Journal of Classification 23:315-334 (2006)
DOI: 10.1007/s00357-006-0018-y

Generation of Random Clusters with Specified Degree of Separation

Weiliang Qiu

Brigham and Women's Hospital, Boston

Harry Joe

University of British Columbia

Abstract: We propose a random cluster generation algorithm that has the desired features: (1) the population degree of separation between clusters and the nearest neighboring clusters can be set to a specified value, based on a separation index; (2) no constraint is imposed on the isolation among clusters in each dimension; (3) the covariance matrices correspond to different shapes, diameters and orientations; (4) the full cluster structures generally could not be detected simply from pair-wise scatterplots of variables; (5) noisy variables and outliers can be imposed to make the cluster structures harder to be recovered. This algorithm is an improvement on the method used in Milligan (1985).

Keywords: Cluster generation; Separation index; Factorial experiment design.

This research was supported by a NIH Grant and a NSERC Discovery Grant. We are grateful to the referees for providing additional references and valuable comments.

Authors' Addresses: W. Qiu, Channing Laboratory, Brigham and Women's Hospital, Boston, MA, USA, e-mail: stwxq@channing.harvard.edu; H. Joe, Department of Statistics, University of British Columbia, Vancouver, BC, Canada.

1. Introduction

To numerically evaluate the performances of clustering methods, simulated data sets are often used. Simulated data sets have known cluster structures so that we can evaluate the performance of a clustering method by checking the agreement between the true partition and the partition obtained by the clustering method. They are also easy to generate, and we can control the noise (e.g. noisy variables, outliers, missing values, and measurement errors, etc.) and produce as many replicates as we want. Furthermore they can be used to determine the situations in which a clustering method works well or doesn't work well.

The qualities of simulated data sets depend on cluster generating algorithms. Many cluster generating methods have been proposed (e.g. Milligan 1985; Gnanadesikan, Kettenring, and Tsao 1995; Zhang, Ramakrishnan, and Livny 1997; Guha, Rastogi, and Shim 1998; Waller, Kaiser, Illian, and Manry 1998; Waller, Underhill, and Kaiser 1999; Tibshirani, Walther, and Hastie 2001). Milligan (1985) and Waller et al. (1999) systematically addressed the problem of cluster generation. Milligan generated clusters from an experimental design point of view: the factors include the number of clusters, the number of dimensions (variables), sizes of clusters, outliers, noisy variables, and measurement errors. In the design, cluster centers and boundaries are generated one dimension at a time. Cluster boundaries are separated by a random quantity in the first dimension. However there is no constraint on the isolation among clusters in other dimensions. Multivariate normal distributions with diagonal covariance matrices are used to generate data points. Data points are rejected if they fall outside the cluster boundaries.

The main limitation of Milligan's (1985) method is that the degree of separation among clusters is not controlled. The degree of separation among clusters is one of the most important factors to check the performances of clustering methods. If a clustering method could work well for closely-spaced clusters, then it is reasonable to believe that this method is better than other clustering methods. If a clustering method could not work well for well-separated clusters, then it is reasonable to believe that its performance is worse than other clustering methods. Therefore it is desirable to control the degree of separation among clusters. Waller et al. (1999) proposed a index called indicator validity to control the average separation among clusters. In this article we use a separation index proposed by Qiu and Joe (2006) to directly control the degree of separation between clusters and the nearest neighboring clusters.

It is quite common in real data sets that covariance matrices are not diagonal and clusters are separated in high-dimensional space but are overlapping in each pair of dimensions. By a random rotation, we can improve Milligan's (1985) method so that clusters might not be visualized by pair-wise scatterplots of variables. However it is not straightforward to improve Milligan's (1985)

method so that the covariance matrices can have different shapes, diameters and orientations, while the degree of separation is controlled to a specified value. At first thought, we can control the degree of separation by specifying the lengths of the gaps between clusters in the first dimension rather than randomly generating the lengths of the gaps. But the gap is equivalent to our degree of separation only when the covariance matrices of clusters are diagonal (case of uncorrelated variables).

We improve the cluster generation method proposed in Milligan (1985) so that the degree of separation between clusters and the nearest neighboring clusters could be set to a specified value while the cluster covariance matrices can be arbitrary positive definite matrices, and so that clusters generated might not be visualized by pair-wise scatterplots of variables.

The remaining sections of the article are organized as follows. Section 2 presents the cluster generating algorithm. Section 3 describes a factorial experiment design to systematically generate simulated data sets. Section 4 gives a verification of the simulated data sets. An illustration of the use of the design for comparing methods of estimating the number of clusters is given in Section 5. Section 6 contains a summary and proposes possible future research topics. Technical details are included in Appendices.

2. Algorithm for Generation of Random Clusters

2.1 Overall Algorithm

In this subsection, we give the overall algorithm for generation of random clusters. We will describe the details in later subsections.

Step 1 Specify the number of non-noisy dimensions p_1, the number of clusters K, the degree of separation J_0 between any cluster and its nearest neighboring cluster, the tuning parameter α for the separation index, the number of noisy variables p_2, the number or ratio of outliers, the lower bound $\lambda_{\min}$ of the eigenvalues for random covariance matrices, the ratio r_λ of the upper bond of the eigenvalues to the lower bound of the eigenvalues for random covariance matrices, and the range of cluster sizes $[n_L,\ n_U]$.

Step 2 Generate cluster centers and random covariance matrices in the p_1 non-noisy dimensions so that neighboring clusters have population separation index J_0 (details are given in Section 2.3).

Step 3 Generate sizes of each cluster randomly from the range $[n_L,\ n_U]$ and generate memberships of each data point.

Step 4 Generate the mean vector and covariance matrix of the noisy variables (details are given in Section 2.4).

Step 5 Apply a random rotation to the cluster means and covariance matrices in Step 2 (details are given in Section 2.5).

Step 6 From Steps 4 and 5, we have cluster means and covariance matrices for all K clusters.

Step 7 Generate random vectors for each of the K clusters from a given family of elliptical distributions.

Step 8 Calculate the population separation index matrices and projection directions for pairs of clusters via the population mean vectors and covariance matrices.

Step 9 Calculate the sample separation index matrices and projection directions via the sample mean vectors and covariance matrices.

Step 10 Generate outliers. The memberships of outliers are assigned as zero (details are given in Section 2.4).

2.2 Degree of Separation

The key concept in the algorithm is the degree of separation. We use the degree of separation based on the separation index proposed by Qiu and Joe (2006). Other separation indices (e.g., Blashfield 1976; Atlas and Overall 1994; Donoghue 1995; Steinley 2003, 2004) could be used instead. The advantage of Qiu and Joe's separation index is that it directly measures the magnitude of the gap or data-sparse area between each pair of clusters. We also tried a probability of overlap between two cluster distributions but this is not as geometrically interpretable.

Denote L_k and U_k as the lower and upper $\alpha/2$ percentiles of projected cluster k, respectively. The quantile version of the separation index is defined as

$$J(\boldsymbol{a}) = \frac{L_2 - U_1}{U_2 - L_1}$$

where $\boldsymbol{a}$ is the projection direction. Figure 1 illustrates that $J(\boldsymbol{a})$ directly measures the gap or data-sparse area between two clusters.

