Package ‘MoMPCA’

October 12, 2022

Type Package

Title Inference and Clustering for Mixture of Multinomial Principal Component Analysis

Version 1.0.1

Author Nicolas Jouvin

Maintainer Nicolas Jouvin <nicolas.jouvin@ec-lyon.fr>

Description Cluster any count data matrix with a fixed number of variables, such as document/term matrices. It integrates the dimension reduction aspect of topic models in the mixture models framework. Inference is done by means of a greedy Classification Variational Expectation Maximisation (C-VEM) algorithm. An Integrated Classification Likelihood (ICL) model selection is designed for selecting the latent dimension (number of topics) and the number of clusters. For more details, see the article of Jouvin et. al. (2020) <arxiv:1909.00721>.

License GPL-3

Encoding UTF-8

LazyData true

Depends R (>= 3.6.0)

Imports methods, topicmodels, tm, Matrix, slam, magrittr, dplyr, stats, doParallel, foreach

Suggests testthat (>= 2.1.0), knitr, markdown, rmarkdown, aricode, ggplot2, tidytex, reshape2

RoxygenNote 7.1.1

VignetteBuilder knitr

Collate 'BBCVEM.R' 'MoMPCA.R' 'benchmarks.R' 'data.R'
'init_functions.R' 'mmpcaClustcontrol_class.R'
'mmpcaClust_class.R' 'mmpca_clust.R'
'mmpca_clust_modelselection.R' 'plot.R' 'simulate_BBC.R'
'utility_functions.R' 'utils-pipe.R'

NeedsCompilation no

Repository CRAN

Date/Publication 2021-01-21 13:10:03 UTC
R topics documented:

- BBCmsg
- benchmarks-functions
- DTMtoSparse
- initializeBeta
- initialize_Y
- mmpcaClust-class
- mmpcaClustcontrol-class
- mmpca_clust
- mmpca_clust_modelselect
- MoMPCA
- plot,mmpcaClust,missing-method
- plot_bound
- plot_topics
- simulate_BBC

Description

Pre-processed BBC articles from the BBC news network.

Usage

data("BBCmsg")

Format

A list of 4 character vectors containing the vectorized and stemmed documents (i.e., unigrams with repetition):

- msg1 the birth of princess Charlotte
- msg2 black holes in astrophysics
- msg3 UK politics
- msg4 cancer diseases in medicine
### Description

These are wrapper to other methods for the clustering of count data. They can be used to initialize the clustering. It is also possible to implement your own benchmark function depending on other packages.

### Usage

- `benchmark.random(dtm, Q, ...)`
- `benchmark.kmeans_lda(dtm, Q, K, nruns = 1, ...)`

### Arguments

- `dtm` an S4 object of class `mmpcaClust`
- `Q` The number of clusters
- `...` Some argument to be consistent with the function’s skeleton: `K` and `nruns` are optional arguments for some of them.
- `K` Number of topics (dimension of the latent space).
- `nruns` Number of restart of the `kmeans()` algorithm.

### Value

A vector of size equal to the number of row of `dtm`, containing a Q-clustering

- `benchmark.random` Random initialisation of the clustering. Arguments `K` and `nruns` are unused
- `benchmarks.kmeans_lda` Cluster the matrix theta obtained by a topicmodels LDA with K topics
initializeBeta

**DTMtoSparse**

*convert a dtm from package tm to sparseMatrix from package Matrix without converting it to full matrix.*

**Description**

convert a dtm from package tm to sparseMatrix from package Matrix without converting it to full matrix.

**Usage**

```r
DTMtoSparse(dtm)
```

**Arguments**

- `dtm` a document-term-matrix from package 'tm'

**Value**

a sparse dgCMatrix from package Matrix

**initializeBeta**

*Beta initialization*

**Description**

Used in the `mmpca_clust()` function to initialize beta. It can be either "random" or "lda". Please note that the `mmpca_clust()` function also allow for a user given beta matrix. In this case, this function is not used.

