Package 'rARPACK'

August 24, 2014

Type Package

Title R wrapper of ARPACK for large scale eigenvalue/vector problems,on both dense and sparse matrices

Version 0.6-0

Date 2014-08-23

Author Yixuan Qiu and authors of the ARPACK library. See file AUTHORS for details.

Maintainer Yixuan Qiu <yixuan.qiu@cos.name>

Description rARPACK is an R wrapper of the ARPACK library
(http://www.caam.rice.edu/software/ARPACK/) to solve large scale
eigenvalue/vector problems. It is typically used to compute a few
eigenvalues/vectors of an n by n matrix, e.g., the k largest eigenvalues,which is usually more efficient than eigen() if k << n. This package
provides the eigs() function which does the similar job as in Matlab,Octave, Python SciPy and Julia. It also provides the svds() function
to calculate the largest k singular values and corresponding
singular vectors of a real matrix. Matrices can be given in either dense or sparse form.

License BSD_3_clause + file LICENSE

Copyright see file COPYRIGHTS

URL https://github.com/yixuan/rARPACK

BugReports https://github.com/yixuan/rARPACK/issues

Depends R (>= 3.0.2)

Imports Matrix (>= 1.1-0), Rcpp (>= 0.11.0)

LinkingTo Rcpp, RcppEigen (>= 0.3.2.2.0)

NeedsCompilation yes

Repository CRAN

Date/Publication 2014-08-24 16:31:58

R topics documented:

eigs 2
svds 5

Index **7**

`eigs` *Find a Specified Number of Eigenvalues/vectors for Square Matrix*

Description

Given an `n` by `n` matrix `A`, function `eigs()` can calculate a limited number of eigenvalues and eigenvectors of `A`. Users can specify the selection criteria by argument `which`, e.g., choosing the `k` largest or smallest eigenvalues and the corresponding eigenvectors.

Currently `eigs()` supports matrices of class "matrix" (the most commonly used matrix type), "dgeMatrix" (general matrix, equivalent to "matrix"), "dgCMatrix" (sparse matrix), "dgRMatrix" (sparse matrix, row oriented) and "dsyMatrix" (symmetric matrix). All classes above except "matrix" are defined in the **Matrix** package.

`eigs_sym()` assumes the matrix is symmetric, and only the lower triangle (or upper triangle, which is controlled by the argument `lower`) is used for computation, which in some cases reduces the workload. Notice that `eigs_sym()` only applies to "ordinary" matrix, i.e., of class "matrix". If you want to calculate eigen values/vectors of matrix of "dsyMatrix" class, use `eigs()` instead.

Usage

```
eigs(A, k, which = "LM", sigma = NULL, opts = list(), ...)

## S3 method for class 'matrix'
eigs(A, k, which = "LM", sigma = NULL, opts = list(),
  ...)

## S3 method for class 'dgeMatrix'
eigs(A, k, which = "LM", sigma = NULL, opts = list(),
  ...)

## S3 method for class 'dgCMatrix'
eigs(A, k, which = "LM", sigma = NULL, opts = list(),
  ...)

## S3 method for class 'dgRMatrix'
eigs(A, k, which = "LM", sigma = NULL, opts = list(),
  ...)

## S3 method for class 'dsyMatrix'
eigs(A, k, which = "LM", sigma = NULL, opts = list(),
  ...)
```

```
eigs_sym(A, k, which = "LM", sigma = NULL, opts = list(),
  ..., lower = TRUE)
```

Arguments

A	The matrix whose eigen values/vectors are to be computed.
k	Number of eigenvalues requested.
which	Selection criteria. See 'Details' below.
sigma	Shift parameter. See 'Details' below.
opts	Control parameters related to the computing algorithm. See 'Details' below.
...	Currently not used.
lower	For symmetric matrices, should the lower triangle or upper triangle be used.

Details

The which argument is a character string that specifies the type of eigenvalues to be computed. Possible values are:

"LM" The k eigenvalues with largest magnitude. Here the magnitude means the euclidean norm of complex numbers.

"SM" The k eigenvalues with smallest magnitude. Here the magnitude means the euclidean norm of complex numbers.

"LR" The k eigenvalues with largest real part.

"SR" The k eigenvalues with smallest real part.

"LI" The k eigenvalues with largest imaginary part.

"SI" The k eigenvalues with smallest imaginary part.

"LA" The k largest (algebraic) eigenvalues, considering any negative sign.

"SA" The k smallest (algebraic) eigenvalues, considering any negative sign.

"BE" Compute k eigenvalues, half from each end of the spectrum. When k is odd, compute more from the high and then from the low end.

eigs() with matrix type "matrix", "dgeMatrix", "dgCMatrix" and "dgRMatrix" can use "LM", "SM", "LR", "SR", "LI" and "SI".

eigs_sym() and eigs() with matrix type "dsyMatrix" can use "LM", "SM", "LA", "SA" and "BE".

