

Package ‘RcppArmadillo’

May 21, 2012

Type Package

Title Rcpp integration for Armadillo templated linear algebra library

Version 0.3.2.0

Date \$Date: 2012-05-21 07:29:28 -0500 (Mon, 21 May 2012) \$

Author Romain Francois, Dirk Eddelbuettel and Doug Bates

Maintainer Romain Francois, Dirk Eddelbuettel and Doug Bates
<RcppArmadillo-authors@r-enthusiasts.com>

Description R and Armadillo integration using Rcpp Armadillo is a templated C++ linear algebra library (by Conrad Sanderson) that aims towards a good balance between speed and ease of use. Integer, floating point and complex numbers are supported, as well as a subset of trigonometric and statistics functions. Various matrix decompositions are provided through optional integration with LAPACK and ATLAS libraries.

A delayed evaluation approach is employed (during compile time) to combine several operations into one, and to reduce (or eliminate) the need for temporaries. This is accomplished through recursive templates and template meta-programming.

This library is useful if C++ has been decided as the language of choice (due to speed and/or integration capabilities), rather than another language.

The RcppArmadillo package includes the header files from the templated Armadillo library (currently version 3.2.0). Thus users do not need to install Armadillo itself in order to use RcppArmadillo.

This Armadillo integration provides a nice illustration of the capabilities of the Rcpp package for seamless R and C++ integration.

Armadillo is licensed under the GNU LGPL version 3 or later, while RcppArmadillo (the Rcpp bindings/bridge to Armadillo) is licenses under the GNU GPL version 2 or later, as is the rest of Rcpp.

License GPL (>= 2)

LazyLoad yes**Depends** R (>= 2.14.0), Rcpp (>= 0.9.10)**LinkingTo** Rcpp**Suggests** inline, RUnit**URL** <http://arma.sourceforge.net/>,
<http://dirk.eddelbuettel.com/code/rcpp.armadillo.html>,
<http://romainfrancois.blog.free.fr/index.php?category/R-package/RcppArmadillo>**Repository** CRAN**Date/Publication** 2012-05-21 14:18:25

R topics documented:

RcppArmadillo-package	2
fastLm	4
RcppArmadillo.package.skeleton	6
Index	8

RcppArmadillo-package *Rcpp/Armadillo bridge*

Description

The package eases the integration of Armadillo types with Rcpp

Details

Package:	RcppArmadillo
Type:	Package
Version:	0.2.36
Date:	2012-03-05
License:	GPL (>= 2)
LazyLoad:	yes

Armadillo

Armadillo is a C++ linear algebra library (matrix maths) aiming towards a good balance between speed and ease of use. Integer, floating point and complex numbers are supported, as well as a subset of trigonometric and statistics functions. Various matrix decompositions are provided through optional integration with LAPACK and ATLAS libraries.

A delayed evaluation approach is employed (during compile time) to combine several operations into one and reduce (or eliminate) the need for temporaries. This is accomplished through recursive templates and template meta-programming.

This library is useful if C++ has been decided as the language of choice (due to speed and/or integration capabilities), rather than another language like Matlab or Octave. It is distributed under a license that is useful in both open-source and commercial contexts.

Armadillo is primarily developed at NICTA (Australia), with contributions from around the world.

RcppArmadillo

RcppArmadillo acts as a bridge between Rcpp and Armadillo, allowing the programmer to write code using armadillo classes that integrate seamlessly with Rcpp.

Using RcppArmadillo

The simplest way to get started is to create a skeleton of a package using RcppArmadillo, this is done by the `RcppArmadillo.package.skeleton` function.