**Usage**

```r
initializeBeta(dtm, init.beta, K, verbose = 0, control_lda_init = NULL)
```

**Arguments**

- `dtm` An object of class `DocumentTermMatrix`
- `init.beta` A string specifying the method, either
  - 'random': Initialization a la Blei et. al. with 1/V coefficient everywhere + a small uniform noise U[0, 1e-10] on every coefficients.
  - 'lda': Recommended. Uses the beta of LDA algorithm via a VEM algorithm, with an initialization of 5 repeats of the gibbs sampling algorithm with 1000 burning iterations and 1000 iterations.
- `K` The number of topics (dimension of the latent space).
The verbosity level. Only prints a message at function activation.

The control for LDA(). Only used when init.beta == 'lda' and initiallized to the default "LDA_VEMcontrol" of the TopicModelcontrol class.

Value

A KxV matrix with each row summing to 1.

Examples

```r
simu = simulate_BBC(N = 100, L = 100)
K = 4
beta = initializeBeta(simu$dtm.full, 'lda', K, verbose = 1)
```

**initialize_Y**  
*Clustering initialization*

**Description**

Perform a DocumentTermMatrix clustering via default routines or allow for user specified function

**Usage**

```r
initialize_Y(dtm, Q, K, init = "random")
```

**Arguments**

- `dtm`  
  An object of class `DocumentTermMatrix`

- `Q`  
  The number of clusters

- `K`  
  The dimension of the latent space. It is mandatory, for compatibility reasons but not always used (e.g. random do not use it).

- `init`  
  Either:
  - 'random': Random initialization.
  - 'kmeans_1da': A Q-kmeans on the latent space (theta matrix) of a K-topic LDA.
  - A user defined function which MUST take the following structure for compatibility `init <- function(dtm, Q, K, nruns, ...)`

**Details**

For more details see `benchmarks-functions`
**Value**

A vector of size equal to the number of row of dtm, containing a Q-clustering.

**Examples**

```r
simu = simulate_BBC(N = 100, L = 100)
Q = 6
K = 4
Y = initialize_Y(simu$dtm.full, Q, K, init = 'kmeans_lda')
```

**Description**

An S4 class representing a fitted mmpca model.

**Details**

The BB-CVEM method is the branch & bound greedy procedure proposed in the original paper of Jouvin et al. [https://arxiv.org/abs/1909.00721](https://arxiv.org/abs/1909.00721). The number of epochs in the n_epochs slot is actually the true number of pass minus 1 (unless max.epochs was reached). Indeed, the last pass before convergence does not change either the bound or the clustering, hence it is removed of the counter.

**Slots**

- **call** A `call` object specifying the call
- **method** The method used in the call
- **clustering** The final partition found by the algorithm
- **controls** An object of class `mmpcaClustcontrol` containing the controls used in the VEM algorithm on the aggregated DTM during the loop. The slots `controls@control_lda_init` where only use when `init.beta == 'lda'`.
- **K** An integer specifying the number of topics.
- **Q** An integer specifying he number of clusters.
- **N** An integer specifying the number of observations.
- **V** An integer specifying the number of variables.
- **beta** The (KxV) topic matrix.
- **gamma** A (QxK) matrix containing the variational parameters of the variational distribution of each $\theta_q$ in its rows.
- **lda_algo** An object of class "LDA" (cf. `TopicModel`) containing the results of the `LDA()` function applied to the aggregated DTM, with control `controls@control_lda_loop`
max. epochs The maximum number of pass through the whole dataset in the algorithm.

logLikelihoods A numeric vector containing the evolution of the variational bound every keep iteration.

keep An integer specifying the . Mostly useful for the plot function.

n_epochs The number of pass through the datasets before convergence. see details

llhood The final value of the variational lower bound.

Yinit The value of the initial partition.

icl The Integrated Classification Likelihood value.

Objects from the class

Object of class "mmpcaClust" are returned by mmpca_clust()

mmpcaClustcontrol-class

mmpcaClustcontrol

Description

An S4 class for mmpca_clust(). It is mainly a wrapper around the class TopicModelcontrol (specifically: LDA_VEMcontrol).

