorithm proceeds as follows:

1. Select the first variable to be the one which provides the most evidence of univariate clustering.
2. Select the second variable to be the one which shows the most evidence of bivariate clustering when combined with the first.
3. Propose the next variable to be the one which shows the most evidence of multivariate clustering when combined with previously selected variables. This variable is accepted if it exhibits more evidence for clustering than not clustering.
4. Propose the variable for removal from the set of selected variables to be the one which shows least evidence of multivariate clustering, removing it if the variable exhibits less evidence for clustering than not clustering.
5. Repeat Steps 3 and 4 until two consecutive steps have been rejected, then stop.

This method essentially computes a BIC statistic that can be used to compare models within and across different numbers of K. Additionally, Raftery and Dean (2006) identify ten separate decompositions of (7) that can be of interest. Unfortunately, the number of potential model comparisons is now a function of the different values of K, the various covariance matrix decompositions, and the number of variables. For example, consider a data set where $V = 10$ and the range of K of interest is $[1, 5]$. The number of all possible subsets of V variables is $2^{10} - 1 = 1023$ (excluding the empty set), resulting in the number of possible model comparisons being $(1023)(5)(10) = 51{,}150$—making it virtually impossible to investigate all possible models across a wide array of data sets.

Thus, for the purposes of this study, it is assumed K is known for two reasons: (1) to help limit the number of possible models investigated by the above model selection procedure; and (2) to help disentangle the performance of the variable selection strategy from the difficult problem of choosing the number of classes. Furthermore, the finite mixture model that is fit to the data will be allowed to fit an arbitrary covariance matrix that is constrained to be equal across all clusters. This restriction should not impede the evaluation of this variable selection strategy because it corresponds to the most complex scenario under which the clusters are generated (see the section below that describes the data generation process). Assuming the value of K is known and restricting the cluster-covariance components to be of the form $\Sigma_k = \Sigma$ reduces the number of possible model comparisons to $2^V - 1$, potentially a large number of comparisons, but definitely more manageable in a broad simulation study.

2.3. Method 3: Scatter Separability (Dy & Brodley, 2004)

Like Methods 1 and 2, Dy and Brodley (2004) utilize finite mixture modeling of multivariate normal distributions for determining the clustering of observations. Furthermore, three additional terms are defined:

$$S_w = \sum_{k=1}^{K} \alpha_k \Sigma_k, \tag{9}$$

$$S_b = \sum_{k=1}^{K} \alpha_k (\mu_k - M_0)(\mu_k - M_0)', \tag{10}$$

$$M_0 = \sum_{k=1}^{K} \alpha_k \mu_k. \tag{11}$$

Clearly, S_w measures how scattered the observations are from their cluster centroids, while S_b measures how scattered the cluster centroids are from the grand mean. Given two variable subsets, $\mathcal{V}^{(1)}$ and $\mathcal{V}^{(2)}$, we can obtain a partition for each subset, $\mathbf{z}^{(1)}$ and $\mathbf{z}^{(2)}$, respectively. Letting $\mathcal{Q}(\mathcal{V}^{(i)}, \mathbf{z}^{(j)}) = \text{trace}(S_w^{-1} S_b)$, where the jth partition is used on the ith subset of variables to compute the quantities in (9), (10), and (11). For two distinct partitions the values

$$\Upsilon_1 = \mathcal{Q}\big(\mathcal{V}^{(1)}, \mathbf{z}^{(1)}\big)\mathcal{Q}(\mathcal{V}^{(2)}, \mathbf{z}^{(1)}), \tag{12}$$

and

$$\Upsilon_2 = \mathcal{Q}\big(\mathcal{V}^{(2)}, \mathbf{z}^{(2)}\big)\mathcal{Q}\big(\mathcal{V}^{(1)}, \mathbf{z}^{(2)}\big) \tag{13}$$

are computed. If $\Upsilon_1 > \Upsilon_2$ ($\Upsilon_2 > \Upsilon_1$), choose variable subset $\mathcal{V}^{(1)}$ ($\mathcal{V}^{(2)}$). Dy and Brodley (2004) implement a sequential forward search procedure that starts with zero variables and sequentially adds one variable at a time. The variable added is the one that provides the largest criterion value when used in combination with the variables already chosen. The search terminates when adding more variables does not improve the criterion.

2.4. Method 4: COSA (Friedman & Meulman, 2004)

The COSA procedure was originally designed to detect subsets of observations that cluster on subsets of the variables rather than all of them simultaneously. For the task at hand, all of the observations cluster on the same subset of variables, resulting in a special case of the original problem. The COSA algorithm implemented in the present study is:

1. Initialize: $\mathbf{W}_{V\times K} = \{1/V\};\ \eta = \xi$.
2. Begin loop {
3. Compute the pairwise distance between the ith and jth objects on the vth variable by

$$\begin{aligned} D_{ij}^{(\eta)}[\mathbf{W}] &= -\eta \log \sum_{v=1}^{V} w_v e^{-d_{ijv}/\eta}, \\ d_{ijv} &= (x_{iv} - x_{jv})^2 / s_k^2, \\ s_k &= \frac{1}{N^2} \sum_{i=1}^{N} \sum_{j=1}^{N} (x_{iv} - x_{jv})^2, \end{aligned} \tag{14}$$

   where $\{w_v \geq 0\}_1^V$ and $\sum_{v=1}^{V} w_v = 1$.
4. Implement the chosen clustering algorithm on the proximity matrix obtained in Step 3.
5. Compute weights for the vth variable on the kth cluster by

$$\begin{aligned} w_{vk} &= \frac{\exp(-S_{vk}/\xi)}{\sum_{v'=1}^{V} \exp(-S_{v'k}/\xi)}, \\ S_{vk} &= \frac{1}{N_k^2} \sum_{i\in C_k} \sum_{j\in C_k} d_{ijv}, \\ w_v &= \sum_{k=1}^{K} \frac{N_k w_{vk}}{N}. \end{aligned} \tag{15}$$

6. $\eta = \eta + \delta\xi$.
7. } Terminate loop when $\mathbf{W}$ stabilizes.

Since in the present study we assume that all of the clusters are defined by the same subset of variables, $\mathbf{W}$ is transformed from a $V \times K$ matrix to a $V \times 1$ vector by a weighted average (weighted by cluster size) of the within-cluster weights. The clustering algorithm implemented was the average-linkage method, while ξ and δ were initialized as .20 and .10, respectively. It should be noted that the most general case of COSA was chosen for the simulation study. Friedman and Meulman (2004) discuss other applications, such as targeted clustering, that may be more appropriate (not to mention more effective) when domain specific knowledge is taken into account.

