Package: vimixr (via r-universe)

May 25, 2026

Title Collapsed Variational Inference for Dirichlet Process (DP) Mixture Model

Version 0.1.2

Description Collapsed Variational Inference for a Dirichlet Process (DP) mixture model with unknown covariance matrix structure and DP concentration parameter. It enables efficient clustering of high-dimensional data with significantly improved computational speed than traditional MCMC methods. The package incorporates 8 parameterisations and corresponding prior choices for the unknown covariance matrix, from which the user can choose and apply accordingly.

Encoding UTF-8

Roxygen list(markdown = TRUE)

License MIT + file LICENSE

Imports ggplot2, patchwork, Rcpp, Rfast, rlang, parallel, stats, irlba

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

Config/testthat/edition 3

URL https://github.com/annesh07/vimixr, https://annesh07.github.io/vimixr/

BugReports https://github.com/annesh07/vimixr/issues

LinkingTo Rcpp, RcppEigen

VignetteBuilder knitr

Config/roxygen2/version 8.0.0

Config/pak/sysreqs make

Repository https://annesh07.r-universe.dev

Date/Publication 2026-05-25 21:26:22 UTC

RemoteUrl https://github.com/annesh07/vimixr

RemoteRef HEAD

RemoteSha 12262d76ac6e54451eb58e8a73de88b8de71701b

Contents

`cum_clustprop` *cum_clustprop*

Description

Calculate the columnwise sum of rowwise cummulative probability

Usage

```
cum_clustprop(P1)
```

Arguments

`P1`	probability matrix

Value

rowwise cummulative probability

cum_clustprop_var *cum_clustprop_var*

Description

Calculate the columnwise sum of rowwise cummulative probability for variance

Usage

```
cum_clustprop_var(P1)
```

Arguments

P1	probability matrix

Value

rowwise cummulative probability for variance

cvi_npmm *Collapsed variational inference for non-parametric Bayesian mixture models*

Description

Collapsed variational inference for non-parametric Bayesian mixture models

Usage

```
cvi_npmm(
  X,
  variational_params,
  prior_shape_alpha,
  prior_rate_alpha,
  post_shape_alpha,
  post_rate_alpha,
  prior_mean_eta,
  post_mean_eta,
  log_prob_matrix = NULL,
  maxit = 100,
  n_inits = 5,
  Seed = NULL,
  parallel = FALSE,
  pca_plot = FALSE,
  covariance_type = "full",
  fixed_variance = FALSE,
```

```
  cluster_specific_covariance = TRUE,
  variance_prior_type = c("IW", "decomposed", "sparse", "off-diagonal normal"),
  ...
)
```

Arguments

X	input data as a matrix
variational_params	number of clusters in the variational distribution
prior_shape_alpha	shape parameter of Gamma prior for the DP concentration parameter alpha. Default is 0.001
prior_rate_alpha	rate parameter of Gamma prior for the DP concentration parameter alpha. Default is 0.001
post_shape_alpha	initial value for posterior update of shape parameter for alpha. Default is 0.001
post_rate_alpha	initial value for posterior update of ratee parameter for alpha. Default is 0.001
prior_mean_eta	mean vector of MVN prior for the DP mean parameters. Default is zero vector
post_mean_eta	initial value of posterior update for the DP mean parameter
log_prob_matrix	logarithm of cluster allocation probability matrix. Default is NULL
maxit	maximum number of iterations. Default is 100
n_inits	Number of random initialisations if log_prob_matrix and other case-specific hyperparameters are NULL. Default is 5
Seed	Seeds for random initialisation; either a vector of n_inits integers or NULL. Default is NULL.
parallel	Logical input for parallelisation. Default is FALSE
pca_plot	Logical input for pca plot. Default is FALSE
covariance_type	covariance matrix is considered diagonal or full. Default is 'full'
fixed_variance	covariance matrix of the data is considered known (fixed) or unknown. Default is FALSE
cluster_specific_covariance	covariance matrix is specific to a cluster allocation or it is same over all cluster choices. Default is TRUE
variance_prior_type	For unknown and full covariance matrix, choice of matrix prior is either Inverse-Wishart ('IW') or Cholesky-decomposed ('decomposed'). For unknown, full and cluster-specific covariance matrix, choice of matrix prior is either Inverse-Wishart ('IW'), element-wise Gamma and Laplace distributed ('sparse') or element-wise Gamma and Normal distributed ('off-diagonal normal')
...	additional parameters, further details given below

