Package: MGMM (via r-universe)

October 25, 2026

Title Missingness-Aware Gaussian Mixture Models

Date 2026-02-26

Version 1.0.1.3

Author Zachary McCaw [aut, cre] (ORCID:
<https://orcid.org/0000-0002-2006-9828>)

Maintainer Zachary McCaw <zmccaw@alumni.harvard.edu>

Description Parameter estimation and classification for Gaussian Mixture Models (GMMs) in the presence of missing data. This package complements existing implementations by allowing for both missing elements in the input vectors and full (as opposed to strictly diagonal) covariance matrices. Estimation is performed using an expectation conditional maximization algorithm that accounts for missingness of both the cluster assignments and the vector components. The output includes the marginal cluster membership probabilities; the mean and covariance of each cluster; the posterior probabilities of cluster membership; and a completed version of the input data, with missing values imputed to their posterior expectations. For additional details, please see McCaw ZR, Julienne H, Aschard H. ``Fitting Gaussian mixture models on incomplete data." <doi:10.1186/s12859-022-04740-9>.

Depends R (>= 3.5.0)

License GPL-3

Encoding UTF-8

LinkingTo Rcpp, RcppArmadillo

Imports cluster, glue, methods, mvnfast, plyr, Rcpp (>= 1.0.3), stats

Suggests testthat (>= 3.0.0), knitr, rmarkdown, withr

VignetteBuilder knitr

Config/build/clean-inst-doc false

RoxygenNote 7.3.3

Config/testthat/edition 3

NeedsCompilation yes

Repository https://cran.r-universe.dev

Date/Publication 2026-02-26 17:20:08 UTC

RemoteUrl https://github.com/cran/MGMM

RemoteRef HEAD

RemoteSha 7773f075e683791f767c0e94d15a6f66350cd875

Contents

CalHar . . . 2
ChooseK . . . 3
ClustQual . . . 4
CombineMIs . . . 5
DavBou . . . 6
FitGMM . . . 7
FitMix . . . 9
FitMVN . . . 10
GenImputation . . . 11
logLik.mix . . . 11
logLik.mvn . . . 12
mean.mix . . . 13
mean.mvn . . . 13
mix-class . . . 14
MixUpdateMeans . . . 14
mvn-class . . . 15
PartitionData . . . 15
print.mix . . . 16
print.mvn . . . 16
ReconstituteData . . . 17
rGMM . . . 17
show,mix-method . . . 19
show,mvn-method . . . 19
vcov.mix . . . 20
vcov.mvn . . . 20

Index **21**

`CalHar` *Calinski-Harabasz Index*

Description

Calculates the Calinski-Harabasz index (ratio of between-cluster to within-cluster dispersion; higher values indicate better separation).

Usage

```
CalHar(data, assign, means)
```

Arguments

data	Numeric matrix of observations.
assign	Vector of cluster assignments.
means	List of cluster mean vectors.

Value

Numeric scalar; higher values indicate better separation.

ChooseK *Cluster Number Selection*

Description

Function to choose the number of clusters k. Examines cluster numbers between k0 and k1. For each cluster number, generates boot bootstrap data sets, fits the Gaussian Mixture Model (FitGMM), and calculates quality metrics (ClustQual). For each metric, determines the optimal cluster number k_opt, and the k_1SE, the smallest cluster number whose quality is within 1 SE of the optimum.

Usage

```
ChooseK(
  data,
  k0 = 2,
  k1 = NULL,
  boot = 100,
  init_means = NULL,
  fix_means = FALSE,
  init_covs = NULL,
  lambda = 0,
  init_props = NULL,
  maxit = 10,
  eps = 1e-04,
  report = TRUE
)
```

Arguments

data	Numeric data matrix.
k0	Minimum number of clusters.
k1	Maximum number of clusters.
boot	Bootstrap replicates.

`init_means`	Optional list of initial mean vectors.
`fix_means`	Fix the means to their starting value? Must provide initial values.
`init_covs`	Optional list of initial covariance matrices.
`lambda`	Optional ridge term added to covariance matrix to ensure positive definiteness.
`init_props`	Optional vector of initial cluster proportions.
`maxit`	Maximum number of EM iterations.
`eps`	Minimum acceptable increment in the EM objective.
`report`	Report bootstrap progress?

Value

List containing `Choices`, the recommended number of clusters according to each quality metric, and `Results`, the mean and standard error of the quality metrics at each cluster number evaluated.

