Package ‘PRISMA’

February 28, 2017

Type Package
Title Protocol Inspection and State Machine Analysis
Version 0.2-6
Date 2017-02-27
Depends R (>= 2.10), Matrix, gplots, methods, ggplot2
Suggests tm (>= 0.6)
Author Tammo Krueger, Nicole Kraemer
Maintainer Tammo Krueger <tammokrueger@googlemail.com>
Description Loads and processes huge text corpora processed with the sally toolbox (<http://www.mlsec.org/sally/>). sally acts as a very fast preprocessor which splits the text files into tokens or n-grams. These output files can then be read with the PRISMA package which applies testing-based token selection and has some replicate-aware, highly tuned non-negative matrix factorization and principal component analysis implementation which allows the processing of very big data sets even on desktop machines.
License GPL (>= 2.0)
NeedsCompilation no
Repository CRAN
Date/Publication 2017-02-28 00:34:16

R topics documented:

PRISMA-package .......................................................... 2
asap ................................................................. 3
corpusToPrisma ....................................................... 3
estimateDimension ...................................................... 4
getDuplicateData ....................................................... 5
getMatrixFactorizationLabels ................................. 5
loadPrismaData ....................................................... 6
plot.prisma .......................................................... 7
plot.prismaDimension ............................................. 8
The PRISMA package is capable of loading and processing huge text corpora processed with the sally toolbox (http://www.mlsec.org/sally/). sally acts as a very fast preprocessor which splits the text files into tokens or n-grams. These output files can then be read with the PRISMA package which applies testing-based token selection and has some replicate-aware, highly tuned non-negative matrix factorization and principal component analysis implementation which allows the processing of very big data sets even on desktop machines.

Details

Package: PRISMA
Type: Package
Version: 0.2
Date: 2013-04-03
License: GPL (>=2.0)

Author(s)

Tammo Krueger, Nicole Kraemer
Maintainer: Tammo Krueger <tammokrueger@googlemail.com>

References


Examples

# please see the vinette for examples
asap  

The ASAP Data Set

Description

Toy data set to show the capabilities of the PRISMA package.

Usage

asap

Format

A prisma object.

Author(s)

Tammo Krueger <tammokrueger@googlemail.com>

References


corpusToPrisma  

Convert tm corpus to PRISMA

Description

Converts a tm corpus object to a PRISMA object.

Usage

corpusToPrisma(corpus, alpha = 0.05, skipFeatureCorrelation = FALSE)

Arguments

corpus a tm corpus
alpha significance level for the feature tests. If NULL, all features are kept.
skipFeatureCorrelation should the grouping of features based on correlation analysis be skipped.

Value

prismaData data object representing the tokenized documents as features x samples matrix.
estimateDimension

Author(s)
Tammo Krueger <tammokrueger@googlemail.com>

Examples
if (require("tm") && packageVersion("tm") >= '0.6') {
  data(thesis)
  thesis
  thesis = corpusToPrisma(thesis, NULL, TRUE)
  thesis
}

Description
Matrix factorization methods compress the original data matrix \( A \in \mathbb{R}^{f,N} \) with \( f \) features and \( N \) samples into two parts, namely \( A = BC \) with \( B \in \mathbb{R}^{f,k} \), \( C \in \mathbb{R}^{k,N} \). The function estimateDimension estimates \( k \) based on a noise model estimated from a scrambled version of the original data matrix.

Usage
estimateDimension(prismaData, alpha = 0.05, nScrambleSamples = NULL)

Arguments
- prismaData: A prismaData object loaded via loadPrismaData
- alpha: Error probability for confidence intervals
- nScrambleSamples: The number of scrambled samples that should be used to estimate the noise model. NULL means to use the complete data set.

Value
- estDim: prismaDimension object that can be printed and plotted.