2.5. *Method 5: Projection Pursuit* (Montanari & Lizzani, 2001)

Montanari and Lizzani (2001) proposed a variable selection procedure for cluster analysis based on the principles of projection pursuit (Friedman, 1987; Friedman & Tukey, 1974; Kruskal, 1969). The basic premise of this approach is to identify a $1 \times V$ unit-length row vector, $\mathbf{a}$, that projects the data into a $1 \times N$ row vector, $\mathbf{y}(\mathbf{a})$, which has a serious departure from normality as measured by a chi-square statistic. More formally, the projection is

$$\mathbf{y}(\mathbf{a}) = \mathbf{a}\mathbf{X}^{'}, \tag{16}$$

and the goal is to find the best $\mathbf{a}$ for the optimization problem:

$$\begin{aligned} &\text{maximize:} \quad \chi^2\big(\mathbf{y}(\mathbf{a})\big), \\ &\text{subject to:} \quad \mathbf{a}\mathbf{a}^{'} = 1. \end{aligned} \tag{17}$$

To compute the χ^2 statistic, Montanari and Lizzani (2001) recommend the creation of $[\sqrt{N}]$ bins, where $[\cdot]$ presumably denotes the nearest integer to "$\cdot$". The bin points are selected to preserve equal areas under the normal curve for each bin. The number of objects in each bin for the projection $\mathbf{y}(\mathbf{a})$ is obtained, and the χ^2 statistic is computed in the usual manner.

Citing the work of Goffe, Ferrier, and Rogers (1994), Montanari and Lizzani (2001) utilized a simulated annealing procedure for the optimization problem posed by (16) and (17). This procedure, which was originally proposed by Corana, Marchesi, Martini, and Ridella (1987), has proven successful for a variety of continuous-variable optimization problems. Although not mentioned by Montanari and Lizzani (2001), the side constraint imposed by (17) does require some minor modification of the original algorithm. Specifically, whenever any variable is perturbed, the vector is no longer of unit length and must be renormalized prior to evaluating the solution. This does not seem to present a major problem, and our projection pursuit MatLab program tends to produce solutions within a rather narrow range. Montanari and Lizzani (2001) compute importance coefficients ($IC(v)$) based on the projection vector that maximizes (17). The coefficients, $IC(v)$ for $1 \leq v \leq V$ are subsequently compared to a threshold and all variables whose coefficients fall short of the threshold are discarded. The simulated annealing heuristic is then reimplemented using only the remaining variables. This process continues until no variables are discarded.

2.6. *Method 6: HINoV* (Carmone et al., 1999)

The HINoV clustering procedure developed by Carmone et al. (1999) consists of the following steps:

1. Choose K.

2. For all $P \in \mathcal{P}(1)$, run Φ[1] random initializations of the K-means algorithm (see, Steinley, 2006a, for a review) and let $\pi(P)$ denote the partition with the best value of $SSE(\pi(P))$ across the Φ initializations, where

$$SSE(\pi(P)) = \sum_{k=1}^{K} \sum_{i \in C_k} \sum_{p \in P} (x_{ip} - \bar{x}_{kp})^2, \tag{18}$$

and

$$\bar{x}_{kp} = \frac{1}{N_k} \sum_{i \in C_k} x_{ip}. \tag{19}$$

3. Compute Hubert and Arabie's (1985) adjusted Rand index, ARI, between all $V(V-1)/2$ pairs of partitions obtained in Step 2. Denote these values $ARI(u, v) = ARI(v, u)$ for $1 \leq u < v \leq V$, where

$$ARI(u, v) = \frac{\binom{N}{2}(\tau_1 + \tau_2) - [(\tau_1 + \tau_3)(\tau_1 + \tau_4) + (\tau_2 + \tau_3)(\tau_2 + \tau_4)]}{\binom{N}{2}^2 - [(\tau_1 + \tau_3)(\tau_1 + \tau_4) + (\tau_2 + \tau_3)(\tau_2 + \tau_4)]}, \tag{20}$$

where τ_1 is the number of object pairs that are in the same cluster for both π_u and π_v (the partitions for variables u and v, respectively), τ_2 is the number of object pairs that are in different clusters in both π_u and π_v, τ_3 is the number of object pairs that are in the same cluster in π_u but different clusters for π_v, and τ_4 is the number of object pairs that are in different clusters in π_u but in the same cluster for π_v.

4. Compute a total pairwise adjusted Rand index, $TOPRI(v)$, for each variable v by summing across all variables

$$TOPRI(v) = \sum_{u \neq v} ARI(u, v). \tag{21}$$

5. Rank the $TOPRI$ values in descending order, and let $\Psi(s) =$ the candidate variable in position s of this ranking $1 \leq s \leq V$. Choose the first s variables based on the following rule:

$$RR(s) = \frac{(TOPRI(\Psi(s)) - TOPRI(\Psi(s+1)))}{(TOPRI(\Psi(s-1)) - TOPRI(\Psi(s)))} \quad \text{for } 2 \leq v \leq V-1. \tag{22}$$

Although Carmone et al. (1999) recommend ranking the $TOPRI(v)$ values in descending order and using a scree-type plot to determine the cutoff point for selected variables, this procedure is not computationally feasible for a large experimental study. Therefore, we formalized Carmone et al.'s method by using the ratio rule in Step 5. A large value of $RR(s)$ suggests that s is a good candidate for the number of variables, because increasing the number of variables from s to $s+1$ produces a much larger decrease in the $TOPRI$ index than increasing from $s-1$ to s.

2.7. *Method 7: VS-KM* (Brusco & Cradit, 2001)

The VS-KM procedure developed by Brusco and Cradit (2001) consists of the following steps:

1. Choose K. Set $threshold = 0$, $g_{\min} = .05$, $g_{\text{fac}} = .5$.
2. For all $P \in \{\mathcal{P}(1) \cup \mathcal{P}(2)\}$, run Φ random initializations of the K-means algorithm and let $\pi(P)$ denote the partition with the best value of $SSE(\pi(P))$ across all Φ initializations.

[1] Φ was set to 50 for M_6 and M_7, while Φ was set to 20 for M_8.

3. Compute the *ARI* (Hubert & Arabie, 1985) for all $V(V-1)/2$ pairs of partitions for $\mathcal{P}(1)$ obtained in Step 2. Denote these values $ARI(u,v) = ARI(v,u)$ for $1 \leq u < v \leq V$. For all $V(V-1)/2$ partitions for $\mathcal{P}(2)$ obtained in Step 2, let $VAF(u,v)$ denote the *VAF* for the two-variable partition obtained using variables u and v, where

$$VAF(\pi(P)) = \frac{(TSS(P) - SSE(\pi(P)))}{TSS(P)}, \tag{23}$$

and

$$TSS(P) = (N-1)\sigma^2(P). \tag{24}$$

4. Choose the variable pair $\{u^*, v^*\}$ based on the criterion:

$$VAF^* = \left[\max_{(u,v)}\big(VAF(u,v)|ARI(u,v) > threshold\big)\right], \tag{25}$$

where $\{u^*, v^*\}$ correspond to the values of u and v that produce VAF^*. Set $s = 2$, $\eta = ARI(u^*, v^*)$, $\mathcal{P}^*(s) = (u^*, v^*)$, and store $\pi(\mathcal{P}^*(s))$.
5. If $s = 2$, go to Step 6; otherwise, set $P = \mathcal{P}^*(s)$ and run Φ random initializations of the K-means algorithm. Let $\pi(\mathcal{P}^*(s))$ denote the partition with the best value of $SSE(\pi(\mathcal{P}^*(s)))$ across the Φ initializations.
6. For all $v \in V \backslash P^*(s)$, compute the *ARI* between partition $\pi(\mathcal{P}^*(s))$ and the single variable partition for variable v obtained in Step 2, denoting these values as $ARI(\mathcal{P}^*(s), v)$ for $v \in V \backslash P^*(s)$.
7. Set $\lambda = \max_v(ARI(\mathcal{P}^*(s), v))$. If $\lambda < g_{\min}$ or $\lambda < \eta \times g_{\text{fac}}$, then return $\mathcal{P}^*(s)$ as the selected subset of variables and STOP; otherwise, go to Step 8.
8. Set $\eta = \lambda$, $\mathcal{P}^*(s+1) = \mathcal{P}^*(s) \cup \{v'\}$, where $v' = v|ARI(\mathcal{P}^*(s), v) = \lambda$. Set $s = s+1$. If $s = V$, then return $\mathcal{P}^*(s)$ as the selected subset of variables and STOP; otherwise, return to Step 5.