See Also

See `ClustQual` for evaluating cluster quality, and `FitGMM` for estimating the GMM with a specified cluster number.

Examples

```
set.seed(100)
mean_list <- list(c(2, 2), c(2, -2), c(-2, 2), c(-2, -2))
data <- rGMM(n = 500, d = 2, k = 4, means = mean_list)
choose_k <- ChooseK(data, k0 = 2, k1 = 6, boot = 10)
choose_k$Choices
```

`ClustQual` *Cluster Quality*

Description

Evaluates cluster quality. Returns the following metrics:

- BIC: Bayesian Information Criterion; lower is better.
- CHI: Calinski-Harabasz index; higher is better.
- DBI: Davies-Bouldin index; lower is better.
- SIL: Mean silhouette width; higher is better.

Usage

```
ClustQual(fit)
```

Arguments

fit	Object of class mix.

Value

List containing the cluster quality metrics.

See Also

See ChooseK for using quality metrics to choose the cluster number.

Examples

```
set.seed(100)

# Data generation
mean_list = list(
c(2, 2, 2),
c(-2, 2, 2),
c(2, -2, 2),
c(2, 2, -2)
)

data <- rGMM(n = 500, d = 3, k = 4, means = mean_list)
fit <- FitGMM(data, k = 4)

# Clustering quality
cluster_qual <- ClustQual(fit)
```

CombineMIs *Combine Multiple Imputations*

Description

Combines point estimates and their estimated sampling (co)variances across multiple imputations using the usual multiple-imputation combining rules.

Usage

```
CombineMIs(points, covs)
```

Arguments

points	List of point estimates (each may be a vector or scalar).
covs	List of estimated sampling covariance matrices (or variances for scalar estimates), one per imputation.

Value

List containing the combined point estimate (`point`) and the combined sampling covariance (`cov`).

Examples

```
set.seed(100)

# Generate data and introduce missingness.
data <- rGMM(n = 25, d = 2, k = 1)
data[1, 1] <- NA
data[2, 2] <- NA
data[3, ] <- NA

# Fit GMM.
fit <- FitGMM(data)

# Lists to store summary statistics.
points <- list()
covs <- list()

# Perform 50 multiple imputations.
# For each, calculate the marginal mean and its sampling variance.
for (i in seq_len(50)) {
  imputed <- GenImputation(fit)
  points[[i]] <- apply(imputed, 2, mean)
  covs[[i]] <- cov(imputed) / nrow(imputed)
}

# Combine summary statistics across imputations.
results <- CombineMIs(points, covs)
```

DavBou *Davies-Bouldin Index*

Description

Calculates the Davies-Bouldin index (average similarity between each cluster and its most similar counterpart; lower values indicate better separation).

Usage

```
DavBou(data, assign, means)
```

Arguments

`data`	Numeric matrix of observations.
`assign`	Vector of cluster assignments.
`means`	List of cluster mean vectors.

Value

Numeric scalar; lower values indicate better separation.

`FitGMM` *Estimate Multivariate Normal Mixture*

Description

Given an $n \times d$ matrix of random vectors, estimates the parameters of a Gaussian Mixture Model (GMM). Accommodates arbitrary patterns of missingness at random (MAR) in the input vectors.

Usage

```
FitGMM(
  data,
  k = 1,
  init_means = NULL,
  fix_means = FALSE,
  init_covs = NULL,
  lambda = 0,
  init_props = NULL,
  maxit = 100,
  eps = 1e-06,
  report = TRUE
)
```

Arguments

`data`	Numeric data matrix.
`k`	Number of mixture components. Defaults to 1.
`init_means`	Optional list of initial mean vectors.
`fix_means`	Fix the means to their starting value? Must provide initial values.
`init_covs`	Optional list of initial covariance matrices.
`lambda`	Optional ridge term added to each component covariance matrix to ensure positive definiteness.
`init_props`	Optional vector of initial cluster proportions.
`maxit`	Maximum number of EM iterations.
`eps`	Minimum acceptable increment in the EM objective.
`report`	Report fitting progress?

Details

Initial values for the cluster means, covariances, and proportions are specified using `init_means`, `init_covs`, and `init_props`, respectively. If the data contain complete observations (rows with no missing elements), `FitGMM` will attempt to initialize these parameters internally using K-means. If there are no complete observations, initial values are required for `init_means`, `init_covs`, and `init_props`.

Value

- For a single component, an object of class `mvn`, containing the estimated mean and covariance, the final objective function, and the imputed data.
- For a multicomponent model $k > 1$, an object of class `mix`, containing the estimated means, covariances, cluster proportions, cluster responsibilities, and observation assignments.

