Package 'clusterGeneration'

May 8, 2026

Version 1.3.8

Date 2023-08-15

Title Random Cluster Generation (with Specified Degree of Separation)

Author Weiliang Qiu <weiliang.qiu@gmail.com>, Harry Joe
<harry@stat.ubc.ca>.

Maintainer Weiliang Qiu <weiliang.qiu@gmail.com>

Depends R (>= 3.5.0), MASS

Description We developed the clusterGeneration package to provide functions
for generating random clusters, generating random
covariance/correlation matrices,
calculating a separation index (data and population version)
for pairs of clusters or cluster distributions, and 1-D and 2-D
projection plots to visualize clusters. The package also
contains a function to generate random clusters based on
factorial designs with factors such as degree of separation,
number of clusters, number of variables, number of noisy
variables.

License GPL (>= 2)

Repository CRAN

Date/Publication 2023-08-16 04:20:02 UTC

NeedsCompilation no

Contents

`genOrthogonal` *Generate An Orthogonal Matrix*

Description

Generate an orthogonal matrix with given dimension.

Usage

```
genOrthogonal(dim)
```

Arguments

`dim`	integer. Dimension of the orthogonal matrix.

Value

An orthogonal matrix with dimension `dim`.

Examples

```
set.seed(12345)
Q = genOrthogonal(3)
print(Q)
A = Q
print(A)
```

`genPositiveDefMat` *GENERATE A POSITIVE DEFINITE MATRIX/COVARIANCE MATRIX*

Description

Generate a positive definite matrix/covariance matrix.

Usage

```
genPositiveDefMat(
  dim,
  covMethod = c("eigen", "onion", "c-vine", "unifcorrmat"),
  eigenvalue = NULL,
  alphad = 1,
  eta = 1,
  rangeVar = c(1, 10),
  lambdaLow = 1,
  ratioLambda = 10)
```

Arguments

`dim`	Dimension of the matrix to be generated.
`covMethod`	Method to generate positive definite matrices/covariance matrices. Choices are "eigen", "onion", "c-vine", or "unifcorrmat"; see details below.
`eigenvalue`	numeric. user-specified eigenvalues when `covMethod = "eigen"`. If `eigenvalue = NULL` and `covMethod = "eigen"`, then eigenvalues will be automatically generated.
`alphad`	parameter for unifcorrmat method to generate random correlation matrix `alphad=1` for uniform. `alphad` should be positive.
`eta`	parameter for "c-vine" and "onion" methods to generate random correlation matrix `eta=1` for uniform. `eta` should be positive.
`rangeVar`	Range for variances of a covariance matrix (see details). The default range is $[1, 10]$ which can generate reasonable variability of variances.
`lambdaLow`	Lower bound on the eigenvalues of cluster covariance matrices. If the argument `covMethod="eigen"`, eigenvalues are generated for cluster covariance matrices. The eigenvalues are randomly generated from the interval [`lambdaLow`, `lambdaLow`$*$`ratioLambda`]. In our experience, `lambdaLow`$= 1$ and `ratioLambda`$= 10$ can give reasonable variability of the diameters of clusters. `lambdaLow` should be positive.
`ratioLambda`	The ratio of the upper bound of the eigenvalues to the lower bound of the eigenvalues of cluster covariance matrices. See `lambdaLow`.

Details

The current version of the function `genPositiveDefMat` implements four methods to generate random covariance matrices. The first method, denoted by "eigen", first randomly generates eigenvalues $(\lambda_1, \ldots, \lambda_p)$ for the covariance matrix ($\boldsymbol{\Sigma}$), then uses columns of a randomly generated orthogonal matrix ($\boldsymbol{Q} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p)$) as eigenvectors. The covariance matrix $\boldsymbol{\Sigma}$ is then contructed as $\boldsymbol{Q} * diag(\lambda_1, \ldots, \lambda_p) * \boldsymbol{Q}^T$.

The remaining methods, denoted as "onion", "c-vine", and "unifcorrmat" respectively, first generates a random correlation matrix ($\boldsymbol{R}$) via the method mentioned and proposed in Joe (2006), then randomly generates variances $(\sigma_1^2, \ldots, \sigma_p^2)$ from an interval specified by the argument `rangeVar`. The covariance matrix $\boldsymbol{\Sigma}$ is then constructed as $diag(\sigma_1, \ldots, \sigma_p) * \boldsymbol{R} * diag(\sigma_1, \ldots, \sigma_p)$.

Value

egvalues	eigenvalues of Sigma
Sigma	positive definite matrix/covariance matrix

Author(s)

Weiliang Qiu <weiliang.qiu@gmail.com>
Harry Joe <harry@stat.ubc.ca>

References

Joe, H. (2006) Generating Random Correlation Matrices Based on Partial Correlations. *Journal of Multivariate Analysis*, **97**, 2177–2189.

Ghosh, S., Henderson, S. G. (2003). Behavior of the NORTA method for correlated random vector generation as the dimension increases. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, **13(3)**, 276–294.

Kurowicka and Cooke, 2006. *Uncertainty Analysis with High Dimensional Dependence Modelling*, Wiley, 2006.

Examples

```
genPositiveDefMat(
  dim = 4,
  covMethod = "unifcorrmat")

aa <- genPositiveDefMat(
dim = 3,
covMethod = "eigen",
eigenvalue = c(3, 2, 1))
print(aa)
print(eigen(aa$Sigma))
```

---

genRandomClust *RANDOM CLUSTER GENERATION WITH SPECIFIED DEGREE OF SEPARATION*

Description

Generate cluster data sets with specified degree of separation. The separation between any cluster and its nearest neighboring cluster can be set to a specified value. The covariance matrices of clusters can have arbitrary diameters, shapes and orientations.

Usage

```
genRandomClust(numClust,
               sepVal = 0.01,
               numNonNoisy = 2,
               numNoisy = 0,
               numOutlier = 0,
               numReplicate = 3,
               fileName = "test",
               clustszind = 2,
               clustSizeEq = 50,
               rangeN = c(50,200),
               clustSizes = NULL,
               covMethod = c("eigen", "onion", "c-vine", "unifcorrmat"),
       eigenvalue = NULL,
               rangeVar = c(1, 10),
               lambdaLow = 1,
               ratioLambda = 10,
               alphad = 1,
               eta = 1,
               rotateind = TRUE,
               iniProjDirMethod = c("SL", "naive"),
               projDirMethod = c("newton", "fixedpoint"),
               alpha = 0.05,
               ITMAX = 20,
               eps = 1.0e-10,
               quiet = TRUE,
               outputDatFlag = TRUE,
               outputLogFlag = TRUE,
               outputEmpirical = TRUE,
               outputInfo = TRUE)
```

Arguments

numClust	Number of clusters in a data set.
sepVal	Desired value of the separation index between a cluster and its nearest neighboring cluster. Theoretically, sepVal can take values within the interval $[-1, 1)$ (In practice, we set sepVal in $(-0.999, 0.999)$). The closer to 1 sepVal is, the more separated clusters are. The default value is 0.01 which is the value of the separation index for two univariate clusters generated from $N(0, 1)$ and $N(0, A)$, respectively, where $A = 4$. sepVal$= 0.01$ indicates a close cluster structure. sepVal$= 0.21(A = 6)$ indicates a separated cluster structure. sepVal$= 0.34(A = 8)$ indicates a well-separated cluster.
numNonNoisy	Number of non-noisy variables.
numNoisy	Number of noisy variables. The default values of numNoisy and numOutlier are 0 so that we get *clean* data sets.
numOutlier	Number or ratio of outliers. If numOutlier is a positive integer, then numOutlier means the number of outliers. If numOutlier is a real number between $(0, 1)$,

then numOutlier means the ratio of outliers, i.e. the number of outliers is equal to round(numOutlier$*n_1$), where n_1 is the total number of non-outliers. If numOutlier is a real number greater than 1, then numOutlier to rounded to an integer. The default values of numNoisy and numOutlier are 0 so that we get 'clean' data sets.