Like the HINoV algorithm, the $ARI(u,v)$ values are computed for all pairs of single-variable partitions in Step 3. In Step 4, the VS-KM algorithm selects a pair of variables that produces the largest *VAF* index, yet also yields a sufficiently large *ARI* index. Although Brusco and Cradit tested a *threshold* of .25, we have found that this frequently resulted in a failure to select any variables, and have obtained better results with *threshold* = 0. Steps 5 through 7 of the VS-KM algorithm attempt to add variables, one at a time. At each iteration, $V - s$ *ARI* values are computed using each of the single-variable partitions for the unselected variables as one partition, and the K-means partition for all currently selected variables as the other partition (Step 5b). The unselected variable whose partition has the strongest agreement with the partition for the current set of selected of variables is added to the selected set provided that two conditions are met: (a) the *ARI* must equal or exceed $g_{\min}$; and (b) the percentage reduction in the *ARI* relative to the last appended variable must be no more than $1 - g_{\text{fac}}$.

2.8. Method 8: Relative Clusterability Weighting with VAF Selection (Steinley & Brusco, 2007)

Steinley and Brusco (2007) introduced a variance-to-range ratio variable weighting method that is used in conjunction with a variable selection algorithm. First, a clusterability index is computed for each variable

$$CI_v = \frac{12 \times \sigma^2(v)}{(r(v))^2}, \tag{26}$$

allowing the relative clusterability of each variable to be computed by

$$RC_v = \frac{CI_v}{\min(CI_v)} \quad \forall\, v = 1, \ldots, V. \tag{27}$$

Thus, the most clusterable variables will have greater values of CI_v. The final weighting is carried out by:

1. Compute CI_v and RC_v for each variable.
2. Transform $\mathbf{X}$ to $\mathbf{X}^*$ by the standard z-score transformation and compute each variable's new range, $r(v^*)$.
3. Complete the transformation procedure by reweighting the variables of $\mathbf{X}^*$ such that the values of RC_v hold in the transformed space. The final transformation of the vth variable is computed by

$$\mathbf{t}_v = \mathbf{x}_v^* \sqrt{\frac{RC_v[r(v_{\min}^*)]^2}{[r(v^*)]^2}}, \tag{28}$$

and then all of the transformed variables are taken together, represented as $\mathbf{T}$.

Furthermore, $\mathbf{T}$ preserves the relative differences in clusterability while placing the values of each variable on a similar scale.

The transformed variables are then used as input into a variable selection program. As a preprocessing step, B standard normal variables with N observations are generated and their respective CI values are computed. The CI values are ordered from $CI_1, CI_2, \ldots, CI_B$ and any variables with CI_v less than $CI_{.95*B}$ are immediately discarded, allowing the Steinley and Brusco (2007) method to handle very large data sets where there is a large number of masking variables.

The culling process based on the CI_v values can be viewed as an initial univariate screening of the variables. This process is subsequently followed by an exhaustive enumeration of all feasible subsets that pass this initial screening. To avoid further notational clutter, we assume that $\mathcal{V}$ (and, accordingly, V) are redefined based on those variables that pass the initial screening. We then apply a variable selection procedure that requires the application of the K-means algorithm (with $\phi = 20$ random initializations) for all feasible subsets of clustering variables. The process is as follows:

1. Choose K. Set $\mathcal{P}^*(s) = \varnothing$ and $VAF^*(s) = 0$ for $1 \leq s \leq V$.
2. For all $P \in \mathcal{P}$ perform Steps 2a and 2b.
   Step 2a. Run Φ random initializations of the K-means algorithm and obtain $SSE(\pi(P))$ and $VAF(\pi(P))$.
   Step 2b. If $VAF(\pi(P)) > VAF^*(s)$ and $|P| = s$, then set $\mathcal{P}^*(s) = P$ and $VAF^*(s) = VAF(\pi(P))$.
   Choose s and $P^*(s)$ based on the following ratio rule:

$$RR(s) = \frac{(VAF^*(s) - VAF^*(s+1))}{(VAF^*(s-1) - VAF^*(s))}. \tag{29}$$

The variable selection algorithm evaluates all feasible subsets of size $1 \leq s \leq V$ in Step 2 and stores the best partition and VAF value for each subset size. The selection of subset size is based on a ratio rule in Step 3. Large values of $RR(s)$ suggest that there is a much greater decrease in VAF obtained from increasing the number of variables from s to $s+1$ than there is from increasing the number of variables from $s-1$ to s. Therefore, we choose the subset of variables, $\mathcal{P}^*(s)$, that produces the maximum value of $RR(s)$.

Although the initial screening process often greatly reduces the number of candidate variables, it is possible that V could remain sufficiently large to preclude exhaustive enumeration of all feasible subsets. In such circumstances, we recommend evaluation of all feasible subsets up to some computationally feasible size, $V' < V$. If it is necessary to select more than V' variables, then we suggest using the subset $\mathcal{P}^*(s = V')$ as a starting point, and incrementally attempt to add variables, one at a time, from that point forward.

3. Simulation Study

3.1. Data Generation

In the simulation study, we wanted to determine the effectiveness of each of the eight procedures described above for selecting the cluster defining variables while simultaneously excluding the masking variables. We generated 20,412 data sets consistent with the OCLUS procedure described by Steinley and Henson (2005), which has been used by a number of previous studies (Steinley, 2003; 2004a; 2006b). The primary advantage of the OCLUS generation method is the ability to generate clusters with known probabilities of overlap. For the present simulation study, each data set was generated with 250 observations and seven factors were systematically manipulated.

The first factor, the number of clusters in the data sets, was examined at three levels, $K = 4$, 6, and 8. The second factor, density (Δ) of the clusters, was tested at three levels: (a) an equal number of objects in each cluster; (b) 10% of the objects; and (c) 60% of the objects in one cluster and the remaining objects equally divided across the remaining clusters. While the third factor, the number of true structure variables, was evaluated at three levels, $V_t = 2$, 4, and 6.

