Journal of Classification 24:99-121 (2007)
DOI: 10.1007/s00357-007-0003-0

Initializing K-means Batch Clustering: A Critical Evaluation of Several Techniques

Douglas Steinley

University of Missouri-Columbia

Michael J. Brusco

University of Florida

Abstract: K-means clustering is arguably the most popular technique for partitioning data. Unfortunately, K-means suffers from the well-known problem of locally optimal solutions. Furthermore, the final partition is dependent upon the initial configuration, making the choice of starting partitions all the more important. This paper evaluates 12 procedures proposed in the literature and provides recommendations for best practices.

Key Words: K-means clustering; Initializations.

1. Introduction

K-means clustering (MacQueen 1967; Hartigan 1975, Chapter 4) is a nonhierarchical clustering procedure that has become the most common technique for partitioning a dataset such that the sum of the within-cluster variances are minimized (see Steinley 2006a, for a comprehensive review). We briefly describe a typical K-means algorithm, indicate several options for initializing the algorithm, compare the procedures, and make several recommendations.

The first author was partially supported by Office of Naval Research Grant #000014-02-1-0877.

Corresponding Author's Address: Department of Psychological Sciences, University of Missouri-Columbia, 210 McAlester Hall, Columbia, Missouri 65211 USA; e-mail: steinleyd@missouri.edu

The K-means method is designed to partition N objects (each having measurements on P variables) into K classes ($C_1, C_2, \ldots, C_K$), where C_k is the set of n_k objects in the k^{th} cluster, and K is given. If $\mathbf{X}_{N\times P} = \{x_{ij}\}_{N\times P}$ denotes the $N \times P$ data matrix, the K-means method constructs these partitions so that the squared Euclidean distance between the row vector for any object and the centroid vector of its respective cluster is no larger than the distances to the centroids of the remaining clusters. The centroid of cluster C_k is a point in P–dimensional space found by averaging the values on each variable over the objects within the cluster. For instance, the centroid value for the j^{th} variable in cluster C_k is

$$\bar{x}_j^{(k)} = \frac{1}{n_k} \sum_{i \in C_k} x_{ij}, \tag{1}$$

and the complete centroid vector for cluster C_k is given by

$$\bar{\mathbf{x}}^{(k)} = (\bar{x}_1^{(k)}, \bar{x}_2^{(k)}, \ldots, \bar{x}_P^{(k)})'. \tag{2}$$

A typical K-means algorithm would operate by the following iterative procedure:

1. K initial seeds are defined by K P-dimensional vectors, $\mathbf{s}^{(k)}$ for $1 \leq k \leq K$.
2. Based on the initial seeds, the squared Euclidean distance, $d^2(x_i, s^{(k)})$, between the i^{th} object and the k^{th} seed vector is obtained. Each object is allocated to the cluster with the minimum squared Euclidean distance to its defining seed.
3. Once all objects have been initially allocated, cluster centroids are calculated as in (2) and the initial seeds are replaced.
4. Objects are compared to each centroid by

$$d^2(x_i, \bar{\mathbf{x}}^{(k)}) = \sum_{j=1}^{P} (x_{ij} - \bar{x}_j^{(k)})^2 \ , \tag{3}$$

   and they are allocated to the cluster whose centroid is closest.
5. New centroids are calculated with the updated cluster membership.
6. Steps 4 and 5 are repeated until no objects can be reallocated to different clusters.

When attempting to find a 'good' partitioning of a dataset through the iterative method just described, we are also trying to minimize the sum of squares error:

$$SSE = \sum_{j=1}^{P} \sum_{k=1}^{K} \sum_{i \in C_k} (x_{ij} - \bar{x}_j^{(k)})^2. \tag{4}$$

If there is an object within C_k such that

$$\frac{n_k}{n_k - 1} d^2(x_i, \bar{\mathbf{x}}^{(k)}) > \frac{n_{k'}}{n_{k'} + 1} d^2(x_i, \bar{\mathbf{x}}^{(k')}) \,, \tag{5}$$

then move the i^{th} object from C_k to cluster $C_{k'}$, and the SE is reduced (see Späth 1980, p. 72). Brusco and Steinley (2005) found this two-step procedure to be quite effective for minimizing (4).

It has long been known that the K-means method is only guaranteed to converge to a locally optimal solution. In fact, Steinley (2003) indicated that, even for a modestly sized dataset ($N = 200, P = 8$), the number of unique local optima could be in the thousands. Therefore, Steinley recommends using several thousand random initializations and choosing the solution that corresponds to the minimum value of SE. Likewise, Brusco (2004) conducted experiments using 10,000 random initializations. Recently, Steinley (2006b) found that the nature of the locally optimal solutions provide insight into the cluster structure of the underlying dataset. Specifically, the number of unique local optima is directly related to the quality of the final partition.

However, the process of repeatedly initializing the same algorithm thousands of times is viewed as an inelegant, brute-force method of arriving at a good, but possibly sub-optimal solution. Thus, several researchers have tried to develop so-called "intelligent" starting configurations (or seed points) to increase the likelihood of arriving at a good final partitioning of the objects. Unfortunately, the performance of these procedures has not been compared or evaluated, disallowing recommendations for the general researcher. The following section introduces several strategies for initializing the K-means algorithm described above (i.e., how are the seed vectors chosen?). Each strategy is then compared in a broad Monte Carlo study, focusing both on the minimization of (4) and the best recovery of the true cluster structure.

2. Initialization Strategies

2.1 I_1: Astrahan 1970

One of the earliest initialization procedures was proposed by Astrahan (1970) and is based on nearest neighbor densities of an object.

1. Define a distance, d_1. A reasonable choice is the average pairwise Euclidean distance
$$d_1 = \frac{1}{n(n-1)} \sum_{i=1}^{P-1} \sum_{j=i+1}^{P} d(x_i, x_j). \tag{6}$$
2. For each data point, x_i, compute the number of other data points that are within d_1 of x_i. This is analogous to counting the number of points that

fall within a circle of radius d_1 centered at x_i. The final number of points in the circle is referred to as the density. The data point with the highest density is chosen as the first cluster seed.

3. The remaining $K-1$ seeds are chosen by decreasing density, as long as they are at least another pre-specified distance, d_2, from all the seeds that have already been chosen. It is acceptable to set $d_2 = d_1$.

