Package 'DPCD'

May 7, 2026

Title Dirichlet Process Clustering with Dissimilarities

Version 0.0.1

Description A Bayesian hierarchical model for clustering dissimilarity data using the Dirichlet process. The latent configuration of objects and the number of clusters are automatically inferred during the fitting process. The package supports multiple models which are available to detect clusters of various shapes and sizes using different covariance structures. Additional functions are included to ensure adequate model fits through prior and posterior predictive checks.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.3

Imports ggplot2, bayesplot, mcclust, cluster, truncnorm

Suggests spelling,

Config/testthat/edition 3

Depends nimble, R (>= 3.5)

URL https://github.com/SamMorrissette/DPCD

BugReports https://github.com/SamMorrissette/DPCD/issues

Language en-US

LazyData true

LazyDataCompression xz

NeedsCompilation no

Author Sam Morrissette [cph, aut, cre]

Maintainer Sam Morrissette <samuel.morrissette01@gmail.com>

Repository CRAN

Date/Publication 2025-12-19 14:00:02 UTC

Contents

`bs_score` *Calculate the Bayesian Silhouette Score*

Description

This function calculates the Bayesian Silhouette (BS) Score for a DPCD model fit using posterior MCMC samples. The BS score can be used to evaluate the clustering quality of a fit and to compare different models.

Usage

```
bs_score(mcmc_samples)
```

Arguments

`mcmc_samples`	An object of class `mcmc` or `mcmc.list` containing posterior samples from a DPCD model fit using `run_dpcd()`. Variables `x` and `z` must be included in the output parameters.

Details

The Bayesian Silhouette Score is computed by calculating the silhouette score for each MCMC iteration based on the latent positions (`x`) and cluster assignments (`z`). The silhouette score measures how similar an object is to its own cluster compared to other clusters. The BS score is then obtained by averaging the silhouette scores across all MCMC iterations. Higher values of the BS score indicate a higher-quality DPCD model in terms of its clustering structure.

Value

A numeric value representing the average silhouette score across all MCMC iterations.

Examples

```
bs_score(mcmc_example)
```

dis_mat_example *Dissimilarity Matrix Example*

Description

A dissimilarity matrix computed with `stats::dist()` on a simulated dataset.

Usage

```
data(dis_mat_example)
```

Format

An object of class `stats::dist()` containing pairwise similarities for ($n = 20$) objects.

Details

The object is intended for examples and vignettes.

Source

Generated by simulating $n = 20$ objects from a two-component mixture distribution and then computing a dissimilarity matrix via `stats::dist()`.

extract_clusters *Extract clusters from MCMC samples*

Description

This function extracts estimated cluster memberships from MCMC samples obtained from a DPCD model fit.

Usage

```
extract_clusters(mcmc_samples)
```

Arguments

`mcmc_samples`	An object of class `mcmc` or `mcmc.list` containing posterior samples from a DPCD model fit using `run_dpcd()`. The variable `z` must be included in the output parameters.

Details

This function uses the cluster membership variable, `z`, from the provided MCMC samples to compute the posterior similarity matrix (PSM) based on the sampled cluster assignments. Using the PSM, it then determines the estimated cluster memberships by maximizing the posterior expected adjusted Rand index, following the method of Fritsch and Ickstadt (2009).

Value

A vector of labels that indicate the estimated cluster membership for each observation.

References

Fritsch, Arno & Ickstadt, Katja. (2009). An Improved Criterion for Clustering Based on the Posterior Similarity Matrix. Bayesian Analysis. 4. doi:10.1214/09-BA414.

See Also

`mcclust::maxpear()`

Examples

```
extract_clusters(mcmc_example)
```

`mcmc_example` *MCMC Example Output*

Description

Posterior samples returned by `run_dpcd()` after fitting an Equal Spherical (ES) model to simulated dissimilarities.

Usage

```
data(mcmc_example)
```

Format

An object of class `mcmc` containing posterior draws for the monitored parameters from the DPCD model fit. It contains 4,000 rows (MCMC iterations) and 110 columns.

Details

The object is intended for examples and vignettes.

