Package 'SPEV'

August 17, 2023

Title Unsmoothed and Smoothed Penalized PCA using Nesterov Smoothing

Version 1.0.0

Maintainer Rebecca Hurwitz <rebeccahurwitz@hsph.harvard.edu>

Description We provide functionality to implement penalized PCA with an option to smooth the objective function using Nesterov smoothing. Two functions are available to compute a user-specified number of eigenvectors. The function unsmoothed_penalized_EV() computes a penalized PCA without smoothing and has three parameters (the input matrix, the Lasso penalty, and the number of desired eigenvectors). The function smoothed_penalized_EV() computes a smoothed penalized PCA using the same parameters and additionally requires the specification of a smoothing parameter. Both functions return a matrix having the desired eigenvectors as columns.

License GPL (>= 2)

Encoding UTF-8

RoxygenNote 7.2.3

Imports stats

NeedsCompilation no

Author Rebecca Hurwitz [aut, cre],
Georg Hahn [ctb]

Repository CRAN

Date/Publication 2023-08-17 14:42:33 UTC

R topics documented:

smoothed_penalized_EV *smoothed_penalized_EV*

Description

This function takes a matrix (m), a lambda value (lambda), the number of desired eigenvectors (k), and a mu value (mu) as input. It then computes eigenvectors 1 to k, penalized by the supplied lambda and smoothed by the Nesterov smoothing function.

Usage

```
smoothed_penalized_EV(m, lambda, k, mu)
```

Arguments

m	A matrix generated from a large dataset.
lambda	A numeric vector of lambda values to use for the penalty.
k	The number of eigenvectors we consider in the analysis.
mu	A number assigned to mu; we are typically using 0.1.

Value

Returns smoothed eigenvectors 1 to k for the specified lambda value.

Examples

```
# Generate a small matrix for testing
m <- matrix(rnorm(100), nrow = 10)
# Call function (using matrix, lambda, mu, and k)
smoothed_penalized_EV(
  m = m,
  lambda = 1,
  k = 2,
  mu = 0.1
)
```

unsmoothed_penalized_EV *unsmoothed_penalized_EV*

Description

This function takes a matrix (m), a lambda value (lambda), and the number of desired eigenvectors (k) as input. It then computes eigenvectors 1 to k, penalized by the supplied lambda.

Usage

```
unsmoothed_penalized_EV(m, lambda, k)
```

Arguments

m	A matrix generated from a large dataset.
lambda	A numeric vector of lambda values to use for the penalty.
k	The number of eigenvectors we consider in the analysis.

Value

Returns eigenvectors 1 to k for the specified lambda value.

Examples

```
# Generate a small matrix for testing
m <- matrix(rnorm(100), nrow = 10)
# Call function (using matrix, lambda, and k)
unsmoothed_penalized_EV(
  m = m,
  lambda = 1,
  k = 2
)
```

Index