If two clusters are generated from p-dimensional elliptical distributions with densities $f_k(\boldsymbol{x}) = |\boldsymbol{\Sigma}_k|^{-1/2} h_p((\boldsymbol{x}-\boldsymbol{\mu}_k)'\boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k))$, $k = 1, 2$, derived from a spherical density $h_p(\boldsymbol{y}'\boldsymbol{y})$ with marginal variances of 1 and means of 0, the separation index function of a projection direction $\boldsymbol{a}$ can be rewritten as

$$J_{12}(\boldsymbol{a}) = \frac{\boldsymbol{a}^T(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) - q_{\alpha/2}\big(\sqrt{\boldsymbol{a}^T\boldsymbol{\Sigma}_1\boldsymbol{a}} + \sqrt{\boldsymbol{a}^T\boldsymbol{\Sigma}_2\boldsymbol{a}}\big)}{\boldsymbol{a}^T(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) + q_{\alpha/2}\big(\sqrt{\boldsymbol{a}^T\boldsymbol{\Sigma}_1\boldsymbol{a}} + \sqrt{\boldsymbol{a}^T\boldsymbol{\Sigma}_2\boldsymbol{a}}\big)},$$

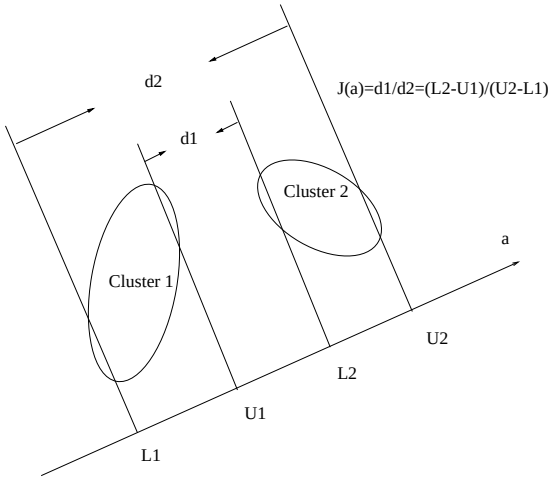

Figure 1. Illustration of the concept of the separation index. L_k and U_k are the lower and upper $\alpha/2$ percentiles of projected cluster k, respectively. $L_2 > U_1$ indicates that two clusters are separated; $L_2 < U_1$ indicates that two clusters are overlapping.

where $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$, $k = 1, 2$, are the mean vectors and covariance matrices for the two clusters, $\alpha \in (0, 0.5)$ is a tuning parameter indicating the percentage of data in the extremes to downweight, $q_{\alpha/2}$ is the upper $\alpha/2$ quantile of the univariate margin of $h_p(\boldsymbol{y}'\boldsymbol{y})$. If the distributions of two clusters are multivariate normal distributions, then $q_{\alpha/2} = z_{\alpha/2}$, where $z_{\alpha/2}$ is the upper $\alpha/2$ quantile of the univariate standard normal distribution.

Let J_{12}^* be the optimal separation index between clusters 1 and 2, i.e., $J_{12}^* = J_{12}(\boldsymbol{a}^*)$, where $\boldsymbol{a}^*$ is the optimal projection direction which maximizes J_{12} (Appendix A briefly describes how to compute $\boldsymbol{a}^*$). Similarly, let $J_{k_1k_2}^*$ be optimal separation index between clusters k_1, k_2.

Let $J_{k_1\,\min}^* = \min_{k_2=1,\ldots,K,k_2\neq k_1} J_{k_1k_2}^*$, where K is the number of clusters. The *degree of separation* then can be measured by the separation indices $J_{k\,\min}^*$, $k = 1, \ldots, K$. If $J_{k\,\min}^*$, $k = 1, \ldots, K$, are all close to zero, then the cluster structure is close. If $J_{k\,\min}^*$, $k = 1, \ldots, K$, are all quite large, then the cluster structure is well-separated. However it is difficult to make a clear-cut decision on whether a cluster structure is "close", "separated", or "well-separated". For the factorial experiment design in Section 3, we regard a cluster structure as *close* if $J_{k\,\min}^* = 0.010$, $k = 1, \ldots, K$, as *separated* if $J_{k\,\min}^* = 0.210$, $k = 1, \ldots, K$, and as *well-separated* if $J_{k\,\min}^* = 0.342$, $k = 1, \ldots, K$. The value 0.010 is the separation index between two clusters, which are generated from two univariate normal distributions $\mathrm{N}(0, 1)$ and $\mathrm{N}(A, 1)$, where $A = 4$. The values 0.210 and 0.342 are the separation indices

corresponding to $A = 6$ and $A = 8$ respectively. The tuning parameter α is equal to 0.05 when calculating these separation indices.

Essentially, the $J^*_{k\min}$ values of 0.01, 0.21 and 0.342 were chosen to match behavior we expect from clustering methods for close, separated and well-separated. In the case of closely-spaced clusters, we expect clustering methods to have difficulties in determining the number of clusters (especially if there are some noisy variables), and in the case of well-separated clusters, we expect good clustering methods to manage fine. Our choices of $J^*_{k\min}$ to match three levels of cluster separation did work out well for the simulation study in Section 5, as well as for other results in Qiu's PhD thesis.

2.3 Allocating Cluster Centers and Generating Covariance Matrices

If there are more than two clusters, then it is not easy to allocate the cluster centers so that the separation indices ($J^*_{k\min}$, $k = 1, \ldots, K$) between any cluster and its nearest neighboring cluster are all equal to J_0, except for the trivial cases where cluster covariance matrices are all equal to a multiple of the identity matrix. To overcome this difficulty, we first allocate cluster centers on the vertices of an equilateral simplex and then adjust the length of the simplex edge so that the minimum separation among clusters is equal to the specified value J_0. Finally we scale covariance matrices (but keep their shapes and orientations) so that the separations between any cluster and its nearest neighboring cluster are also equal to the specified value J_0.

A *simplex* in a p_1-dimensional space contains $p_1 + 1$ vertices $\boldsymbol{v}_1$, ..., $\boldsymbol{v}_{p_1+1}$. These p_1+1 vertices are linearly dependent. However all its proper subsets are linearly independent. A simplex is a line segment, a triangle, and a tetrahedron in one-, two-, and three-dimensional space respectively. The lengths of the simplex edges are not necessarily equal.

With the following cluster-center-allocation algorithm, we can obtain mean vectors and covariance matrices of clusters so that the population separation indices $J^*_{k\min}$, $k = 1, \ldots, K$, are all equal to J_0.

Cluster-Center-Allocation Algorithm

Step (a) Generate K covariance matrices $\boldsymbol{\Sigma}_k$ in p_1 dimensions, $k = 1, \ldots, K$.