The sigma argument is used in the shift-and-invert mode. When sigma is not NULL, the selection criteria specified by argument which will apply to $1/(\lambda - \sigma)$ where λ are the eigenvalues of A. For example, if A is positive definite and $\sigma = 0$, then which = "LM" will select the largest values of $1/\lambda$, which turns out to select the smallest eigenvalues of A. This method is preferable to which = "SM" in that ARPACK is good at finding large eigenvalues rather than finding small ones. More explanation of the shift-and-invert mode can be found in the SciPy document, http://docs.scipy.org/doc/scipy/reference/tutorial/arpack.html.

The opts argument is a list that can supply any of the following parameters:

`ncv` Number of Lanzcos basis vectors to use. More vectors will result in faster convergence, but with greater memory use. For general matrix, `ncv` must satisfy $k + 2 \leq ncv \leq n$, and for symmetric matrix, the constraint is $k < ncv \leq n$. Default is `min(n, max(2*k+1, 20))`.

`tol` Precision parameter. Default is 1e-10.

`maxitr` Maximum number of iterations. Default is 1000.

`retvec` Whether to compute eigenvectors. If FALSE, only calculate and return eigenvalues.

Value

A list of converged eigenvalues and eigenvectors.

`values`	Computed eigenvalues.
`vectors`	Computed eigenvectors. `vectors[, j]` corresponds to `values[j]`.
`nconv`	Number of converged eigenvalues.
`niter`	Number of iterations in the computation.

Author(s)

Yixuan Qiu <http://statr.me>

See Also

`eigen()`, `svd()`, `svds()`

Examples

```
n = 20;
k = 5;

## Will have complex eigenvalues
set.seed(111);
A1 = matrix(rnorm(n^2), n);
eigs(A1, k);

## Only have real eigenvalues,
## since A2 is symmetric
A2 = crossprod(A1);
eigs(A2, k);
eigs_sym(A2, k);

## Find the smallest (in absolute value) k eigenvalues of A2
eigs_sym(A2, k, which = "SM")
## Another way to do this: use the sigma argument
eigs_sym(A2, k, sigma = 0)
## The results should be the same,
## but the latter method is far more stable on large matrices

### more examples in examples/eigs.R ###
```

svds *(Truncated) Singular Value Decomposition of a Matrix*

Description

Given an m by n matrix A, function svds() can find its largest k singular values and the corresponding singular vectors. It's also called the truncated singular value decomposition since it only contains a subset of the whole singular triplets.

Currently svds() supports matrices of class "matrix", "dgeMatrix", "dgCMatrix", "dgRMatrix" and "dsyMatrix". All classes above except "matrix" are defined in the **Matrix** package, representing general matrix, sparse matrix (column oriented), sparse matrix (row oriented) and symmetric matrix respectively. Note that when A is symmetric, SVD reduces to eigen decomposition, so you may consider using eigs() instead.

Usage

```
svds(A, k, nu = k, nv = k, opts = list(), ...)

## S3 method for class 'matrix'
svds(A, k, nu = k, nv = k, opts = list(), ...)

## S3 method for class 'dgeMatrix'
svds(A, k, nu = k, nv = k, opts = list(), ...)

## S3 method for class 'dgCMatrix'
svds(A, k, nu = k, nv = k, opts = list(), ...)

## S3 method for class 'dgRMatrix'
svds(A, k, nu = k, nv = k, opts = list(), ...)

## S3 method for class 'dsyMatrix'
svds(A, k, nu = k, nv = k, opts = list(), ...)
```

Arguments

A	The matrix whose SVD is to be computed.
k	Number of singular values requested.
nu	Number of left singular vectors to be computed. This must be between 0 and k.
nv	Number of right singular vectors to be computed. This must be between 0 and k.
opts	Control parameters related to the computing algorithm. See Details below.
...	Currently not used.

Details

The `opts` argument is a list that can supply any of the following parameters:

`ncv` Number of Lanzcos basis vectors to use. More vectors will result in faster convergence, but with greater memory use. `ncv` must be satisfy $k < ncv \leq p$ where `p = min(m, n)`. Default is `min(p, max(2*k+1, 20))`.

`tol` Precision parameter. Default is 1e-10.

`maxitr` Maximum number of iterations. Default is 1000.

Value

A list with the following components:

`d`	A vector of the computed singular values.
`u`	An `m` by `nu` matrix whose columns contain the left singular vectors. If `nu == 0`, `NULL` will be returned.
`v`	An `n` by `nv` matrix whose columns contain the right singular vectors. If `nv == 0`, `NULL` will be returned.
`nconv`	Number of converged singular values.
`niter`	Number of iterations.

Author(s)

Yixuan Qiu <http://statr.me>

See Also

`eigen()`, `svd()`, `eigs()`.

Examples

```
m = 100;
n = 20;
k = 5;
set.seed(111);
A = matrix(rnorm(m * n), m);
svds(A, k);
svds(A, k, nu = 0, nv = 3);

### more examples in examples/svds.R ###
```

Index

∗Topic **array**
eigs, 2
svds, 5

eigen, *4*, *6*
eigs, 2, *5*, *6*
eigs_sym *(*eigs*)*, 2

svd, *4*, *6*
svds, *4*, 5