Details

The following models are supported in `vimixr`, listing their required input arguments in `...` when calling `cvi_npmm()`:

- **Known covariance**
  - *diagonal covariance* We need the following additional arguments:
    `cov_data`**: a non-negative diagonal matrix, representing the covariance of the data**

    `prior_precision_scalar_eta`**: a non-negative scalar, representing the precision prior for the DP mean par**

    `post_precision_scalar_eta`**: initial value for the posterior update of precision for the DP mean parameter**
  - *full covariance* We need the following additional arguments:
    `cov_data`**: a positive definite matrix, representing the covariance of the data**
    `prior_cov_eta`**: a positive definite matrix, representing the covariance prior for the DP mean parameters**

    `post_cov_eta`**: initial value for the posterior update of covariance for the DP mean parameters**
- **Unknown covariance (Global)**
  - *diagonal covariance* We need the following additional arguments:
    `prior_shape_scalar_cov`**: a non-negative scalar, representing the shape parameter of Gamma prior for the**

    `prior_rate_scalar_cov`**: a non-negative scalar, representing the rate parameter of Gamma prior for the pr**

    `post_shape_scalar_cov`**: initial value for posterior update of precision shape parameter**

    `post_rate_scalar_cov`**: initial value for posterior update of precision rate parameter**

    `prior_precision_scalar_eta`**: a non-negative scalar, representing the precision prior for the DP mean par**

    `post_precision_scalar_eta`**: initial value for the posterior update of precision for the DP mean parameter**
  - *Inverse-Wishart* We need the following additional arguments:
    `prior_df_cov`**: a scalar as the degree of freedom parameter of the Inverse-Wishart prior, Default value D+2**

    `prior_scale_cov`**: positive-definite matrix as the scale parameter of the Inverse-Wishart prior**

    `post_df_cov`**: initial value for the posterior update of degree of freedom**
    `post_scale_cov`**: initial value for the posterior update of scale matrix**
    `prior_cov_eta`**: a positive definite matrix, representing the covariance prior for the DP mean parameters**

    `post_cov_eta`**: initial value for the posterior update of covariance for the DP mean parameters**
  - *Cholesky-decomposition* We need the following additional arguments:

`prior_shape_diag_decomp`**: a non-negative scalar as the shape parameter of Gamma prior for diagonal ele**

`prior_rate_diag_decomp`**: a non-negative scalar as the rate parameter of Gamma prior for diagonal eleme**

`prior_mean_offdiag_decomp`**: a scalar as the mean parameter of Normal prior for off-diagonal elements of**

`prior_var_offdiag_decomp`**: a non-negative scalar as the variance parameter of Normal prior for off-diago**

`post_shape_diag_decomp`**: initial value for posterior update of the shape parameter for diagonal elements**

`post_rate_diag_decomp`**: initial value for posterior update of the rate parameter for diagonal elements**

`post_mean_offdiag_decomp`**: initial value for posterior update of the mean parameter for off-diagonal elem**

`post_var_offdiag_decomp`**: initial value for posterior update of the variance parameter for off-diagonal ele**

`prior_cov_eta`**: a positive definite matrix, representing the covariance prior for the DP mean parameters**

`post_cov_eta`**: initial value for the posterior update of covariance for the DP mean parameters**

- **Unknown covariance (cluster-specific)**
  - *Inverse Wishart* We need the following additional arguments:
    `prior_df_cs_cov`**: a vector representing degree of freedom parameters for each cluster-specific Inverse-Wi**

    `prior_scale_cs_cov`**: an array of positive-definite matrices representing scale matrix parameters for each**

    `post_df_cs_cov`**: initial value for posterior update of the degree of freedom parameters**

    `post_scale_cs_cov`**: initial value for posterior update of the scale matrix parameters**

    `scaling_cov_eta`**: a non-negative scaling factor for covariance matrix of the DP mean parameters**

  - *Element-wise Gamma and Laplace prior* We need the following additional arguments:
    `prior_shape_d_cs_cov`**: a non-negative vector as shape parameters for cluster-specific Gamma priors of th**