See Also

See `rGMM` for data generation, and `ChooseK` for selecting the number of clusters.

Examples

```
# Single component without missingness
# Bivariate normal observations
sigma <- matrix(c(1, 0.5, 0.5, 1), nrow = 2)
data <- rGMM(n = 1e3, d = 2, k = 1, means = c(2, 2), covs = sigma)
fit <- FitGMM(data, k = 1)

# Single component with missingness
# Trivariate normal observations
mean_list <- list(c(-2, -2, -2), c(2, 2, 2))
sigma <- matrix(c(1, 0.5, 0.5, 0.5, 1, 0.5, 0.5, 0.5, 1), nrow = 3)
data <- rGMM(n = 1e3, d = 3, k = 2, means = mean_list, covs = sigma)
fit <- FitGMM(data, k = 2)

# Two components without missingness
# Trivariate normal observations
mean_list <- list(c(-2, -2, -2), c(2, 2, 2))
sigma <- matrix(c(1, 0.5, 0.5, 0.5, 1, 0.5, 0.5, 0.5, 1), nrow = 3)
data <- rGMM(n = 1e3, d = 3, k = 2, means = mean_list, covs = sigma)
fit <- FitGMM(data, k = 2)

# Four components with missingness
# Bivariate normal observations
# Note: Fitting is slow.
mean_list <- list(c(2, 2), c(2, -2), c(-2, 2), c(-2, -2))
sigma <- 0.5 * diag(2)
data <- rGMM(
n = 1000,
d = 2,
k = 4,
pi = c(0.35, 0.15, 0.15, 0.35),
miss = 0.1,
means = mean_list,
covs = sigma)
fit <- FitGMM(data, k = 4)
```

`FitMix` *Fit Multivariate Mixture Distribution*

Description

Given a matrix of random vectors, estimates the parameters for a mixture of multivariate normal distributions. Accommodates arbitrary patterns of missingness, provided the elements are missing at random (MAR).

Usage

```
FitMix(
  data,
  k = 2,
  init_means = NULL,
  fix_means = FALSE,
  init_covs = NULL,
  lambda = 0,
  init_props = NULL,
  maxit = 100,
  eps = 1e-06,
  report = FALSE
)
```

Arguments

`data`	Numeric data matrix.
`k`	Number of mixture components. Defaults to 2.
`init_means`	Optional list of initial mean vectors.
`fix_means`	Fix means to their starting values? Must initialize.
`init_covs`	Optional list of initial covariance matrices.
`lambda`	Optional ridge term added to covariance matrix to ensure positive definiteness.
`init_props`	Optional vector of initial cluster proportions.
`maxit`	Maximum number of EM iterations.
`eps`	Minimum acceptable increment in the EM objective.
`report`	Report fitting progress?

Value

Object of class `mix`.

FitMVN *Fit Multivariate Normal Distribution*

Description

Given a matrix of n x d-dimensional random vectors, possibly containing missing elements, estimates the mean and covariance of the best fitting multivariate normal distribution.

Usage

```
FitMVN(
  data,
  init_mean = NULL,
  fix_mean = FALSE,
  init_cov = NULL,
  lambda = 0,
  maxit = 100,
  eps = 1e-06,
  report = TRUE
)
```

Arguments

data	Numeric data matrix.
init_mean	Optional initial mean vector.
fix_mean	Fix the mean to its starting value? Initial values must be provided if TRUE.
init_cov	Optional initial covariance matrix.
lambda	Optional ridge term added to covariance matrix to ensure positive definiteness.
maxit	Maximum number of EM iterations.
eps	Minimum acceptable increment in the EM objective.
report	Report fitting progress?

Value

An object of class mvn.

GenImputation *Generate Stochastic Imputation*

Description

Generates a single stochastic imputation of the data from a fitted GMM. Observed values are unchanged; missing values are drawn from the conditional distribution given the observed data (or from the marginal distribution for fully missing rows). For multiple imputation, call this function repeatedly and combine results using `CombineMIs`.

Usage

```
GenImputation(fit)
```

Arguments

`fit`	Fitted model of class `mvn` or `mix` (e.g. from `FitGMM`).

Value

Numeric matrix with the same dimensions as `fit@Data`, with missing values imputed. If the fitted data have no missing values, returns the original data unchanged.