numReplicate — Number of data sets to be generated for the same cluster structure specified by the other arguments of the function genRandomClust. The default value 3 follows the design in Milligan (1985).

fileName — The first part of the names of data files that record the generated data sets and associated information, such as cluster membership of data points, labels of noisy variables, separation index matrix, projection directions, etc. (see details). The default value of fileName is 'test'.

clustszind — Cluster size indicator. clustszind$= 1$ indicates that all cluster have equal size. The size is specified by the argument clustSizeEq. clustszind$= 2$ indicates that the cluster sizes are randomly generated from the range specified by the argument rangeN. clustszind$= 3$ indicates that the cluster sizes are specified via the vector clustSizes. The default value is 2 so that the generated clusters are more realistic.

clustSizeEq — Cluster size. If the argument clustszind$= 1$, then all clusters will have the equal number clustSizeEq of data points. The value of clustSizeEq should be large enough to get non-singular cluster covariance matrices. We recommend the clustSizeEq is at least $10 * p$, where p is the total number of variables (including both non-noisy and noisy variables). The default value 100 is a reasonable cluster size.

rangeN — The range of cluster sizes. If clustszind$= 2$, then cluster sizes will be randomly generated from the range specified by rangeN. The lower bound of the number of clusters should be large enough to get non-singular cluster covariance matrices. We recommend the minimum cluster size is at least $10 * p$, where p is the total number of variables (including both non-noisy and noisy variables). The default range is $[50, 200]$ which can produce reasonable variability of cluster sizes.

clustSizes — The sizes of clusters. If clustszind$= 3$, then cluster sizes will be specified via the vector clustSizes. We recommend the minimum cluster size is at least $10 * p$, where p is the total number of variables (including both non-noisy and noisy variables). The user needs to specify the value of clustSizes. Therefore, we set the default value of clustSizes as NULL.

covMethod — Method to generate covariance matrices for clusters (see details). The default method is 'eigen' so that the user can directly specify the range of the *diameters* of clusters.

eigenvalue — numeric. user-specified eigenvalues when covMethod = "eigen". If eigenvalue = NULL and covMethod = "eigen", then eigenvalues will be automatically generated.

rangeVar — Range for variances of a covariance matrix (see details). The default range is $[1, 10]$ which can generate reasonable variability of variances.

lambdaLow — Lower bound of the eigenvalues of cluster covariance matrices. If the argument "covMethod="eigen"", we need to generate eigenvalues for cluster covariance matrices. The eigenvalues are randomly generated from the interval

[lambdaLow, lambdaLow∗ratioLambda]. In our experience, lambdaLow= 1 and ratioLambda= 10 can give reasonable variability of the diameters of clusters. lambdaLow should be positive.

ratioLambda — The ratio of the upper bound of the eigenvalues to the lower bound of the eigenvalues of cluster covariance matrices. If the argument covMethod="eigen", we need to generate eigenvalues for cluster covariance matrices. The eigenvalues are randomly generated from the interval [lambdaLow, lambdaLow∗ratioLambda]. In our experience, lambdaLow= 1 and ratioLambda= 10 can give reasonable variability of the diameters of clusters. ratioLambda should be larger than 1.

alphad — parameter for unifcorrmat method to generate random correlation matrix alphad=1 for uniform. alphad should be positive.

eta — parameter for "c-vine" and "onion" methods to generate random correlation matrix eta=1 for uniform. eta should be positive.

rotateind — Rotation indicator. rotateind=TRUE indicates randomly rotating data in non-noisy dimensions so that we may not detect the full cluster structure from pairwise scatter plots of the variables.

iniProjDirMethod — Indicating the method to get initial projection direction when calculating the separation index between a pair of clusters (c.f. Qiu and Joe, 2006a, 2006b).
iniProjDirMethod="SL", the default, indicates the initial projection direction is the sample version of the SL's projection direction (Su and Liu, 1993, JASA) $(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$
iniProjDirMethod="naive" indicates the initial projection direction is $\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1$

projDirMethod — Indicating the method to get the optimal projection direction when calculating the separation index between a pair of clusters (c.f. Qiu and Joe, 2006a, 2006b).
projDirMethod="newton" indicates we use the modified Newton-Raphson method to search the optimal projection direction (c.f. Qiu and Joe, 2006a). This requires the assumptions that both covariance matrices of the pair of clusters are positive-definite. If this assumption is violated, the "fixedpoint" method could be used. The "fixedpoint" method iteratively searches the optimal projection direction based on the first derivative of the separation index to the projection direction (c.f. Qiu and Joe, 2006b).

alpha — Tuning parameter reflecting the percentage in the two tails of a projected cluster that might be outlying. We set alpha= 0.05 like we set the significance level in hypothesis testing as 0.05.

ITMAX — Maximum iteration allowed when iteratively calculating the optimal projection direction. The actual number of iterations is usually much less than the default value 20.

eps — Convergence threshold. A small positive number to check if a quantitiy q is equal to zero. If $|q| <$eps, then we regard q is equal to zero. eps is used to check if an algorithm converges. The default value is $1.0e-10$.

quiet — A flag to switch on/off the outputs of intermediate results and/or possible warning messages. The default value is TRUE.

outputDatFlag — Indicates if data set should be output to file.

outputLogFlag — Indicates if log info should be output to file.

`outputEmpirical`
: Indicates if empirical separation indices and projection directions should be calculated. This option is useful when generating clusters with sizes which are not large enough so that the sample covariance matrices may be singular. Hence, by default, `outputEmpirical=TRUE`.

`outputInfo`
: Indicates if theoretical and empirical separation information data frames should be output to a file with format `[fileName]_info.log`.

Details

The function `genRandomClust` is an implementation of the random cluster generation method proposed in Qiu and Joe (2006a) which improve the cluster generation method proposed in Milligan (1985) so that the degree of separation between any cluster and its nearest neighboring cluster 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 separation between a pair of clusters is measured by the separation index proposed in Qiu and Joe (2006b).

The current version of the function `genRandomClust` implements two methods to generate covariance matrices for clusters. The first method, denoted by `eigen`, first randomly generates eigenvalues $(\lambda_1, \ldots > \lambda_p)$ for the covariance matrix ($\boldsymbol{\Sigma}$), then uses columns of a randomly generated orthogonal matrix ($\boldsymbol{Q} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p)$) as eigenvectors. The covariance matrix $\boldsymbol{\Sigma}$ is then contructed as $\boldsymbol{Q} * diag(\lambda_1, \ldots, \lambda_p) * \boldsymbol{Q}^T$. The second method, denoted as "unifcorrmax", first generates a random correlation matrix ($\boldsymbol{R}$) via the method proposed in Joe (2006), then randomly generates variances $(\sigma_1^2, \ldots, \sigma_p^2)$ from an interval specified by the argument `rangeVar`. The covariance matrix $\boldsymbol{\Sigma}$ is then constructed as $diag(\sigma_1, \ldots, \sigma_p) * \boldsymbol{R} * diag(\sigma_1, \ldots, \sigma_p)$.

For each data set generated, the function `genRandomClust` outputs four files: data file, log file, membership file, and noisy set file. All four files have the same format: `[fileName]_[i].[extension]`, where i indicates the replicate number, and '`extension`' can be '`dat`', '`log`', '`mem`', and '`noisy`'.

The data file with file extension '`dat`' contains $n + 1$ rows and p columns, where n is the number of data points and p is the number of variables. The first row is the variable names. The log file with file extension '`log`' contains information such as cluster sizes, mean vectors, covariance matrices, projection directions, separation index matrices, etc. The membership file with file extension '`mem`' contains n rows and one column of cluster memberships for data points. The noisy set file with file extension '`noisy`' contains a row of labels of noisy variables.