    `prior_rate_d_cs_cov`**: a non-negative matrix as rate parameter for cluster-specific Gamma prior of the di**

    `prior_var_offd_cs_cov`**: a non-negative vector as variance parameter for cluster-specific Laplace priors o**

    `post_shape_d_cs_cov`**: initial value for posterior update of the diagonal shape parameters**

    `post_rate_d_cs_cov`**: initial value for posterior update of the diagonal rate parameters**

    `post_var_offd_cs_cov`**: initial value for sum, squared sum and log sum of the off-diagonal variance param**

`scaling_cov_eta`**: a non-negative scaling factor for covariance matrix of the DP mean parameters**

– *Element-wise Gamma and Normal prior* We need the following additional arguments:
`prior_shape_d_cs_cov`**: a non-negative vector as shape parameters for cluster-specific Gamma priors of th**

`prior_rate_d_cs_cov`**: a non-negative matrix as rate parameter for cluster-specific Gamma prior of the di**

`prior_var_offd_cs_cov`**: a non-negative scalar as variance parameter for cluster-specific Normal priors of**

`post_shape_d_cs_cov`**: initial value for posterior update of the diagonal shape parameters**

`post_rate_d_cs_cov`**: initial value for posterior update of the diagonal rate parameters**

`post_mean_offd_cs_cov`**: initial value for posterior update of the off-diagonal mean parameters**

`scaling_cov_eta`**: a non-negative scaling factor for covariance matrix of the DP mean parameters**

Value

`[vimixr()]` returns a `list` with the following elements:

- `alpha`: posterior DP concentration parameter
- `Cluster number`: number of clusters from posterior probability allocation matrix
- `Cluster Proportion`: cluster proportions from posterior probability allocation matrix
- `log Probability matrix`: log of posterior probability allocation matrix
- `ELBO`: Optimisation of the ELBO function
- `Iterations`: Number of iterations required for convergence
- `PCA_viz`: A PCA `[ggplot2]` plot to visualize the clustering of data based on cluster labels
- `ELBO_viz`: A line `[ggplot2]` plot to visualize the ELBO optimisation

Examples

```
X <- rbind(matrix(rnorm(100, m=0, sd=0.5), ncol=2),
           matrix(rnorm(100, m=3, sd=0.5), ncol=2))

#for fixed-diagonal
res <- cvi_npmm(X, variational_params = 20, prior_shape_alpha = 0.001,
         prior_rate_alpha = 0.001, post_shape_alpha = 0.001,
         post_rate_alpha = 0.001, prior_mean_eta = matrix(0, 1, ncol(X)),
         post_mean_eta = matrix(0.001, 20, ncol(X)),
         log_prob_matrix = t(apply(matrix(-3, nrow(X), 20), 1,
                             function(x){x/sum(x)})), maxit = 100,
         fixed_variance = TRUE, covariance_type = "diagonal",
         prior_precision_scalar_eta = 0.001,
         post_precision_scalar_eta = matrix(0.001, 20, 1),
         cov_data = diag(ncol(X)))
 summary(res)
```

```
plot(res)
```

`CVI_update_function` *Update of the variational parameters*

Description

Update of the variational parameters

Usage

```
CVI_update_function(
  fixed_variance = FALSE,
  covariance_type = "diagonal",
  cluster_specific_covariance = TRUE,
  variance_prior_type = c("IW", "decomposed", "sparse", "off-diagonal normal"),
  X,
  inverts,
  params
)
```

Arguments

`fixed_variance`	whether the covariance is fixed or estimated. Default is `FALSE` which means it is estimated.
`covariance_type`	The assumed type of the covariance matrix. Can be either `"diagonal"` if it is the identify multiplied by a scalar, or `"full"` for a fully unspecified covariance matrix.
`cluster_specific_covariance`	whether the the covariance is shared across estimated clusters or is cluster specific. Default is `TRUE` which means it is cluster specific.
`variance_prior_type`	character string specifying the type of prior distribution for the covariance when cluster_specific_covariance is `TRUE`. Can be either `"IW"` or `"decomposed"` if `cluster_specific_covariance` is `FALSE`, and can be either `"IW"`, `"sparse"` or `"off-diagonal normal"` otherwise.
`X`	the data matrix
`inverts`	a list of inverses
`params`	a list of required arguments