Examples

```
set.seed(100)

# Generate data and introduce missingness.
data <- rGMM(n = 25, d = 2, k = 1)
data[1, 1] <- NA
data[2, 2] <- NA
data[3, ] <- NA

# Fit GMM.
fit <- FitGMM(data)

# Generate imputation.
imputed <- GenImputation(fit)
```

logLik.mix *Log-Likelihood for Fitted GMM*

Description

Returns the final EM objective; for models with missing data this is not the exact log-likelihood of the observed data.

Usage

```
## S3 method for class 'mix'
logLik(object, ...)
```

Arguments

`object`	A `mix` object.
`...`	Unused.

Value

Numeric scalar (the stored EM objective).

`logLik.mvn` *Log-Likelihood for Fitted MVN Model*

Description

Returns the final EM objective; for models with missing data this is not the exact log-likelihood of the observed data.

Usage

```
## S3 method for class 'mvn'
logLik(object, ...)
```

Arguments

`object`	A `mvn` object.
`...`	Unused.

Value

Numeric scalar (the stored EM objective).

mean.mix *Mean for Fitted GMM*

Description

Mean for Fitted GMM

Usage

```
## S3 method for class 'mix'
mean(x, ...)
```

Arguments

x	A mix object.
...	Unused.

Value

List of estimated cluster mean vectors.

mean.mvn *Mean for Fitted MVN Model*

Description

Mean for Fitted MVN Model

Usage

```
## S3 method for class 'mvn'
mean(x, ...)
```

Arguments

x	A mvn object.
...	Unused.

Value

The estimated mean vector.

mix-class *Mixture Model Class*

Description

Defines a class to hold Gaussian Mixture Models.

Slots

`Assignments` Maximum a posteriori assignments.

`Completed` Completed data, with missing values imputed to their posterior expectations.

`Components` Number of components.

`Covariances` List of fitted cluster covariance matrices.

`Data` Original data, with missing values present.

`Density` Density of each component at each example.

`Means` List of fitted cluster means.

`Objective` Final value of the EM objective.

`Proportions` Fitted cluster proportions.

`Responsibilities` Posterior membership probabilities for each example.

MixUpdateMeans *Mean Update for Mixture of MVNs with Missingness.*

Description

Paper eq. (7): $\mu_j^{(r+1)} = (1/n_j) \sum_i \gamma_{ij} \hat{y}_{ij}$ (responsibility-weighted average of working responses).

Usage

```
MixUpdateMeans(split_data, means, covs, gamma)
```

Arguments

`split_data`	Data partitioned by missingness.
`means`	List of component means.
`covs`	List of component covariances.
`gamma`	List of component responsibilities.

Value

List containing the updated component means.

mvn-class *Multivariate Normal Model Class*

Description

Defines a class to hold multivariate normal models.

Slots

`Completed` Completed data, with missing values imputed to their posterior expectations.

`Covariance` Fitted covariance matrix.

`Data` Original data, with missing values present.

`Mean` Fitted mean vector.

`Objective` Final value of the EM objective.

PartitionData *Partition Data by Missingness Pattern*

Description

Splits the input data into complete cases, incomplete cases (at least one missing value), and empty cases (all values missing). Useful for custom workflows or inspecting missingness patterns.

Usage

```
PartitionData(data)
```

Arguments

`data`	A numeric matrix or data frame (coerced to matrix).

Value

List containing:

- The original row and column names: ‘orig_row_names‘, ‘orig_col_names‘.
- The original row and column numbers: ‘n_row‘ and ‘n_col‘.
- The complete cases ‘data_comp‘.
- The incomplete cases ‘data_incomp‘.
- The empty cases ‘data_empty‘.
- Counts of complete ‘n0‘, incomplete ‘n1‘, and empty ‘n2‘ cases.
- Initial order of the observations ‘init_order‘.

print.mix *Print Fitted GMM*

Description

Print method for objects of class `mix`.

Usage

```
## S3 method for class 'mix'
print(x, ...)
```

Arguments

`x`	A `mix` object.
`...`	Unused.

Value

Invisibly returns `x`.

print.mvn *Print Fitted MVN Model*

Description

Print Fitted MVN Model

Usage

```
## S3 method for class 'mvn'
print(x, ...)
```

Arguments

`x`	A `mvn` object.
`...`	Unused.

Value

Invisibly returns `x`.

ReconstituteData *Reconstitute Data*

Description

Reassembles a data matrix from the list returned by PartitionData, restoring the original row order and dimension names.

Usage

```
ReconstituteData(split_data)
```

Arguments

split_data	List returned by PartitionData.