Source

Generated by fitting an Equal Spherical (ES) DPCD model to the dissimilarities calculated from a small ($n = 20$) simulated dataset with two mixture components.

plot_objects *Plot the Object Configuration*

Description

Generates a plot of the posterior mean of the latent coordinates (x) from a DPCD model fit, aligned to a specified target matrix using a Procrustes transformation.

Usage

```
plot_objects(mcmc_samples, target_matrix, show_clusters = TRUE, ...)
```

Arguments

mcmc_samples	An object of class mcmc or mcmc.list containing posterior samples from a DPCD model fit using run_dpcd(). Variable x must be included in the output parameters.
target_matrix	A matrix used as the target for aligning the posterior latent coordinates (x) via a Procrustes transformation.
show_clusters	Logical argument indicating whether to colour points by their cluster membership. If TRUE, then z must be included in mcmc_samples.
...	Additional arguments to be passed to plot() (2 dimensions) or pairs() (higher dimensions).

Details

Since the latent coordinates are non-identifiable due to invariance of Euclidean distances to rotation, reflection, and translation, this function first aligns the posterior samples of x to a specified target matrix using a Procrustes transformation. Then, it computes the posterior mean of the aligned latent coordinates and generates a plot. If show_clusters is set to TRUE, points are coloured according to their cluster memberships, which is estimated through maximizing the posterior expected adjusted Rand index (Fritsch and Ickstadt, 2009).

Value

A scatter plot (for 2-dimensional latent space) or pairs plot (for higher dimensions) of the object configuration.

References

Fritsch, Arno & Ickstadt, Katja. (2009). An Improved Criterion for Clustering Based on the Posterior Similarity Matrix. Bayesian Analysis. 4. doi:10.1214/09-BA414.

Examples

```
target_matrix <- cmdscale(dis_mat_example, k = 2)
plot_objects(mcmc_example, target_matrix, show_clusters = TRUE)
```

post_predictive *Posterior Predictive Check*

Description

This function simulates dissimilarities from the posterior predictive distribution of a specified DPCD model and optionally plots the density of the simulated dissimilarities against the observed dissimilarities.

Usage

```
post_predictive(
  mcmc_samples,
  dis_matrix,
  nsim = 1000,
  scale = TRUE,
  plot = TRUE
)
```

Arguments

mcmc_samples	An object of class mcmc or mcmc.list containing posterior samples from a DPCD model fit using run_dpcd(). Both the latent positions x and the error variance sigma_sq must be included in mcmc_samples.
dis_matrix	A distance structure such as that returned by stats::dist or a full symmetric matrix containing the dissimilarities.
nsim	Number of datasets to simulate from the posterior predictive distribution.
scale	Logical argument indicating whether to scale the dissimilarities so that the maximum value is 1.
plot	Logical argument indicating whether to plot the simulated dissimilarities against the observed dissimilarities. See details for more information.

Details

A posterior predictive check is used to assess if datasets drawn from the posterior predictive distribution are consistent with the observed data. Posterior predictive checks differ from prior predictive checks in that they incorporate information from the observed data. If the model fits the data well, the observed dissimilarities should look similar to dissimilarities simulated from the posterior predictive distribution.

If plot = TRUE, a plot is created to compare the density of the observed dissimilarities to the densities of the dissimilarities simulated from the posterior predictive distribution using bayesplot::ppc_dens_overlay().

See run_dpcd() for details on the DPCD models and hyperparameters.

Value

A matrix of simulated dissimilarities from the posterior predictive distribution with `nsim` rows and `n * (n-1) / 2` columns, where `n` is the number of objects (i.e. the number of rows/columns of `dis_matrix`).

References

Gabry, J., Simpson, D., Vehtari, A., Betancourt, M., & Gelman, A. (2019). Visualization in Bayesian workflow. Journal of the Royal Statistical Society A, 182(2), 389–402. https://doi.org/10.1111/rssa.12378

See Also

`run_dpcd()`

Examples

```
ppc <- post_predictive(mcmc_example, dis_mat_example, nsim = 100, plot = TRUE)
```

---

prior_predictive *Prior Predictive Check*

Description

This function simulates dissimilarities from the prior predictive distribution of a specified DPCD model and optionally plots the density of the simulated dissimilarities against the observed dissimilarities.