Value

Updated parameters

eBa0 *Root for a0 hyper-parameter for Sparse DPMM*

Description

Root for a0 hyper-parameter for Sparse DPMM

Usage

```
eBa0(
  logP,
  X,
  a_min = min(1e-08, 1/ncol(X)),
  a_max = max(1e+06, ncol(X)),
  grid_points = min(ncol(X), 10000)
)
```

Arguments

logP	log of probability allocation matrix
X	observed data
a_min	minimum value of a0 for grid search
a_max	maximum value of a0 for grid search
grid_points	number of points for grid search

Value

No return value, called for side effects.

elbo_fixed_diagonal *ELBO calculating functions depending on type of model for covariance matrix*

Description

ELBO calculating functions depending on type of model for covariance matrix

Usage

```
elbo_fixed_diagonal(X, inverts, params)
```

Arguments

X	the data matrix
inverts	a list of inverses
params	a list of required arguments

Value

No return value, called for side effects.

`ELBO_function` *General ELBO function*

Description

General ELBO function

Usage

```
ELBO_function(
  fixed_variance = FALSE,
  covariance_type = "diagonal",
  cluster_specific_covariance = TRUE,
  variance_prior_type = c("IW", "decomposed", "sparse", "off-diagonal normal"),
  X,
  inverts,
  params
)
```

Arguments

`fixed_variance`	whether the covariance is fixed or estimated. Default is `FALSE` which means it is estimated.
`covariance_type`	The assumed type of the covariance matrix. Can be either `"diagonal"` if it is the identify multiplied by a scalar, or `"full"` for a fully unspecified covariance matrix.
`cluster_specific_covariance`	whether the the covariance is shared across estimated clusters or is cluster specific. Default is `TRUE` which means it is cluster specific.
`variance_prior_type`	character string specifying the type of prior distribution for the covariance when cluster_specific_covariance is `TRUE`. Can be either `"IW"` or `"decomposed"` if `cluster_specific_covariance` is `FALSE`, and can be either `"IW"`, `"sparse"` or `"off-diagonal normal"` otherwise.
`X`	the data matrix
`inverts`	a list of inverses
`params`	a list of required arguments

Value

ELBO values

generate_log_prob *Generate random log Probability matrix if not provided*

Description

Generate random log Probability matrix if not provided

Usage

```
generate_log_prob(N, T0, seed0)
```

Arguments

N	rows of the data matrix
T0	variational clusters
seed0	seed for generating log Probability matrix

Value

No return value, called for side effects.

log_sum_exp *Log-sum-exponential computation on the log probability allocation matrix*

Description

Log-sum-exponential computation on the log probability allocation matrix

Usage

```
log_sum_exp(Plog)
```

Arguments

Plog	log probability allocation matrix

Value

per sample ordered log probability allocation matrix

lower_tri_stats *lower_tri_stats*

Description

Extract lower diagonal elements of a Matrix, and perform sum, squared sum and log sum

Usage

```
lower_tri_stats(M)
```

Arguments

M	matrix

Value

a vector of sum, squared sum and log sum elements

mat_mult *mat_mult*

Description

Calculate matrix multiplication with optional transposition.

Usage

```
mat_mult(A, B, transpose_A = FALSE, transpose_B = FALSE)
```

Arguments

A	matrix or vector
B	matrix or vector
transpose_A	transpose A before multiplying
transpose_B	transpose B before multiplying

Value

A %*% B (or variant), as vector if either input was a vector

`mat_mult_t` *mat_mult_t*

Description

Calculate a combination of matrix multiplications

Usage

```
mat_mult_t(A, B, C)
```

Arguments

`A`	matrix
`B`	matrix
`C`	matrix

Value

A %% *B* %% t(C)

`params_check` *Function to check the list of type-specific arguments*

Description

Function to check the list of type-specific arguments

Usage

```
params_check(
  params,
  fixed_variance = FALSE,
  covariance_type = "diagonal",
  cluster_specific_covariance = TRUE,
  variance_prior_type = c("IW", "decomposed", "sparse", "off-diagonal normal")
)
```