2.2 I_2: Bradley and Fayyad 1998

Bradley and Fayyad (1998) proposed a bootstrap like procedure for determining the initial seeds:

1. First, choose S random sub-samples of the original data matrix, $\mathbf{X}$, denoted as $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_S$. Bradley and Fayyad (1998) recommend $S = 10$, where each of the 10 sub-samples contained one-tenth of the number of observations as $\mathbf{X}$.
2. Cluster each of the S sub-samples with the K-means algorithm (where a random starting point is used), obtaining S centroid matrices, where the i^{th} centroid matrix is $\mathbf{C}^{(i)}_{K\times P}$ (each row represents a centroid for one of the K clusters). Create the new dataset,

$$\mathbf{X}^*_{SK\times P} = \begin{bmatrix} \mathbf{C}^{(1)} \\ \mathbf{C}^{(2)} \\ \vdots \\ \mathbf{C}^{(s)} \end{bmatrix}.$$

3. Cluster $\mathbf{X}^*$ S times with the K-means algorithm, using each of the $\mathbf{C}^{(i)}$ as starting configurations. The centroids, $\mathbf{C}^*$ corresponding to the clustering of $\mathbf{X}^*$ with the starting point $\mathbf{C}^{(i)}$ that led to the smallest value of SSE are retained as the initial seed configuration.

Bradley and Fayyad (1998) argue that this leads to a refined starting seed that is not corrupted by outliers or other influential data points.

2.3 I_3: Faber 1994

The method to initialize K-means proposed by Faber (1994) is to choose a random sample of K data points from the whole population of data points. The main argument behind this procedure is that the initial seeds will be actual data points instead of a set of arbitrarily chosen data points contained in the data space. Since the initial seeds are data points, denser areas of the data are more likely to be represented by starting seeds.

2.4 I_4: Hand and Krzanowski 2005

Hand and Krzanowski (2005) proposed a procedure that refines the final solution of the K-means algorithm. Specifically:

1. Conduct a K-means clustering from a given starting point and note the resulting partition, $\mathcal{P}$, as well as its associated value of SSE. For this implementation of the K-means algorithm, a random starting configuration is used.
2. Set $i = 1$, $\alpha = .3$, and $\beta = .95$.
3. Perturb the solution from Step 1 by moving each object to a different cluster (randomly chosen) with probability α. Repeat the K-means clustering, noting the configuration, $\mathcal{P}_i$ and criterion value SSE_i.
4. Set $i = i + 1$ and replace α by $\alpha\beta$. If $SSE_i < SSE$ then $SSE = SSE_i$ and $\mathcal{P} = \mathcal{P}_i$. Stop if $i = 100$.

Note, that at the termination of Step 4, the partitioning of the objects is complete (i.e., the seed selection is embedded within the procedure). Hand and Krzanowski (2005) also recommend terminating the algorithm if 10 successive SSE_i values are the same; however, we will implement the full 100 iterations to give the above procedure more opportunities to find a better final partition.

2.5 I_5 : SPSS 2003

SPSS (1999, pp. 308–315) uses the first K observations in $\mathbf{X}$ as the initial seeds. If the data are randomly entered into SPSS, then this procedure is the same as I_3. However, if the data are not randomly ordered, systematic bias can be introduced into the choice of initial cluster seeds.

2.6 I_6: Milligan 1980

Milligan (1980) recommended using the solution from Ward's (1963) hierarchical cluster analysis as a starting configuration for the K-means procedure, which has been widely supported in the literature (Arabie and Hubert 1992, 1994; Huberty, DiStefano and Kamphaus 1997; Milligan 1980, 1996; Milligan and Sokol 1980; Punj and Stewart 1983; Waller, Kaiser, Illian, and Manry 1998).

2.7 I_7: Mirkin 2005

Mirkin (2005, p. 88) indicated that I_7 (the MaxMin procedure) has proved to work well in real and simulated datasets. The procedure is implemented as follows:

1. For all pairs of objects, x_i and x_j, find the maximum value of $d^2(x_i, x_j)$. The pair with the maximum distance between them are chosen as the first pair of cluster seeds, $\mathbf{s}_1$ and $\mathbf{s}_2$.
2. Let K^* be the number of seeds chosen thus far. For each of the remaining objects, x_i that have not been chosen as cluster seeds, calculate $d^2(x_i, \mathbf{s}^{(k^*)})$ ($k^* = 1, \ldots, K^*$). For each object, find the minimum value of $d^2(x_i, \mathbf{s}^{(k^*)})$, denoted as $d^2_m(x_i)$.
3. The next cluster seed, s_{K^*+1} is the object such that $d^2_m(x_i)$ is a maximum. If $K^* < K$, then $K^* = K^* + 1$, and return to Step 2. If $K^* = K$, the process is terminated.

2.8 I_8: Mirkin 2005

Mirkin (2005, p. 93) proposed a procedure referred to as Intelligent K-means. The "intelligence" is due to how the cluster seeds are chosen.

1. Set $k = 1$ and I_k is the original set of objects.

2a. In the original description of the procedure, Mirkin (2005) suggested shifting the variables to have mean zero and standardizing by the range. However, this step is excluded in the present treatment as not to confuse the underlying mechanism driving the recovery of clusters (i.e., standardization or initialization). Instead, the variables are only shifted to have mean of zero.

2b. Using all objects in I_k, put a candidate seed point, s, as the point that is most distant from the origin.

2c. Compare the distance between each object, x_i, with s and with the origin. Assign x_i to C_k (the cluster associated with seed point s) if $d^2(x_i, s) < d^2(x_i, 0)$, where $d^2(x_i, 0)$ is the distance of x_i from the origin.

2d. Compute the centroid of C_k, $\bar{\mathbf{x}}^{(k)}$, and determine if it is different than s. If s and $\bar{\mathbf{x}}^{(k)}$ differ, then $s = \bar{\mathbf{x}}^{(k)}$ and return to Step 2c. Otherwise, go to Step 2e.

2e. Output C_k and $\bar{\mathbf{x}}^{(k)}$ as the most deviant cluster.

3. Set $k = k + 1$. If $k < K$, put $I_k = I_k - C_k$, and return to Step 2b. If $k = K$, go to Step 4.
4. Conduct K-means with initial seeds equal to $\bar{\mathbf{x}}^{(k)}$ ($k = 1, \ldots, K$).

For both I_7 and I_8, Mirkin (2005, Chapter 3) has other alternative stopping criteria. Namely, (a) the number of clusters is chosen when a pre-specified amount of the variance is accounted for – similar to a procedure for choosing the number of components in principal components analysis, and (b) the contribution of additional clusters becomes too small. Throughout the simulation to follow, the number of clusters is known. Furthermore, it is assumed that all objects are generated from a known cluster solution, thus, we have chosen to

terminate the procedure when K cluster seeds are chosen. This helps place all of the procedures on equal footing.