Author(s)
Tammo Krueger <tammokrueger@googlemail.com>

References
getDuplicateData

Examples
# please see the vignette for examples

getDuplicateData Restores Data with Duplicates

Description
The loadPrismaData function triggers a feature selection and data combination methods which subsequently remove duplicate entries for efficient representation of the data. The getDuplicateData rebuilds the data matrix with explicit representation of all duplicate entries.

Usage
getDuplicateData(prismaData)

Arguments
prismaData prisma data loaded via loadPrismaData

Value
dataWithDuplicates Data matrix containing explicit copies of all duplicates.

Author(s)
Tammo Krueger <tammokrueger@googlemail.com>

Examples
data(asap)
dataWithDuplicates = getDuplicateData(asap)

getMatrixFactorizationLabels

Convert Coordinates of Matrix Factorization to Labels

Description
Given a matrix factorization object $A = BC$, this function returns for each document the index of the inner dimension which has the maximal coordinate. Thus, it converts the fuzzy clustering found in the columns of the $C$ matrix into a hard clustering by returning the position with the maximal coordinate value.
loadPrismaData

Usage
getMatrixFactorizationLabels(prismaMF)

Arguments
prismaMF a matrix factorization object.

Value
labels vector containing the label assignment for each document.

Author(s)
Tammo Krueger <tammokrueger@googlemail.com>

See Also
prismaNMF

Description
Loads files generated by the sally tool (see http://www.mlsec.org/sally/) and represents
the data as binary token/ngrams x documents matrix. After loading, statistical tests are applied to find
features which are not volatile nor constant. Co-occurring features are grouped to further compactify
the data. See system.file("extdata","sallyPreprocessing.py", package="PRISMA")
for a Python script which generates the corresponding .fsally file from a .sally file which reduce the
loading time via loadPrismaData considerably.

Usage
loadPrismaData(path, maxLines = -1, fastSally = TRUE,
alpha = 0.05, skipFeatureCorrelation=FALSE)

Arguments
path path of the data file without the .sally extension. loadPrisma loads path.sally or
path.fsally depending on the fastSally switch.
maxLines maximal number of lines to read from the data file. -1 means to read all lines.
fastSally should the fsally file be used, which drastically decreases loading time.
alpha significance level for the feature tests. If NULL, all features are kept.
skipFeatureCorrelation should the grouping of features based on correlation analysis be skipped.
Value

prismaData data object representing the tokenized documents as features x samples matrix.

Author(s)

Tammo Krueger <tammokrueger@googlemail.com>

References

See http://www.mlsec.org/sally/ for the sally utility.

Examples

# please see the vingette for examles
# please see system.file("extdata","asap.tar.gz", package="PRISMA") for
# an example sally output

data(asap)
print(asap)
plot(asap)
plot.prismaDimension  Generics For PRISMA Objects

Description

Print and plot generic for the PRISMA dimension objects.

Usage

```r
## S3 method for class 'prismaDimension'
print(x, ...)
## S3 method for class 'prismaDimension'
plot(x, ...)
```

Arguments

- `x` PRISMA dimension object generated via `estimateDimension`
- `...` not used

Author(s)

Tammo Krueger <tammokrueger@googlemail.com>

See Also

`estimateDimension`, `prismaHclust`, `prismaDuplicatePCA`, `prismaNMF`

Examples

```
# please see the vignette for examples
```

plot.prismaMF  Generics For PRISMA Objects

Description

Print and plot generic for the PRISMA matrix factorization objects.

Usage

```r
## S3 method for class 'prismaMF'
plot(x, nlines = NULL, baseIndex = NULL, sampleIndex = NULL,
     minValue = NULL, noRowClustering = FALSE, noColClustering = FALSE, type
     = c("base", "coordinates"), ...)
```
prismaDuplicatePCA

Arguments

- **x**: PRISMA matrix factorization object
- **nLines**: number of lines that should be plotted
- **baseIndex**: which bases should be plotted
- **sampleIndex**: which samples should be plotted
- **minValue**: cut-off value, i.e., every value smaller than minValue won’t be shown
- **noRowClustering**: don’t cluster the rows
- **noColClustering**: don’t cluster the columns
- **type**: show the base (**type = "base"**, i.e. the $B$ matrix) or show the coordinate (**type = "coordinates"**, i.e. the $C$ matrix).
- **...**: not used

Author(s)

Tammo Krueger &lt;tammokrueger@googlemail.com&gt;

See Also

- `estimatedimension`, `prismaHclust`, `prismaDuplicatePCA`, `prismaNMF`

Examples

```r
# please see the vignette for examples
```

Description

Efficient implementation of a replicate-aware principal component analysis (PCA).