Step (b) Construct a p_1-dimensional equilateral simplex whose edges have length 2. The first two vertices are $\boldsymbol{v}_1 = -\boldsymbol{e}_1$ and $\boldsymbol{v}_2 = \boldsymbol{e}_1$ respectively, where the $p_1 \times 1$ vector $\boldsymbol{e}_1 = (1, 0, \ldots, 0)^T$. Denote the j-th vertex as $\boldsymbol{v}_j$, $j = 1, \ldots, p_1 + 1$. A method for construction of the other vertices is given below.

Step (c) If $K \leq p_1 + 1$, then take the first K vertices of the simplex as initial

cluster centers. If $K > p_1 + 1$, then we start adding vertices from the following sequence after $\boldsymbol{v}_{p_1+1}$ until all K cluster centers are allocated:

$$\begin{array}{llll} \boldsymbol{v}_2 + 2 * \boldsymbol{e}_1, \ \ldots, & \boldsymbol{v}_{p_1+1} + 2 * \boldsymbol{e}_1, \boldsymbol{v}_2 + 4 * \boldsymbol{e}_1, & \ldots, \boldsymbol{v}_{p_1+1} + 4 * \boldsymbol{e}_1, \\ \boldsymbol{v}_2 + 6 * \boldsymbol{e}_1, \ \ldots, & \boldsymbol{v}_{p_1+1} + 6 * \boldsymbol{e}_1, \ \ldots, & \ldots, \ \ldots, \end{array}$$

Essentially this just keeps on adding points on a shifted symmetric simplex.

Step (d) Calculate the separation index matrix $\boldsymbol{J}^*_{K\times K}$.

Step (e) Scale the length of the simplex edge by a scalar c_1 so that the minimum separation index $\min_{i\neq j} J^*_{ij}$ among pairs of clusters is equal to J_0.

Step (f) Compute the separation indices, $J^*_{k\,\min}$, $k = 1, \ldots, K$, between a cluster and its nearest neighboring cluster, and obtain $k^* = \arg \max_{k=1,\ldots K} J^*_{k\,\min}$. If $J^*_{k^*\,\min} > J_0$, then go to Step (g). Otherwise scaling is complete.

Step (g) Scale the covariance matrix $\boldsymbol{\Sigma}_{k^*}$ by a scalar c_2 so that $J^*_{k^*\,\min} = J_0$. Go back to Step (f).

The vertices of a p-dimensional equilateral simplex, whose edge length is $L = 2$ and first two vertices are $\boldsymbol{v}_1 = -\boldsymbol{e}_1$ and $\boldsymbol{v}_2 = \boldsymbol{e}_1$, are not unique. In the following, we provide a set of such vertices. Suppose that we already obtain the coordinates of the vertices $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{k-1}$, $2 < k \leq p + 1$. Then the coordinates of the vertex $\boldsymbol{v}_k$ is obtained by:

$$\begin{aligned} v_{ki} &= \bar{v}_{ki}, \ i = 1, \ldots, k-2, \\ v_{kj} &= 0, \ j = k, \ldots, p, \\ v_{k,k-1} &= \left\{ 4 - \frac{1}{k-1} \sum_{i=1}^{k-1} \left(\boldsymbol{v}_i - \bar{\boldsymbol{v}}_{k-1}\right)^T \left(\boldsymbol{v}_i - \bar{\boldsymbol{v}}_{k-1}\right) \right\}^{1/2}, \end{aligned} \tag{2.1}$$

where $\bar{\boldsymbol{v}}_{k-1} = \frac{1}{k-1} \sum_{i=1}^{k-1} \boldsymbol{v}_i = (\bar{v}_{k-1,1}, \ldots, \bar{v}_{k-1,p})^T$. Appendix B gives the derivation of this formula.

The eigenvalues and eigenvectors determine the diameter, shape and orientation corresponding to a positive-definite covariance matrix $\boldsymbol{\Sigma}$. Therefore, we generate random positive-definite covariance matrices by generating random eigenvalues and eigenvectors. The relation between a $p \times p$ positive-definite covariance matrix $\boldsymbol{\Sigma}$ and its eigenvalues and eigenvectors is $\boldsymbol{\Sigma} = \boldsymbol{Q}\boldsymbol{\Lambda}\boldsymbol{Q}^T$, where $\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_p)$, $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p > 0$ are eigenvalues of $\boldsymbol{\Sigma}$, and the j-th column of the matrix $\boldsymbol{Q}$ is the normalized eigenvector of $\boldsymbol{\Sigma}$ corresponding to the eigenvalue λ_j. Note that $\boldsymbol{Q}$ is a $p \times p$ orthogonal matrix

such that $\boldsymbol{Q}\boldsymbol{Q}^T = \boldsymbol{I}_p$ and $\boldsymbol{Q}^T\boldsymbol{Q} = \boldsymbol{I}_p$, where $\boldsymbol{I}_p$ is the p-dimensional identity matrix.

We can generate p eigenvalues uniformly from a bounded interval whose lower bound is positive. In our experience, the range $[\lambda_{\min} = 1, \lambda_{\max} = 10]$ can give reasonable variability for the diameters of clusters. Therefore, we set $\lambda_{\min} = 1$ and $r_\lambda = \lambda_{\max}/\lambda_{\min} = 10$ for the simulated data sets mentioned in the subsequent sections.

To generate a $p \times p$ orthogonal matrix $\boldsymbol{Q}$, we can first generate a $p \times p$ lower triangle matrix $\boldsymbol{M}$ whose diagonal elements are all non-zero. Then we use the Gram-Schmidt Orthogonalization (Kotz and Johnson 1983, Vol. 3, pp. 478) to transform the lower triangle matrix $\boldsymbol{M}$ to an orthogonal matrix.

Note that other methods can be used to generate random positive definite covariance matrices. For example, we can generate covariance matrices based on correlation matrices (e.g., Waller et al. 1999; Joe 2006).

2.4 Constructing Noisy Variables and Outliers

There is no unified definition of a noisy variable. Milligan (1985) assumed that noisy variables are uniformly distributed and are independent of each other and of non-noisy variables. For the cluster generating, we assume that noisy variables are normally distributed and independent of non-noisy variables. However, noisy variables are not necessarily independent of each other.

Like Milligan (1985), we require that the variations of noisy variables in the generated data sets are similar to those of non-noisy variables. If noisy variables have smaller variations than those of non-noisy variables, then we implicitly downweight noisy variables. Hence the data sets would be less challenging.

Denote the $p_1 \times p_1$ matrix $\boldsymbol{\Sigma}^*$ as the covariance matrix of non-noisy variables and the $p_2 \times p_2$ matrix $\boldsymbol{\Sigma}_0$ as the covariance matrix of noisy variables. One possible way to make the variations of noisy variables similar to those of non-noisy variables is to make the ranges of eigenvalues of $\boldsymbol{\Sigma}_0$ similar to those of $\boldsymbol{\Sigma}^*$.