When generating clusters, population covariance matrices are all positive-definite. However sample covariance matrices might be semi-positive-definite due to small cluster sizes. In this case, the function `genRandomClust` will automatically use the "fixedpoint" method to search the optimal projection direction.

The current version of the function `genPositiveDefMat` implements four methods to generate random covariance matrices. The first method, denoted by "eigen", first randomly generates eigenvalues $(\lambda_1, \ldots, \lambda_p)$ for the covariance matrix ($\boldsymbol{\Sigma}$), then uses columns of a randomly generated orthogonal matrix ($\boldsymbol{Q} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p)$) as eigenvectors. The covariance matrix $\boldsymbol{\Sigma}$ is then contructed as $\boldsymbol{Q} * diag(\lambda_1, \ldots, \lambda_p) * \boldsymbol{Q}^T$.

The remaining methods, denoted as "onion", "c-vine", and "unifcorrmat" respectively, first generates a random correlation matrix ($\boldsymbol{R}$) via the method mentioned and proposed in Joe (2006), then randomly generates variances $(\sigma_1^2, \ldots, \sigma_p^2)$ from an interval specified by the argument `rangeVar`. The covariance matrix $\boldsymbol{\Sigma}$ is then constructed as $diag(\sigma_1, \ldots, \sigma_p) * \boldsymbol{R} * diag(\sigma_1, \ldots, \sigma_p)$.

Value

The function outputs four data files for each data set (see details).

This function also returns separation information data frames `infoFrameTheory` and `infoFrameData` based on population and empirical mean vectors and covariance matrices of clusters for all the data sets generated. Both `infoFrameTheory` and `infoFrameData` contain the following seven columns:

`Column 1:`	Labels of clusters $(1, 2, \ldots, numClust)$, where $numClust$ is the number of clusters for the data set.
`Column 2:`	Labels of the corresponding nearest neighbors.
`Column 3:`	Separation indices of the clusters to their nearest neighboring clusters.
`Column 4:`	Labels of the corresponding farthest neighboring clusters.
`Column 5:`	Separation indices of the clusters to their farthest neighbors.
`Column 6:`	Median separation indices of the clusters to their neighbors.
`Column 7:`	Data file names with format `[fileName]_[i]`, where i indicates the replicate number.

The function also returns three lists: `datList`, `memList`, and `noisyList`.

`datList:`	a list of data matrices for generated data sets.
`memList:`	a list of luster memberships for data points for generated data sets.
`noisyList:`	a list of sets of noisy variables for generated data sets.

Note

This function might be take a while to complete.

Author(s)

Weiliang Qiu <weiliang.qiu@gmail.com>
Harry Joe <harry@stat.ubc.ca>

References

Joe, H. (2006) Generating Random Correlation Matrices Based on Partial Correlations. *Journal of Multivariate Analysis*, **97**, 2177–2189.

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

Qiu, W.-L. and Joe, H. (2006a) Generation of Random Clusters with Specified Degree of Separaion. *Journal of Classification*, **23**(2), 315-334.

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

Su, J. Q. and Liu, J. S. (1993) Linear Combinations of Multiple Diagnostic Markers. *Journal of the American Statistical Association*, **88**, 1350–1355.

Ghosh, S., Henderson, S. G. (2003). Behavior of the NORTA method for correlated random vector generation as the dimension increases. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, **13(3)**, 276–294.

Kurowicka and Cooke, 2006. *Uncertainty Analysis with High Dimensional Dependence Modelling*, Wiley, 2006.

Examples

```
## Not run:
tmp1 <- genRandomClust(
       numClust = 7,
       sepVal = 0.3,
       numNonNoisy = 5,
                       numNoisy = 3,
       numOutlier = 5,
       numReplicate = 2,
       fileName = "chk1")

## End(Not run)
## Not run:
tmp2 <- genRandomClust(
       numClust = 7,
       sepVal = 0.3,
       numNonNoisy = 5,
                       numNoisy = 3,
       numOutlier = 5,
       numReplicate = 2,
                       covMethod = "unifcorrmat",
       fileName = "chk2")

## End(Not run)
## Not run:
tmp3 <- genRandomClust(
       numClust = 2,
       sepVal = -0.1,
       numNonNoisy = 2,
                       numNoisy = 6,
       numOutlier = 30,
       numReplicate = 1,
                       clustszind = 1,
       clustSizeEq = 80,
       rangeVar = c(10, 20),
                       covMethod = "unifcorrmat",
       iniProjDirMethod = "naive",
                       projDirMethod = "fixedpoint",
       fileName = "chk3")

## End(Not run)
```

getSepProj *OPTIMAL PROJECTION DIRECTION AND CORRESPONDING SEPARATION INDEX FOR PAIRS OF CLUSTERS*

Description

Optimal projection direction and corresponding separation index for pairs of clusters.

Usage

```
getSepProjTheory(
 muMat,
 SigmaArray,
                 iniProjDirMethod = c("SL", "naive"),
                 projDirMethod = c("newton", "fixedpoint"),
                 alpha = 0.05,
 ITMAX = 20,
 eps = 1.0e-10,
 quiet = TRUE)

getSepProjData(
       y,
       cl,
               iniProjDirMethod = c("SL", "naive"),
               projDirMethod = c("newton", "fixedpoint"),
               alpha = 0.05,
       ITMAX = 20,
       eps = 1.0e-10,
       quiet = TRUE)
```

Arguments

muMat	Matrix of mean vectors. Rows correspond to mean vectors for clusters.
SigmaArray	Array of covariance matrices. SigmaArray[,,i] record the covariance matrix of the i-th cluster.
y	Data matrix. Rows correspond to observations. Columns correspond to variables.
cl	Cluster membership vector.
iniProjDirMethod	Indicating the method to get initial projection direction when calculating the separation index between a pair of clusters (c.f. Qiu and Joe, 2006a, 2006b). iniProjDirMethod="SL" indicates the initial projection direction is the sample version of the SL's projection direction (Su and Liu, 1993) $(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$ iniProjDirMethod="naive" indicates the initial projection direction is $\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1$

projDirMethod	Indicating the method to get the optimal projection direction when calculating the separation index between a pair of clusters (c.f. Qiu and Joe, 2006a, 2006b). projDirMethod="newton" indicates we use the Newton-Raphson method to search the optimal projection direction (c.f. Qiu and Joe, 2006a). This requires the assumptions that both covariance matrices of the pair of clusters are positive-definite. If this assumption is violated, the "fixedpoint" method could be used. The "fixedpoint" method iteratively searches the optimal projection direction based on the first derivative of the separation index to the project direction (c.f. Qiu and Joe, 2006b).
alpha	Tuning parameter reflecting the percentage in the two tails of a projected cluster that might be outlying. We set alpha$= 0.05$ like we set the significance level in hypothesis testing as 0.05.
ITMAX	Maximum iteration allowed when to iteratively calculate the optimal projection direction. The actual number of iterations is usually much less than the default value 20.
eps	Convergence threshold. A small positive number to check if a quantitiy q is equal to zero. If $\|q\|$ <eps, then we regard q as equal to zero. eps is used to check if an algorithm converges. The default value is $1.0e-10$.
quiet	A flag to switch on/off the outputs of intermediate results and/or possible warning messages. The default value is TRUE.

Details

When calculating the optimal projection direction and corresponding optimal separation index for a pair of cluster, if one or both cluster covariance matrices is/are singular, the 'newton' method can not be used. In this case, the functions getSepProjTheory and getSepProjData will automatically use the 'fixedpoint' method to search the optimal projection direction, even if the user specifies the value of the argument projDirMethod as 'newton'. Also, multiple initial projection directions will be evaluated.

