Package ‘vrnmf’

August 20, 2021

Title Volume-Regularized Structured Matrix Factorization

Version 1.0.0

Description
Implements a set of routines to perform structured matrix factorization with minimum volume constraints. The NMF procedure decomposes a matrix \( X \) into a product \( C \times D \). Given conditions such that the matrix \( C \) is non-negative and has sufficiently spread columns, then volume minimization of a matrix \( D \) delivers a correct and unique, up to a scale and permutation, solution \((C, D)\). This package provides both an implementation of volume-regularized NMF and "anchor-free" NMF, whereby the standard NMF problem is reformulated in the covariance domain. This algorithm was applied in Vladimir B. Seplyarskiy Ruslan A. Soldatov, et al. "Population sequencing data reveal a compendium of mutational processes in the human germ line". Science, 12 Aug 2021. <doi:10.1126/science.aba7408>. This package interacts with data available through the 'simulatedNMF' package, which is available in a 'drat' repository. To access this data package, see the instructions at <https://github.com/kharchenkolab/vrnmf>. The size of the 'simulatedNMF' package is approximately 8 MB.

License GPL-3

Encoding UTF-8

Depends R (>= 3.5.1)

Imports graphics, ica (>= 1.0), lpSolveAPI (>= 5.5.2.0), nls,
parallel (>= 3.5.1), quadprog (>= 1.5), stats

Suggests knitr (>= 1.28), rmarkdown (>= 2.1), testthat

RoxygenNote 7.1.1

URL https://github.com/kharchenkolab/vrnmf

BugReports https://github.com/kharchenkolab/vrnmf/issues

Author Ruslan Soldatov [aut],
Peter Kharchenko [aut],
Evan Biederstedt [cre, aut]

Maintainer Evan Biederstedt <evan.biederstedt@gmail.com>

NeedsCompilation no

Repository CRAN

Date/Publication 2021-08-20 12:40:02 UTC
AnchorFree

R topics documented:

AnchorFree . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2
factor_intensities . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 3
infer_intensities . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
projection_onto_simplex . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 5
sim_factors . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 6
volnmf_det . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 7
volnmf_estimate . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 8
volnmf_logdet . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
volnmf_main . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 11
volnmf_procrustes . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 13
volnmf_simplex_col . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 14
volnmf_simplex_row . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15
vol_preprocess . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 15

Index 17

**AnchorFree**

Non-negative tri-factorization of co-occurrence matrix using minimum volume approach.

Description

AnchorFree method tri-factorizes (co-occurrence) matrix in a product $P C \ast E \ast t(C)$ of non-negative matrices $C$ and $E$ such that matrix $E$ has minimum volume and columns of matrix $C$ equal to 1.

Usage

```
AnchorFree(
  vol,
  n.comp = 3,
  init = NULL,
  init.type = "diag",
  n.iter = 30,
  err.cut = 1e-30,
  verbose = FALSE
)
```

Arguments

- **vol**: An output object of `vol_preprocess()`. The method factorizes co-occurrence matrix `vol$P`.
- **n.comp**: An integer. Number of components to extract (by default 3). Defines number of columns in matrix $C$. (default=3)
- **init**: A numeric matrix. Initial matrix $M$. (default=3)
### Details

Implementation closely follows (Fu X et al., IEEE Trans Pattern Anal Mach Intell., 2019).

### Value

List of objects:

- $C, E$: Factorization matrices.
- $P_{est}$: Estimate of $\text{vol}$ co-occurrence matrix $P_{est} = C \ast E \ast t(C)$.
- $M, \det M$: Auxiliary matrix $M$ and its determinant.
- $\text{init.type}$: Type of initialization of matrix $M$ that was used.

### Examples

```r
small_example <- sim_factors(5, 5, 5)
vol <- vol_preprocess(t(small_example$X))
vol.anchor <- AnchorFree(vol)
```

---

**factor_intensities**

Infer a matrix of non-negative intensities in NMF with offset/nmf-offset.