Slots

control_lda_init Object of class "LDA_VEMcontrol": specifies the controls of the VEM algorithm used for the initialization of beta.

control_lda_loop Object of class "LDA_VEMcontrol": specifies the controls for the VEM algorithm used after a swap in the branch & bound.

mmpca_clust

Greedy procedures for joint inference and clustering in MMPCA

Description

Perform clustering of count data using the MMPCA model.
Usage

```r
mmpca_clust(
  dtm,
  Q,
  K,
  model = NULL,
  Yinit = "random",
  method = "BBCVEM",
  init.beta = "lda",
  keep = 1L,
  max.epoehs = 10L,
  verbose = 1L,
  nruns = 1L,
  mc.cores = max(1, (detectCores() - 1))
)
```

Arguments

dtm an NxV DocumentTermMatrix with term-frequency weighting.

Q The number of clusters

K The number of topics (latent space dimension)

model A given model in which to take the controls for the VE-steps in the greedy procedure. If NULL, a model of class mmpcaClust is created with default controls (see mmpcaClustcontrol class for more details).

Yinit Parameter for the initialization of Y. It can be either:

- a string or a function specifying the initialization procedure. It should be one of ('random', 'kmeans_lda'). See benchmarks-functions functions for more details.
- A vector of length N with Q modalities, specifying the initialization clustering.

method The clustering algorithm to be used. Only "BBCVEM" is available : it corresponds to the branch and bound C-VEM of the original article.

init.beta Parameter for the initialization of the matrix beta. It can be either:

- a string specifying the initialization procedure. It should be one of ('random', 'lda'). See initializeBeta() for more details.
- A KxV matrix with each row summing to 1.

keep The evolution of the bound is tracked every keep iteration

max.epoehs Specifies the maximum number of pass allowed on the whole dataset.

verbose verbosity level

nruns number of runs of the algorithm (default to 1) : the run achieving the best evidence lower bound is selected.

mc.cores The number of CPUs to use when fitting in parallel the different models (only for non-Windows platforms). Default is the number of available cores minus 1.
Value

An object of class "mmpcaClust" containing the fitted model.

mmpca_clust_modelselect

Model selection for MMPCA

Description

A wrapper on mmpca_clust() to perform model selection with an Integrated Classification Likelihood (ICL) criterion.

Usage

mmpca_clust_modelselect(
  dtm, Qs, Ks, Yinit = "random", method = "BBCVEM", init.beta = "lda", keep = 1L, max.epochs = 10L, verbose = 1L, nruns = 5L, mc.cores = (detectCores() - 1)
)

Arguments

dtm an NxV DocumentTermMatrix with term-frequency weighting.
Qs The vector of clusters to be tested.
Ks The number of topics to be tested.
Yinit Parameter for the initialization of Y. It can be either:
  • a string or a function specifying the initialization procedure. It should be one of ("random", "kmeans_lda"). See benchmarks-functions functions for more details.
  • (Only when Qs is a singleton) A vector of length N with Q modalities, specifying the initialization clustering.
method The clustering algorithm to be used. Only "BBCVEM" is available: it corresponds to the branch and bound C-VEM of the original article.
init.beta Parameter for the initialization of the matrix beta. It can be either:
  • a string specifying the initialization procedure. It should be one of ("random", "lda"). See initializeBeta() for more details.
MoMPCA

- (Only when Ks is a singleton) A KxV matrix with each row summing to 1.

keep

The evolution of the bound is tracked every keep iteration.

max.epochs

Specifies the maximum number of pass allowed on the whole dataset.

verbose

verbosity level.

nrns

number of runs of the algorithm for each (K,Q) pair (default to 1): the run achieving the best evidence lower bound is selected.

mc.cores

The number of CPUs to use when fitting in parallel the different models. Default is the number of available cores minus 1.

Value

- An object of class "mmpcaClust" containing the best selected model.
- A matrix containing the value of the ICL for each pair (K,Q).