If we assume that data points in non-noisy dimensions are from a mixture of distributions with the density function $f(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k f_k(\boldsymbol{x})$, where $0 \le \pi_k \le 1$ and $\sum_{k=1}^{K} \pi_k = 1$, then the covariance matrix $\boldsymbol{\Sigma}^*$ of the mixture of distributions is $\boldsymbol{\Sigma}^* = \sum_{k=1}^{K} \pi_k \boldsymbol{\Sigma}_k + \sum_{k<k'} \pi_k \pi'_k \left(\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'}\right)\left(\boldsymbol{\mu}_k - \boldsymbol{\mu}_{k'}\right)^T$, where $\boldsymbol{\mu}_k$ and $\boldsymbol{\Sigma}_k$ are the mean vector and covariance matrix of the k-th component of the mixture of distributions. We can randomly generate the eigenvalues of $\boldsymbol{\Sigma}_0$ from the interval $[\lambda^*_{p_1}, \lambda^*_1]$, where p_1 is the number of non-noisy variables, $\lambda^*_{p_1}$ and λ^*_1 are the minimum and maximum eigenvalues of the matrix $\boldsymbol{\Sigma}^*$. In this way, the variations of noisy variables would be similar to those of non-noisy variables.

The mean vector $\boldsymbol{\mu}^* = \sum_{k=1}^{K} \pi_k \boldsymbol{\mu}_k$ of the mixture of distributions can be used to generate the $p_2 \times 1$ mean vector $\boldsymbol{\mu}_0$ of the noisy variables. For example, we can randomly generate the p_2 elements of $\boldsymbol{\mu}_0$ from the interval $[\min_{1 \le j \le p_1} \mu_j^*,\ \max_{1 \le j \le p_1} \mu_j^*]$.

Once we generate the mean vectors and covariance matrices of non-noisy and noisy variables, we can randomize the labels of variables to make the generated data sets closer to real data sets.

Outliers, like noisy variables, are frequently encountered in real data sets, and they may affect the recovery of true cluster structures. Therefore any cluster generating algorithm should provide a function to produce outliers for simulated data sets.

For simplicity, we generate outliers from a distribution whose marginal distributions are independent uniform distributions. The outliers are generated for the whole data set instead of for each cluster. The range of the j-th marginal uniform distribution depends on the range of non-outliers in the j-th dimension. We set the range as $[\hat{\mu}_j - 4\hat{\sigma}_j, \hat{\mu}_j + 4\hat{\sigma}_j]$, where $\hat{\mu}_j$ and $\hat{\sigma}_j$ are the sample mean and standard deviation of the j-th variable respectively.

2.5 Rotating Data Points

In most cases, we could not detect the numbers of clusters of real data sets in high-dimensional spaces by lower-dimensional scatterplots of variables. However, the simulated data sets produced by the methods mentioned in the cluster analysis literature do not always have this property. For example, we can easily detect the numbers of clusters in data sets generated by Milligan (1985) from the scatterplots of the first variable versus any one of other variables. The data sets generated by our algorithm might have the same problem.

To improve the simulated data sets so that we could not detect the numbers of clusters by pair-wise scatterplots of variables, we can simply transform these data sets by random rotations. To rotate a data point $\boldsymbol{x}$, we can apply the transformation $\boldsymbol{y} = \boldsymbol{Q}\boldsymbol{x}$, where $\boldsymbol{Q}$ is an orthogonal matrix. We can use the method proposed in Section 2.3 to generate an orthogonal matrix. We only rotate non-noisy variables and do not rotate noisy variables because otherwise it is possible that noisy variables are no longer noisy after rotation. Hence the rotation leads to simulated data sets that are more representative of real data sets. For evaluation by simulation of some aspects of clustering algorithms, the rotation step might not be needed.

2.6 An Illustration of the Cluster-Center-Allocation Algorithm

Suppose that we would like generate a data set that has 5 clusters in a 2-dimensional space with close structure ($J_0 = 0.01$). We first generate 5

(random) covariance matrices using the method described in subsection 2.3, after specifying $\lambda_{\min} = 1$ and $r_\lambda = 10$. The covariance matrices are:

$$\begin{pmatrix} 6.11 & 0.46 \\ 0.46 & 4.63 \end{pmatrix}, \begin{pmatrix} 4.98 & 0.86 \\ 0.86 & 6.11 \end{pmatrix}, \begin{pmatrix} 7.70 & 1.90 \\ 1.90 & 3.19 \end{pmatrix}, \begin{pmatrix} 8.43 & -0.02 \\ -0.02 & 8.38 \end{pmatrix}, \begin{pmatrix} 5.76 & 0.10 \\ 0.10 & 6.24 \end{pmatrix}.$$

Then we construct an equilateral simplex and a shifted equilateral simplex with edges length 2 in a 2-dimensional space (see Figure 2).

The initial separation index matrix is $\boldsymbol{A}_1$. After scaling the lengths of the simplices by a factor 5.35, the separation index matrix becomes $\boldsymbol{A}_2$ with the minimum (non-diagonal) value equal to $J_0 = 0.01$.

$$\boldsymbol{A}_1 = \begin{matrix} -1.00 & -0.64 & -0.63 & -0.45 & -0.47 \\ -0.64 & -1.00 & -0.58 & -0.67 & -0.66 \\ -0.63 & -0.58 & -1.00 & -0.47 & -0.67 \\ -0.45 & -0.67 & -0.47 & -1.00 & -0.68 \\ -0.47 & -0.66 & -0.67 & -0.68 & -1.00 \end{matrix},$$

$$\boldsymbol{A}_2 = \begin{matrix} -1.00 & 0.08 & 0.1 & 0.34 & 0.32 \\ 0.08 & -1.00 & 0.17 & 0.03 & 0.04 \\ 0.10 & 0.17 & -1.00 & 0.31 & 0.03 \\ 0.34 & 0.03 & 0.31 & -1.00 & 0.01 \\ 0.32 & 0.04 & 0.03 & 0.01 & -1.00 \end{matrix}.$$

Next, we scale covariance matrices so that the separations between any cluster and its nearest neighboring cluster are also equal to J_0. Because $J^*_{1\min} = 0.08 = \arg\max_{k=1,\ldots,5} J^*_{k\min}$, we scale the covariance matrix of cluster 1 by a scalar 1.63. Then $J^*_{3\min} = 0.03 = \arg\max_{k=1,\ldots,5} J^*_{k\min}$, and we scale covariance matrix of cluster 3 by a scalar 1.20. The final separation index matrix becomes:

$$\begin{pmatrix} -1.00 & 0.01 & 0.01 & 0.29 & 0.26 \\ 0.01 & -1.00 & 0.15 & 0.03 & 0.04 \\ 0.01 & 0.15 & -1.00 & 0.30 & 0.01 \\ 0.29 & 0.03 & 0.30 & -1.00 & 0.01 \\ 0.26 & 0.04 & 0.01 & 0.01 & -1.00 \end{pmatrix}.$$

3. A Factorial Experiment Design

One application of our cluster generating algorithm is to systematically study the performances of clustering-related methods (such as clustering methods or methods to estimate the number of clusters), which could be affected by factors such as (1) the number of clusters; (2) the degree of separation; (3) the number p_1 of non-noisy variables; and (4) the number p_2 of noisy variables. To know the impact of these factors, a factorial experiment design like in Milligan (1985) is needed. Our cluster generating algorithm can generate data sets with desired cluster structures for factorial experiment designs.