The fourth factor (and likely the most influential across methods—see Steinley, 2003, 2004a, 2006b; Steinley & Henson, 2005; Steinley and McDonald, 2007), the average probability of overlap between clusters on each true structure variable, assumed six levels, $p(O) = 0, .05, .15, .25, .35$, and $.45$. Here cluster overlap is defined as the amount of overlap (probabilistically) existing on each of the V_t dimensions between two clusters. For two clusters (C_k and C_{k^*}) some points from each cluster occupy the same region of space for the vth variable with probability $p_v^{kk^*}$. If p^{kk^*} is the probability of overlap between any pair of clusters across all clusters (i.e., $p^{kk^*} = \sum_{v=1}^{V} p_v^{kk^*}$), then we can define an overall probability of overlap, $p(O)$, as the average probability of overlap between all adjacent clusters. Steinley and Henson (2005) propose first generating each of the V_t dimensions independently to allow for an exact control of cluster overlap. Since each dimension is generated independently, the clusters can be thought of as existing on a continuum for that dimension. Thus, there are $K - 1$ adjacent clusters and $p(O)$ is then defined as $p(O) = 1/(K-1)\sum p^{kk^*}$, where C_k and C_{k^*} are adjacent clusters.

Then, if within-cluster correlation (see below) is desired, Steinley and Henson (2005) prove that a rotation of the clusters does not alter the overlap between the clusters. Each cluster was generated from multivariate normal distributions (as originally proposed by Milligan, 1985). When $p(O) = 0$, the means of each adjacent cluster were separated by six standard deviations on each of the V_t dimensions (creating data sets that have effectively no overlap between clusters). As V_t increases, the clusters become more separated (see Steinley & Henson, 2005, for a proof), creating clusters that were well separated (i.e., internally cohesive and externally isolated, see Cormack, 1971). As $p(O)$ increases, the clusters begin to overlap on each dimension, becoming indistinguishable and more difficult to recover. This type of "noise" in the cluster structure is a more exact manner (again, see Steinley & Henson, 2005, for a proof) to control cluster overlap than adding random noise or outliers as originally proposed by Milligan, 1980).

The fifth factor was the degree of within-cluster correlation present and had two conditions: (a) $\Sigma_k = \mathbf{I}_{V_t}$—the within-cluster correlations were set to zero (i.e., variables are independent given cluster membership is known); and (b) $\Sigma_k = \Sigma$—each cluster has the same covariance matrix (a necessary restriction if exact cluster overlap is to be preserved—see Steinley & Henson, 2005); however, that covariance matrix is arbitrary. The off-diagonal elements in (b) were each chosen from a continuous uniform distribution on the range $[.3, .8]$.

The sixth factor, the number of masking variables, was evaluated at three levels, $V_m = 2$, 4, and 6. The seventh factor, the distribution of the masking variables, $f(V_m)$, was tested at seven levels:

(a) all V_m masking variables were independently generated from an F-distribution with 1 degree of freedom for both the numerator and denominator,
(b) all V_m masking variables were independently generated from a gamma distribution with one degree for both the numerator and denominator,
(c) $x_{v(m)} \sim N_{V_m}(\mathbf{0}, \mathbf{I}_{V_m})$;
(d) $x_{v(m)} \sim N_{V_m}(\mathbf{0}, .25 * \mathbf{J}_{V_m} + .75 * \mathbf{I}_{V_m})$;
(e) $x_{v(m)} \sim N_{V_m}(\mathbf{0}, .50 * \mathbf{J}_{V_m} + .50 * \mathbf{I}_{V_m})$;
(f) $x_{v(m)} \sim N_{V_m}(\mathbf{0}, .75 * \mathbf{J}_{V_m} + .25 * \mathbf{I}_{V_m})$; and
(g) $x_{v(m)} \sim N_{V_m}(\mathbf{0}, \Sigma_{ii} = U(1, 20);\ \Sigma_{ij} = 0)$.

The first two conditions (a) and (b) represent different degrees of skewed masking variables, while conditions (c)–(f) represent masking variables with different degrees of correlation—zero, low, medium, and high, respectively—and (g) indicates that each of the variances of the masking variables were independently drawn from a continuous uniform distribution on the range $[1, 20]$. This resulted in a $3 \times 3 \times 3 \times 6 \times 2 \times 3 \times 7 = 6804$ distinct data scenarios. In addition, three replications were made of each scenario, resulting in 20,412 data sets.

3.2. *Performance Measures*

The performance of each of the eight methods, henceforth referred to as M_1–M_8, was evaluated on all 20,412 data sets. Performance was measured in three different manners:

Recall: The number of relevant variables in the selected subset divided by the total number of relevant variables. Recall will be denoted by $\mathcal{R}$.
Precision: The number of relevant variables in the selected subset divided by the total number of variables selected. Precision will be denoted by $\mathcal{PR}$.
Cluster Recovery: The ability of each procedure to return the true cluster structure based on the subset of variables the procedure selected. The degree of true recovery is measured by *ARI*, which assumes a value of unity when there is perfect recovery of the true cluster structure and a value of zero when recovery is equal to random chance. Steinley (2004b) indicated that the adjusted Rand index should be the measure of choice when evaluating cluster recovery.

Since M_1 and M_4 use weighting schemes for the variables instead of a pure selection procedure that requires a variable to be in or out, some care must be taken in computing the numerator for $\mathcal{R}$ and $\mathcal{PR}$. The fact that the weights, w_v, for the variables are constrained to be in the range [0,1], means that the numerator can be calculated by summing the weights, $\sum_{v=1}^{V} w_v$.

To calculate the *ARI*, cluster membership must be provided by the procedures. M_4 is the only method to independently provide a pure clustering of the objects upon termination of the algorithm. For the three procedures based upon mixture models, M_1–M_3, cluster membership was determined by assigning objects to the clusters in which the posterior probability is highest (see Bartholomew & Knott, 1999). For M_5–M_8, a K-means cluster analysis ($\Phi = 100$) (see Steinley, 2006a, for a complete review) was conducted on the selected variables, and the resultant cluster memberships were used to compute *ARI*.

3.3. *Results*

3.3.1. Overall Performance of Methods. Across methods, the three measures exhibit a moderate level of correlation: $\text{Cor}(\mathcal{R}, \mathcal{PR}) = 0.644$, $\text{Cor}(\mathcal{R}, ARI) = 0.639$, $\text{Cor}(\mathcal{PR}, ARI) = 0.712$ with $p < .0001$ for all correlations. Additionally, the correlations between the recovery of the methods are presented in Table 1. Although some of the correlations are moderate in nature, many of them are much smaller than one would expect, indicating that the eight methods exhibit a great deal of differential performance. None of the correlations are large enough (the largest being $\text{Cor}(M_6, M_8) = .769$) to allow the assertion that one method may serve as a substitution or

TABLE 1.
Correlation matrix for *ARI* of variable selection methods.

	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
M_1	—							
M_2	.212	—						
M_3	.116	.474	—					
M_4	.445	.449	.395	—				
M_5	.456	.512	.483	.559	—			
M_6	.421	.218	.118	.438	.414	—		
M_7	.419	.264	.189	.410	.523	.693	—	
M_8	.326	.182	.074	.367	.383	.769	.711	—

Note: All correlations are significant at $p < .0001$.

TABLE 2.
Mean performance measures for the eight variable selection methods.

Method	$\mathcal{R}$	$\mathcal{PR}$	*ARI*	% Perfect
M_1	.9112	.8127	.6070	6.33
M_2	.3262	.5521	.4334	6.43
M_3	.5063	.6122	.3864	1.73
M_4	.6596	.7432	.7211	23.45
M_5	.7317	.6992	.7082	30.92
M_6	.9264	.9511	.8514	38.25
M_7	.9832*	.8406	.8507	38.41
M_8	.8932	.9636*	.8611*	38.46*

Note: *Best performing method for the particular performance measure.

predictor for another method (for instance, both M_6 and M_8 only account for 59% of the variance in the other).

