Package 'EfficientMaxEigenpair'

December 7, 2016

Type Package

Title Efficient Initials for Computing the Maximal Eigenpair

Version 0.1.0

Date 2016-12-03

Author Mu-Fa Chen <mfchen@bnu.edu.cn>

Maintainer Xiao-Jun Mao <maoxj.ki@gmail.com>

Description An implementation for using efficient initials to compute the maximal eigenpair in R. It provides two algorithms to find the efficient initials under two cases: the tridiagonal matrix case and the general matrix case. Besides, it also provides algorithms for the next to the maximal eigenpair under these two cases.

License MIT + file LICENSE

URL http://github.com/mxjki/EfficientMaxEigenpair

BugReports http://github.com/mxjki/EfficientMaxEigenpair/issues

Depends R (>= 3.3.2), stats

Encoding UTF-8

RoxygenNote 5.0.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2016-12-07 13:46:29

R topics documented:

eff.ini.maxeig.general 2
eff.ini.maxeig.tri 3
eff.ini.seceig.general 4
eff.ini.seceig.tri 5

EfficientMaxEigenpair 6
ray.quot 6
tridiag 7

Index **8**

`eff.ini.maxeig.general`

General matrix maximal eigenpair

Description

Calculate the maximal eigenpair for the general matrix.

Usage

```
eff.ini.maxeig.general(A, v0_tilde = NULL, z0 = NULL, z0numeric,
  improved = F, xi = 1, digit.thresh = 6)
```

Arguments

`A`	The input general matrix.
`v0_tilde`	The unnormalized initial vector $\tilde{v0}$.
`z0`	The type of initial z_0 used to calculate the approximation of $\rho(Q)$. There are three types: 'fixed', 'Auto' and 'numeric' corresponding to three choices of z_0 in paper.
`z0numeric`	The numerical value assigned to initial z_0 as an approximation of $\rho(Q)$ when z_0='numeric'.
`improved`	With improved=T, the improved algorithm to calculate initial approximating eigenpairs is used.
`xi`	The coefficient used to form the convex combination of δ_1^{-1} and $(v_0, -Q * v_0)_\mu$, it should between 0 and 1.
`digit.thresh`	The precise level of output results.

Value

A list of eigenpair object are returned, with components z and v.

`z`	The approximating sequence of the maximal eigenvalue.
`v`	The approximating sequence of the corresponding eigenvector.

See Also

`eff.ini.maxeig.tri` for the tridiagonal matrix maximal eigenpair.

Examples

```
A = matrix(c(1, 1, 3, 2, 2, 2, 3, 1, 1), 3, 3)
eff.ini.maxeig.general(A, v0_tilde = rep(1, dim(A)[1]), z0 = 'fixed')

A = matrix(c(1, 1, 3, 2, 2, 2, 3, 1, 1), 3, 3)
eff.ini.maxeig.general(A, v0_tilde = rep(1, dim(A)[1]), z0 = 'Auto')

##Symmetrizing A converge to second largest eigenvalue
A = matrix(c(1, 3, 9, 5, 2, 14, 10, 6, 0, 11, 11, 7, 0, 0, 1, 8), 4, 4)
S = (t(A) + A)/2
N = dim(S)[1]
a = diag(S[-1, -N])
b = diag(S[-N, -1])
c = rep(NA, N)
c[1] = -diag(S)[1] - b[1]
c[2:(N - 1)] = -diag(S)[2:(N - 1)] - b[2:(N - 1)] - a[1:(N - 2)]
c[N] = -diag(S)[N] - a[N - 1]

z0ini = eff.ini.maxeig.tri(a, b, c, xi = 7/8, improved = TRUE)$z[1]
eff.ini.maxeig.general(A, v0_tilde = rep(1, dim(A)[1]), z0 = 'numeric',
z0numeric = 28 - z0ini)
```

`eff.ini.maxeig.tri` *Tridiagonal matrix maximal eigenpair*

Description

Calculate the maximal eigenpair for the tridiagonal matrix.

Usage

```
eff.ini.maxeig.tri(a, b, c, xi = 1, improved = F, digit.thresh = 6)
```

Arguments

a	The lower diagonal vector.
b	The upper diagonal vector.
c	The shifted main diagonal vector. The corresponding unshift diagonal vector is -c(b[1] + c[1], a[1:N - 1] + b[2:N] + c[2:N], a[N] + c[N + 1]) where N+1 is the dimension of matrix.
xi	The coefficient used to form the convex combination of δ_1^{-1} and $(v_0, -Q * v_0)_\mu$, it should between 0 and 1.
improved	With improved=T, the improved algorithm to calculate initial approximating eigenpairs is used.
digit.thresh	The precise level of output results.

Value

A list of eigenpair object are returned, with components z and v.

z	The approximating sequence of the maximal eigenvalue.
v	The approximating sequence of the corresponding eigenvector.

See Also

eff.ini.maxeig.general for the general matrix maximal eigenpair.

Examples

```
a = c(1:7)^2
b = c(1:7)^2
c = rep(0, length(a) + 1)
c[length(a) + 1] = 8^2
eff.ini.maxeig.tri(a, b, c, xi = 1)
```

---

eff.ini.seceig.general

General conservative matrix maximal eigenpair

Description

Calculate the next to maximal eigenpair for the general conservative matrix.