For example, we can consider a factorial experiment design containing four factors listed in Table 1. The number of clusters and number of non-noisy

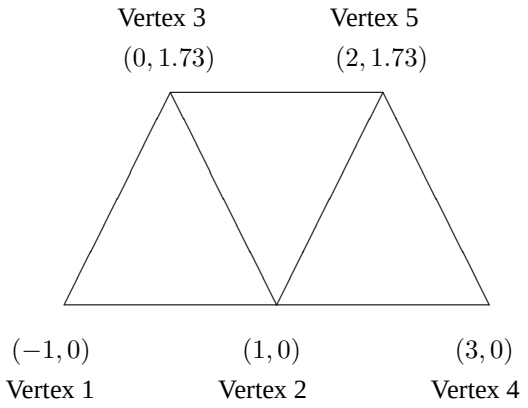

Figure 2. A equilateral simplex and a shifted equilateral simplex with edge length 2.

Table 1. The factors and their levels in a factorial experiment design

Factors	Levels
Number of clusters	3, 6, 9
Degree of separation	close, separated, well-separated
Number of non-noisy variables p_1	4, 8, 20
Number of noisy variables p_2	1, $0.5p_1$, p_1
Totally $3 \times 3 \times 3 \times 3 = 81$ cells in the design.	

variables were considered in Milligan's (1985) design. We add the degree of separation as a factor because by intuition, it will affect a lot the performances of clustering-related methods. It is well-known (e.g. Gordon 1981, Section 2.4.5; Milligan 1989; Gnanadesikan et al. 1995) that noisy variables may mask true cluster structure. Therefore we explicitly add the number of noisy variables as a factor. Unlike Milligan (1985), the cluster size and outlier are not considered in this design.

As mentioned in subsection 2.2, we regard a cluster structure as *close* if $J^*_{k\min} = 0.010$, $k = 1, \ldots, K$, as *separated* if $J^*_{k\min} = 0.210$, $k = 1, \ldots, K$, and as *well-separated* if $J^*_{k\min} = 0.342$, $k = 1, \ldots, K$.

Following Milligan (1985), we generate three replicates for each cell of the design. So the design produces $3 \times 3 \times 3 \times 3 \times 3 = 3 \times 81 = 243$ simulated data sets. When generating these 243 data sets, we set $\alpha = 0.05$, $\lambda_{\min} = 1$, and $r_\lambda = 10$. We randomly generate cluster sizes from the intervals $[10p,\ 10p + 100]$, where p is the total number of variables (including both noisy and non-noisy variables). The number of outliers is set to be zero. The data points are generated from mixtures of multivariate normal distributions.

Note that for generating random vectors with non-normal distributions, the methods given in Cario and Nelson (1997) and Devroye (1986) can be used.

4. Verification of Simulated Data Sets

Once we generate a simulated data set with a specified separation index J_0, we need to verify if the sample separation indices $J^*_{k\min}$, $k = 1, \ldots, K$, are close to J_0, where K is the number of clusters in the data set.

For the design discussed in Section 3, there are 81 data sets each for close, separated and well-separated cluster structures. For the 81 data sets with close cluster structures, we put all the estimated separation indices $\hat{J}^{*i}_{k\min}$, $k = 1, \ldots, K_i$, $i = 1, \ldots, 81$, into a set $S_{\hat{J}c}$. Similarly, we obtain the set $S_{\hat{J}s}$ for data sets with separated cluster structures and the set $S_{\hat{J}w}$ for data sets with well-separated cluster structures. We expect that all elements in the sets $S_{\hat{J}c}$, $S_{\hat{J}s}$, and $S_{\hat{J}w}$ are close to 0.010, 0.210, and 0.342 respectively.

Let S be any one of the sets $S_{\hat{J}c}$, $S_{\hat{J}s}$, and $S_{\hat{J}w}$. Denote s_i as the i-th element of the set S and m as the number of elements in the set S. The estimates of the bias and mean-squared error (MSE) for the specified degree of separation J_0 are defined as $\widehat{\text{bias}}\,(S) = \overline{S} - J_0$, $\widehat{\text{MSE}}\,(S) = \widehat{\text{Var}}\,(S) + \widehat{\text{bias}}\,(S)^2$, where $\overline{S} = m^{-1}\sum_{i=1}^m s_i$ and $\widehat{\text{Var}}\,(S) = (m-1)^{-1}\sum_{i=1}^m (s_i - \overline{S})^2$, are the sample mean and variance of the set S respectively. The specified degree of separation J_0 can take values 0.010, 0.210, or 0.342.

Table 2 lists the results for the sets $S_{\hat{J}c}$, $S_{\hat{J}s}$, and $S_{\hat{J}w}$. We can see that the sample degrees of separation of the data sets are close to the specified degree of separation.

In addition to the separation indices between clusters and the nearest neighbors, the separation indices between clusters and their other *direct neighboring clusters* can also provide useful information about the degree of separation of the cluster structure in a data set. In our algorithm, a cluster k_2 is a *direct neighboring cluster* of the cluster k_1 if the distance between the two cluster centers is equal to L, where L is the edge length of the symmetric simplices.

Let $\mathcal{N}(k)$ be the set of vertices that are neighbors of vertex k (or set of clusters that are direct neighbors of cluster k). When we generate the 81 data sets with population separation index J_0, we recorded the separation indices between clusters and their farthest direct neighboring clusters $\hat{J}^{*i}_{k\max} = \max_{k'\in\mathcal{N}(k)} \hat{J}^{*i}_{k,k'}$, where K_i is the number of clusters of the i-th data set, $i = 1, \ldots, 81$. We denote these separation indices as the *farthest separation indices*. We also record the median of the separation indices between clusters and their direct neighboring clusters $\hat{J}^{*i}_{k\,\text{med}} = \text{median}_{\,k'\in\mathcal{N}(k)} \hat{J}^{*i}_{k,k'}$, where K_i is the number of clusters of the i-th data set, $i = 1, \ldots, 81$. We denote these separation indices as the *median separation indices*. We put all the sample farthest separation indices of the 81 data sets with close cluster structures into the set $S_{\hat{J}cf}$. Similarly, the set $S_{\hat{J}sf}$ for separated, $S_{\hat{J}wf}$ for well-separated. Similarly, we put the sample median separation indices into the sets $S_{\hat{J}cm}$, $S_{\hat{J}sm}$, and

Table 2. The sample means and standard deviations of the sets $S_{\hat{J}_c}$, $S_{\hat{J}_s}$, and $S_{\hat{J}_w}$ as well as the corresponding estimates of biases and squared roots of MSEs of J_0.

J_0	mean (sd)	bias	$\sqrt{\text{MSE}}$
0.010	0.013 (0.023)	0.003	0.023
0.210	0.213 (0.020)	0.003	0.020
0.342	0.345 (0.017)	0.003	0.018

$S_{\hat{J}_{wm}}$ respectively. The means and standard deviations of these sets are listed in Tables 3 and 4.