**Description**

`factor_intensities` estimates a non-negative matrix $D$ that optimizes the objective function $F = \| X - C \ast D - \text{offset} \|^2$, where offset is either column-specific offset or a "1-rank nmf term": product of row vector and column vector.
Usage

factor_intensities(
  C,
  X,
  fit.nmf = TRUE,
  fit.factor = FALSE,
  qp.exact = FALSE,
  n.iter = 200,
  qp.iter = 10,
  rel.error.cutoff = 1e-05,
  extrapolate = TRUE,
  extrapolate.const = TRUE,
  extrapolate.convex = FALSE,
  q.factor = 1,
  verbose = TRUE,
  n.cores = 1
)

Arguments

C Numeric matrices.
X Numeric matrices.
fit.nmf A boolean. Fit both intensities and spectrum of the offset residuals.
fit.factor A boolean. Fit only spectrum of the offset residuals (keep intensities constant across samples).
qp.exact A boolean. Estimate intensities using exact quadratic programming (qp.exact = TRUE) or inexact QP via gradient decent with extrapolation (qp.exact = FALSE).
n.iter An integer. Number of iterations.
qp.iter = 1e+1 An integer. Number of iterations of inexact QP.
rel.error.cutoff A numeric. Relative error cutoff between iterations to stop iterations.
extrapolate A boolean. Use Nesterov-like extrapolation at each iteration.
extrapolate.const A boolean. Use extrapolation scheme that adds a constant extrapolation q.factor (described below) at each iteration.
extrapolate.convex A boolean. Use Nesterov extrapolation scheme.
q.factor A numeric. Specification of a a constant extrapolation factor used in case of extrapolate.const = T.
verbose A boolean. Print per-iteration information (by default TRUE).
n.cores An integer. Number of cores to use.

Value

Fitted matrix D.
infer_intensities

Infer a matrix of non-negative intensities in NMF

Description

infer_intensities estimates a non-negative matrix $D$ that optimizes the objective function $F = ||X - C \ast D||^2$ using per-row quadratic programming.

Usage

infer_intensities(C, X, esign = "pos", n.cores = 1)

Arguments

C Numeric matrices.
X Numeric matrices.
esign A character. Keep elements of matrix $D$ non-negative ("pos") or not ("all"). (default="pos")
n.cores An integer. Number of cores to use. (default=1)

Value

Fitted matrix $D$.

projection_onto_simplex

Project vector onto a probabilistic simplex.

Description

projection_onto_simplex projects a vector $\text{unproj}$ onto a probabilistic simplex of sum $\text{bound}$.

Usage

projection_onto_simplex(unproj, bound)

Arguments

unproj A numeric vector. An unprojected vector
bound A numeric. Sum of projected vector elements.

Value

A projected vector.
**Description**

`sim_factors` simulates non-negative factorization matrices `C` and `D` under a variety of conditions to explore factorization `X = C * D + noise`.

**Usage**

```r
sim_factors(
  m, n, r,
  simplex = "col",
  distr = "unif",
  frac.zeros = 0.4,
  condition = FALSE,
  noise = 0
)
```

**Arguments**

- `m` Integers. Size of matrices. Matrix `C` has a size of `m*r` and matrix `D` has a size of `r*n`.
- `n` Integers. Size of matrices. Matrix `C` has a size of `m*r` and matrix `D` has a size of `r*n`.
- `r` Integers. Size of matrices. Matrix `C` has a size of `m*r` and matrix `D` has a size of `r*n`.
- `simplex` A character. Either columns ("col") or rows ("row") of matrix `C` are projected onto unit simplex. (default="col")
- `distr` A character. Distribution to simulate matrix entries: "unif" for uniform and "exp" for exponential distributions. (default="unif")
- `frac.zeros` A numeric. Fraction of zeros in matrix `C`. It promotes sufficient scattering of matrix column/row vectors. (default=0.4)
- `condition` A boolean. Generate more well-conditioned matrix `R`. (default=FALSE)
- `noise` A numeric. Standard deviation of gaussian noise to add. (default=0e-4)