Specifically, $2 + 2p$ projection directions will be evaluated. The first projection direction is the "naive" direction $\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1$. The second projection direction is the "SL" projection direction $(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$. The next p projection directions are the p eigenvectors of the covariance matrix of the first cluster. The remaining p projection directions are the p eigenvectors of the covariance matrix of the second cluster.

Each of these $2 + 2 * p$ projection directions are in turn used as the initial projection direction for the 'fixedpoint' algorithm to obtain the optimal projection direction and the corresponding optimal separation index. We also obtain $2 + 2 * p$ separation indices by projecting two clusters along each of these $2 + 2 * p$ projection directions.

Finally, the projection direction with the largest separation index among the $2 * (2 + 2 * p)$ optimal separation indices is chosen as the optimal projection direction. The corresponding separation index is chosen as the optimal separation index.

Value

sepValMat	Separation index matrix
projDirArray	Array of projection directions for each pair of clusters

Author(s)

Weiliang Qiu <weiliang.qiu@gmail.com>
Harry Joe <harry@stat.ubc.ca>

References

Qiu, W.-L. and Joe, H. (2006a) Generation of Random Clusters with Specified Degree of Separaion. *Journal of Classification*, **23**(2), 315-334.

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

Su, J. Q. and Liu, J. S. (1993) Linear Combinations of Multiple Diagnostic Markers. *Journal of the American Statistical Association*, **88**, 1350–1355.

Examples

```
n1 <- 50
mu1 <- c(0, 0)
Sigma1 <- matrix(c(2, 1, 1, 5), 2, 2)
n2 <- 100
mu2 <- c(10, 0)
Sigma2 <- matrix(c(5, -1, -1, 2), 2, 2)
projDir <- c(1,  0)
muMat <- rbind(mu1,  mu2)
SigmaArray <- array(0,  c(2, 2, 2))
SigmaArray[, , 1] <- Sigma1
SigmaArray[, , 2] <- Sigma2

a <- getSepProjTheory(
    muMat = muMat,
    SigmaArray = SigmaArray,
    iniProjDirMethod = "SL")
# separation index for cluster distributions 1 and 2
a$sepValMat[1, 2]
# projection direction for cluster distributions 1 and 2
a$projDirArray[1, 2, ]

library(MASS)
y1 <- mvrnorm(n1, mu1, Sigma1)
y2 <- mvrnorm(n2, mu2, Sigma2)
y <- rbind(y1, y2)
cl <- rep(1:2, c(n1, n2))

b <- getSepProjData(
  y = y,
  cl = cl,
  iniProjDirMethod = "SL",
  projDirMethod = "newton")
# separation index for clusters 1 and 2
b$sepValMat[1, 2]
# projection direction for clusters 1 and 2
b$projDirArray[1, 2, ]
```

`nearestNeighborSepVal` *SEPARATON INFORMATION MATRIX*

Description

Separation information matrix containing the nearest neighbor and farthest neighbor of each cluster.

Usage

```
nearestNeighborSepVal(sepValMat)
```

Arguments

`sepValMat`	a `K` by `K` matrix, where `K` is the number of clusters. `sepValMat[i,j]` is the separation index between cluster `i` and `j`.

Value

This function returns a separation information matrix containing `K` rows and the following six columns, where `K` is the number of clusters.

`Column 1:`	Labels of clusters $(1, 2, \ldots, numClust)$, where $numClust$ is the number of clusters for the data set.
`Column 2:`	Labels of the corresponding nearest neighbors.
`Column 3:`	Separation indices of the clusters to their nearest neighboring clusters.
`Column 4:`	Labels of the corresponding farthest neighboring clusters.
`Column 5:`	Separation indices of the clusters to their farthest neighbors.
`Column 6:`	Median separation indices of the clusters to their neighbors.

Author(s)

Weiliang Qiu <weiliang.qiu@gmail.com>
Harry Joe <harry@stat.ubc.ca>

References

Qiu, W.-L. and Joe, H. (2006a) Generation of Random Clusters with Specified Degree of Separaion. *Journal of Classification*, **23**(2), 315-334.

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

Examples

```
n1 <- 50
mu1 <- c(0, 0)
Sigma1 <- matrix(c(2, 1, 1, 5), 2, 2)
n2 <- 100
mu2 <- c(10, 0)
Sigma2 <- matrix(c(5, -1, -1, 2), 2, 2)
n3 <- 30
mu3 <- c(10, 10)
Sigma3 <- matrix(c(3, 1.5, 1.5, 1), 2, 2)

projDir <- c(1, 0)
muMat <- rbind(mu1, mu2, mu3)
SigmaArray <- array(0, c(2, 2, 3))
SigmaArray[, , 1] <- Sigma1
SigmaArray[, , 2] <- Sigma2
SigmaArray[, , 3] <- Sigma3

tmp <- getSepProjTheory(
muMat = muMat,
SigmaArray = SigmaArray,
iniProjDirMethod="SL")
sepValMat <- tmp$sepValMat
nearestNeighborSepVal(sepValMat = sepValMat)
```

`plot1DProjection` *PLOT A PAIR OF CLUSTERS AND THEIR DENSITY ESTIMATES, WHICH ARE PROJECTED ALONG A SPECIFIED 1-D PROJECTION DIRECTION*

Description

Plot a pair of clusters and their density estimates, which are projected along a specified 1-D projection direction.

Usage

```
plot1DProjection(
 y1,
 y2,
 projDir,
                  sepValMethod = c("normal", "quantile"),
 bw = "nrd0",
                  xlim = NULL,
 ylim = NULL,
                  xlab = "1-D projected clusters",
 ylab = "density estimates",
                  title = "1-D Projected Clusters and their density estimates",
```

```
                font = 2,
font.lab = 2,
cex = 1.2,
cex.lab = 1.2,
cex.main = 1.5,
                lwd = 4,
lty1 = 1,
lty2 = 2,
pch1 = 18,
pch2 = 19,
col1 = 2,
col2 = 4,
                type = "l",
alpha = 0.05,
eps = 1.0e-10,
quiet = TRUE)
```

Arguments

y1	Data matrix of cluster 1. Rows correspond to observations. Columns correspond to variables.
y2	Data matrix of cluster 2. Rows correspond to observations. Columns correspond to variables.
projDir	1-D projection direction along which two clusters will be projected.
sepValMethod	Method to calculate separation index for a pair of clusters projected onto a 1-D space. sepValMethod="quantile" indicates the quantile version of separation index will be used: $sepVal = (L_2 - U_1)/(U_2 - L_1)$ where L_i and U_i, $i = 1, 2$, are the lower and upper alpha/2 sample percentiles of projected cluster i. sepValMethod="normal" indicates the normal version of separation index will be used: $sepVal = [(xbar_2 - xbar_1) - z_{\alpha/2}(s_1 + s_2)]/[(xbar_2 - xbar_1) + z_{\alpha/2}(s_1 + s_2)]$, where $xbar_i$ and s_i are the sample mean and standard deviation of projected cluster i.
bw	The smoothing bandwidth to be used by the function density.
xlim	Range of X axis.
ylim	Range of Y axis.
xlab	X axis label.
ylab	Y axis label.
title	Title of the plot.
font	An integer which specifies which font to use for text (see par).
font.lab	The font to be used for x and y labels (see par).
cex	A numerical value giving the amount by which plotting text and symbols should be scaled relative to the default (see par).
cex.lab	The magnification to be used for x and y labels relative to the current setting of 'cex' (see par).

cex.main	The magnification to be used for main titles relative to the current setting of 'cex' (see par).
lwd	The line width, a positive number, defaulting to '1' (see par).
lty1	Line type for cluster 1 (see par).
lty2	Line type for cluster 2 (see par).
pch1	Either an integer specifying a symbol or a single character to be used as the default in plotting points for cluster 1 (see points).
pch2	Either an integer specifying a symbol or a single character to be used as the default in plotting points for cluster 2 (see points).
col1	Color to indicates cluster 1.
col2	Color to indicates cluster 2.
type	What type of plot should be drawn (see plot).
alpha	Tuning parameter reflecting the percentage in the two tails of a projected cluster that might be outlying.
eps	A small positive number to check if a quantitiy q is equal to zero. If $\|q\| <$eps, then we regard q as equal to zero. eps is used to check the denominator in the formula of the separation index is equal to zero. Zero-value denominator indicates two clusters are totally overlapped. Hence the separation index is set to be -1. The default value of eps is $1.0e-10$.
quiet	A flag to switch on/off the outputs of intermediate results and/or possible warning messages. The default value is TRUE.