2.9 I_9: Su and Dy 2004

Su and Dy (2004) introduced a technique that is based on the iterative application of principal components. The procedure proceeds as:

1. Starting from a single cluster, divide it into two sub-clusters.
2. Choose the sub-cluster with the largest within-cluster variance as the next cluster to divide.
3. At each split, for the selected cluster, C_k, we first project $x_i \in C_k$ onto the first principal direction of $x_i \in C_k$, as defined by the first principal component (where first refers to the principal component associated with the largest eigenvalue). The transformed version of x_i on the first principal component is denoted as y_i.
4. Divide C_k into two sub-clusters $C_k^{(1)}$ and $C_k^{(2)}$ according to the rule: If $y_i \leq \bar{y}$, assign x_i to $C_k^{(1)}$; otherwise, assign x_i to $C_k^{(2)}$.
5. Repeat steps 2 through 4 until K clusters are found.

2.10 I_{10}: SAS 2004

The procedure implemented in SAS is a simpler version of I_1.

1. Define a distance, d_1. A reasonable choice is the average pairwise Euclidean distance computed in Equation (7).
2. Set $k = 1$. The first cluster seed is chosen as the first observation in $\mathbf{X}$.
3. If $k = K$, then stop.
4. If $k < K$, the $k + 1$ seed is chosen as the first observation that is at least distance d_1 from each of the previous k seeds. Set $k = k + 1$ and return to Step 3. Furthermore, it is indicated that a good value for d_1 must be chosen; however, no recommendations are given.

2.11 I_{11}: Likas, Vlassis, and Verbeek 2003

Likas, Vlassis, and Verbeek (2003) introduced what they termed as the "global K-means algorithm".

1. Set $k = 1$. Compute the grand mean.
2. Set $k = k + 1$. Using the K-means clustering algorithm, cluster the data into k clusters N different times ($i = 1, \ldots, N$). For the i^{th} clustering initialize the K-means algorithm with the following k centroids: (a) the centroids obtained from the $k - 1$ clustering and (b) observation x_i.

3. Choose the final solution for k to be the clustering that returns a minimum for (5). If $k = K$, stop; otherwise return to Step 2.

Likas et al. (2003) touted this procedure, indicating that it was both deterministic and does not rely on any initial conditions or empirically adjustable parameters. Furthermore, after several experiments and applications, Likas et al. (2003) claim that this procedure is "...experimentally optimal". Steinley (2006a) seriously questioned this claim, while Hansen, Ngai, Cheung, and Mladenovic (2005) proved that the global K-means procedure cannot be guaranteed to find the optimal solution.

2.12 I_{12}: Steinley 2003

The following initialization procedure is based on recommendations by Steinley (2003).

1. Randomly divide the data into K clusters, where observation x_i has equal probability of being placed in any one cluster.
2. Compute the initial cluster seeds based on the random division in Step 1.
3. Repeat multiple times, choosing the partition corresponding to the minimum value of (5).

Steinley (2003) recommended conducting the above procedure upwards of 5,000 times; however, in the present context, that would give I_{12} an unfair advantage. Instead, we chose to repeat the above initialization for a reasonable period of time. For this study, the final solution was the best solution found within 20 seconds.

3. Methods: Simulation I

The first simulation provides a focused study that describes the performance of the various initialization techniques across a range of data structures. The general procedure for evaluating cluster analytic methods in the studies mentioned above is to generate data with a known cluster structure, apply the clustering technique of interest to the data, and compare the recovered cluster structure with the known, true cluster structure. To understand the general robustness of the K-means method, it is necessary to examine its performance on several data structures. To provide a wide breadth of data and increase the generalizability of the present study, the following factors were varied:

1. Number of clusters.
2. Number of variables.
3. Distribution of variables.

4. Number of observations.
5. Relative cluster density.
6. Type of multidimensional overlap between clusters.
7. Probability clusters will overlap.

The levels of the factors and each of the factors are described in turn.

3.1 Description of Factors

3.1.1 Number of Clusters

The number of clusters, K, assumed the values from $K = 4, 6,$ and 8.

3.1.2 Number of Variables

The number of variables, P, assumed the values $P = 4, 6, 8,$ and 10.

3.1.3 Distribution of Variables

The generation procedure developed by Steinley and Henson (2005) allows for the generation of clusters from several different distributional families. For this study, the observations were drawn from five different distributional families: normal with equal variances, normal with unequal variances, triangular distributions, uniform distributions, and a mixed distribution. The normal with equal variances generates variables with a covariance matrix proportional to the identity. The normal with unequal variances initially generates data with a diagonal covariance matrix (with no restrictions on the values of the variances), but through arbitrary rotations different correlation structures are achieved. The triangular distribution generates data from a discrete triangular distribution that results in skewed data. The uniform distribution generates data from a continuous uniform distribution, while the mixed condition randomly selects one of the four aforementioned distributions for each of the variables – resulting in datasets where each variable may be drawn from a different distribution, creating more realistic datasets. It is suspected, from previous studies (see Steinley 2006b, for a recent exploration in the effect of distributional assumptions on the performance of K-means), that the K-means clustering algorithm will perform optimally when the data are generated from normal distributions with equal variances (see Steinley 2006a, for a discussion of the theoretical background that also supports this claim).

3.1.4 Number of Observations

The number of observations for the first simulation are set at 200 as both Brusco (2004) and Steinley (2003; 2006b) found the number of observations

to have negligible impact on the recovery of the true cluster structure over this range of clusters and dimensionality. When investigating a larger number of clusters and variables (see the second simulation) than previously analyzed in the literature, the number of observations are varied to determine whether data set size influences the performance of the procedures under investigation.

3.1.5 Relative Cluster Density

Following Milligan and Cooper (1988) and Steinley (2003), the relative cluster density assumed three levels: (a) all clusters had the same number of observations, (b) one cluster had 60% of the observations while the remaining observations were evenly divided among the remaining clusters, and (c) one cluster had 10% of the observations while the remaining observations were evenly divided among the remaining clusters.

3.1.6 Type of Multidimensional Overlap Between Clusters

Operationalized by Steinley and Henson (2005), the type of overlap between clusters can be of two different kinds: marginal or joint. Marginal overlap is defined by allowing clusters to overlap on some dimensions (i.e., variables), but not all dimensions simultaneously. The most familiar cluster generation method using this type of overlap is Milligan's (1985) method where clusters are not allowed to overlap on the first dimension, but they are allowed to overlap on all others. This restriction prevents clusters from overlapping in the joint P-variate space (i.e., on all dimensions at once). Operationalized, marginal overlap can be defined as the probability that some interval on the j^{th} dimensions contains at least one observation from each of the two classes under consideration. This notion is readily arrived at by only investigating the overlap between two clusters, C_k and C_{k^*}, a dimension at a time. Thus, if the overlap between the clusters on the j^{th} dimension is defined as $p_j^{kk^*}$, the overall marginal overlap is defined as