Table 3 shows that the median separation indices tend to be close to the specified degree of separation. This is desirable since we want the separation indices between clusters and their direct neighboring clusters to be as close to the specified degree of separation as possible.

However, the farthest separation indices in Table 4 tend to be much larger than the specified separation indices; this is because if $K > p$, then not all cluster centers can be neighboring vertices of a simplex.

5. Illustration of Use of Factorial Experiment Design

In this section, we use the 243 simulated data sets generated by the design proposed in Section 3 to compare the performances of four number-of-cluster-estimation methods *CH*, *Silhouette*, *Hartigan*, and *KL* (see Tibshirani et al., 2001). We use a *modified kmeans* clustering method to obtain partitions. In this *modified kmeans* method, we first obtain 10 partitions by using *kmeans* method given the number of clusters. Then we choose as the final partition the partition which has the minimum average within cluster distance.

For the *CH*, *Silhouette* and *KL* methods, we first obtain 19 partitions with consecutive numbers of clusters starting from 2 and ending with 20. Then we choose the number of clusters which optimizes the *CH* index, *Silhouette* index, or *KL* index. For the *Hartigan* method, we obtain a sequence of partitions with the number of clusters starting from 2 until the *Hartigan* index is less than or equal to 10.

To measure the performances of these number-of-cluster-estimation methods, we record the numbers and sizes of underestimates and overestimates. Denote δ as the difference $\hat{K} - K$, where K is the true number of clusters (i.e., the number of component distributions of the mixture of normal distributions used to generate the data set) and $\hat{K}$ is the estimated number of clusters. If $\hat{K} < K$, then the size of underestimate is $-\delta$. If $\hat{K} > K$, then the size of overestimate is δ. We also calculate the values of five external indices (Rand index, Hubert and Arabie's adjusted Rand index, Morey and Agresti's adjusted Rand index,

Table 3. Means and standard deviations of the median separation indices.

J_0	mean (sd)	set
0.010	0.076 (0.053)	$S_{\hat{J}_{cm}}$
0.210	0.267 (0.048)	$S_{\hat{J}_{sm}}$
0.342	0.397 (0.043)	$S_{\hat{J}_{wm}}$

Table 4. Means and standard deviations of the farthest separation indices.

J_0	mean (sd)	set
0.010	0.167 (0.113)	$S_{\hat{J}_{cf}}$
0.210	0.348 (0.094)	$S_{\hat{J}_{sf}}$
0.342	0.466 (0.083)	$S_{\hat{J}_{wf}}$

Fowlkes and Mallows index, and Jaccard index (Milligan 1986)) to measure the agreements between the obtained partitions with the true partitions. The closer to 1 the values of the external indices are, the better the agreements are. The perfect agreement has index value 1.

The total numbers and sizes of underestimates and overestimates of the number of clusters for the 243 data sets are summarized in Table 5. The second and third columns are the total numbers (sizes) of underestimates and overestimates of the number of clusters for the 243 data sets in which noisy variables are deleted, while the fourth and fifth columns are results obtained with all noisy variables.

The average values and corresponding standard errors of the five external indices are summarized in Tables 6 and 7. Tables 5, 6 and 7 show that the magnitude of gaps will affect the recovery of the true cluster structures. As the magnitude of gaps decreases, the performances of the number-of-cluster-estimation methods get worse. Also noisy variables will affect the recovery of the true cluster structures. This simulation study also shows that the generated data sets are challenging. The *Hartigan* and *KL* methods do poorly with overestimation even for separated cluster structures and no noise. The *CH* method does not overestimate but underestimates the number of clusters when the clusters are close and when there is noise (this property is not bad as these are the situations where we expect it is harder to find boundaries among clusters). Overall, the *Silhouette* method is best but can overestimate as well as underestimate.

In several previous simulation studies (e.g., Brusco and Cradit 2001; Milligan 1988; Steinley 2004), the results of the simulation studies were analyzed via ANOVA. To see the effect of the different factors, we also calculate the ANOVA tables. Although the values for different external indices sometimes are quite different, the patterns of the values of the external indices across dif-

Table 5. The numbers and sizes of underestimates and overestimates for the 243 data sets (m_- and s_- are total the number and size of underestimates while m_+ and s_+ are the total number and size of overestimates). The clustering method used is a modified *kmeans* method.

method	without noise		with noise	
	m_- (s_-)	m_+ (s_+)	m_- (s_-)	m_+ (s_+)
close cluster structure				
CH	36 (189)	0 (0)	61 (271)	0 (0)
Silhouette	8 (29)	10 (13)	18 (84)	7 (14)
Hartigan	0 (0)	81 (1251)	0 (0)	81 (1023)
KL	6 (31)	28 (179)	9 (40)	41 (392)
separated cluster structure				
CH	0 (0)	0 (0)	22 (101)	0 (0)
Silhouette	0 (0)	0 (0)	10 (52)	6 (34)
Hartigan	0 (0)	81 (827)	0 (0)	81 (975)
KL	1 (4)	17 (100)	9 (46)	31 (261)
well-separated cluster structure				
CH	0 (0)	2 (2)	17 (91)	1 (1)
Silhouette	0 (0)	2 (2)	9 (48)	3 (13)
Hartigan	0 (0)	81 (758)	0 (0)	81 (978)
KL	1 (7)	10 (63)	10 (52)	28 (226)

ferent number-of-cluster-estimation methods are similar. So we only consider the factor effects on the Hubert and Arabie's adjusted Rand index.

Table 8 has a partial ANOVA table for a linear model that includes all 4 factors shown in Table 1 plus the number-of-cluster estimation methods, and up to third order interactions. Note the the summary table is informative, even though the homoscedasticity assumption for the linear model is not valid. The main effects are all highly statistically significant, and more meaningful is the effect size, given in the last column. The effect size η^2 (Kirk 1982; Tabachnick and Fidell 1989) is the proportion of variance in the dependent variable that is attributed to a factor or interaction. That is, $\eta^2 = SS_{\text{factor}}/SS_{\text{total}}$. Table 8 includes also the few interactions with the largest F ratios and η^2.

We can see from the ANOVA table that as expected *a priori*, that the factor degree of separation has a large effect size; also the number of non-noisy variables and the number-of-cluster estimation methods have large effects on the adjusted Rand index.