The mean recall, precision, and recovery for the eight methods are presented in Table 2. Across all performance measures, the top methods were M_6, M_7, and M_8. M_8 exhibits the greatest recovery and precision, while M_7 indicates the greatest level of recall and somewhat lower level of precision than both M_6 and M_8. This relationship indicates that making the most out of each variable selected may be more important than selecting a large number of relevant variables.

The lower level of recall of M_8 may be attributed to the preprocessing nature of the procedure that discards all variables that do not have a high enough relative clusterability index. Due to the random nature of the preprocessing procedure, relevant variables may occasionally be discarded. However, it seems that the trade-off of increased precision results in both better average recovery and more frequent instances where the cluster solution was perfectly recovered (i.e., $ARI = 1.00$).

After examining the overall performance of the methods, it is informative to determine if method performance is dependent upon specific situations (i.e., performance varies with the factor levels). The individual performances are examined by the levels of each factor, beginning with the factors with the most influence and ending with the factors with the least influence. Furthermore, since recovery is usually of paramount importance, we conducted an MANOVA (see Table 3) on the true cluster structure recovery, supplementing findings on the *ARI* with discussions on recall and precision. The between data sets effects can be thought of as the influence of the design factors across all variable selection methods. To simplify the discussion, only main effects are modeled and discussed. Furthermore, given the large sample size, it was expected that

TABLE 3.
MANOVA for eight variable selection methods on *ARI*.

Effect	Source	*DF*	*SS*	*F*	$\hat{\eta}^2$
Between data sets effects					
	$f(V_m)$	6	2276.23	5516.31	.361
	$p(O)$	5	1447.03	4208.16	.229
	V_t	2	1053.11	7656.43	.167
	Σ_k	2	55.99	814.19	.009
	V_m	2	46.64	339.10	.007
	K	2	21.41	155.62	.003
	Δ	2	4.19	30.52	.001
	Error	20391	1402.34		
Within data sets effects (univariate tests)					
	Method (M)	7	5023.68	20145.4	.398
	M$^*f(V_m)$	42	1440.18	962.54	.114
	M$^*\Delta$	14	716.29	1436.21	.056
	M$^*V_t$	14	131.72	264.12	.010
	M$^*p(O)$	35	118.59	95.11	.010
	M$^*V_m$	14	54.88	110.03	.004
	M$^*\Sigma_k$	7	30.02	120.39	.002
	M$^*K$	14	29.49	59.12	.002
	Error	142737	5094.01		

all factors would be statistically significant; therefore, all effects were evaluated with respect to their estimated effect sizes, $\hat{\eta}^2$. Finally, the factors are ordered by decreasing effect size, $\hat{\eta}^2$.

The factor having the most impact on the recovery capabilities of the methods is the distribution of the masking variables (see Table 4). When $f(V_m)$ is an F distribution, M_8 drastically outperforms the other seven selection procedures, indicating that many of the variable selection techniques have difficulty in the presence of skewed data. In fact, the F distribution results in the poorest performance for all other methods, while it is the distribution under which M_8 exhibits the best performance. Furthermore, two of the three mixture modeling techniques (M_2 and M_3) and the projection pursuit technique (M_5) perform at essentially chance levels under the F distribution. However, there is marked a improvement under the other distributions, with M_6–M_8 exhibiting cluster recovery at consistent levels (excluding the high correlation level where M_6 shows some degradation in recovery). Whenever the masking variables arise from a normal distribution, regardless of correlation structure, M_5 also performs quite well. Although still at an adequate level, M_4 seems to perform a step below M_6–M_8 under all distributions except the F distribution, where it performs much worse. Finally, the mixture modeling techniques, M_1–M_3 all perform at a disappointing level, with none of them ever approaching the performance of the other top variable selection procedures.

The effect of the overlap factor lends itself to a very straightforward interpretation: as the probability of overlap between the clusters increases, the recovery capabilities of the procedures decreases. The top three methods, M_6–M_8, were more resilient to increases in cluster overlap than the other five selection procedures. M_1, M_4, and M_5 exhibit a slightly greater decrease in relation to overlap; whereas, M_2 and M_3 perform extremely poorly with regard to increases in cluster overlap. In fact, if the heuristic cut-off value of $ARI = .65$ is considered, the minimum *ARI* for acceptable cluster recover (see Steinley, 2004b), none of the mixture model variable selection techniques provide acceptable cluster recovery, regardless of the amount of overlap;

TABLE 4.
Performance for eight selection methods by $f(V_m)$.

Method	Measure	$F_{1,1}$	$\Gamma_{1,1}$	$N_{\rho_{ij}=0}$	$N_{\rho_{ij}=.25}$	$N_{\rho_{ij}=.50}$	$N_{\rho_{ij}=.75}$	$N_{\sigma^2\sim U(1,20)}$
M_1	$\mathcal{R}$	.6273	.9628	.9762	.9767	.9614	.9005	.9732
	$\mathcal{PR}$	.3990	.6477	.9850	.9817	.9266	.7665	.9821
	ARI	.3317	.6652	.6703	.6678	.6576	.5662	.6901
M_2	$\mathcal{R}$	.1099	.4243	.3981	.3768	.3269	.2568	.3908
	$\mathcal{PR}$	.1758	.7119	.6701	.6387	.5566	.4517	.6594
	ARI	.0246	.4527	.5668	.5455	.4800	.3926	.5715
M_3	$\mathcal{R}$	.0829	.6578	.5616	.5597	.5594	.5614	.5607
	$\mathcal{PR}$	.1228	.8623	.6537	.6567	.6606	.6630	.6662
	ARI	.0507	.4335	.4386	.4447	.4461	.4443	.4464
M_4	$\mathcal{R}$	.3490	.7341	.7610	.7436	.7075	.6121	.7096
	$\mathcal{PR}$	.3238	.8617	.8296	.8271	.8155	.7099	.8343
	ARI	.3432	.8205	.8340	.8186	.7785	.6481	.8044
M_5	$\mathcal{R}$	.0056	.6647	.8905	.8819	.8901	.8937	.8950
	$\mathcal{PR}$	.0075	.6863	.8553	.8559	.8377	.7979	.8531
	ARI	.0044	.6712	.8528	.8573	.8614	.8619	.8481
M_6	$\mathcal{R}$	.9431*	.9441	.9438	.9445	.9335	.8432	.9320
	$\mathcal{PR}$	.9637	.9724*	.9738	.9701*	.9523	.8512	.9743
	ARI	.8043	.8679	.8694	.8714	.8684	.8107	.8671
M_7	$\mathcal{R}$	.8859	1.000*	1.000*	1.000*	1.000*	1.000*	.9969*
	$\mathcal{PR}$	.8821	.8176	.8212	.8203	.8202	.8247	.8977
	ARI	.7156	.8711*	.8718*	.8757*	.8766*	.8756*	.8683*
M_8	$\mathcal{R}$	.9191	.8523	.8916	.8866	.8954	.8920	.9149
	$\mathcal{PR}$	1.000*	.9197	.9640*	.9581	.9620*	.9645*	.9766*
	ARI	.8712*	.8480	.8597	.8619	.8650	.8632	.8589

Note: *Best performing method for the particular performance measure within each distribution.

whereas, M_6–M_8 consistently return average values of *ARI* greater than .65 across all levels of cluster overlap.