$$P_m = \sum_{C_k \neq C_{k^*}} p_j^{kk^*} / (K-1) \, . \tag{7}$$

Steinley and Henson (2005) relaxed this condition and allowed clusters to overlap on all dimensions (defined as joint overlap), resulting in a more realistic technique for generating clusters. Contrary to marginal overlap, joint overlap is defined as the amount of overlap existing in all P dimensions simultaneously between two classes. Joint overlap is the probability that observations from each class are jointly in a bounded subspace, $\mathbf{R}^{\mathbf{P}}_{\mathbf{sp}}$ in $\mathbf{R}^{\mathbf{P}}$ (the P-dimensional space of the data). For two classes, C_k and C_{k^*} joint overlap is defined as

$$p^{kk^*} = Prob[x_{kjm} \in \mathbf{R}^{\mathbf{P}}_{\mathbf{sp}}] \times Prob[x_{k^*jm} \in \mathbf{R}^{\mathbf{P}}_{\mathbf{sp}}] \tag{8}$$

for $C_k \neq C_{k^*}$, all j ($j = 1, \ldots, P$), and some m, denoting the shared volume for classes C_k and C_{k^*} in $\mathbf{R^P}$. Thus, for two classes some points from both classes occupy the same region of space in $\mathbf{R^P}$ with probability p^{kk^*}. If (9) is the probability of joint overlap between any pair of classes, then we can define an overall probability of joint overlap, P_j, as the average probability of overlap between adjacent classes. Furthermore, Steinley and Henson (2005) define the number of classes that overlap as $K - 1$, resulting in

$$P_j = \sum_{C_k \neq C_{k^*}} p^{kk^*}/(K-1)\,. \tag{9}$$

Many previous types of cluster generation procedures are special cases of their method. Additionally, in their conclusion, Steinley and Henson (2005) speculated that the type of cluster overlap will have the largest effect on determining the recoverability of a cluster structure (this conjecture has been borne out in the recent experiments presented in Steinley 2006b).

Also, it should be noted that K-means clustering is designed to find non-overlapping groups (i.e., it contains a hard classification where an object is either assigned to a cluster or it is not assigned to a cluster). The inclusion of overlap between the clusters allows for the gradual investigation of how the performance of the procedures being investigated withstand data analytic situations that are not ideal. Otherwise, if all clusters were generated to be well-separated, there would be little (if any) variance in the responses, making the decision of which initialization procedure to use quite unclear. However, if the goal is to find overlapping structures we point the reader in the direction of the following techniques: COSA (Friedman and Meulman 2004), finite-mixture modeling (McLachlan and Peel 2000), and fuzzy K-means (Bezdek 1974), among others.

3.1.7 Probability Clusters Will Overlap

The probability that two clusters overlapped (either marginal or joint overlap) ranged from $P = 0$ to 0.40 in steps of 0.10. The probability of the overlap between the clusters is manipulated by simultaneously altering the relevant parameters of the distribution from which the data are being generated. For example, when generating clusters from uniform distributions, the upper and lower bounds of the of the respective distributions are manipulated to provide the desired probability of overlap; whereas, when generating clusters from the normal with equal variances only the means of the distributions are manipulated. The m-files (i.e., programs for use in the MATLAB computing environment) provided by Steinley and Henson (2005) were used to generate the cluster structures used in all simulations. To alleviate the need of standardization of variables, each variable was generated with comparable ranges by manipulating the appropriate parameters (which varied depending on which distribution

was being generated). Finally, for continuous distributions that are bounded by positive/negative infinity an overlap with probability equal to .001 is considered to be the same as no (i.e., "0") overlap. The exact generation of the various distributions is based on multi-variable calculus – for a complete description, the interested reader should refer to Steinley and Henson (2005).

3.2 Results

3.2.1 Analysis Procedure

After the datasets were generated, the K-means procedure outlined above was used to partition the data. Each dataset was clustered 12 times, once for each starting procedure discussed above. As in other studies that explore the properties of a clustering algorithm, it is assumed the number of clusters, K, is known. Both the final SSE and the agreement of the chosen solution and the "true" cluster structure, as computed using the Hubert-Arabie adjusted Rand index (Hubert and Arabie 1985), are recorded. The adjusted Rand index (ARI) is a measure of agreement corrected for chance and assumes a value of unity when there is perfect agreement between the two partitions and a value of zero when recovery is at a chance level. The ARI has been shown to have sound statistical properties and is highly recommended in cluster validation research (see Milligan and Cooper 1986; Steinley 2004b)[1]. As in several previous cluster validation studies, the values of ARI will be analyzed in an analysis of variance (ANOVA) setting using a completely-crossed design with three replications per cell (see Brusco 2004; Brusco and Cradit 2001; Donoghue 1995; Milligan and Cooper 1988; Steinley 2004a; 2004b), resulting in $3 \times 4 \times 5 \times 3 \times 5 \times 2 = 1,800$ different unique levels and a subsequent $5,400$ different datasets to be analyzed.

3.2.2 Overall Summary

When analyzing the results for the performance of the methods I_1–I_{12}, there are numerous alternatives. We begin with a description of the overall performance on both objective function minimization and true cluster recovery (see Table 1). Table 1 provides both the average rankings (where a lower ranking is better) within all 5,400 datasets for both SSE and ARI. Finally, an average ranking across both outcomes is given. The methods are arranged from best to worst in terms of the average ranking. Furthermore, the methods are grouped with like performing methods. We see that initializing K-means with Ward's (1963) method performs the best and is followed closely by the multiple ran-

1. The quality of the solutions were also assessed using Cramer's normalized χ^2 measure; however, the results were identical to those obtained by the ARI in terms of which procedures performed the best. Thus, the final results are reported in the form of the more commonly used ARI measure.

Table 1. Rankings of Twelve Initialization Methods: Primary Simulation

Method	Average Ranking	SSE Ranking	ARI Ranking (Mean ARI)
I_6	4.69	3.93	5.45 (.6678)
I_{12}	4.74	3.78	5.70 (.6606)
I_4	5.26	4.28	6.23 (.6317)
I_{11}	5.26	4.91	5.62 (.6585)
I_7	5.70	5.15	6.26 (.6409)
I_2	6.54	7.27	5.81 (.6541)
I_1	6.93	6.71	7.16 (.5934)
I_3	7.21	6.99	7.43 (.5729)
I_9	7.45	8.60	6.31 (.6313)
I_8	7.52	8.54	6.49 (.6274)
I_5	8.01	9.32	6.71 (.6317)
I_{10}	8.68	8.52	8.83 (.5214)

dom initialization strategy (Steinley 2003). This ordering is slightly different than that found in Steinley (2003); however, this is most likely due to the time constraint limitation placed on I_{12} in the present study. Given more random initializations, as in previous studies (Brusco 2004; Steinley 2003, 2004a), it is expected that I_{12} will outperform I_6.