The important steps are

- Include the `RcppArmadillo.h` header file, which also includes `armadillo.h`
- Depend and link to Rcpp and RcppArmadillo by adding these lines to the DESCRIPTION file:

```
Depends: Rcpp (>= 0.8.0), RcppArmadillo (>= 0.2.0)
LinkingTo: Rcpp, RcppArmadillo
```

- Link against the Rcpp, blas and lapack libraries, by adding this line in the Makevars

```
PKG_LIBS = $(shell $(R_HOME)/bin/Rscript -e "Rcpp:::LdFlags()" ) $(LAPACK_LIBS) $(BLAS_LIBS) $(FLI
```

and this line to the Makevars.win:

```
PKG_LIBS = $(shell $(R_HOME)/bin/${R_ARCH_BIN}/Rscript.exe -e "Rcpp:::LdFlags()") $(LAPACK_LIBS) $
```

Options: ATLAS and Boost

Armadillo can optionally use atlas and boost. Please refer to the Armadillo documentation <http://arma.sourceforge.net/>

RcppArmadillo does not force these features. it is up to the third party package to enable these them for their package. If your package wants to make use of these features, you can add a subset of these lines before `#include <RcppArmadillo.h>`

```
#define ARMA_USE_BOOST
#define ARMA_USE_BOOST_DATE
#define ARMA_USE_ATLAS
```

when you do so, you must make sure that the corresponding features are available, which typically involves writing a configure file. Refer to Writing R Extensions for the procedure.

Support

Please use the Rcpp-devel mailing list on r-forge for questions about RcppArmadillo (subscribe first). <https://lists.r-forge.r-project.org/cgi-bin/mailman/listinfo/rcpp-devel>

Questions about armadillo itself should be directed to the armadillo forum <http://sourceforge.net/apps/phpbb/arma/>

Author(s)

For RcppArmadillo: Romain Francois, Dirk Eddelbuettel and Doug Bates

Maintainer: Romain, Dirk and Doug <RcppArmadillo-authors@r-enthusiasts.com>

For Armadillo: Conrad Sanderson

References

Armadillo project: <http://arma.sourceforge.net/>

Conrad Sanderson, *Armadillo: An Open Source C++ Linear Algebra Library for Fast Prototyping and Computationally Intensive Experiments*. Technical Report, NICTA, 2010.

fastLm

Bare-bones linear model fitting function

Description

fastLm estimates the linear model using the solve function of Armadillo linear algebra library.

Usage

```
fastLmPure(X, y)

fastLm(X, ...)
## Default S3 method:
fastLm(X, y, ...)
## S3 method for class 'formula'
fastLm(formula, data = list(), ...)
```

Arguments

y	a vector containing the explained variable.
X	a model matrix.
formula	a symbolic description of the model to be fit.
data	an optional data frame containing the variables in the model.
...	not used

Details

Linear models should be estimated using the `lm` function. In some cases, `lm.fit` may be appropriate.

The `fastLmPure` function provides a reference use case of the Armadillo library via the wrapper functions in the **RcppArmadillo** package.

The `fastLm` function provides a more standard implementation of a linear model fit, offering both a default and a formula interface as well as `print`, `summary` and `predict` methods.

Lastly, one must be careful in timing comparisons of `lm` and friends versus this approach based on Armadillo. The reason that Armadillo can do something like `lm.fit` faster than the functions in the stats package is because Armadillo uses the Lapack version of the QR decomposition while the stats package uses a *modified* Linpack version. Hence Armadillo uses level-3 BLAS code whereas the stats package uses level-1 BLAS. However, Armadillo will either fail or, worse, produce completely incorrect answers on rank-deficient model matrices whereas the functions from the stats package will handle them properly due to the modified Linpack code.

An example of the type of situation requiring extra care in checking for rank deficiency is a two-way layout with missing cells (see the examples section). These cases require a special pivoting scheme of “pivot only on (apparent) rank deficiency” which is not part of conventional linear algebra software.

Value

`fastLmPure` returns a list with three components:

<code>coefficients</code>	a vector of coefficients
<code>stderr</code>	a vector of the (estimated) standard errors of the coefficient estimates
<code>df.residual</code>	a scalar denoting the degrees of freedom in the model

`fastLm` returns a richer object which also includes the residuals, fitted values and call argument similar to the `lm` or `rlm` functions..

Author(s)

Armadillo is written by Conrad Sanderson. RcppArmadillo is written by Romain Francois, Dirk Eddelbuettel and Douglas Bates.