The effects of the remaining five factors (number of true variables, number of masking variables, cluster density, and within-cluster correlation) on cluster recovery are displayed in Table 5. M_8 outperforms all the other procedures across all factor levels for the remaining four factors except for two levels: (1) for $K = 8$, M_7 has an average *ARI* .0043 greater than M_8; and (2) for $V_m = 2$, M_6 has an average *ARI* .0020 greater than M_8 (with M_6 and M_7 exhibiting similar degrees of performance as M_8 across the remaining factor levels). Given the stellar performance of M_8, the discussion of recovery, precision, and recall will be restricted to this procedure. In general, as the number of clusters increased, *ARI* decreased. The drop in *ARI* was accompanied by a small decrease in $\mathcal{R}$ (.9643, .8773, .8378); however, $\mathcal{PR}$ (.9518, .9674, .9716) remained fairly constant for $K = 4, 6$, and 8, respectively. This indicates that the variable selection procedure is probably not responsible for the degradation of the *ARI*; whereas, the decrease in *ARI* is more likely attributed to the properties of the K-means clustering algorithm (see Steinley, 2006b, for similar results concerning the K-means clustering algorithm).

Increasing the number of true variables resulted in dramatic increases in *ARI*. Altering the number of true variables had no immediate effect on the precision of M_8 ($\mathcal{PR} \approx .96$ for all V_t); however, as V_t increased, $\mathcal{R}$ decreased ($\mathcal{R} = .9642, .8773$, and .8378 for $V_t = 2, 4$, and 6,

TABLE 5.
Recovery of eight selection methods by number of clusters, number of true and masking variables, and cluster density.

Factor	Level	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8
K	4	.5955	.4753	.3813	.7317	.7372	.8665	.8561	.8853*
	6	.6118	.4339	.3875	.7347	.6995	.8483	.8511	.8577*
	8	.6138	.3911	.3903	.6969	.6879	.8392	.8449*	.8406
V_t	2	.4752	.3562	.3585	.5604	.5856	.7255	.7401	.7517*
	4	.6400	.4492	.4008	.7614	.7328	.8798	.8669	.8840*
	6	.7060	.4949	.3998	.8415	.8062	.9488	.9452	.9477*
V_m	2	.6449	.4723	.4242	.7765	.7035	.8577*	.8469	.8557
	4	.6091	.4269	.3863	.7202	.7099	.8583	.8521	.8627*
	6	.5671	.4010	.3485	.6665	.7111	.8381	.8532	.8651*
Δ	Equal	.6662	.3877	.3007	.7056	.7128	.8868	.8809	.8977*
	10%	.6573	.3523	.2982	.7074	.7060	.8787	.8758	.8874*
	60%	.4976	.5604	.5601	.7503	.7059	.7887	.7954	.7984*
Σ	$\Sigma_k = \mathbf{I}$	.6160	.3944	.3622	.6888	.6950	.8355	.8306	.8487*
	$\Sigma_k = \Sigma$	.5980	.4724	.4104	.7534	.7214	.8672	.8709	.8737*

*Best performing method for the particular performance measure within each level of all factors.

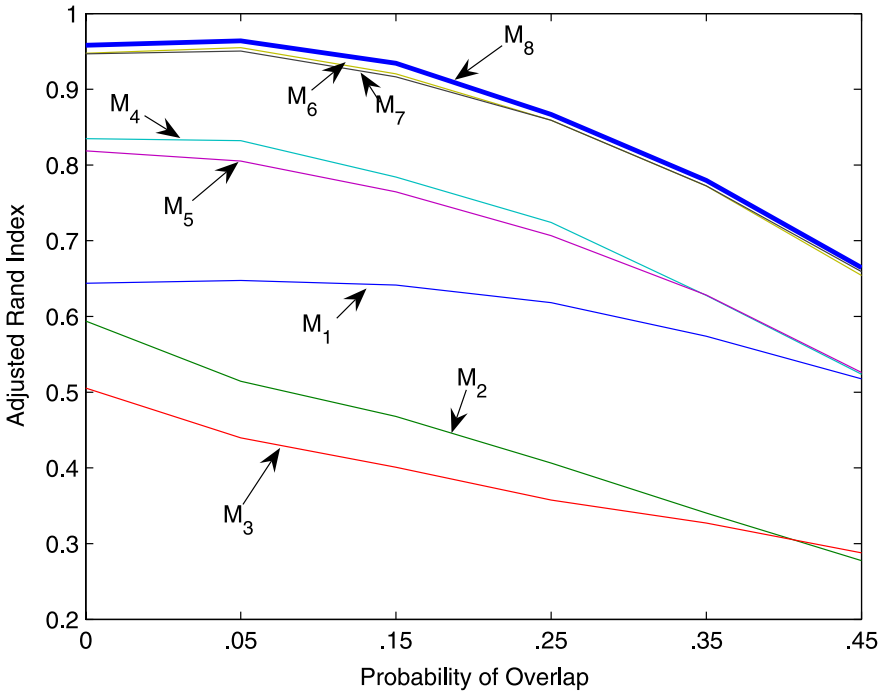

FIGURE 1.
Effect of overlap on variable selection methods.

respectively). This does not indicate any shortcomings on the part of M_8's performance, merely signaling that M_8 selects only the subset of variables with the most clustering information. Once again, this is likely due to the pre-processing procedure based on the *CI* values. Furthermore, the general increase in *ARI* is supported by previous studies as well (see Milligan, 1980; Steinley, 2006b). Altering V_m has no effect on either recall ($\mathcal{R} \approx .89$ for all V_m) or precision ($\mathcal{PR} \approx .96$ for all V_m). Almost counterintuitively, *ARI* increases slightly as V_m increases. This is likely due to greater information available for M_8 to determine what subsets of variables due not contain any clustering information. This will have the largest effect when some subset of masking variables pass the original screening of the *CI* values.

Just as changing the number of masking variables did not have an effect on recall and precision, changing the density of the clusters did not have a noticeable effect on either measure ($\mathcal{R} \approx .89$ and $\mathcal{PR} \approx .96$ for all Δ). The degradation of the *ARI* can be attributed to the difficulty that the K-means procedure has in finding many small clusters in the presence of one large cluster (see Steinley, 2006b, for similar results). However, even the relatively poor performance of M_8 in the context of one large, dominating cluster is better than the performance of the other seven procedures. Finally, it is seen that the within-cluster covariance structure does not have a large effect on the performance of any of the measures (i.e., it only accounts for about 1% of the between methods effects). This is likely due to the fact that the more critical information is the nature of the masking variables, the distance between the clusters, and the amount of true cluster information present. Once again, for M_8, the $\mathcal{R} \approx .96$ for each condition while $\mathcal{PR} \approx .92$ for the uncorrelated variables condition and approximately .87 for the correlated variables condition.