Usage

```
prior_predictive(
  dis_matrix,
  model_name = c("UU", "EU", "UD", "ED", "US", "ES"),
  p = 2,
  trunc_value = 15,
  hyper_params = NULL,
  scale = TRUE,
  nsim = 1000,
  plot = TRUE
)
```

Arguments

`dis_matrix`	A distance structure such as that returned by stats::dist or a full symmetric matrix containing the dissimilarities.
`model_name`	The DPCD model from which to draw prior predictive samples. Must be one of "UU", "EU", "UD", "ED", "US", or "ES".

p	The dimension of the space in which the objects are embedded. Must be at least 2.
trunc_value	The truncation level for the stick-breaking representation of the Dirichlet process.
hyper_params	A named list of hyperparameter values. See details for more information.
scale	Logical argument indicating whether to scale the dissimilarities so that the maximum value is 1.
nsim	Number of datasets to simulate from the prior predictive distribution.
plot	Logical argument indicating whether to plot the simulated dissimilarities against the observed dissimilarities. See details for more information.

Details

A prior predictive check is used to assess if datasets drawn from the prior predictive distribution are consistent with the observed data. Most of the mass of the prior predictive distribution should be placed on plausible values of the dissimilarities, while little or no mass should be placed on implausible values.

If `plot = TRUE`, a plot is created to compare the density of the observed dissimilarities to the densities of the dissimilarities simulated from the prior predictive distribution using `bayesplot::ppc_dens_overlay()`.

See `run_dpcd()` for details on the DPCD models and hyperparameters.

Value

A matrix of simulated dissimilarities from the prior predictive distribution with `nsim` rows and `n * (n-1) / 2` columns, where `n` is the number of objects (i.e. the number of rows/columns of `dis_matrix`).

References

Gabry, J., Simpson, D., Vehtari, A., Betancourt, M., & Gelman, A. (2019). Visualization in Bayesian workflow. Journal of the Royal Statistical Society A, 182(2), 389–402. https://doi.org/10.1111/rssa.12378

See Also

`run_dpcd()`

Examples

```
ppc <- prior_predictive(dis_mat_example, "UU", p = 2, nsim = 100, plot = TRUE)
```

procrustes *Procrustes Transformation*

Description

Aligns a given object configuration to a target object configuration using a Procrustes transformation.

Usage

```
procrustes(X, Y)
```

Arguments

X	The target configuration.
Y	The configuration to be aligned to X.

Details

This function performs a Procrustes transformation to align a given configuration, Y, to the target configuration, X, using a combination of translation and rotation. The transformation aims to minimize the sum of squared differences between the two configurations.

X and Y should be numeric matrices of the same dimension.

Value

The transformed version of Y aligned to X.

Examples

```
X <- matrix(rnorm(20), ncol = 2)
rotation_matrix <- matrix(c(cos(pi/4), -sin(pi/4), sin(pi/4), cos(pi/4)), ncol = 2)
Y <- X %*% rotation_matrix + 2
Y_transformed <- procrustes(X, Y)
```

run_dpcd *Run Dirichlet Process Clustering with Dissimilarities*

Description

This function fits an infinite mixture model to dissimilarity data using a Dirichlet Process prior. The model is constructed and MCMC sampling is performed using the nimble package. Currently, there are six different models available.

Usage

```
run_dpcd(
  dis_matrix,
  model_name = c("UU", "EU", "UD", "ED", "US", "ES"),
  p = 2,
  trunc_value = 15,
  hyper_params = NULL,
  init_params = NULL,
  output_params = c("x", "z", "pi", "mu", "Sigma", "sigma_sq"),
  scale = TRUE,
  WAIC = TRUE,
  nchains = 1,
  niter = 10000,
  nburn = 0,
  ...
)
```

