

Package: nmfbin (via r-universe)

October 12, 2024

Title Non-Negative Matrix Factorization for Binary Data

Version 0.2.1

Description Factorize binary matrices into rank-k components using the logistic function in the updating process. See e.g. Tomé et al (2015) <[doi:10.1007/s11045-013-0240-9](https://doi.org/10.1007/s11045-013-0240-9)> .

License MIT + file LICENSE

Encoding UTF-8

Language en-GB

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

URL <https://michalovadek.github.io/nmfbin/>

BugReports <https://github.com/michalovadek/nmfbin/issues>

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Config/testthat/edition 3

Repository <https://michalovadek.r-universe.dev>

RemoteUrl <https://github.com/michalovadek/nmfbin>

RemoteRef HEAD

RemoteSha 8689b530e7eac7dc11e88ed6e2c49ad6dbe581a6

Contents

nmfbin	2
Index	4

nmfbin

*Logistic Non-negative Matrix Factorization***Description**

This function performs Logistic Non-negative Matrix Factorization (NMF) on a binary matrix.

Usage

```
nmfbin(
  X,
  k,
  optimizer = "mur",
  init = "nndsvd",
  max_iter = 1000,
  tol = 1e-06,
  learning_rate = 0.001,
  verbose = FALSE,
  loss_fun = "logloss",
  loss_normalize = TRUE,
  epsilon = 1e-10
)
```

Arguments

<code>X</code>	A binary matrix (m x n) to be factorized.
<code>k</code>	The number of factors (components, topics).
<code>optimizer</code>	Type of updating algorithm. <code>mur</code> for NMF multiplicative update rules, gradient for gradient descent, <code>sgd</code> for stochastic gradient descent.
<code>init</code>	Method for initializing the factorization. By default Nonnegative Double Singular Value Decomposition with average densification.
<code>max_iter</code>	Maximum number of iterations for optimization.
<code>tol</code>	Convergence tolerance. The optimization stops when the change in loss is less than this value.
<code>learning_rate</code>	Learning rate (step size) for the gradient descent optimization.
<code>verbose</code>	Print convergence if TRUE.
<code>loss_fun</code>	Choice of loss function: <code>logloss</code> (negative log-likelihood, also known as binary cross-entropy) or <code>mse</code> (mean squared error).
<code>loss_normalize</code>	Normalize loss by matrix dimensions if TRUE.
<code>epsilon</code>	Constant to avoid $\log(0)$.

Value

A list containing:

- *W*: The basis matrix ($m \times k$). The document-topic matrix in topic modelling.
- *H*: The coefficient matrix ($k \times n$). Contribution of features to factors (topics).
- *c*: The global threshold. A constant.
- *convergence*: Divergence (loss) from X at every *iter* until *tol* or *max_iter* is reached.

Examples

```
## Not run:  
# Generate a binary matrix  
m <- 100  
n <- 50  
X <- matrix(sample(c(0, 1), m * n, replace = TRUE), m, n)  
  
# Set the number of factors  
k <- 4  
  
# Factorize the matrix with default settings  
result <- nmfbin(X, k)  
  
## End(Not run)
```

Index

nmfbin, [2](#)