The within data sets effects can be thought of as highlighting the factors that differ the most across the variable selections method. Of course, we see that the most influential effect is which method is employed (refer back to Table 2 for individual mean recovery, recall, and precision values). Given that M_8 was found to perform the best, post-hoc tests between M_8 and all other methods found that M_8 significantly (after using a standard Bonferroni correction for multiple tests) outperformed the other variable selection methods. The effect sizes (in terms of Cohen's d) for these differences range from large (.89, 1.14, and 1.41 for M_1–M_3, respectively), to medium (.44 for both M_4 and M_5), and small (.07 and .06, for M_6 and M_7, respectively). All the within data sets interactions in Table 3 are not particularly interesting and can easily be investigated using Tables 4 and 5. For instance, the interaction between method and masking variable density is understood by observing that for M_8 there are very small mean differences across the masking variable densities (.0232 to be exact), while some of the other procedures have large mean differences across the different factor levels–for example, M_5 ranges from .0044 to .8619. Similar results can be found when investigating the other method by factor interactions.

4. Computation Time

Computation times for each of the methods are provided in Table 6. The times (in seconds) are displayed by the three most influential factors: the number of clusters, the number of true variables, and the number of masking variables. Each cell in Table 6 represents the time to select the variables for one data set where the conditions not displayed were set such that the probability of cluster overlap was zero, the relative density was equal, the masking variables were normally distributed and uncorrelated, and the within-cluster correlations were set to zero.[2] The average time per data set for M_1–M_8 were .2668, 167.83, 445.96, 51.81, 121.60, .0966, .3518, and .2002, respectively. When considering that each method was implemented for 20,412 data sets, the overall

[2]We found that varying these latter four conditions did not noticeably affect the computation time for any of the methods.

TABLE 6.
Computation time (in seconds).

		$K=4$			$K=6$			$K=8$		
		$V_t=2$	$V_t=4$	$V_t=6$	$V_t=2$	$V_t=4$	$V_t=6$	$V_t=2$	$V_t=4$	$V_t=6$
$V_m=2$	M_1	0.11	0.16	0.17	0.17	0.28	0.28	0.25	0.27	0.27
	M_2	26.22	22.25	41.55	51.42	614.86	163.02	148.77	124.09	255.81
	M_3	91.08	154.38	340.14	178.94	334.20	471.78	252.28	445.70	759.47
	M_4	4.69	4.80	4.77	7.05	4.74	4.73	4.83	72.27	110.67
	M_5	66.54	103.78	144.33	69.34	115.17	144.58	58.09	105.59	164.64
	M_6	0.05	0.06	0.08	0.04	0.06	0.09	0.05	0.06	0.09
	M_7	0.06	0.22	0.38	0.08	0.25	0.50	0.09	0.27	0.48
	M_8	0.05	0.09	0.34	0.05	0.09	0.38	0.05	0.11	0.41
$V_m=4$	M_1	0.18	0.25	0.19	0.23	0.25	0.31	0.25	0.31	0.33
	M_2	42.75	63.46	47.62	130.50	257.19	134.41	128.97	137.59	469.97
	M_3	250.73	244.05	280.56	322.80	556.50	634.97	445.91	598.86	645.59
	M_4	113.97	4.77	4.84	98.45	94.72	95.84	16.10	49.58	42.91
	M_5	80.91	67.33	121.49	58.05	142.91	163.19	61.52	131.72	210.71
	M_6	0.08	0.09	0.11	0.08	0.09	0.13	0.08	0.09	0.13
	M_7	0.17	0.33	0.48	0.17	0.31	0.55	0.11	0.50	0.56
	M_8	0.03	0.09	0.70	0.05	0.09	0.38	0.03	0.11	0.41
$V_m=6$	M_1	0.17	0.19	0.30	0.22	0.27	0.38	0.33	0.42	0.67
	M_2	48.95	59.09	51.82	217.06	229.93	278.45	257.27	214.33	314.16
	M_3	242.89	438.42	344.22	383.06	582.14	975.70	605.95	673.56	787.14
	M_4	116.09	4.47	4.89	102.38	100.28	101.45	106.65	52.58	46.44
	M_5	93.14	125.63	144.89	104.23	160.43	223.36	92.06	134.77	194.75
	M_6	0.09	0.13	0.14	0.11	0.11	0.16	0.11	0.14	0.16
	M_7	0.17	0.39	0.64	0.20	0.44	0.61	0.25	0.55	0.73
	M_8	0.03	0.09	0.34	0.03	0.09	0.78	0.06	0.11	0.41

computation times for M_1–M_8 across the entire simulation become approximately 91 minutes, 40 days, 105 days, 12 days, 29 days, 33 minutes, 120 minutes, and 68 minutes, respectively. This results in approximately 186 days of computation time.

All of the programs were written as *m*-files in MATLAB 7 and obtained from the original authors if possible. The program for M_1 was obtained from Law et al. (2004), while the program for M_2 used the *m*-files developed by Martinez and Martinez (2001, chapter 8; 2005, chapter 6). Programs implementing M_7 and M_8 were obtained from Brusco and Cradit (2001) and Steinley and Brusco (2007), respectively. On the other hand, M_3–M_6 were programmed by the authors. It is possible that all programs could be written in different languages or made more efficient in general; however, we would expect the same general results to hold. M_6–M_8 have a computational advantage because the foundation of the variable selection techniques is the K-means algorithm, which is known to be very fast and very efficient (see Steinley, 2006a); whereas, M_2 and M_3 are estimating models that rely on the EM algorithm (which is known to be computationally burdensome, see McLachlan & Krishnan, 1997, chapter 1). Furthermore, M_2, M_3, and M_5 are reestimating these models each time a variable is considered for inclusion or exclusion from the set of clustering variables. This latter reason likely explains why M_1, which also relies on the EM algorithm, is much faster to implement because it estimates the inclusion statistics, ϕ_v, within the context of the model estimation and it only estimates the model once. Finally, it is very reassuring that the best performing methods are also the quickest to converge.

5. Discussion

This paper compares eight contemporary variable selection techniques from several different literatures, including statistics, machine learning, and psychology. The most effective method was the procedure proposed by Steinley and Brusco (2007), while the worst performing methods were those based on finite mixture modeling.

The Steinley and Brusco (2007) procedure (M_8) takes advantage of a new variable weighting technique that determines the relative clusterability of each variable in the system, which proves to be particularly advantageous in the presence of skewed random noise. Closely following the performance of M_8 were the two procedures (M_6 and M_7) based upon finding sets of variables that, when partitioned, produce similar cluster structures. Although fourth best, COSA's (M_4) performance is somewhat remarkable because of the fact that the COSA procedure was not designed for blind variable selection. In fact, it has been shown that COSA performs quite well when used for its intended purpose: finding subsets of objects whose cluster structures are defined by different subsets of variables, preferably with some degree of prior knowledge. The current research illustrates the shortcomings of a possible (perhaps likely) misuse of the COSA algorithm. The projection pursuit variable selection procedure (M_5) performed fifth best and primarily failed due to its inherent nature. Projection pursuit is designed to find the most interesting structures (i.e., non-normal). Subsequently, the procedure's worst performance is observed when the random noise is very non-normal, allowing projection pursuit to denote the non-normal random noise as more "interesting" than the underlying cluster structure. The three worst performing were based on the theory of mixture models for finding the cluster structure. Of those methods, the procedure that builds the variable weighting scheme into the EM algorithm (M_1) performs the best. The final two methods, M_2 and M_3, performed drastically worse than the other six methods.