6. Discussion

In this article, we use the degree of separation among clusters based on the separation index proposed by Qiu and Joe (2006) to develop a cluster generating algorithm which can generate clusters with a specified degree of separa-

Table 6. The average values (corresponding standard errors) of the five external indices* for the 243 data sets without noisy variables.

method	HA	MA	Rand	FM	Jaccard
close cluster structure					
CH	0.54 (0.32)	0.54 (0.32)	0.78 (0.18)	0.68 (0.22)	0.53 (0.28)
Silhouette	0.74 (0.17)	0.74 (0.17)	0.90 (0.11)	0.81 (0.11)	0.68 (0.15)
Hartigan	0.32 (0.09)	0.33 (0.09)	0.83 (0.08)	0.73 (0.06)	0.24 (0.06)
KL	0.65 (0.23)	0.65 (0.23)	0.88 (0.11)	0.73 (0.16)	0.58 (0.21)
separated cluster structure					
CH	0.98 (0.01)	0.98 (0.01)	1.00 (0.00)	0.99 (0.01)	0.97 (0.01)
Silhouette	0.98 (0.01)	0.98 (0.01)	1.00 (0.00)	0.99 (0.01)	0.97 (0.01)
Hartigan	0.50 (0.17)	0.50 (0.16)	0.86 (0.09)	0.62 (0.11)	0.40 (0.14)
KL	0.90 (0.19)	0.90 (0.19)	0.97 (0.08)	0.92 (0.13)	0.87 (0.21)
well-separated cluster structure					
CH	1.00 (0.01)	1.00 (0.01)	1.00 (0.00)	1.00 (0.01)	1.00 (0.02)
Silhouette	1.00 (0.01)	1.00 (0.01)	1.00 (0.00)	1.00 (0.01)	1.00 (0.01)
Hartigan	0.53 (0.18)	0.53 (0.18)	0.87 (0.09)	0.64 (0.12)	0.43 (0.16)
KL	0.94 (0.17)	0.94 (0.17)	0.98 (0.07)	0.96 (0.12)	0.93 (0.19)

* HA, MA, Rand, FM, and Jaccard represent Hubert and Arabie's adjusted Rand index, Morey and Agresti's adjusted Rand index, Rand index, Fowlkes and Mallows index, and Jaccard index, respectively.

tion. An application to the estimation of the number of clusters shows that the generated cluster structures are challenging.

The design proposed in Section 5 is just a simple example of an experiment design which is based on our random cluster generation algorithm proposed in Section 2. Other designs can be considered. For example, we can add cluster size and proportion of outliers as two additional factors in the design.

We didn't mention all the details of the random cluster generation algorithm. More detailed information can be found in the help files of the R package *clusterGeneration* that we wrote to implement the random cluster generation algorithm proposed in this article. *clusterGeneration* allows the user to have more control over several factors. For example, the user is allowed (1) to generate cluster size randomly from a range; (2) or to generate clusters with equal size; or (3) to specify each cluster size.

Currently, we assume that all variables are continuous type and clusters are symmetric about their centers. To make the generated cluster structures closer to real data sets, we will investigate in our future research on how to generate cluster structures with mixed-type variables and clusters of other shapes.

Table 7. The average values (corresponding standard errors) of the five external indices* with noisy variables

method	HA	MA	Rand	FM	Jaccard
close cluster structure					
CH	0.37 (0.30)	0.37 (0.30)	0.69 (0.17)	0.57 (0.22)	0.40 (0.26)
Silhouette	0.65 (0.25)	0.66 (0.25)	0.86 (0.15)	0.75 (0.16)	0.61 (0.21)
Hartigan	0.30 (0.07)	0.31 (0.07)	0.83 (0.08)	0.44 (0.06)	0.23 (0.04)
KL	0.47 (0.27)	0.47 (0.27)	0.83 (0.12)	0.58 (0.21)	0.41 (0.24)
separated cluster structure					
CH	0.80 (0.32)	0.80 (0.32)	0.90 (0.17)	0.86 (0.21)	0.79 (0.30)
Silhouette	0.86 (0.27)	0.86 (0.27)	0.94 (0.14)	0.90 (0.18)	0.85 (0.26)
Hartigan	0.43 (0.14)	0.43 (0.14)	0.85 (0.08)	0.56 (0.10)	0.33 (0.12)
KL	0.68 (0.34)	0.68 (0.34)	0.90 (0.13)	0.75 (0.26)	0.65 (0.35)
well-separated cluster structure					
CH	0.84 (0.31)	0.84 (0.31)	0.92 (0.16)	0.89 (0.21)	0.84 (0.30)
Silhouette	0.90 (0.25)	0.90 (0.25)	0.95 (0.13)	0.94 (0.16)	0.90 (0.24)
Hartigan	0.44 (0.16)	0.45 (0.15)	0.85 (0.09)	0.57 (0.11)	0.35 (0.13)
KL	0.71 (0.34)	0.71 (0.34)	0.90 (0.14)	0.78 (0.25)	0.69 (0.35)

* HA, MA, Rand, FM, and Jaccard represent Hubert and Arabie's adjusted Rand index, Morey and Agresti's adjusted Rand index, Rand index, Fowlkes and Mallows index, and Jaccard index, respectively.

Appendix

A. Finding the Optimal Projection Direction

In this section, we give a brief description on how to find the optimal projection direction for the optimization problem

$$\boldsymbol{a}^* = \arg \max_{\boldsymbol{a}^T(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) > 0} J_{12}(\boldsymbol{a}). \qquad (A.1)$$

A detailed proof can be found in Qiu and Lee (2005). The idea is to first transform the constrained optimization problem (A.1) into an unconstrained optimization problem

$$\min_{\boldsymbol{y}} g(\boldsymbol{y}), \qquad (A.2)$$

where

$g(\boldsymbol{y}) = \sqrt{g_1(\boldsymbol{y})} + \sqrt{g_2(\boldsymbol{y})}$,
$g_1(\boldsymbol{y}) = \boldsymbol{y}^T\boldsymbol{y} + 1$,

Table 8. ANOVA for Hubert and Arabie's adjusted Rand index

	Df	Sum Sq	Mean Sq	F value	η^2
#C = No. of clusters	2	0.785	0.392	17	0.007
DS = Degree of Separation	2	14.504	7.252	314	0.137
#NNV = No. of non-noisy vars.	2	16.113	8.056	349	0.152
#NV = No. of noisy variables	2	1.218	0.609	26	0.011
#CEM = #Cluster est. method	3	21.597	7.199	312	0.203
#C × #NNV	4	7.469	1.867	81	0.070
DS × #NNV	4	1.855	0.464	20	0.017
#C × #CEM	6	7.250	1.208	52	0.068
DS × #CEM	6	2.865	0.478	21	0.027
#NNV × #CEM	6	3.352	0.559	24	0.032
Residuals	808	18.661	0.023		

$g_2(\boldsymbol{y}) = \left(\boldsymbol{y} + \boldsymbol{V}_{22}^{-1}\boldsymbol{v}_{21}\right)^T \boldsymbol{V}_{22}\left(\boldsymbol{y} + \boldsymbol{V}_{22}^{-1}\boldsymbol{v}_{21}\right) + c_2$,
$c_2 = v_{11} - \boldsymbol{v}_{21}^T\boldsymbol{V}_{22}^{-1}\boldsymbol{v}_{21}$,
and $\boldsymbol{V} = \boldsymbol{Q}_2^T\boldsymbol{Q}_1^T\boldsymbol{\Sigma}_2\boldsymbol{Q}_1\boldsymbol{Q}_2 = \begin{pmatrix} v_{11} & \boldsymbol{v}_{12} \\ \boldsymbol{v}_{21} & \boldsymbol{V}_{22} \end{pmatrix}$,
$\boldsymbol{Q}_1$ is a $p \times p$ nonsingular matrix such that $\boldsymbol{Q}_1^T\boldsymbol{\Sigma}_1\boldsymbol{Q}_1 = \boldsymbol{I}_G$, $\boldsymbol{Q}_2$ is a $p \times p$ orthogonal matrix such that $\boldsymbol{Q}_2^T\left[\boldsymbol{Q}_1^T(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)\right] = c_1\boldsymbol{e}_1$, $\boldsymbol{e}_1$ is a $p \times 1$ vector whose elements are all equal to zero except the first element is equal to 1, $c_1 = \|\boldsymbol{Q}_2^T\boldsymbol{Q}_1^T(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)\| > 0$, and the norm $\|\boldsymbol{z}\|$ is defined as $\sqrt{\boldsymbol{z}^T\boldsymbol{z}}$.