Arguments

`params`	the list of required parameters
`fixed_variance`	whether covariance is assumed fixed or not; can be TRUE or FALSE
`covariance_type`	structure of covariance matrix; can be "diagonal" or "full"
`cluster_specific_covariance`	whether covariance matrix is cluster specific or not; can be TRUE or FALSE
`variance_prior_type`	prior distribution for the covariance matrix; can be "IW" or "decomposed" when cluster_specific_covariance = FALSE, or can be "IW", "sparse" or "off-diagonal normal" otherwise

Value

stops the code if the required list of arguments are not present

---

`plot.CVIoutput` *S3 plotting function for* `CVIoutput`*objects'*

Description

S3 plotting function for `CVIoutput`objects'

Usage

```
## S3 method for class 'CVIoutput'
plot(x, ...)
```

Arguments

`x`	a CVIoutput object
`...`	additional arguments

Value

A ggplot object representing visualisation

quadratic_form_diag *quadratic_form_diag*

Description

Calculate a combination of matrix multiplications

Usage

```
quadratic_form_diag(A, B)
```

Arguments

A	matrix
B	matrix

Value

diag(A %% *B* %% t(A))

run_single *CVI implementation for one set of initial parameters*

Description

CVI implementation for one set of initial parameters

Usage

```
run_single(
  config,
  X,
  N,
  D,
  T0,
  prior_shape_alpha,
  prior_rate_alpha,
  post_shape_alpha,
  post_rate_alpha,
  prior_mean_eta,
  post_mean_eta,
  fixed_variance,
  covariance_type,
  cluster_specific_covariance,
  variance_prior_type,
```

```
  maxit,
  varargs
)
```

Arguments

config	List of inputs that are generated if not user-provided
X	the data matrix
N	samples of X
D	dimensions of X
T0	variational clusters
prior_shape_alpha	shape parameter of Gamma prior for the DP concentration parameter alpha. Default is 0.001
prior_rate_alpha	rate parameter of Gamma prior for the DP concentration parameter alpha. Default is 0.001
post_shape_alpha	initial value for posterior update of shape parameter for alpha. Default is 0.001
post_rate_alpha	initial value for posterior update of ratee parameter for alpha. Default is 0.001
prior_mean_eta	mean vector of MVN prior for the DP mean parameters. Default is zero vector
post_mean_eta	initial value of posterior update for the DP mean parameter
fixed_variance	covariance matrix of the data is considered known (fixed) or unknown.
covariance_type	covariance matrix is considered diagonal or full.
cluster_specific_covariance	covariance matrix is specific to a cluster allocation or it is same over all cluster choices.
variance_prior_type	For unknown and full covariance matrix, choice of matrix prior is either Inverse-Wishart ('IW') or Cholesky-decomposed ('decomposed'). For unknown, full and cluster-specific covariance matrix, choice of matrix prior is either Inverse-Wishart ('IW'), element-wise Gamma and Laplace distributed ('sparse') or element-wise Gamma and Normal distributed ('off-diagonal normal')
maxit	Maximum number of iterations for variational updates
varargs	List of case specific parameters

Value

a `list` with the following elements:

- `alpha`: posterior DP concentration parameter
- `Cluster number`: number of clusters from posterior probability allocation matrix
- `Cluster Proportion`: cluster proportions from posterior probability allocation matrix

- `log Probability matrix`: log of posterior probability allocation matrix
- `ELBO`: Optimisation of the ELBO function
- `Iterations`: Number of iterations required for convergence

`sparse_cov_op` *sparse_cov_op*

Description

Calculate the sum, squared sum and log sum of off-diagonal vector elements from the covariance array

Usage

```
sparse_cov_op(X, P, inv_C0, L1)
```

Arguments

`X`	data matrix
`P`	probability allocation matrix
`inv_C0`	matrix corresponding diagonal elements of the cluster precision matrices
`L1`	cluster mean matrix

Value

likelihood term calculation in elbo

`sweep_3D` *sweep_3D*

Description

A C++ alternative of sweep() function from base R

Usage

```
sweep_3D(A, R, dims, n_threads = 4L)
```

Arguments

`A`	a 3D array
`R`	a vector
`dims`	dimensions in 3D
`n_threads`	number of threads

Value

sweep(A, 3, R, "*")

`t_mat_mult` *t_mat_mult*

Description

Calculate a combination of matrix multiplications

Usage

```
t_mat_mult(A, B, C)
```

Arguments

`A`	matrix
`B`	matrix
`C`	matrix

Value

t(A) %% *B* %% C

Index