Details

The ticks along X axis indicates the positions of points of the projected two clusters. The positions of L_i and U_i, $i = 1, 2$, are also indicated on X axis, where L_i and U_i are the lower and upper $\alpha/2$ sample percentiles of cluster i if sepValMethod="quantile". If sepValMethod="normal", $L_i = xbar_i - z_{\alpha/2} s_i$, where $xbar_i$ and s_i are the sample mean and standard deviation of cluster i, and $z_{\alpha/2}$ is the upper $\alpha/2$ percentile of standard normal distribution.

Value

sepVal	value of the separation index for the projected two clusters along the projection direction projDir.
projDir	projection direction. To make sure the projected cluster 1 is on the left-hand side of the projected cluster 2, the input projDir might be changed to -projDir.

Author(s)

Weiliang Qiu <weiliang.qiu@gmail.com>
Harry Joe <harry@stat.ubc.ca>

References

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

See Also

plot2DProjection viewClusters

Examples

```
n1 <- 50
mu1 <- c(0,0)
Sigma1 <- matrix(c(2, 1, 1, 5), 2, 2)
n2 <- 100
mu2 <- c(10, 0)
Sigma2 <- matrix(c(5, -1, -1, 2), 2, 2)
projDir <- c(1,  0)

library(MASS)
set.seed(1234)
y1 <- mvrnorm(n1, mu1, Sigma1)
y2 <- mvrnorm(n2, mu2, Sigma2)
y <- rbind(y1, y2)
cl <- rep(1:2, c(n1, n2))

b <- getSepProjData(
    y = y,
    cl = cl,
    iniProjDirMethod = "SL",
    projDirMethod = "newton")
# projection direction for clusters 1 and 2
projDir <- b$projDirArray[1, 2, ]

plot1DProjection(
 y1 = y1,
 y2 = y2,
 projDir = projDir)
```

---

plot2DProjection *PLOT A PAIR OF CLUSTERS ALONG A 2-D PROJECTION SPACE*

Description

Plot a pair of clusters along a 2-D projection space.

Usage

```
plot2DProjection(
 y1,
 y2,
 projDir,
                  sepValMethod = c("normal", "quantile"),
                  iniProjDirMethod = c("SL", "naive"),
```

```
                projDirMethod = c("newton", "fixedpoint"),
                xlim = NULL,
 ylim = NULL,
                xlab = "1st projection direction",
                ylab = "2nd projection direction",
                title = "Scatter plot of 2-D Projected Clusters",
                font = 2,
 font.lab = 2,
 cex = 1.2,
 cex.lab = 1,
 cex.main = 1.5,
                lwd = 4,
 lty1 = 1,
 lty2 = 2,
 pch1 = 18,
 pch2 = 19,
 col1 = 2,
 col2 = 4,
                alpha = 0.05,
 ITMAX = 20,
 eps = 1.0e-10,
 quiet = TRUE)
```

Arguments

y1	Data matrix of cluster 1. Rows correspond to observations. Columns correspond to variables.
y2	Data matrix of cluster 2. Rows correspond to observations. Columns correspond to variables.
projDir	1-D projection direction along which two clusters will be projected.
sepValMethod	Method to calculate separation index for a pair of clusters projected onto a 1-D space. sepValMethod="quantile" indicates the quantile version of separation index will be used: $sepVal = (L_2 - U_1)/(U_2 - L_1)$ where L_i and U_i, $i = 1, 2$, are the lower and upper alpha/2 sample percentiles of projected cluster i. sepValMethod="normal" indicates the normal version of separation index will be used: $sepVal = [(xbar_2 - xbar_1) - z_{\alpha/2}(s_1 + s_2)]/[(xbar_2 - xbar_1) + z_{\alpha/2}(s_1 + s_2)]$, where $xbar_i$ and s_i are the sample mean and standard deviation of projected cluster i.
iniProjDirMethod	Indicating the method to get initial projection direction when calculating the separation index between a pair of clusters (c.f. Qiu and Joe, 2006a, 2006b). iniProjDirMethod="SL" indicates the initial projection direction is the sample version of the SL's projection direction (Su and Liu, 1993) $(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)^{-1}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$ iniProjDirMethod="naive" indicates the initial projection direction is $\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1$
projDirMethod	Indicating the method to get the optimal projection direction when calculating the separation index between a pair of clusters (c.f. Qiu and Joe, 2006a, 2006b). projDirMethod="newton" indicates we use the Newton-Raphson method to

search the optimal projection direction (c.f. Qiu and Joe, 2006a). This requires the assumptions that both covariance matrices of the pair of clusters are positive-definite. If this assumption is violated, the "fixedpoint" method could be used. The "fixedpoint" method iteratively searches the optimal projection direction based on the first derivative of the separation index to the project direction (c.f. Qiu and Joe, 2006b).

xlim	Range of X axis.
ylim	Range of Y axis.
xlab	X axis label.
ylab	Y axis label.
title	Title of the plot.
font	An integer which specifies which font to use for text (see par).
font.lab	The font to be used for x and y labels (see par).
cex	A numerical value giving the amount by which plotting text and symbols should be scaled relative to the default (see par).
cex.lab	The magnification to be used for x and y labels relative to the current setting of 'cex' (see par).
cex.main	The magnification to be used for main titles relative to the current setting of 'cex' (see par).
lwd	The line width, a positive number, defaulting to '1' (see par).
lty1	Line type for cluster 1 (see par).
lty2	Line type for cluster 2 (see par).
pch1	Either an integer specifying a symbol or a single character to be used as the default in plotting points for cluster 1 (see points).
pch2	Either an integer specifying a symbol or a single character to be used as the default in plotting points for cluster 2 (see points).
col1	Color to indicates cluster 1.
col2	Color to indicates cluster 2.
alpha	Tuning parameter reflecting the percentage in the two tails of a projected cluster that might be outlying.
ITMAX	Maximum iteration allowed when iteratively calculating the optimal projection direction. The actual number of iterations is usually much less than the default value 20.
eps	A small positive number to check if a quantitiy q is equal to zero. If $\|q\| <$eps, then we regard q as equal to zero. eps is used to check the denominator in the formula of the separation index is equal to zero. Zero-value denominator indicates two clusters are totally overlapped. Hence the separation index is set to be -1. The default value of eps is $1.0e-10$.
quiet	A flag to switch on/off the outputs of intermediate results and/or possible warning messages. The default value is TRUE.

Details

To get the second projection direction, we first construct an orthogonal matrix with first column `projDir`. Then we rotate the data points according to this orthogonal matrix. Next, we remove the first dimension of the rotated data points, and obtain the optimal projection direction `projDir2` for the rotated data points in the remaining dimensions. Finally, we rotate the vector `projDir3=(0, projDir2)` back to the original space. The vector `projDir3` is the second projection direction.