Although the comparative performances of each of the methods is quite interesting, the major limitation to the current research is the fact that the number of clusters is assumed to be known. The number of clusters is very rarely known in advance and must be determined by the researcher. One of the most popular procedures for choosing the number of clusters when implementing finite mixture models is using the BIC within the larger context of model comparison (see Raftery & Dean, 2006). One promising statistic for choosing the number of clusters in K-means clustering is the gap-statistic (Tibshirani, Walther, & Hastie, 2001). However, these are only two among many methods for choosing the number of clusters (see McLachlan & Peel, 2000, chapter 6, for additional procedures in finite mixture modeling; see Steinley, 2006a, for numerous methods for choosing the number of clusters in the context of K-means clustering). Unfortunately, extensive comparisons have not been conducted, and including the decision of the number of clusters within the simulation study would have likely confounded the results obtained and made the inferences about the variable selection procedures much more difficult. Regardless of the method used to choose the number of clusters, any such method will be prone to error, and this error will result in the precision, recall, and *ARI* reported herein to be inflated (i.e., the true values of the performance measures are likely to be lower). However, given the degree that M_6, M_7, and M_8 outperform the other methods, it is quite unlikely that the worst performing methods will become the best performing methods (and vice versa) when choosing the number of clusters is incorporated into the decision making process.

Lastly, the present study opens several avenues of future research that are currently underway. First, when the number of variables dramatically increases (say more than 100), many of the procedures will become increasingly more inefficient. We suspect M_8 will continue to perform well in these situations due to the preprocessing component that ranks the variables in terms of clusterability. Second, it would be worthwhile to investigate the dependence of the recovery capability of the more successful methods on the objective function and the algorithm. In the

former instance, it is possible that minimizing a different criterion other than *SSE* (such as $|\mathbf{W}|$, $\text{tr}(\mathbf{W}^{-1}\mathbf{B})$, etc.) may be more effective. In the latter instance, the traditional K-means algorithm is simple and efficient; however, more sophisticated algorithms (such as simulated annealing, genetic algorithms, tabu search, etc.) may be more effective in minimizing *SSE* or an alternative criterion. Finally, the results indicate that, excluding the F-distribution, M_7 outperforms all of the other methods. Thus, to maximize recovery, it seems that some combination of the elements of M_8 and M_7 is called for. An effective combination is currently being pursued.

References

Banfield, J.D., & Raftery, A.E. (1993). Model-based Gaussian and non-Gaussian clustering. *Biometrics*, *49*, 803–821.

Bartholomew, D.J., & Knott, M. (1999). *Latent variable models and factor analysis*. London: Arnold.

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 segment definition by identifying noisy variables. *Journal of Marketing Research*, *36*, 501–509.

Cormack, R.M. (1971). A review of classification. *Journal of the Royal Statistical Society, Series A*, *134*, 321–367.

Dempster, A.P., Laird, N.M., & Rubin, D.B. (1977). Maximum likelihood from incomplete data via the E-M algorithm. *Journal of the Royal Statistical Society, Series B*, *39*, 1–38.

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

De Soete, G., DeSarbo, W.S., & Carroll, J.D. (1985). Optimal variable weighting for hierarchical clustering: An alternative least-squares algorithm. *Journal of Classification*, *2*, 173–192.

Donoghue, J.R. (1990). Univariate screening measures for cluster analysis. *Multivariate Behavioral Research*, *30*, 385–427.

Dy, J.G., & Brodley, C.E. (2004). Feature selection for unsupervised learning. *Journal of Machine Learning Research*, *5*, 845–889.

Fowlkes, E.B., & Mallows, C.L. (1983). A method for comparing two hierarchical clusterings. *Journal of the American Statistical Association*, *78*, 553–569.

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

Friedman, J.H. (1987). Exploratory projection pursuit. *Journal of the American Statistical Association*, *82*, 249–266.

Friedman, J.H., & Meulman, J.J. (2004). Clustering objects on subsets of variables. *Journal of the Royal Statistical Society, Series B*, *66*, 1–25.

Friedman, J.H., & Tukey, J.W. (1974). A projection pursuit algorithm for exploratory data analysis. *IEEE Transactions on Computing*, *23*, 881–890.

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

Goffe, W.L., Ferrier, G.D., & Rogers, J. (1994). Global optimization of statistical functions with simulated annealing. *Journal of Econometrics*, *60*, 65–99.

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.

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

Kruskal, J.B. (1969). Toward a practical method which helps uncover the structure of a set of observations by finding the line transformation which optimizes a new index of condensation. In R.C. Milton, & J.A. Nelder (Eds.), *Statistical Computation* (pp. 427–440). New York: Academic Press.

Law, M.H.C., Figueiredo, M.A.T., & Jain, A.K. (2004). Simultaneous feature selection and clustering using mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*, 1154–1166.

Martinez, W.L., & Martinez, A.R. (2001). *Computational statistics handbook with MATLAB*. Boca Raton: Chapman & Hall.

Martinez, W.L., & Martinez, A.R. (2005). *Exploratory data analysis with MATLAB*. Boca Raton: Chapman & Hall.

McLachlan, G.J., & Basford, K.E. (1988). *Mixture models: Inference and applications to clustering*. New York: Dekker.

McLachlan, G.J., & Krishnan, T. (1997). *The EM algorithm and extensions*. New York: Wiley.

McLachlan, G.J., & Peel, D. (2000). *Finite mixture models*. New York: Wiley.

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. (1985). An algorithm for generating artificial test clusters. *Psychometrika*, *50*, 23–127.

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

Montanari, A., & Lizzani, L. (2001). A projection pursuit approach to variable selection. *Computational Statistics & Data Analysis*, *35*, 463–473.

Raftery, A.E., & Dean, N. (2006). Variable selection for model-based clustering. *Journal of the American Statistical Association*, *101*, 168–178.

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

Steinley, D. (2004a). Standardizing variables in K-means clustering. In D. Banks, L. House, F.R. McMorris, P. Arabie, & W. Gaul (Eds.), *Classification, clustering, and data mining applications* (pp. 53–60). New York: Springer.

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

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

Steinley, D. (2006b). Profiling local optima in K-means clustering: Developing a diagnostic technique. *Psychological Methods*, *11*, 178–192.

Steinley, D., & Brusco, M.J. (2007, in press). A new variable weighting and selection procedure for K-means cluster analysis. *Psychometrika*.

Steinley, D., & Henson, R. (2005). OCLUS: An analytic method for generating clusters with known overlap. *Journal of Classification*, *22*, 221–250.

Steinley, D., & McDonald, R.P. (2007). Examining factor score distributions to determine the nature of latent spaces. *Multivariate Behavioral Research*, *42*, 133–156.

Tibshirani, R., Walther, G., & Hastie, T. (2001). Estimating the number of clusters in a data set via the gap statistic. *Journal of the Royal Statistical Society, Series B*, *63*, 411–423.

van Buuren, S.V., & Heiser, W.J. (1989). Clustering N objects into K groups under an optimal scaling of variables. *Psychometrika*, *54*, 699–706.

Manuscript received 6 JUL 2005
Final version received 3 JUL 2006