**Value**

List of simulated matrices:

- `X.noise`, `X` - noisy and original matrix `X` to decompose.
- `C`, `D` - factorization matrices.
**volnmf_det**  
*Update volume-regularized matrix R using det volume approximation*

**Description**

volnmf_det finds matrix R that minimizes objective $||X-C\cdot R||^2 + w.\text{vol}\cdot \det(R)$

**Usage**

```r
volnmf_det(
  C,
  X,
  R,
  posit = FALSE,
  w.vol = 0.1,
  eigen.cut = 1e-16,
  err.cut = 0.001,
  n.iter = 1000
)
```

**Arguments**

- **C**

- **X**

- **R**

- **posit**
  A boolean. Set up (TRUE) or not (FALSE) non-negative constraints on matrix R. (default=TRUE)

- **w.vol**
  A numeric. Volume (det) weight in objective function. (default=0.1)

- **eigen.cut**
  A numeric. Threshold on eigenvalue of SVD eigenvectors. (default=1e-16)

- **err.cut**
  A numeric. Stop algorithm if relative erro in R between iteration is less than err.cut. (default=1e-3)

- **n.iter**
  An integer. Number of iterations. (default=1e+3)

**Value**

An updated matrix R.
volnmf_estimate provides alternating optimization of volume-regularized factorization of a matrix $B$ using the following objective function: $F = \| B Q - C R \|^2 + w_{vol} \cdot volume(R)$. Matrix $C$ is required to be non-negative and having either column or row vectors on the simplex. Matrix $R$ can optionally have non-negativity constraint. Matrix $Q$ can optionally be identity matrix or any unitary.

Usage

```r
code

```

Arguments

- **B**: A numeric matrix. A matrix to factorize (by default NULL). If not given than matrix $B$ is taken to be a square root decomposition of $P = B \cdot t(B)$.
- **C**: Numeric matrices. Initial matrices for optimization.
volnmf_estimate

R Numeric matrices. Initial matrices for optimization.
Q Numeric matrices. Initial matrices for optimization.
domain A character. Optimize unitary rotation matrix Q ("covariance") or keep it as identity matrix (as in standard NMF). By default "covariance".
volf A character. Function that approximate volume. Can have values of "logdet" or "det" (by default "logdet").
R.majorate A boolean. Majorate logdet each iteration of volnmf_logdet() (by default FALSE).
wvol A numeric. A weight of volume-regularized term volume(R).
delta A numeric. Logdet regularization term log(det(R) + delta) (by default 1e-8).
n.iter An integer. Logdet regularization term log(det(R) + delta) (by default 1e-8).
err.cut A numeric. Relative error in determinant between iterations to stop algorithm (by default 1e-8).
vol.iter An integer. Number of iterations to update volume-regularized matrix R at each alternating step.
c.iter An integer. Number of iterations to update simplex matrix C at each alternating step.
extrapolate A numeric. Do Nesterov extrapolation inside blocks of R and C optimization (by default TRUE).
accelerate A numeric. Do acceleration each update after R and C blocks estimated via Nesterov-like extrapolation.
acc.C A numeric. Acceleration parameter of matrix C.
acc.R A numeric. Acceleration parameter of matrix R.
C.constraint A character. Constraint either sum of columns ("col") or sum of rows ("row") to be equal to C.bound (By default "col").
C.bound A numeric. A simplex constraint on matrix C vectors.
R.constraint A character. Set up non-negativity ("pos") constraint on elements of R (by default "pos", alternative "no").
verbose A boolean. Print per-iteration information (by default FALSE)
record A numeric. Record parameters every 'record' iterations (by default NULL).
Canchor A matrix. A matrix of anchor components (unused currently). (default=NULL)
Ctrue A matrix. Correct matrix C if known. Useful for benchmark.
mutation.run A boolean. Assess goodness of solution using reflection test if mutation.run=TRUE (applicable only to analysis of mutation patterns). (default=FALSE)