Usage

```
eff.ini.seceig.general(Q, z0 = NULL, c1 = 1000, digit.thresh = 6)
```

Arguments

Q	The input general matrix.
z0	The type of initial z_0 used to calculate the approximation of $\rho(Q)$. There are two types: 'fixed' and 'Auto' corresponding to two choices of z_0 in paper.
c1	A large constant.
digit.thresh	The precise level of output results.

Value

A list of eigenpair object are returned, with components z and v.

z	The approximating sequence of the maximal eigenvalue.
v	The approximating sequence of the corresponding eigenvector.

Note

The conservativity of matrix $Q = (q_{ij})$ means that the sums of each row of matrix Q are all 0.

See Also

`eff.ini.seceig.tri` for the tridiagonal matrix next to the maximal eigenpair.

Examples

```
Q = matrix(c(-30, 1/5, 11/28, 55/3291, 30, -17, 275/42, 330/1097,
0, 84/5, -20, 588/1097, 0, 0, 1097/84, -2809/3291), 4, 4)
eff.ini.seceig.general(Q, z0 = 'Auto', digit.thresh = 5)
eff.ini.seceig.general(Q, z0 = 'fixed', digit.thresh = 5)
```

---

eff.ini.seceig.tri *Tridiagonal matrix next to the maximal eigenpair*

Description

Calculate the next to maximal eigenpair for the tridiagonal matrix whose sums of each row should be 0.

Usage

```
eff.ini.seceig.tri(a, b, xi = 1, digit.thresh = 6)
```

Arguments

a	The lower diagonal vector.
b	The upper diagonal vector.
xi	The coefficient used in the improved initials to form the convex combination of δ_1^{-1} and $(v_0, -Q * v_0)_\mu$, it should between 0 and 1.
digit.thresh	The precise level of output results.

Value

A list of eigenpair object are returned, with components z and v.

z	The approximating sequence of the maximal eigenvalue.
v	The approximating sequence of the corresponding eigenvector.

Note

The sums of each row of the input tridiagonal matrix should be 0.

See Also

`eff.ini.seceig.general` for the general conservative matrix next to the maximal eigenpair.

Examples

```
a = c(1:7)^2
b = c(1:7)^2

eff.ini.seceig.tri(a, b, xi = 0)
eff.ini.seceig.tri(a, b, xi = 1)
eff.ini.seceig.tri(a, b, xi = 2/5)
```

`EfficientMaxEigenpair` *EfficientMaxEigenpair: A package for computating the maximal eigenpair for a matrix.*

Description

The EfficientMaxEigenpair package provides some auxillary functions and four categories of important functions: `tridiag`, `ray.quot` `eff.ini.maxeig.tri`, `eff.ini.maxeig.general`, `eff.ini.seceig.tri` and `eff.ini.seceig.general`.

EfficientMaxEigenpair functions

`tridiag`: Generate tridiagonal matrix Q based on three input vectors.

`ray.quot`: Rayleigh quotient iteration algorithm to computing the maximal eigenpair of matrix Q.

`eff.ini.maxeig.tri`: Calculate the maximal eigenpair for the tridiagonal matrix.

`eff.ini.maxeig.general`: Calculate the maximal eigenpair for the general matrix.

`eff.ini.seceig.tri`: Calculate the next to maximal eigenpair for the tridiagonal matrix whose sums of each row should be 0.

`eff.ini.seceig.general`: Calculate the next to maximal eigenpair for the general conservative matrix.

`ray.quot` *Rayleigh quotient iteration*

Description

Rayleigh quotient iteration algorithm to computing the maximal eigenpair of matrix Q.

Usage

```
ray.quot(Q, mu, v0_tilde, zstart, digit.thresh = 6)
```

Arguments

Q	The input matrix to find the maximal eigenpair.
mu	A vector.
v0_tilde	The unnormalized initial vector $\tilde{v0}$.
zstart	The initial z_0 as an approximation of $\rho(Q)$.
digit.thresh	The precise level of output results.

Value

A list of eigenpair object are returned, with components 'z' and 'v'.

z	The approximating sequence of the maximal eigenvalue.
v	The approximating sequence of the corresponding eigenvector.

Examples

```
Q = matrix(c(1, 1, 3, 2, 2, 2, 3, 1, 1), 3, 3)
ray.quot(Q, mu=rep(1,dim(Q)[1]), v0_tilde=rep(1,dim(Q)[1]), zstart=6,
 digit.thresh = 6)
```

---

tridiag *Tridiagonal matrix*

Description

Generate tridiagonal matrix Q based on three input vectors.

Usage

```
tridiag(upper, lower, main)
```

Arguments

upper	The upper diagonal vector.
lower	The lower diagonal vector.
main	The main diagonal vector.

Value

A tridiagonal matrix is returned.

Examples

```
a = c(1:7)^2
b = c(1:7)^2
c = -c(1:8)^2
tridiag(b, a, c)
```

Index

eff.ini.maxeig.general, 2, *4*, *6*
eff.ini.maxeig.tri, *2*, 3, *6*
eff.ini.seceig.general, 4, *5*, *6*
eff.ini.seceig.tri, *5*, 5, *6*
EfficientMaxEigenpair, 6
EfficientMaxEigenpair-package
        *(*EfficientMaxEigenpair*)*, 6

ray.quot, *6*, 6

tridiag, *6*, 7