Arguments

dis_matrix	A distance structure such as that returned by stats::dist or a full symmetric matrix containing the dissimilarities.
model_name	The DPCD model to fit. Must be one of "UU" (unequal unrestricted), "EU" (equal unrestricted), "UD" (unequal diagonal), "ED" (equal diagonal), "US" (unequal spherical), or "ES" (equal spherical). See details for a brief description of each model.
p	The dimension of the space in which the objects are embedded. Must be at least 2.
trunc_value	The truncation level for the stick-breaking representation of the Dirichlet process.
hyper_params	A named list of hyperparameter values. See details for more information.
init_params	A named list of initial values for model parameters. See details for more information.
output_params	A character vector of model parameters to save in the output. See details for more information.
scale	Logical argument indicating whether to scale the dissimilarities so that the maximum value is 1.
WAIC	Logical argument indicating whether to compute the Watanabe-Akaike Information Criterion (WAIC) for model comparison.
nchains	Number of MCMC chains to run.
niter	Number of MCMC iterations to run.
nburn	Number of MCMC burn-in iterations.
...	Additional arguments passed to nimble::nimbleMCMC() from the nimble package.

Details

Dirichlet Process Clustering with Dissimilarities (DPCD) models dissimilarity data using an infinite mixture model with a Dirichlet Process prior. The six available covariance structures for mixture components are:

- **"UU"**: Unequal Unrestricted — each component has its own unrestricted covariance matrix.
- **"EU"**: Equal Unrestricted — components share a common unrestricted covariance matrix.
- **"UD"**: Unequal Diagonal — each component has its own diagonal covariance matrix.
- **"ED"**: Equal Diagonal — components share a common diagonal covariance matrix.
- **"US"**: Unequal Spherical — each component has its own spherical covariance matrix.
- **"ES"**: Equal Spherical — components share a common spherical covariance matrix.

The `hyper_params` list allows users to specify custom hyperparameter values. Some hyperparameters are common across all models, while others depend on the selected covariance structure.

Common hyperparameters:

- `alpha_0`: Concentration parameter for the Dirichlet Process prior.
- `a_0`, `b_0`: Shape and scale parameters for the Inverse-Gamma prior on the measurement error parameter.
- `lambda`: Scaling parameter for the prior on component means.
- `mu_0`: Mean vector for the prior on component means.

Model-specific hyperparameters:

- `nu_0` and `Psi_0`(degrees of freedom and scale matrix for the Inverse-Wishart prior) - UU and EU only.
- `alpha_tau` and `beta_tau` (shape and scale parameters for the Inverse-Gamma prior) - UD, ED, US, and ES only.

The `init_params` list allows users to supply initial values for model parameters to assist MCMC convergence. The following parameters may be initialized:

- `x`: `n × p` matrix of latent positions.
- `sigma_sq`: Scalar measurement error variance.
- `mu`: `trunc_value × p` matrix of component means.
- `Sigma`: `p × p` covariance matrix.
- `tau_sq`: Scalar variance parameter (for "US" and "ES" only).
- `tau_vec`: Length-`p` variance vector (for "UD" and "ED" only).
- `beta`: Length `trunc_value-1` vector of stick-breaking weights.
- `z`: Length-`n` vector of cluster assignments.

Default values are used for both `hyper_params` and `init_params` if none are supplied.

The `output_params` vector specifies which model parameters should be saved in the MCMC output. Valid names include `"beta"`, `"pi"`, `"z"`, `"mu"`, `"Sigma"`, `"sigma_sq"`, `"x"`, and `"delta"`.

Value

Posterior samples are returned a coda `mcmc` object, unless `nchains > 1`, in which case the posterior samples are returned as a coda `mcmc.list` object. If `WAIC = TRUE`, a named list is returned containing the posterior samples and the WAIC value.

Examples

```
# Fit the unequal unrestricted model with default settings
mcmc_samples <- run_dpcd(dis_mat_example, "UU", p = 2, niter = 10000, nburn = 2000)
summary(mcmc_samples)

# Fit the equal spherical model with custom hyperparameters and initial values
custom_hyper_params <- list(alpha_tau = 0.01, beta_tau = 0.01)
custom_init_params <- list(sigma_sq = 0.5)
mcmc_samples_es <- run_dpcd(dis_mat_example, "ES", p = 2,
                            hyper_params = custom_hyper_params,
                            init_params = custom_init_params,
                            niter = 10000, nburn = 2000, WAIC = TRUE)
```

Index