Value

List of objects:
C,R,Q, E Factorization matrices.
iter, err Number of iterations and relative per-iteration error err in matrix C.
info.record a list of objects that record and store state of matrices each record iterations.
**volnmf_logdet** 

Update volume-regularized matrix \( \mathbf{R} \) using logdet volume approximation.

**Description**

volnmf_logdet finds matrix \( \mathbf{R} \) that minimizes objective \( \| \mathbf{X} - \mathbf{C} \mathbf{R} \|^2 + w.\text{vol}\log(\det(\mathbf{R}) + \delta) \).

**Usage**

```r
volnmf_logdet(
  \( \mathbf{C} \),
  \( \mathbf{X} \),
  \( \mathbf{R} \),
  \( \text{R.constraint} = "\text{pos}" \),
  \( \text{majorate} = \text{FALSE} \),
  \( \text{extrapolate} = \text{TRUE} \),
  \( qmax = 100 \),
  \( w.\text{vol} = 0.1 \),
  \( \delta = 1 \),
  \( \text{err.cut} = 0.001 \),
  \( n.\text{iter} = 1000 \)
)
```

**Arguments**

- **\( \mathbf{C} \)**: Numeric Matrices. Matrices involved in objective function. Matrix \( \mathbf{R} \) serves as initialization.
- **\( \mathbf{X} \)**: Numeric Matrices. Matrices involved in objective function. Matrix \( \mathbf{R} \) serves as initialization.
- **\( \mathbf{R} \)**: Numeric Matrices. Matrices involved in objective function. Matrix \( \mathbf{R} \) serves as initialization.
- **\( \text{R.constraint} \)**: A character. Set up ('pos') or not ('no') non-negative constraints on matrix \( \mathbf{R} \) (by default 'pos').
- **\( \text{majorate} \)**: A boolean. Majorate logdet each iteration (by default FALSE).
- **\( \text{extrapolate} \)**: A boolean. Use Nesterov acceleration (by default FALSE, currently is not supported).
- **\( qmax \)**: A numeric. Maximum asymptotic \( (1 - 1/qmax) \) of extrapolation step.
- **\( w.\text{vol} \)**: A numeric. Volume (logdet) weight in objective function.
- **\( \delta \)**: A numeric. Determinant pseudocount in objective function.
- **\( \text{err.cut} \)**: A numeric. Stop algorithm if relative erro in \( \mathbf{R} \) between iteration is less than \( \text{err.cut} \).
- **\( n.\text{iter} \)**: An integer. Number of iterations.
Value

An updated matrix R.

---

**Volume-regularized NMF**

**Description**

`volnmf_main` enables volume-regularized factorization of a matrix B using the following objective function: 

\[ F = \|B \ast Q - C \ast R\|_2^2 + w.\text{vol} \ast \text{volume}(R) \]

Matrix C is required to be non-negative and having either column or row vectors on the simplex. Matrix R can optionally have non-negativity constraint. Matrix Q can optionally be identity matrix or any unitary. The latter option is used to decompose co-occurrence matrix `vol_P`.