It is not surprising that I_4 ranks closely behind I_{12} as it is basically the same starting procedure on a smaller scope. I_{11} performs reasonably well; however, we see that it is not "experimentally optimal" and should not be recommended over the simpler I_{12}, the best initialization strategy in terms of minimizing SSE. I_2–I_8 are more complicated methods for choosing the initial seeds; unfortunately, they perform, perhaps as expected, worse than the simpler methods. Most surprisingly and worrisome, the procedures implemented in SPSS (I_5) and SAS (I_{10}) perform worse than the other methods. Finally, in terms of purely minimizing the objective function at hand, I_{12} is unequivocally recommended.

Of additional interest is to determine when different methods lead to better recovery. For instance, if I_4 always led to the best recovery, it would have a rank of 1; however, the observed rank of 4.69 indicates that, at times, I_4 is outperformed by other methods. The present section explores the differential performances of the twelve methods under investigation. First, a multivariate analysis of variance was conducted where the twelve initializations were treated as multiple measurements on the same dataset (see Table 2). The first half of Table 2 presents the between datasets effects, while the second half of Table 2 presents the within datasets effects.

3.2.3 Between Datasets Effects

The between datasets effects can be thought of as the influence of the design factors across all initialization methods. In addition to all main effects,

Table 2. MANOVA for 12 Initialization Methods

Effect	Source	DF	SS	F	$\hat{\eta}^2$
Between Datasets Effects					
	Type of Overlap	1	469.17	1754.46	.165
	Distribution	4	451.82	422.39	.159
	Probability of Overlap	4	419.40	104.85	.147
	Type*Overlap	4	313.42	78.35	.110
	Relative Cluster Density	2	182.25	91.13	.064
	Number of Clusters	2	79.95	39.98	.028
	Number of Variables	3	20.85	6.95	***
	Error	5384	911.86		
Within Datasets Effects (univariate tests)					
	Initialization Method	11	107.21	672.93	.085
	Method*Density	22	151.57	475.66	.120
	Method*Distribution	44	69.64	109.27	.055
	Method*Overlap	44	51.89	81.41	.041
	Method*Type	11	16.61	104.27	.013
	Method*Type*Overlap	44	14.33	22.86	.011
	Method*Clusters	22	5.67	17.80	***
	Method*Variables	33	1.41	2.95	***
	Error	59213	857.68		

$*** =<< .01$

the interaction of type of overlap by probability of overlap was included. No other interactions were included due to the fact that Steinley (2005) indicated that other interactions were of little consequence. Furthermore, given the large sample size, it was expected that 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, $\hat{\eta}^2$. [2]

The type of overlap, joint or marginal, has the largest effect on cluster recovery ($ARI_{joint} = .53$ and $ARI_{marginal} = .71$), respectively, supporting Steinley (2005) and confirming Steinley and Henson's (2005) conjecture that

2. We would like to thank the Editor for indicating that de Craen, Commandeur, Frank, and Heiser (2006) found a small interaction between "distribution" and "density", where the distribution factor in the de Craen et al. study referred to the departure from sphericity under the multivariate normal distribution. Upon reanalysis, this effect was found to be quite small for the between-subjects effect ($\hat{\eta}^2 = .0038$) and slightly larger for the within-subjects effect ($\hat{\eta}^2 = .0173$). Upon closer inspection, it is seen that this interaction reflects the same departure from sphericity as found by De Craen et al., resulting in lower overall recovery for all methods when the distribution is the normal with unequal variances (the most non-spherical) than the normal with equal variances (perfect sphericity). While the decline in performance when switching between distributions was seen at all density levels, the interaction occurred because a few of the initialization procedures exhibited quite poor performance for the spherical and non-spherical multivariate normal distributions when there was one small cluster — namely, I_1, I_3, I_4, and I_{10} performed poorly under both conditions.

multidimensional overlap will be the most significant predictor of cluster recovery. The second most influential factor was the type of distribution that generated the cluster structure, where the respective ARI's were: Uniform = .68, Triangular = .63, Normal (Equal Variances) = .74, Normal (Unequal Variances) = .50, and Mixed = .56, indicating a large differences depending on the type of distribution. Overall, as the probability of overlap increases from 0 to .4, the average recovery decreases, going from an $ARI = .79$ to an $ARI = .56$, respectively. However, when tempered by type of overlap we see that, in the presence of joint overlap there is a decrease an ARI from .81 to .40 as probability of overlap goes from 0 to .4; on the other hand, in the presence of marginal overlap, ARI only decreases from .75 to .71, indicating very little influence when clusters do not overlap on all dimensions. When clusters are of equal size or there is one large cluster (comprising of 60% of the observations), the average ARI was .66; however, there was difficulty in recovery of the true cluster structure when there was one small cluster ($ARI = .55$). Finally, as the number of clusters varied from $K = 4, 6,$ and 8 while the number of variables varied from $P = 4, 6, 8$ and 10, there was very little effect on cluster recovery, ARI's of (.67, .62, .58) and (.60, .62, .63, 64), respectively.

3.2.4 Within Datasets Effects

Perhaps more informative is the determination of what methods are effective under which conditions. The lower half of Table 2 indicates that there is a significant difference between the initialization procedures in terms of cluster recovery. Furthermore, it is shown that the largest differences exist for cluster density and the type of distribution from which the clusters arise. Compared with these two within datasets factors, the effects of the factors are smaller and, for the most part, inconsequential. Unfortunately, since there are 12 methods, 6 main effects, and one interaction that together possess 32 levels, a reporting of all means for all methods would require the evaluation of 384 different values — a prohibitively large and mostly uninformative task. Instead, Table 3 presents the effect sizes ($\hat{\eta}^2$) for each of the methods from twelve independent univariate ANOVAs (one for each initialization method).

Examining the relative magnitude of the effect sizes indicates which methods were most "affected" by which factors. Thus, large values $\hat{\eta}^2$ indicates an initialization method is sensitive to different levels of the respective factor, while small values of $\hat{\eta}^2$ indicate that the method is relatively unaffected by changes in a particular factor level. For instance, it is immediately apparent that the recovery capabilities of each of the methods are not sensitive to changes in the number of variables; whereas, for relative cluster density there is a large discrepancy in terms of sensitivity, with I_3 being extremely sensitive ($\hat{\eta}^2 = .26$) while I_2 and I_9 were relatively unaffected by different cluster

Table 3. $\hat{\eta}^2$'s for Factor Levels by Method

Factor	I_1	I_2	I_3	I_4	I_5	I_6	I_7	I_8	I_9	I_{10}	I_{11}	I_{12}
Type of Overlap	.14	.19	.11	.13	.07	.11	.09	.11	.13	.07	.14	.14
Distribution	.09	.14	.07	.12	.07	.20	.20	.13	.08	.05	.16	.15
Probability of Overlap	.10	.11	.06	.13	.02	.14	.16	.13	.02	.15	.16	.14
Type*Overlap	.05	.07	.06	.07	.11	.10	.07	.10	.09	.04	.10	.10
Relative Cluster Density	.15	***	.26	.14	.06	.03	.03	.04	.01	.19	.06	.07
Number of Clusters	.01	.02	.03	.02	.01	.02	.02	.04	.02	.02	.02	.02
Number of Variables	***	***	***	***	***	***	***	***	***	***	***	***

$*** =<< .01$

densities ($\hat{\eta}^2 \leq .01$). Table 4 provides a correlation like matrix that contains the average pairwise ARI between methods where the columns and rows have been permuted to correspond to the order of performance provided in Table 1. The first four methods (I_6, I_{12}, I_4, and I_{11}) have a high degree of correspondence among their final solutions, and for the most part, they can be viewed as interchangeable. Many of these relationships are to be expected given the similarities of the initialization methods.