The ticks along X axis indicates the positions of points of the projected two clusters. The positions of L_i and U_i, $i = 1, 2$, are also indicated on X axis, where L_i and U_i are the lower and upper $\alpha/2$ sample percentiles of cluster i if `sepValMethod="quantile"`. If `sepValMethod="normal"`, $L_i = xbar_i - z_{\alpha/2} s_i$, where $xbar_i$ and s_i are the sample mean and standard deviation of cluster i, and $z_{\alpha/2}$ is the upper $\alpha/2$ percentile of standard normal distribution.

Value

`sepValx`	value of the separation index for the projected two clusters along the 1st projection direction.
`sepValy`	value of the separation index for the projected two clusters along the 2nd projection direction.
`Q2`	1st column is the 1st projection direction. 2nd column is the 2nd projection direction.

Author(s)

Weiliang Qiu <weiliang.qiu@gmail.com>
Harry Joe <harry@stat.ubc.ca>

References

Qiu, W.-L. and Joe, H. (2006a) Generation of Random Clusters with Specified Degree of Separaion. *Journal of Classification*, **23**(2), 315-334.

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

See Also

`plot1DProjection` `viewClusters`

Examples

```
n1 <- 50
mu1 <- c(0,0)
Sigma1 <- matrix(c(2, 1, 1, 5), 2, 2)
n2 <- 100
mu2 <- c(10, 0)
Sigma2 <- matrix(c(5, -1, -1, 2), 2, 2)
projDir <- c(1,  0)

library(MASS)
set.seed(1234)
```

```
y1 <- mvrnorm(n1, mu1, Sigma1)
y2 <- mvrnorm(n2, mu2, Sigma2)
y <- rbind(y1, y2)
cl <- rep(1:2, c(n1, n2))

b <- getSepProjData(
    y = y,
    cl = cl,
    iniProjDirMethod = "SL",
    projDirMethod = "newton")
# projection direction for clusters 1 and 2
projDir <- b$projDirArray[1,2,]

par(mfrow = c(2,1))
plot1DProjection(
 y1 = y1,
 y2 = y2,
 projDir = projDir)
plot2DProjection(
 y1 = y1,
 y2 = y2,
 projDir = projDir)
```

`rcorrmatrix` *GENERATE A RANDOM CORRELATION MATRIX BASED ON RANDOM PARTIAL CORRELATIONS*

Description

Generate a random correlation matrix based on random partial correlations.

Usage

```
rcorrmatrix(d, alphad = 1)
```

Arguments

`d`	Dimension of the matrix. `d` should be a non-negative integer.
`alphad`	α parameter for partial of $1, d$ given $2, \ldots, d-1$, for generating random correlation matrix based on the method proposed by Joe (2006), where d is the dimension of the correlation matrix. The default value `alphad`$= 1$ leads to a random matrix which is uniform over space of positive definite correlation matrices. Each correlation has a $Beta(a, a)$ distribution on $(-1, 1)$ where $a = alphad + (d-2)/2$. `alphad` should be a positive number.

Value

A correlation matrix.

Author(s)

Weiliang Qiu <weiliang.qiu@gmail.com>
Harry Joe <harry@stat.ubc.ca>

References

Joe, H. (2006) Generating Random Correlation Matrices Based on Partial Correlations. *Journal of Multivariate Analysis*, **97**, 2177–2189.

Examples

```
rcorrmatrix(3)
rcorrmatrix(5)
rcorrmatrix(5, alphad = 2.5)
```

`sepIndex` *MEASURE THE MAGNITUDE OF THE GAP OR SPARSE AREA BETWEEN A PAIR OF CLUSTERS ALONG THE SPECIFIED PROJECTION DIRECTION*

Description

Measure the magnitude of the gap or sparse area between a pair of clusters (or cluster distributions) along the specified projection direction.

Usage

```
sepIndexTheory(
       projDir,
       mu1,
       Sigma1,
       mu2,
       Sigma2,
               alpha = 0.05,
       eps = 1.0e-10,
       quiet = TRUE)

sepIndexData(
     projDir,
     y1,
     y2,
     alpha = 0.05,
     eps = 1.0e-10,
     quiet = TRUE)
```

Arguments

projDir	Projection direction.
mu1	Mean vector of cluster 1.
Sigma1	Covariance matrix of cluster 1.
mu2	Mean vector of cluster 2.
Sigma2	Covariance matrix of cluster 2.
y1	Data matrix of cluster 1. Rows correspond to observations. Columns correspond to variables.
y2	Data matrix of cluster 2. Rows correspond to observations. Columns correspond to variables.
alpha	Tuning parameter reflecting the percentage in the two tails of a projected cluster that might be outlying. We set alpha$= 0.05$ like we set the significance level in hypothesis testing as 0.05.
eps	Convergence threshold. A small positive number to check if a quantitiy q is equal to zero. If $\|q\| <$eps, then we regard q is equal to zero. eps is used to check if an algorithm converges. The default value is $1.0e-10$.
quiet	A flag to switch on/off the outputs of intermediate results and/or possible warning messages. The default value is TRUE.

Value

The value of the separation index defined in Qiu and Joe (2006).

Author(s)

Weiliang Qiu <weiliang.qiu@gmail.com>
Harry Joe <harry@stat.ubc.ca>

References

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

Examples

```
n1<-50
mu1<-c(0,0)
Sigma1<-matrix(c(2,1,1,5),2,2)
n2<-100
mu2<-c(10,0)
Sigma2<-matrix(c(5,-1,-1,2),2,2)
projDir<-c(1, 0)
sepIndexTheory(projDir, mu1, Sigma1, mu2, Sigma2)

library(MASS)
y1 <- mvrnorm(n1, mu1, Sigma1)
y2 <- mvrnorm(n2, mu2, Sigma2)
```

```
sepIndexData(
     projDir = projDir,
     y1 = y1,
     y2 = y2)
```

---

simClustDesign *DESIGN FOR RANDOM CLUSTER GENERATION WITH SPECIFIED DEGREE OF SEPARATION*

Description

Generating data sets via a factorial design, which has factors: degree of separation, number of clusters, number of non-noisy variables, number of noisy variables. The separation between any cluster and its nearest neighboring clusters can be set to a specified value. The covariance matrices of clusters can have arbitrary diameters, shapes and orientations.

Usage

```
simClustDesign(numClust = c(3,6,9),
               sepVal = c(0.01, 0.21, 0.342),
               sepLabels = c("L", "M", "H"),
               numNonNoisy = c(4,8,20),
               numNoisy = NULL,
               numOutlier = 0,
               numReplicate = 3,
               fileName = "test",
               clustszind = 2,
               clustSizeEq = 50,
               rangeN = c(50,200),
               clustSizes = NULL,
               covMethod = c("eigen", "onion", "c-vine", "unifcorrmat"),
       eigenvalue = NULL,
               rangeVar = c(1, 10),
               lambdaLow = 1,
               ratioLambda = 10,
               alphad = 1,
               eta = 1,
               rotateind = TRUE,
               iniProjDirMethod = c("SL", "naive"),
               projDirMethod = c("newton", "fixedpoint"),
               alpha = 0.05,
               ITMAX = 20,
               eps = 1.0e-10,
               quiet = TRUE,
               outputDatFlag = TRUE,
               outputLogFlag = TRUE,
               outputEmpirical = TRUE,
               outputInfo = TRUE)
```