Examples

```r
## generate data with the BBCmsg
simu = simulate_BBC(N = 100, L = 250)
## Define a grid
Qs = 5:6
Ks = 3:4
## Run model selection with MoMPCA
res <- mmpca_clust_modelselect(simu$dtm.full, Qs = Qs, Ks = Ks,
                                Yinit = 'kmeans_lda',
                                init.beta = 'lda',
                                method = 'BBCVEM',
                                max.epochs = 7,
                                nruns = 2,
                                verbose = 1,
                                mc.cores = 2)
```

MoMPCA

MoMPCA: Greedy clustering of count data through a mixture of multinomial PCA

Description


Details

The main entry point is the `mmpca_clust()` function to perform the clustering.
Description
Use ggplot2 if available.

Usage

## S4 method for signature 'mmpcaClust,missing'
plot(x, type = "topics", ...)

Arguments

x an S4 object of class mmpcaClust
type Either:
  • 'topics' (default): Show the top topic words of topic matrix. See plot_topics documentation for more details.
  • 'bound': plot the lower bound evolution during the greedy procedure. See plot_bound documentation for more details.

... optional argument specifying the number of words to display and the entropy correction to apply when calling plot_topics().

Value

a plot

plot_bound Bound evolution plot

Description
Plot lower bound evolution

Usage

plot_bound(res)

Arguments

res An S4 object of class mmpcaClust

Value

a ggplot2 object if ggplot2 is available. Plot on the device otherwise.
plot_topics

Description
Plot topic matrix

Usage
```
plot_topics(res, s = 2, n_words = 10)
```

Arguments
- `res` An S4 object of class `mmpcaClust`
- `s` an entropy correction parameter for the topic matrix. It is applied to the beta matrix before sorting the words by highest probability. The greater, the more emphasis is put towards words contributing a lot to the entropy of a topic. Set s=1 to ignore.
- `n_words` the number of words to display per topic.

Value
a ggplot2 object

simulate_BBC

Description
This function simulate from the MMPCA model with an additional noise parameter epsilon. The number of cluster is Q=6 for K=4 topics. The parameter beta is taken to be the row normalized document-term matrix of 4 BBC messages contained in BBCmsg.

Usage
```
simulate_BBC(N, L, epsilon = 0, lambda = 1, theta_true = NULL)
```

Arguments
- `N` number of observations.
- `L` vector of length N containing the total count per observations. Duplicated if integer.
- `epsilon` The noise level in the latent space. Quantify how far the distribution is from `theta_true`
simulate_BBC

lambda A parameter quantifying the class proportion. lambda=1 means balanced cluster sizes, lower means that the last clusters are bigger, with an geometric decay in cluster size for the first ones.

theta_true The true parameter theta for the simulation. If NULL (default) then it is initialized to the default value of the experimental section of the paper.

Value

A list with names

- dtm.full: A DocumentTermMatrix object containing the simulated document-term matrix
- Ytruth: the simulated partition
- theta_true The parameter of the simulation

Examples

```r
simu <- simulate_BBC(N = 100, L = 200, epsilon = 0, lambda = 1)
dtm <- simu$dtm.full
Ytruth <- simu$Ytruth
```
Index

* datasets
  BBCmsg, 2

BBCmsg, 2
benchmark.kmeans_lda
  (benchmarks-functions), 3
benchmark.random
  (benchmarks-functions), 3
benchmarks-functions, 3
call, 6
DocumentTermMatrix, 4, 5, 8, 9, 13
DTMtoSparse, 4
initialize_Y, 5
initializeBeta, 4, 8, 9
kmeans, 3
LDA, 5, 6
mmpca_clust, 4, 7, 7, 9, 10
mmpca_clust_modelselect, 9
mmpcaClust, 3, 8–12
mmpcaClust-class, 6
mmpcaClustcontrol, 6, 8
mmpcaClustcontrol-class, 7
MoMPCA, 10
plot,mmpcaClust,missing-method, 11
plot_bound, 11, 11
plot_topics, 11, 12

simulate_BBC, 12

TopicModel, 6
TopicModelcontrol, 5, 7