**Usage**

```r
volnmf_main(
  vol,
  B = NULL,
  volnmf = NULL,
  n.comp = 3,
  n.reduce = n.comp,
  do.nmf = TRUE,
  iter.nmf = 100,
  seed = NULL,
  domain = "covariance",
  volf = "logdet",
  wvol = NULL,
  delta = 1e-08,
  n.iter = 500,
  err.cut = 1e-16,
  vol.iter = 20,
  c.iter = 20,
  extrapolate = TRUE,
  accelerate = FALSE,
  acc.C = 4/5,
  acc.R = 3/4,
  C.constraint = "col",
  C.bound = 1,
  R.constraint = "pos",
  R.majorate = FALSE,
  C.init = NULL,
  R.init = NULL,
  Q.init = NULL,
  anchor = NULL,
  Ctrue = NULL,
```
verbose = TRUE,
record = 100,
verbose.nmf = FALSE,
record.nmf = NULL,
mutation.run = FALSE
)

Arguments

vol An output object of vol_preprocess().
B A numeric matrix. A matrix to factorize (by default NULL). If not given than matrix B is taken to be a square root decomposition of \( P = B \times t(B) \).
volnmf An output object of volnmf.main. An option is useful to re-estimate solution using different parameters (by default NULL).
n.comp An integer. Number of components to extract (by default 3). Defines number of columns in matrix \( C \).
n.reduce An integer. Dimensional reduction of matrix B (number of columns) if taken as a square root decomposition of volP (by default equal to n.comp).
do.nmf A boolean. Estimate standard solution with w.vol=0 as initialization before applying volume regularization (by default TRUE).
iternmf An integer. Number of iterations to get solution with w.vol=0 if the former requested (by default 1,000).
seed An integer. Fix seed.
domain A character. Optimize unitary rotation matrix \( Q \) ("covariance") or keep it as identity matrix (as in standard NMF). By default "covariance".
volf A character. Function that approximate volume. Can have values of "logdet" or "det" (by default "logdet").
wvol A numeric. A weight of volume-regularized term \( \text{volume}(R) \).
delta A numeric. Logdet regularization term \( \log(\text{det}(R) + \text{delta}) \) (by default 1e-8).
n.iter An integer. Number of iterations (by default 1,000).
err.cut A numeric. Relative error in determinant between iterations to stop algorithm (by default 1e-8).
vol.iter An integer. Number of iterations to update volume-regularized matrix \( R \) at each alternating step.
c.iter An integer. Number of iterations to update simplex matrix \( C \) at each alternating step.
extrapolate A numeric. Do Nesterov extrapolation inside blocks of \( R \) and \( C \) optimization (by default TRUE).
accelerate A numeric. Do acceleration each update after \( R \) and \( C \) blocks estimated via Nesterov-like extrapolation.
acc.C A numeric. Acceleration parameter of matrix \( C \).
acc.R A numeric. Acceleration parameter of matrix \( R \).
volnmf_procrustes

C.constraint A character. Constraint either sum of columns ("col") or sum of rows ("row") to be equal to C.bound (By default "col").
C.bound
A numeric. A simplex constraint on matrix C vectors.
R.constraint A character. Set up non-negativity ("pos") constraint on elements of R (by default "pos", alternative "no").
R.majorate A boolean. Majorate logdet each iteration of volnmf_logdet() (by default FALSE).
anchor An output object of AnchorFree(). Object is used optionally to initialize matrices (by default NULL).
Ctrue A matrix. Correct matrix C if known. Useful for benchmark.
verbose A boolean. Print per-iteration information (by default FALSE).
record A numeric. Record parameters every 'record' iterations (by default NULL).
verbose.nmf A boolean. Print per-iteration information for standard NMF (by default FALSE).
record.nmf A numeric. Record parameters every 'record' iterations for standard NMF (by default NULL).
mutation.run A boolean. Assess goodness of solution using reflection test if mutation.run=TRUE (applicable only to analysis of mutation patterns).

Value

List of objects:
C,R,Q Factorization matrices.
C.init,R.init,Q.init Initialization matrices for volume-regularized optimization.
C.rand,R.rand,Q.rand Random initialization matrices for NMF optimization (w.vol=0).
rec a list of objects that record and store state of matrices each record iterations.

---

volnmf_procrustes  Procrustes algorithm estimates orthonormal transformation between two matrices.