Value

Numeric matrix with the same dimensions and row order as the original data passed to PartitionData.

rGMM *Generate Data from Gaussian Mixture Models*

Description

Generates an $n \times d$ matrix of multivariate normal random vectors with observations (examples) as rows. If $k = 1$, all observations belong to the same cluster. If $k > 1$ the observations are generated via a two-step procedure. First, the cluster membership is drawn from a multinomial distribution, with mixture proportions specified by pi. Conditional on cluster membership, the observation is drawn from a multivariate normal distribution, with cluster-specific mean and covariance. The cluster means are provided using means, and the cluster covariance matrices are provided using covs. If $miss > 0$, missingness is introduced, completely at random, by setting that proportion of elements in the data matrix to NA.

Usage

```
rGMM(n, d = 2, k = 1, pi = NULL, miss = 0, means = NULL, covs = NULL)
```

Arguments

n	Observations (rows).
d	Observation dimension (columns).
k	Number of mixture components. Defaults to 1.
pi	Mixture proportions. If omitted, components are assumed equiprobable.
miss	Proportion of elements missing, $miss \in [0, 1)$.

means	Either a prototype mean vector, or a list of mean vectors. Defaults to the zero vector.
covs	Either a prototype covariance matrix, or a list of covariance matrices. Defaults to the identity matrix.

Value

Numeric matrix with observations as rows. Row numbers specify the true cluster assignments.

See Also

For estimation, see FitGMM.

Examples

```
set.seed(100)
# Single component without missingness.
# Bivariate normal observations.
cov <- matrix(c(1, 0.5, 0.5, 1), nrow = 2)
data <- rGMM(n = 1e3, d = 2, k = 1, means = c(2, 2), covs = cov)

# Single component with missingness.
# Trivariate normal observations.
mean_list <- list(c(-2, -2, -2), c(2, 2, 2))
cov <- matrix(c(1, 0.5, 0.5, 0.5, 1, 0.5, 0.5, 0.5, 1), nrow = 3)
data <- rGMM(n = 1e3, d = 3, k = 2, means = mean_list, covs = cov)

# Two components without missingness.
# Trivariate normal observations.
mean_list <- list(c(-2, -2, -2), c(2, 2, 2))
cov <- matrix(c(1, 0.5, 0.5, 0.5, 1, 0.5, 0.5, 0.5, 1), nrow = 3)
data <- rGMM(n = 1e3, d = 3, k = 2, means = mean_list, covs = cov)

# Four components with missingness.
# Bivariate normal observations.
mean_list <- list(c(2, 2), c(2, -2), c(-2, 2), c(-2, -2))
cov <- 0.5 * diag(2)
data <- rGMM(
n = 1000,
d = 2,
k = 4,
pi = c(0.35, 0.15, 0.15, 0.35),
miss = 0.1,
means = mean_list,
covs = cov)
```

show,mix-method *Show for Fitted Mixture Models*

Description

Show for Fitted Mixture Models

Usage

```
## S4 method for signature 'mix'
show(object)
```

Arguments

object	A mix object.

show,mvn-method *Show for Multivariate Normal Models*

Description

Show for Multivariate Normal Models

Usage

```
## S4 method for signature 'mvn'
show(object)
```

Arguments

object	A mvn object.

vcov.mix *Covariance for Fitted GMM*

Description

Covariance for Fitted GMM

Usage

```
## S3 method for class 'mix'
vcov(object, ...)
```

Arguments

object	A mix object.
...	Unused.

Value

List of estimated cluster covariance matrices.

vcov.mvn *Covariance for Fitted MVN Model*

Description

Covariance for Fitted MVN Model

Usage

```
## S3 method for class 'mvn'
vcov(object, ...)
```

Arguments

object	A mvn object.
...	Unused.

Value

The estimated covariance matrix.

Index

CalHar, 2
ChooseK, 3, *5*, *8*
ClustQual, *3*, 4, *4*
CombineMIs, 5, *11*

DavBou, 6

FitGMM, *3, 4*, 7, *11*, *18*
FitMix, 9
FitMVN, 10

GenImputation, 11

logLik.mix, 11
logLik.mvn, 12

mean.mix, 13
mean.mvn, 13
mix-class, 14
MixUpdateMeans, 14
mvn-class, 15

PartitionData, 15, *17*
print.mix, 16
print.mvn, 16

ReconstituteData, 17
rGMM, *8*, 17

show,mix-method, 19
show,mvn-method, 19

vcov.mix, 20
vcov.mvn, 20