Arguments

numClust	Vector of the number of clusters for data sets in the design.
sepVal	Vector of desired values of the separation index between clusters and their nearest neighboring clusters. Each element of sepVal can take values within the interval [-1, 1). The closer to 1 an element of sepVal is, the more separated the pair of clusters are. The values $0.01, 0.21, 0.34$ are the values of the separation index for two univariate clusters generated from $N(0,1)$ and $N(0,A)$, where $A = 4, 6, 8$, respectively. sepVal$= 0.01(A = 4)$ indicates a close cluster structure. sepVal$= 0.21(A = 6)$ indicates a separated cluster structure. sepVal$= 0.34(A = 8)$ indicates a well-separated cluster.
sepLabels	Labels for "close", "separated", and "well-separated" cluster structures. By default, "L" (low) means "close", "M" (medium) means "separated", "H" (high) means "well-separated".
numNonNoisy	Vector of the number of non-noisy variables.
numNoisy	Vectors of the number of noisy variables. The default value of numNoisy is NULL so that the program can automatically assign the value of numNoisy as a vector with elements $1, round(p1/2), p1$.
numOutlier	The number or ratio of outliers. If numOutlier is a positive integer, then numOutlier means the number of outliers. If numOutlier is a real number between $(0, 1)$, then numOutlier means the ratio of outliers, i.e. the number of outliers is equal to round(numOutlier$*n_1$), where n_1 is the total number of non-outliers. If numOutlier is a real number greater than 1, then numOutlier is rounded to an integer.
numReplicate	Number of data sets to be generated for the same cluster structure specified by the other arguments of the function genRandomClust. The default value 3 follows the design in Milligan (1985).
fileName	The first part of the names of data files that record the generated data sets and associated information, such as cluster membership of data points, labels of noisy variables, separation index matrix, projection directions, etc. (see details). The default value of fileName is 'test'.
clustszind	Cluster size indicator. clustszind$= 1$ indicates that all cluster have equal size. The size is specified by the argument clustSizeEq. clustszind$= 2$ indicates that the cluster sizes are randomly generated from the range specified by the argument rangeN. clustszind$= 3$ indicates that the cluster sizes are specified via the vector clustSizes. The default value is 2 so that the generated clusters are more realistic.
clustSizeEq	Cluster size. If the argument clustszind$= 1$, then all clusters will have the equal number clustSizeEq of data points. The value of clustSizeEq should be large enough to get non-singular cluster covariance matrices. We recommend the clustSizeEq is at least $10 * p$, where p is the total number of variables (including both non-noisy and noisy variables). The default value 100 is a reasonable cluster size.
rangeN	The range of cluster sizes. If clustszind$= 2$, then cluster sizes will be randomly generated from the range specified by rangeN. The lower bound of the number of clusters should be large enough to get non-singular cluster covariance

matrices. We recommend the minimum cluster size is at least $10 * p$, where p is the total number of variables (including both non-noisy and noisy variables). The default range is $[50, 200]$ which can produce reasonable variability of cluster sizes.

`clustSizes` The sizes of clusters. If `clustszind`$= 3$, then cluster sizes will be specified by the vector `clustSizes`. We recommend the minimum cluster size is at least $10 * p$, where p is the total number of variables (including both non-noisy and noisy variables). The user needs to specify the value of `clustSizes`. Therefore, we set the default value of `clustSizes` as `NULL`.

`covMethod` Method to generate covariance matrices for clusters (see details). The default method is 'eigen' so that the user can directly specify the range of the *diameters* of clusters.

`eigenvalue` numeric. user-specified eigenvalues when `covMethod = "eigen"`. If `eigenvalue = NULL` and `covMethod = "eigen"`, then eigenvalues will be automatically generated.

`rangeVar` Range for variances of a covariance matrix (see details). The default range is $[1, 10]$ which can generate reasonable variability of variances.

`lambdaLow` Lower bound of the eigenvalues of cluster covariance matrices. If the argument `covMethod="eigen"`, we need to generate eigenvalues for cluster covariance matrices. The eigenvalues are randomly generated from the interval [`lambdaLow`, `lambdaLow`$*$`ratioLambda`]. In our experience, `lambdaLow`$= 1$ and `ratioLambda`$= 10$ can give reasonable variability of the diameters of clusters. `lambdaLow` should be positive.

`ratioLambda` The ratio of the upper bound of the eigenvalues to the lower bound of the eigenvalues of cluster covariance matrices. If the argument `covMethod="eigen"`, we need to generate eigenvalues for cluster covariance matrices. The eigenvalues are randomly generated from the interval [`lambdaLow`, `lambdaLow`$*$`ratioLambda`]. In our experience, `lambdaLow`$= 1$ and `ratioLambda`$= 10$ can give reasonable variability of the diameters of clusters. `ratioLambda` should be larger than 1.

`alphad` parameter for unifcorrmat method to generate random correlation matrix `alphad=1` for uniform. `alphad` should be positive.

`eta` parameter for "c-vine" and "onion" methods to generate random correlation matrix `eta=1` for uniform. `eta` should be positive.

`rotateind` Rotation indicator. `rotateind=TRUE` indicates randomly rotating data in non-noisy dimensions so that we may not detect the full cluster structure from pairwise scatter plots of the variables.

`iniProjDirMethod`
Indicating the method to get initial projection direction when calculating the separation index between a pair of clusters (c.f. Qiu and Joe, 2006a, 2006b).
`iniProjDirMethod="SL"`, the default, indicates the initial projection direction is the sample version of the SL's projection direction (Su and Liu, 1993, JASA) $(\boldsymbol{\Sigma}_1 + \boldsymbol{\Sigma}_2)^{-1} (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)$
`iniProjDirMethod="naive"` indicates the initial projection direction is $\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1$

`projDirMethod` Indicating the method to get the optimal projection direction when calculating the separation index between a pair of clusters (c.f. Qiu and Joe, 2006a, 2006b).

projDirMethod="newton" indicates we use the modified Newton-Raphson method to search the optimal projection direction (c.f. Qiu and Joe, 2006a). This requires the assumptions that both covariance matrices of the pair of clusters are positive-definite. If this assumption is violated, the "fixedpoint" method could be used. The "fixedpoint" method iteratively searches the optimal projection direction based on the first derivative of the separation index to the projection direction (c.f. Qiu and Joe, 2006b).

`alpha` Tuning parameter reflecting the percentage in the two tails of a projected cluster that might be outlying. We set `alpha`$= 0.05$ like we set the significance level in hypothesis testing as 0.05.

`ITMAX` Maximum iteration allowed when to iteratively calculating the optimal projection direction. The actual number of iterations is usually much less than the default value 20.

`eps` Convergence threshold. A small positive number to check if a quantitiy q is equal to zero. If $|q| <$`eps`, then we regard q as equal to zero. `eps` is used to check if an algorithm converges. The default value is $1.0e-10$.

`quiet` A flag to switch on/off the outputs of intermediate results and/or possible warning messages. The default value is `TRUE`.

`outputDatFlag` Indicates if data set should be output to file.

`outputLogFlag` Indicates if log info should be output to file.

`outputEmpirical`
Indicates if empirical separation indices and projection directions should be calculated. This option is useful when generating clusters with sizes which are not large enough so that the sample covariance matrices may be singular. Hence, by default, `outputEmpirical=TRUE`.

`outputInfo` Indicates if theoretical and empirical separation information data frames should be output to a file with format `[fileName]_info.log`.

Details

The function `simClustDesign` is an implementation of the design for generating random clusters proposed in Qiu and Joe (2006a). In the design, the degree of separation between any cluster and its nearest neighboring cluster 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 separation between a pair of clusters is measured by the separation index proposed in Qiu and Joe (2006b).

The current version of the function `simClustDesign` implements two methods to generate covariance matrices for clusters. The first method, denoted by `eigen`, first randomly generates eigenvalues $(\lambda_1, \ldots > \lambda_p)$ for the covariance matrix ($\boldsymbol{\Sigma}$), then uses columns of a randomly generated orthogonal matrix ($\boldsymbol{Q} = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_p)$) as eigenvectors. The covariance matrix $\boldsymbol{\Sigma}$ is then contructed as $\boldsymbol{Q} * diag(\lambda_1, \ldots, \lambda_p) * \boldsymbol{Q}^T$. The second method, denoted as `unifcorrmat`, first generates a random correlation matrix ($\boldsymbol{R}$) via the method proposed in Joe (2006), then randomly generates variances $(\sigma_1^2, \ldots, \sigma_p^2)$ from an interval specified by the argument `rangeVar`. The covariance matrix $\boldsymbol{\Sigma}$ is then constructed as $diag(\sigma_1, \ldots, \sigma_p) * \boldsymbol{R} * diag(\sigma_1, \ldots, \sigma_p)$.