However, it may be the case that the investigator is interested in more than the final partitioning (e.g., perhaps the appropriateness of clustering the dataset, etc.) or perhaps the investigator has prior knowledge concerning the dataset under examination. To aid in determining which starting procedure to implement, the following strategy is undertaken. First, if little or nothing is known about the dataset, it is suggested to use either I_6 or I_{12} as an initialization strategy. The former provides a quick solution for moderately sized datasets, while multiple implementations of the latter are recommended for larger datasets. Furthermore, if I_{12} is used, we recommend using several thousand random restarts instead of limiting the time to a pre-specified value for two reasons: (a) as Steinley (2003, 2005) illustrates even moderately sized datasets can have several thousand local optima, and (b) Steinley (2005) indicates that the distribution of local optima is informative with respect to the quality (in terms of ARI) of the final cluster solution. Thus, overall, even though the ARI for I_6 was greater by .0072, we recommend the continued use of I_{12}.

On the other hand, if prior knowledge exists about one of the factors, then particular starting procedures may be more appropriate than the continued use of I_6 or I_{12} For each of the factor levels, we indicate the best method and, how much better the recovery is (in terms of average ARI) than I_6 and I_{12}. For instance, for type of overlap the results are: (a) for joint overlap, I_6 ($ARI_{I_6} - ARI_{I_6} = 0$, $ARI_{I_6} - ARI_{I_{12}} = .0185$), and (b) for marginal overlap, I_{11} ($ARI_{I_{11}} - ARI_{I_6} = .0064$, $ARI_{I_{11}} - ARI_{I_{12}} = .0064$). From a purely technical perspective, I_{11} outperforms both I_6 and I_{12} (under marginal overlap); however, from a substantive point of view, I_6 and I_{12} are still recommended in both

Table 4. Pairwise ARI Between Methods

	I_6	I_{12}	I_4	I_{11}	I_7	I_2	I_1	I_3	I_9	I_8	I_5	I_{10}
I_6	1.00											
I_{12}	.85	1.00										
I_4	.80	.85	1.00									
I_{11}	.84	.84	.80	1.00								
I_7	.83	.79	.76	.80	1.00							
I_2	.78	.81	.79	.78	.74	1.00						
I_1	.73	.76	.76	.74	.70	.74	1.00					
I_3	.70	.72	.72	.71	.67	.69	.70	1.00				
I_9	.68	.69	.67	.68	.66	.72	.64	.64	1.00			
I_8	.71	.72	.70	.70	.68	.70	.66	.62	.62	1.00		
I_5	.67	.66	.63	.66	.65	.69	.60	.61	.79	.59	1.00	
I_{10}	.65	.66	.67	.66	.65	.64	.68	.66	.57	.59	.52	1.00

overlap scenarios as the mean increase in ARI when using I_{11} has little, if any, practical importance.

For type of distribution, the results are: (a) uniform distribution, I_6 (0, .0321), (b) triangular distribution, I_9 (.0499, .0345), (c) normal with equal variances, I_6 (0, .0440), (d) normal with unequal variances, I_9 (.0279, .0221), and (3) mixed distributions, I_9 (.0299, .0215). For clusters with more equal shapes, such as uniform distributions or normal distributions with equal variances, I_6 and I_{12} can continue to be recommended. As the data become more irregularly shaped, I_9 performs the best; however, I_9 is ranked 9^{th} in overall performance, raising questions about its use under a broad range of scenarios. Furthermore, when the clusters are oddly shaped, I_{12} provides recovery almost nearly as good. Finally, the risk in using I_9 is amplified when its relatively low pairwise agreement with I_6 and I_{12} is noted (see Table 3), indicating that it does not have a high correspondence with the best performing methods.

For probability of overlap, I_6 and I_{12} performed the best across all conditions. Thus, the results are not presented as the recommendations would not change. When clusters were of equal size, the best performing method was I_{12} (.0100, 0). When there was one large cluster, I_{11} negligibly performed better (.0071, .0023). However, if it is suspected that there is one small cluster, I_9 notably performs better than I_6 and I_{12} (.0635, .0951). Across all levels for the number of clusters and for the number of variables, I_6 and I_{12} are the top two performing methods.

4. Methods: Simulation II

In this simulation, the focus shifts to data sets that are much larger in terms of observations, variables, and clusters. In order to facilitate such a simu-

lation, an alternative experimental design had to be adapted. Due to the nature of some of the initialization techniques, the same completely-crossed block design of the factors adopted in the first simulation study would result in a set of simulations that would be much too burdensome in terms of computational time. The factors varied in this follow-up simulation are

1. Number of clusters ($K = 5, 10,$ and 20).
2. Number of variables ($P = 25, 50,$ and 125).
3. The total sample size varied from $N = 200, 1000,$ and 5000.
4. Type of multidimensional overlap between clusters (marginal and joint overlap as in Simulation I).
5. Probability clusters will overlap (the same levels of Simulation I).

The distribution of the variables within clusters are chosen from the normal distribution with equal variances. Likewise, the relative density of the clusters were set to be equal (i.e., each cluster contains the same number of observations). A completely crossed design of the factors described above results in $3 \times 3 \times 3 \times 2 \times 5 = 270$ distinct levels. Repeating the strategy above and replicating each combination of factors three times, results in 810 distinct data sets to be analyzed.

4.1 Results

The analysis conducted for the second simulation are the same as the first simulation. In terms of cluster recovery, the ranking of the twelve methods for the larger data sets are (with mean ARI in parentheses): $I_6(.9958) > I_7(.9754) > I_{11}(.8883) > I_4(.8247) > I_3(.7941) > I_8(.7476) > I_1(.6875) > I_{12}(.6753) > I_2(.6404) > I_{10}(.5543) > I_5(.5030) > I_9(.4650)$. Once again, it is seen that I_6 completely dominates the other solutions, followed closely by I_7 and I_{11}. What is particularly important to note is that I_{12} falls considerably in the ranking. The reason for the poorer performance of I_{12} is quite understandable. When the sizes of the data sets increases (either in terms of the number of clusters, variables, or observations), the required computation time to implement several thousand random initializations increases the imposed time constraint of 20 seconds (this issue is revisited in a subsequent section).