We can show that the objective function of this unconstrained optimization problem (A.2) is a strictly convex function if the $p \times p$ covariance matrices $\boldsymbol{\Sigma}_1$ and $\boldsymbol{\Sigma}_2$ are positive definite and that the $p \times 1$ mean vectors $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ are different. Thus the optimization problem (A.2) has a unique critical point and the unique critical point is the minimum point. Hence the optimization problem (A.1) has the unique maximum point.

We can use the Newton-Raphson method to obtain the minimum point of (A.2). The initial value of $\boldsymbol{y}$ can be taken as $\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1$.

B. Generating Vertices of a p-Dimensional Simplex

We first describe how to obtain the third vertex for the p-dimensional simplex. Since the simplex is equilateral, we have $(\boldsymbol{v}_3 - \boldsymbol{v}_1)^T(\boldsymbol{v}_3 - \boldsymbol{v}_1) = 4$, $(\boldsymbol{v}_3 - \boldsymbol{v}_2)^T(\boldsymbol{v}_3 - \boldsymbol{v}_2) = 4$. By adding the two equations, we can obtain $\boldsymbol{v}_3^T\boldsymbol{v}_3 - 2\boldsymbol{v}_3\bar{\boldsymbol{v}}_2 = 4 - \frac{1}{2}\sum_{i=1}^2 \boldsymbol{v}_k^T\boldsymbol{v}_k$, where $\bar{\boldsymbol{v}}_2 = \frac{1}{2}\sum_{i=1}^2 \boldsymbol{v}_k = (\bar{v}_{21}, \ldots, \bar{v}_{2p})^T$. Let $v_{31} = \bar{v}_{21}$ and $v_{33} = \cdots = v_{3p} = 0$. Then we can get $v_{32}^2 = 4 - \frac{1}{2}\sum_{i=1}^2 \boldsymbol{v}_k^T\boldsymbol{v}_k + \bar{v}_{21}^2 + 2v_{32}\bar{v}_{22}$. Note that $\bar{v}_{22} = 0$. Hence

$$v_{32} = \left\{4 - \left[\tfrac{1}{2}\sum_{k=1}^{2} \boldsymbol{v}_k^T \boldsymbol{v}_k - \bar{\boldsymbol{v}}_2^T \bar{\boldsymbol{v}}_2\right]\right\}^{1/2}$$
$$= \left\{4 - \tfrac{1}{2}\sum_{k=1}^{2} \left(\boldsymbol{v}_k - \bar{\boldsymbol{v}}_2\right)^T \left(\boldsymbol{v}_k - \bar{\boldsymbol{v}}_2\right)\right\}^{1/2}.$$

By using the same technique, we can obtain the coordinates of the k-th vertex $\boldsymbol{v}_k$ given $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{k-1}$, $2 < k \leq p+1$ (see Formula (2.1)).

References

ATLAS, R.S. and OVERALL, J.E. (1994), "Comparative Evaluation of Two Superior Stopping Rules for Hierarchical Cluster Analysis", *Psychometrika, 59*, 581–591.

BLASHFIELD, R.K. (1976), "Mixture Model Test of Cluster Analysis: Accuracy of Four Agglomerative Hierarchical Methods", *Psychological Bulletin, 83*, 377–388.

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

CARIO, M.C. and NELSON, B.L.(1997), "Modeling and Generating Random Vectors with Arbitrary Marginal Distributions and Correlation Matrix", Tech. rep., Department of Industrial Engineering and Management Sciences, Northwestern University, Evanston, Ill.

DEVROYE, L. (1986), *Non-Uniform Random Variate Generation*, Springer-Verlag: New York.

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

GNANADESIKAN, R., KETTENRING, J.R., and TSAO, S.L.(1995), "Weighting and Selection of Variables for Cluster Analysis", *Journal of Classification, 12*, 113–136.

GORDON, A. D. (1981), *Classification: Methods for the Exploratory Analysis of Multivariate Data*, Chapman & Hall.

GUHA, S., RASTOGI, R., and SHIM K. (1998), "CURE: An Efficient Clustering Algorithm for Large Databases", in *Proceedings of the ACM SIGMOD Conference on Management of Data*, pp. 73–84.

JOE, H. (2006), "Generating Random Correlation Matrices Based on Partial Correlations", *Journal of Multivariate Analysis*, in press.

KIRK, R.E. (1982), *Experimental Design: Procedures for the Behavioral Sciences* (2nd ed.), Belmont, CA: Brooks/Cole.

KOTZ, S. and JOHNSON, N.L. (eds.) (1983), *Encyclopedia of Statistical Sciences*, New York: Wiley.

MILLIGAN, G. W. (1985), "An Algorithm for Generating Artificial Test Clusters", *Psychometrika, 50*, 123–127.

MILLIGAN, G.W. (1989), "A Validation Study of a Variable Weighting Algorithm for Cluster Analysis", *Journal of Classification, 6*, 53–71.

MILLIGAN, G.W. 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.

QIU, W.-L. and JOE, H. (2006), "Separation Index and Partial Membership for Clustering", *Computational Statistics and Data Analysis, 50*, 585–603.

QIU, W.-L. and LEE, M.-L. T. (2005), "A New Classification Method Based on a Separation Index, with Applications in Genomics", Submitted.

STEINLEY, D. (2003), "Local Optima in k-means Clustering: What You Don't Know May Hurt You", *Psychological Methods, 8*, 294–304.

STEINLEY, D. (2004), "Properties of the Hubert-Arabie Adjusted Rand Index", *Psychological Methods, 9*, 386–396.

TABACHNICK, B.G. (1989), *Using Multivariate Statistics* (2nd ed.), New York: Harper & Row.

TIBSHIRANI, R., WALTHER, G., and HASTIE, T. (2001), "Estimating the Number of Clusters in a Dataset Via the Gap Statistic". *Journal of the Royal Statistical Society: Series B, 63*, 411–423.

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

WALLER, N.G., UNDERHILL, J.M., and KAISER, H. (1999), "A Method for Generating Simulated Plasmodes and Artifical Test Clusters with User-defined Shape, Size, and Orientation", *Multivariate Behavioral Research, 34*, 123–142.

ZHANG, T., RAMAKRISHNAN, R., and LIVNY, M. (1997), "A New Data Clustering Algorithm and its Applications", *Data Mining and Knowledge Discovery, 1*, 141–182.