Package: LRMF3 (via r-universe)

August 27, 2024

Title Low Rank Matrix Factorization S3 Objects

Version 0.1.0.9000

Description Provides S3 classes to represent low rank matrix decompositions.

License MIT + file LICENSE

URL https://github.com/RoheLab/LRMF3

BugReports https://github.com/RoheLab/LRMF3/issues

Depends Matrix, R (>= 3.1)

Imports glue

Suggests covr, testthat

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

Config/testthat/edition 3

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

RemoteUrl https://github.com/rohelab/lrmf3

RemoteRef HEAD

RemoteSha d0a325959e6157bcff35193a7a09b4ae708e1b33

Contents

as_fa_like *Coerce an object to a factor analysis like factorization*

Description

Coerce an object to a factor analysis like factorization

Usage

```
as_fa_like(x, ...)

## S3 method for class 'list'
as_fa_like(x, ...)
```

Arguments

x	Object to coerce
...	Ignored.

Value

Object as `svd_like()` object.

as_svd_like *Coerce an object to LRMF class*

Description

Coerce an object to LRMF class

Usage

```
as_svd_like(x, ...)

## S3 method for class 'list'
as_svd_like(x, ...)
```

Arguments

x	Object to coerce
...	Ignored.

Value

Object as `svd_like()` object.

dim_and_class *Description array size and type*

Description

Description array size and type

Usage

```
dim_and_class(x)
```

Arguments

x	Matrix or vector

Value

Description as character vector

fa_like *Create a Factor Analysis-like low rank matrix factorization object*

Description

A low rank matrix factorization of a matrix `X` is parameterized by `X ~= X %*% B %*% t(Y)`. The object is "factor analysis-like" because the middle matrix in the decomposition is arbitrary rather than diagonal.

Usage

```
fa_like(Z, B, Y, subclasses = NULL, ...)
```

Arguments

Z	A *matrix* of embeddings for each observation.
B	A mixing *matrix* describing how observation embeddings and topics interact. Does not have to be diagonal!
Y	A *matrix* describing the compositions of various topics or factors.
subclasses	A character vector of subclasses. Optional, defaults to `NULL`.
...	Optional additional items to pass to the constructor.

Examples

```
s <- svd(as.matrix(trees))

fa_like(s$u, diag(s$d), s$v)
```

ml100k *MovieLens 100K dataset*

Description

Standard benchmarking dataset for recommendation systems. 100k movie ratings on 1682 movies by 943 users. Each user has rated at least 20 movies.

Usage

```
ml100k
```

Format

An object of class `dgCMatrix` with 943 rows and 1682 columns.

Details

Stored as a `Matrix::dgCMatrix` object, which is a sparse matrix. Each row corresponds to a user and each column to a movie.

References

F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015), 19 pages. DOI=http://dx.doi.org/10.1145/2827872

https://grouplens.org/datasets/movielens/100k/

svd_like *Create a SVD-like low rank matrix factorization object*

Description

A low rank matrix factorization of a matrix `X` is parameterized by `X ~= u %*% diag(d) %*% t(v)`. The object is "svd-like" because the middle matrix in the decomposition must be strictly diagonal.

Usage

```
svd_like(u, d, v, subclasses = NULL, ...)
```

Arguments

`u`	A *matrix* "left singular-ish" vectors.
`d`	A *vector* of "singular-ish" values.
`v`	A *matrix* of "right singular-ish" vectors.
`subclasses`	A character vector of subclasses. Optional, defaults to `NULL`.
`...`	Optional additional items to pass to the constructor.

Examples

```
s <- svd(as.matrix(trees))

# using the constructor directly
svd_like(s$u, s$d, s$v)

# coercing svd-like lists
as_svd_like(s)
```

Index