For each data set generated, the function `simClustDesign` outputs four files: data file, log file, membership file, and noisy set file. All four files have the same format:

[fileName]J[j]G[g]v[p1]nv[p2]out[numOutlier]_[numReplicate].[extension]
where 'extension' can be 'dat', 'log', 'mem', or 'noisy'. 'J' indicates separation index, with 'j' indicating the level of the factor 'separation index'; 'G' indicates number of clusters, with 'g' indicating the level of the factor 'number of clusters'; 'v' indicates the number of non-noisy variables, with 'p1' indicating the level of the factor 'number of non-noisy variables'; 'nv' indicates the number of noisy variables, with 'p2' indicating the level of the factor 'number of noisy variables'; 'out' indicates number of outliers, with 'numOutlier' indicating the value of the argument numOutlier of the function simClustDesign; 'numReplicate' indicates the value of the argument numReplicate of the function simClustDesign.

The data file with file extension 'dat' contains $n + 1$ rows and p columns, where n is the number of data points and p is the number of variables. The first row is the variable names. The log file with file extension 'log' contains information such as cluster sizes, mean vectors, covariance matrices, projection directions, separation index matrices, etc. The membership file with file extension 'mem' contains n rows and one column of cluster memberships for data points. The noisy set file with file extension 'noisy' contains a row of labels of noisy variables.

When generating clusters, population covariance matrices are all positive-definite. However sample covariance matrices might be semi-positive-definite due to small cluster sizes. In this case, the function genRandomClust will automatically use the "fixedpoint" method to search the optimal projection direction.

Value

The function outputs four data files for each data set (see details).

This function also returns separation information data frames infoFrameTheory and infoFrameData based on population and empirical mean vectors and covariance matrices of clusters for all the data sets generated. Both infoFrameTheory and infoFrameData contain the following seven columns:

Column 1:	Labels of clusters $(1, 2, \ldots, numClust)$, where $numClust$ is the number of clusters for the data set.
Column 2:	Labels of the corresponding nearest neighbors.
Column 3:	Separation indices of the clusters to their nearest neighboring clusters.
Column 4:	Labels of the corresponding farthest neighboring clusters.
Column 5:	Separation indices of the clusters to their farthest neighbors.
Column 6:	Median separation indices of the clusters to their neighbors.
Column 7:	Data file names with format [fileName]J[j]G[g]v[p1]nv[p2]out[numOutlier]_[numReplicate] (see details).

The function also returns three lists: datList, memList, and noisyList.

datList:	a list of lists of data matrices for generated data sets.
memList:	a list of lists of cluster memberships for data points for generated data sets.
noisyList:	a list of lists of sets of noisy variables for generated data sets.

Note

The speed of this function might be slow.

Author(s)

Weiliang Qiu <weiliang.qiu@gmail.com>
Harry Joe <harry@stat.ubc.ca>

References

Joe, H. (2006) Generating Random Correlation Matrices Based on Partial Correlations. *Journal of Multivariate Analysis*, **97**, 2177–2189.

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

Qiu, W.-L. and Joe, H. (2006a) Generation of Random Clusters with Specified Degree of Separaion. *Journal of Classification*, **23**(2), 315-334.

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

Su, J. Q. and Liu, J. S. (1993) Linear Combinations of Multiple Diagnostic Markers. *Journal of the American Statistical Association*, **88**, 1350–1355

Examples

```
## Not run:
tmp <- simClustDesign(
      numClust = 3,
                      sepVal = c(0.01, 0.21),
                      sepLabels = c("L", "M"),
                      numNonNoisy = 4,
                      numOutlier = 0,
                      numReplicate = 2,
                      clustszind = 2)
## End(Not run)
```

---

viewClusters *PLOT ALL CLUSTERS IN A 2-D PROJECTION SPACE*

Description

Plot all clusters in a 2-D projection space.

Usage

```
viewClusters(
     y,
     cl,
     outlierLabel = 0,
             projMethod = "Eigen",
     xlim = NULL,
     ylim = NULL,
```

```
            xlab = "1st projection direction",
            ylab = "2nd projection direction",
            title = "Scatter plot of 2-D Projected Clusters",
            font = 2,
      font.lab = 2,
      cex = 1.2,
      cex.lab = 1.2)
```

Arguments

y	Data matrix. Rows correspond to observations. Columns correspond to variables.
cl	Cluster membership vector.
outlierLabel	Label for outliers. Outliers are not involved in calculating the projection directions. Outliers will be represented by red triangles in the plot. By default, outlierLabel=0.
projMethod	Method to construct 2-D projection directions. projMethod="Eigen" indicates that we project data to the 2-dimensional space spanned by the first two eigenvectors of the between cluster distance matrix $B = \frac{2}{k_0}\sum_{i=1}^{k_0}\Sigma_i + \frac{2}{k_0(k_0-1)}\sum_{i<j}(\theta_i - \theta_j)(\theta_i-\theta_j)^T$. projMethod="DMS" indicates that we project data to the 2-dimensional space spanned by the first two eigenvectors of the between cluster distance matrix $B = \sum_{i=2}^{k_0}\sum_{j=1}^{i-1} n_i n_j(\theta_i-\theta_j)(\theta_i-\theta_j)^T$. "DMS" method is proposed by Dhillon et al. (2002).
xlim	Range of X axis.
ylim	Range of Y axis.
xlab	X axis label.
ylab	Y axis label.
title	Title of the plot.
font	An integer which specifies which font to use for text (see par).
font.lab	The font to be used for x and y labels (see par).
cex	A numerical value giving the amount by which plotting text and symbols should be scaled relative to the default (see par).
cex.lab	The magnification to be used for x and y labels relative to the current setting of 'cex' (see par).

Value

B	Between cluster distance matrix measuring the between cluster variation.
Q	Columns of Q are eigenvectors of the matrix B.
proj	Projected clusters in the 2-D space spanned by the first 2 columns of the matrix Q.

Author(s)

Weiliang Qiu <weiliang.qiu@gmail.com>
Harry Joe <harry@stat.ubc.ca>

References

Dhillon I. S., Modha, D. S. and Spangler, W. S. (2002) Class visualization of high-dimensional data with applications. *computational Statistics and Data Analysis*, **41**, 59–90.

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

See Also

plot1DProjection plot2DProjection

Examples

```
n1 <- 50
mu1 <- c(0, 0)
Sigma1 <- matrix(c(2, 1, 1, 5), 2, 2)
n2 <- 100
mu2 <- c(10, 0)
Sigma2 <- matrix(c(5, -1, -1, 2), 2, 2)
n3 <- 30
mu3 <- c(10, 10)
Sigma3 <- matrix(c(3, 1.5, 1.5, 1), 2, 2)
n4 <- 10
mu4 <- c(0, 0)
Sigma4 <- 50*diag(2)

library(MASS)
set.seed(1234)
y1 <- mvrnorm(n1, mu1, Sigma1)
y2 <- mvrnorm(n2, mu2, Sigma2)
y3 <- mvrnorm(n3, mu3, Sigma3)
y4 <- mvrnorm(n4, mu4, Sigma4)
y <- rbind(y1, y2, y3, y4)
cl <- rep(c(1:3, 0), c(n1, n2, n3, n4))

par(mfrow=c(2,1))
viewClusters(y = y, cl = cl)
viewClusters(y = y, cl = cl, projMethod = "DMS")
```

Index