References

Armadillo project: <http://arma.sourceforge.net/>

See Also

[lm](#), [lm.fit](#)

Examples

```

data(trees, package="datasets")

## bare-bones direct interface
flm <- fastLmPure( cbind(1, log(trees$Girth)), log(trees$Volume) )
print(flm)

## standard R interface for formula or data returning object of class fastLm
flmmod <- fastLm( log(Volume) ~ log(Girth), data=trees)
summary(flmmod)

## case where fastLm breaks down
dd <- data.frame(f1 = gl(4, 6, labels = LETTERS[1:4]),
                 f2 = gl(3, 2, labels = letters[1:3]))[-(7:8), ]
xtabs(~ f2 + f1, dd) # one missing cell
mm <- model.matrix(~ f1 * f2, dd)
kappa(mm) # large, indicating rank deficiency
set.seed(1)
dd$y <- mm %*% seq_len(ncol(mm)) + rnorm(nrow(mm), sd = 0.1)
summary(lm(y ~ f1 * f2, dd)) # detects rank deficiency
summary(fastLm(y ~ f1 * f2, dd)) # some huge coefficients

```

RcppArmadillo.package.skeleton

Create a skeleton for a new package that intends to use RcppArmadillo

Description

RcppArmadillo.package.skeleton automates the creation of a new source package that intends to use features of RcppArmadillo.

It is based on the [package.skeleton](#) function which it executes first.

Usage

```

RcppArmadillo.package.skeleton(name = "anRpackage", list = character(),
environment = .GlobalEnv, path = ".", force = FALSE, namespace = TRUE,
code_files = character(), example_code = TRUE)

```

Arguments

name	See package.skeleton
list	See package.skeleton
environment	See package.skeleton
path	See package.skeleton
force	See package.skeleton
namespace	See package.skeleton
code_files	See package.skeleton
example_code	If TRUE, example c++ code using RcppArmadillo is added to the package

Details

In addition to [package.skeleton](#) :

The 'DESCRIPTION' file gains a Depends line requesting that the package depends on Rcpp and RcppArmadillo and a LinkingTo line so that the package finds Rcpp and RcppArmadillo header files.

The 'NAMESPACE', if any, gains a useDynLib directive.

The 'src' directory is created if it does not exist and a 'Makevars' file is added setting the environment variable 'PKG_LIBS' to accommodate the necessary flags to link with the Rcpp library.

If the `example_code` argument is set to TRUE, example files 'rcpparma_hello_world.h' and 'rcpparma_hello_world.cpp' are also created in the 'src'. An R file 'rcpparma_hello_world.R' is expanded in the 'R' directory, the `rcpparma_hello_world` function defined in this file makes use of the C++ function 'rcpparma_hello_world' defined in the C++ file. These files are given as an example and should eventually be removed from the generated package.

Value

Nothing, used for its side effects

References

Read the *Writing R Extensions* manual for more details.

Once you have created a *source* package you need to install it: see the *R Installation and Administration* manual, [INSTALL](#) and [install.packages](#).

See Also

[package.skeleton](#)

Examples

```
## Not run:  
RcppArmadillo.package.skeleton( "foobar" )  
  
## End(Not run)
```

Index

- *Topic **interface**
 - RcppArmadillo-package, [2](#)
- *Topic **package**
 - RcppArmadillo-package, [2](#)
- *Topic **programming**
 - RcppArmadillo-package, [2](#)
 - RcppArmadillo.package.skeleton, [6](#)
- *Topic **regression**
 - fastLm, [4](#)

fastLm, [4](#)
fastLmPure (fastLm), [4](#)

INSTALL, [7](#)
install.packages, [7](#)

lm, [5](#)
lm.fit, [5](#)

package.skeleton, [6](#), [7](#)

RcppArmadillo (RcppArmadillo-package), [2](#)
RcppArmadillo-package, [2](#)
RcppArmadillo.package.skeleton, [3](#), [6](#)
RcppArmadilloExample
 (RcppArmadillo-package), [2](#)
r1m, [5](#)