The superior performance of I_6 is quite extraordinary. First, I_6 perfectly recovered the cluster structure 86.4% of the time. Additionally, the minimum ARI for any of the data sets for I_6 was .8664, a value of recovery Steinley (2004b) indicated was quite good. When conducting an ANOVA, there is little variability to be explained by the factors. Surprisingly, the increase in the number of clusters does not lead to a deterioration in the overall cluster recovery (only a decrease of .3% in ARI–.997 to .994 from 5 clusters to 20 clusters). Similar results are found across the entire set of factors.

This exhibited robustness is *not generalizable* across the other methods. For instance, while the minimum value of I_6 is .8664, the minimum value of I_7 (the next best performing procedure) is .6544, with the worst solution on any of the 810 data sets being an $ARI = 0.0074$ and given by I_{10}. There is little argument that any procedure other than I_6 should be used as an initialization procedure. The only true limitation is the amount of memory available to hold the $N \times N$ proximity matrix. For instance, using MATLAB 7.04 on a PC with a 2.6 GHz processor and 2 GB of RAM, we were able to analyze data sets with up to 12,000 observations without using any data compression techniques. If the size limitations are exceeded, it would be recommended to initialize the K-means algorithm with I_{11} (the second best performing method, I_7, requires the pairwise distance between all observations, which would result in the same size limitations encountered by I_6).

5. Using I_{12}

The flexibility and additional information (in terms of how the distribution of locally optimal solutions relates to cluster recovery – see Steinley 2006b) afforded by using I_{12} should be utilized until it becomes computationally infeasible. The restriction of 20 seconds of computation time was only imposed to make the simulation study more feasible (as can be seen from Table 5, there are several instances when one random initialization of the K-means procedure exceeds the 20 second cutoff). This section relates the size of data sets with the required computation time for the K-means implementation.

The values in Table 5 indicate the average computation time for the selected combinations for one random initialization of I_{12}. If the specific combination between the number of variables and the number of clusters for a given sample size did not occur in the first or second simulation, then the value provided is the average of ten random initializations for that particular cell combination. What is seen quickly is what is expected: (a) as the number of clusters increases the average time increases (moving across columns within sample size), (b) as the number of variables increases the average computation time increases (moving down rows within columns), and (c) as the sample size increases the computation time increases (comparing like-positioned values across sample size blocks).

Steinley (2003, 2006b) recommends implementing several thousand iterations due to the prevalence of locally optimal solutions. The advantage of this is two-fold: (1) smaller values of SSE are more likely to be realized, and (2) more importantly, a sense of the distribution of locally optimal solutions is obtained, and that distribution can help inform the validity of the final solution. To get a rough idea of how long it would take to complete a "batch of iterations" for a certain combination of the factors (sample size, number of clusters and num-

Table 5. Computation Time for Random Initialization of K-means (in seconds)

	N = 200			N = 1,000			N = 5,000		
	Clusters			Clusters			Clusters		
Variables	4	8	20	4	8	20	4	8	20
4	.0073	.2301	.8932	.0297	1.6942	4.9744	.3740	6.2391	53.6229
6	.0081	.2492	1.0383	.0333	3.5610	6.7985	.4818	7.8315	85.8936
8	.0121	.2422	1.2015	.0753	4.7215	11.0251	.7320	9.2635	95.5127
10	.0127	.2945	1.6995	.1671	4.7239	13.5869	.9853	13.9428	164.6304
25	.0143	.7928	4.1263	.4907	6.1390	37.3077	1.9994	13.9134	210.5844
125	.0806	4.0058	34.0057	1.0649	34.2305	493.9347	11.0465	465.8757	2241.9232
250	.2886	6.0753	68.1105	3.4705	85.3312	813.3204	20.3711	581.1952	3980.2358

ber of variables), the values in Table 5 can be multiplied by the total number of desired iterations. For instance, assume that we are interested in conducting 1000 random initializations. For the three sample sizes (N = 200, N = 1,000, and N = 5,000), the range of times are $[7.3\text{sec}, 18.92\text{hrs}]$, $[29.7\text{sec}, 9.4\text{days}]$, and $[37.4\text{sec}, 46.07\text{days}]$, respectively.

Many of these times become infeasible in a broad simulation study, forcing the implementation of a cutoff for maximum amount of time allowed to run. For instance, if 1,000 iterations were chosen instead of a time cut-off, the 90 data sets in the 250 variables and 20 clusters condition of the second simulation would have taken more than 11 years! However, in the context of "real-world" data analysis, when only one data set is being analyzed, several of the data set sizes may become feasible to analyze (perhaps depending on the patience of the researcher and the importance of the problem).

6. Conclusion

Without much reservation, the method of multiple random starting points can be recommended for general use. The present study shows this method of initialization performs very well when compared to several more complicated procedures. Additionally, Steinley (2003, 2004a) indicated the good performance of random multiple initializations, while Steinley (2006b) showed that the distribution of the local optima created from multiple initializations is useful in determining the quality of the cluster solution [3]. If the size of the data

3. The one exception, as found by the present study, is when one cluster only contains 10% of the observations while all the other clusters are larger and equally sized. In this situation, the starting rule should be based on the iterative application of principal components. However, given the overall performance of this initialization technique, the researcher should be certain that one of the clusters is much smaller than the other clusters. In general, cluster analysis is an exploratory technique where the user will not know the nature of the clusters beforehand; however, certain situations can be imagined. For instance, in psychological research, it might be known (either from previous research or theoretical motivations), that one of the clusters in a broad-based representative sample is comprised of severely mentally ill patients (which would almost certainly be a low percentage of all the observations)

set, the number of variables, or the number of clusters precludes implementing I_{12} numerous times and it becomes impossible to "estimate" the distribution of locally optimal solution, then initializing K-means with Ward's method is recommended (i.e., the use of I_6). Finally, if the number of objects makes the creation of the $N \times N$ proximity matrix impossible, then it is advised to use global K-means (I_{11}).