Usage

`prismaDuplicatePCA(prismaData)`

Arguments

- **prismaData**: PRISMA data for which a PCA should be calculated

Value

- **prismaPCA**: Matrix factorization object $A = B C$, in which the factors are calculated by a replicate-aware PCA
Author(s)
Tammo Krueger <tammokrueger@googlemail.com>

Examples

# please see the vingette for examles

prismaHclust  Matrix Factorization Based on Hierarchical Clustering

Description
A matrix factorization $A = BC$ based on the results of hclust is constructed, which holds the mean feature values for each cluster in the matrix $B$ and the indication of the cluster in the matrix $C$ for each data point (i.e. each data point is represented by its assigned cluster center).

Usage
prismaHclust(prismaData, ncomp, method = "single")

Arguments
- **prismaData**: PRISMA data for which a clustering should be calculated.
- **ncomp**: the number of components that should be extracted.
- **method**: the method used for clustering.

Value
- **prismaHclust**: Matrix factorization object containing $B$ and $C$ resulting from the hierarchical clustering of the data.

Author(s)
Tammo Krueger <tammokrueger@googlemail.com>

See Also
hclust

Examples

# please see the vingette for examles
Matrix Factorization Based on Replicate-Aware NMF

Description

Matrix factorization $A = BC$ with strictly positive matrices $B, C$ which minimize the reconstruction error $\|A - BC\|$. This replicate-aware version of the non-negative matrix factorization (NMF) is based on the alternating least squares approach and exploits the replicate information to speed up the calculation.

Usage

prismanmf(prismaData, ncomp, time = 60, pca.init = TRUE, doNorm = TRUE, oldResult = NULL)

Arguments

- **prismaData**: PRISMA data for which a NMF should be calculated.
- **ncomp**: either an integer or prismaDimension object specifying the inner dimension of the matrix factorization.
- **time**: seconds after which the calculation should end.
- **pca.init**: should the $B$ matrix be initialized by a PCA.
- **doNorm**: should the $B$ matrix normalized (i.e. all columns have the Euclidean length of 1).
- **oldResult**: re-use results of a previous run, i.e. $B$ and $C$ are pre-initialized with the values of this previous matrix factorization object.

Value

- **prismanmf**: Matrix factorization object containing the $B$ and $C$ matrix.

Author(s)

Tammo Krueger <tammokrueger@googlemail.com>

References


Examples

# please see the vinette for examples
The Thesis Data Set

Description

The 15 sections of a thesis (see references) as a tm-corpus.

Usage

thesis

Format

A tm-corpus.

Author(s)

Tammo Krueger <tammokrueger@googlemail.com>

References

Index

*Topic **datasets**
  asap, 3
  thesis, 12
*Topic **package**
  PRISMA-package, 2

asap, 3

corpusToPrisma, 3

estimateDimension, 4, 7–9

getDuplicateData, 5, 5
getMatrixFactorizationLabels, 5

hclust, 10

loadPrismaData, 5, 6, 6, 7

plot.prisma, 7
plot.prismaDimension, 8
plot.prismaMF, 8
print.prisma(plot.prisma), 7
print.prismaDimension
  (plot.prismaDimension), 8
PRISMA (PRISMA-package), 2
PRISMA-package, 2
prismaDuplicatePCA, 7–9, 9
prismaHclust, 7–9, 10
prismaNMF, 6–9, 11

thesis, 12