Description

volnmf_procrustes finds orthonormal matrix Q that minimizes objective ||A-B*Q||^2

Usage

volnmf_procrustes(A, B)
Arguments

A | Numeric Matrices. Orthonormal transformation convert matrix B in matrix A.
B | Numeric Matrices. Orthonormal transformation convert matrix B in matrix A.

Value

An optimal orthonormal transformation matrix Q.

Description

volnmf_simplex_col finds non-negative matrix C that minimizes the objective $||X-C*\mathbf{R}||^2$ under constraints that columns of C equal to 1 using local approximation with extrapolation.

Usage

```r
volnmf_simplex_col(
  X,
  R,
  C.prev = NULL,
  bound = 1,
  extrapolate = TRUE,
  err.cut = 1e-10,
  n.iter = 10000,
  qmax = 100
)
```

Arguments

X | Numeric Matrices. Matrices involved in the objective function.
R | Numeric Matrices. Matrices involved in the objective function.
C.prev | Numeric Matrices. Matrices involved in the objective function. Matrix C.prev serves as initialization. (default=NULL)
bound | A numeric. Equality constraint on columns of matrix C. (default=1)
extrapolate | A boolean. Use extrapolation after local approximation. (default=TRUE)
err.cut | A numeric. Stop iterations if relative error between iterations is less than err.cut (parameter is not active now). (default=1e-10)
n.iter | An integer. Number of iterations. (default=1000)
qmax | A numeric. Maximum asymptotic (1 - 1/qmax) of extrapolation step.

Value

An updated matrix C.
volnmf_simplex_row

Update of a matrix in NMF with equality constraints on rows.

Description

volnmf_simplex_row finds non-negative matrix $C$ that minimizes the objective $\|X-C*R\|^2$ under constraints that rows of $C$ equal to 1 using per-row quadratic programming.

Usage

```r
volnmf_simplex_row(X, R, C.prev = NULL, meq = 1)
```

Arguments

- **X**: Numeric Matrices. Matrices involved in the objective function.
- **R**: Numeric Matrices. Matrices involved in the objective function.
- **C.prev**: Numeric Matrices. Matrices involved in the objective function. Matrix $C.prev$ serves as initialization. (default=NULL)
- **meq**: An integer 0 or 1. Require equality (meq=1) or inequality (meq=0) constraint on rows (by default 1).

Value

An updated matrix $C$.

vol_preprocess

Preprocess the data for downstream volume analysis.

Description

vol_preprocess Routine normalizes the data (as requested), estimates covariance and SVD decomposition.

Usage

```r
vol_preprocess(X, col.norm = "sd", row.norm = NULL, pfactor = NULL)
```

Arguments

- **X**: A numeric matrix. Covariance is estimated for column vectors of $X$.
- **col.norm**: A character. Specifies column normalization strategy (by default "sd"). NULL to avoid normalization.
- **row.norm**: A character. Specifies row normalization strategy (by default NULL).
- **pfactor**: A numeric A factor to normalize co-occurence matrix (by default NULL). Row normalization follows column normalization. NULL to avoid normalization.
Value

A list of objects that include normalized matrix \( X \).process, row and column normalization factors \( \text{row.factors} \) and \( \text{col.factors} \), covariance matrix \( P_0 \), covariance matrix \( P \) normalized to maximum value \( \text{pfactor} \), orthonormal basis \( U \) and vector of eigenvalues \( \text{eigens} \).

Examples

```r
small_example <- sim_factors(5, 5, 5)
vol <- vol_preprocess(t(small_example$X))
```
Index

AnchorFree, 2
factor_intensities, 3
infer_intensities, 5
projection_onto_simplex, 5
sim_factors, 6
vol_preprocess, 15
volnmf_det, 7
volnmf_estimate, 8
volnmf_logdet, 10
volnmf_main, 11
volnmf_procrustes, 13
volnmf_simplex_col, 14
volnmf_simplex_row, 15