As with all simulation studies concerning cluster analysis, decisions had to be made about what to include in the investigation. An assumption that is almost uniformally made across these types studies is that the number of clusters are known ahead of time. Clearly, in a "real-world" data analytic situation, the number of clusters is not known ahead of time and must be estimated. Naturally, the extra step of estimating the number of clusters will have some effect on the results obtained from this simulation. Milligan and Cooper (1985) studied several techniques for determining the number of clusters that require calculations based on successive clusterings of the data (usually over a range of possible numbers of clusters). Each clustering of the data is evaluated and a final decision is reached. Logically, if using K-means to both estimate the number of clusters and obtain the final clustering, it becomes imperative to avoid poor solutions so the number of clusters can be chosen with confidence. Thus, for estimating the number of clusters in the context of K-means, we recommend using the methods suggested above to choose the number of clusters. Additionally, decisions had to be made about how many initializations strategies to include in the simulation study. In the interest of breadth, a variety of the most popular strategies were studied; however, those included in the analysis do not exhaust all possible initialization heuristics for K-means clustering (for example, we would like to thank one of the anonymous reviewers for indicating two other possible initialization strategies: the "Build algorithm" by Kaufmann and Rousseeuw 1990, and a Ward-like divisive algorithm described in Section 4.2 of Mirkin 2005).

Historically, cluster recovery in general has been the main concern when comparing different procedures for obtaining clusters (see Balashirikan, Cooper, Jacob, and Lewis 1994; Milligan 1980; Milligan and Cooper 1986, 1988; Waller et al. 1998, among others). This criterion is in fact the most reasonable when comparing clustering procedures that are designed to optimize different criteria (for example, if one was comparing complete-link clustering to K-means clusters); however, in the present context, all of the initialization procedures are designed to minimize (4). Given the global goal of minimizing SSE that all of the initialization procedures have, a certain degree of confusion tends to arise when there is a disjunction between relative performance in terms of SSE and ARI. Since the goal is to minimize SSE, in our opinion, I_{12} should always be chosen since it performs the best in this task. The fact that other procedures

may have higher levels of ARI while not having minimum SSE should sound a bullhorn into the clustering research community: "Perhaps solely minimizing SSE will not lead to the best cluster recovery". Thus, instead of choosing an initialization procedure that may happen to stumble upon a better cluster solution some of the time, other criteria should be investigated and new criteria should be developed.

References

ASTRAHAN, M.M. (1970), *Speech Analysis by Clustering, Or the Hyperphome Method*, Stanford Artificial Intelligence Project Memorandum AIM-124, Stanford, CA: Stanford University.

BALAKRISHNAN, P.V., COOPER, M.C., JACOB, V.S., and LEWIS, P.A. (1994), "A Study of Classification Capabilities of Neural Networks Using Unsupervised Learning: A Comparison With K-means Clustering", *Psychometrika, 59*, 509–525.

BEZDEK, J.C. (1974), "Cluster Validity With Fuzzy Sets", *Journal of Cybernetics, 3*, 58–73.

BRADLEY, P.S., and FAYYAD, U.M. (1998), "Refining Initial Points for k-means Clustering", *Proceedings 15th International Conference on Machine Learning*, San Francisco: Morgan Kaufmann, 91–99.

BRUSCO, M.J. (2004), "Clustering Binary Data in the Presence of Masking Variables", *Psychological Methods, 9*, 510–523.

BRUSCO, M.J., and CRADIT, J.D. (2001), "A Variable Selection Heuristic for K-means Clustering", *Psychometrika, 66*, 249–270.

BRUSCO, M.J., and STEINLEY, D. (2005), "A Comparison of Heuristic Procedures for Minimum Within-cluster Sums of Squares Partitioning", *Manuscript Submitted for Publication*.

DE CRAEN, S., COMMANDEUR, J.J.F., FRANK, L.E., and HEISER, W.J. (2006), "Effects of Group Size and Lack of Sphericity on the Recovery of Clusters in K-means Cluster Analysis", *Multivariate Behavioral Research, 41*, 127–146.

DONOGHUE, J.R. (1995), "Univariate Screening Measures for Cluster Analysis", *Multivariate Behavioral Research, 30*, 385–427.

FABER, V. (1994), "Clustering and the Continuous K-means Algorithm", *Los Alamos Science, 22*, 138–144.

FRIEDMAN, J.H., and MEULMAN, J.J. (2004), "Clustering Objects on Subsets of Variables", *Journal of the Royal Statistical Society, B, 66*, 1–25.

HAND, D.J., and KRZANOWSKI, W.J. (2005), "Optimising k-means Clustering Results with Standard Software Packages", *Computational Statistics and Data Analysis, 49*, 969–973.

HANSEN, P., NGAI, E., CHEUNG, B.K., and MLADENOVIC, N. (2005), "Analysis of Global K-means, An Incremental Heuristic for Minimum Sum-of-squares Clustering", *Journal of Classification, 22*, 287–310.

HARTIGAN, J.A. (1975), *Clustering Algorithms*, New York: John Wiley and Sons.

HUBERT, L.J. and ARABIE, P. (1985), "Comparing Partitions", *Journal of Classification, 2*, 193–218.

KAUFMAN, L., and ROUSSEEUW , P. (1990), *Finding Groups in Data: An Introduction to Cluster Analysis*, New York: Wiley.

LIKAS, A., VLASSIS, N., and VERBEEK, J. (2003), "The Global K-means Clustering Algorithm", *Pattern Recognition, 36*, 451–461.

MACQUEEN, J. (1967), "Some Methods of Classification and Analysis of Multivariate Observations", in Eds. L.M. Le Cam and J. Neyman, *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*,Berkeley, CA: University of California Press, 281–297.

MCLACHLAN, G.J., and PEEL, D. (2000). *Finite Mixture Modeling*, New York: Wiley.

MILLIGAN, G.W. (1980), "The Validation of Four Ultrametric Clustering Algorithms", *Pattern Recognition, 12*, 41–50.

MILLIGAN, G.W., and 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., and 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., and COOPER, M.C. (1988), "A Study of Standardization of Variables in Cluster Analysis", *Journal of Classification, 5*, 181–204.

MIRKIN, B. (2005), *Clustering For Data Mining: A Data Recovery Approach*, London: Chapman and Hall.

SAS (2004), "The FASTCLUS Procedure", in *SAS/STAT 9.1 user's guide volume 2*, Cary, NC: SAS Institute Inc.

SPÄTH, H. (1980), *Cluster Analysis Algorithms for Data Reduction and Classification of Objects*, New York: Wiley.

SPSS (2003), *SPSS 12.0 Command Syntax Reference*, Chicago: SPSS, Inc.

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 Eds., D. Banks, L. House, F.R. McMorris, P. Arabie, and W. Gaul, *Classification, Clustering, and Data Mining Applications*, New York: Springer, 53–60.

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, and HENSON, R. (2005), "OCLUS: An Analytic Method for Generating Clusters with Known Overlap", *Journal of Classification, 22*, 221–250.

SU, T., and DY, J.G. (2004), "Another Look at Non-random Methods for Initializing K-means Clustering", *Proceedings of the* 16^{th} *IEEE International Conference on Tools with Artificial Intelligence*, 784–786.

WALLER, N.G., KAISER, H.A., ILLIAN, J. B., and 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.