Package 'knitroR'

January 5, 2015

Type Package

Title R integration of Knitro

Version 1.0

Date 2014-12-21

Author Jan Tilly

Maintainer Jan Tilly <jtilly@econ.upenn.edu>

Description This package provides an R integration of knitro via Rcpp. Knitro has an excellent C++ implementation. This package passes user defined R functions on to the C++ interface. To use Knitro you need to have a valid license.

License GPL (>= 2)

Imports Rcpp (>= 0.11.3)

LinkingTo Rcpp

R topics documented:

knitro 1
knitroCpp 2
knitro_ms 3

Index **5**

`knitro` *Call the knitro C++ interface*

Description

This function passes user defined R functions on to the C++ interface

Usage

```
knitro(objFun = NULL, objGrad = NULL, c_equality = NULL,
  c_inequality = NULL, jac = NULL, jacIndexCons = NULL,
  jacIndexVars = NULL, x0 = NA, lb = NULL, ub = NULL,
  optionsFile = "options.opt")
```

Arguments

objFun	is a scalar valued R function that returns the objective function
objGrad	is a vector-valued R function with the gradient
c_equality	is a vector-valued R function with equality constraints
c_inequality	is a vector-valued R function with inequality constraints
jac	is a vector with the content of the Jacobian (sparse)
jacIndexCons	refers to each element of jac and contains the number of the constraint it refers to. Indexing is C++ compatible, i.e. the first constraint has index 0
x0	is a vector with starting values
optionsFile	is the path and filename of the options file. If it does not exist, the function will create it
jacIndexCons	refers to each element of jac and contains the number of the variable it refers to. Indexing is C++ compatible, i.e. the first variable has index 0

Value

a list with the final estimates, the function value, and Knitro's exit status

---

knitroCpp *Knitro C++ Wrapper*

Description

This function is the standard C++ wrapper around knitro. It defines the object KTR_new and defines a callback function that is used to evaluate the objective function, the constraints, and gradients. The only deviation from the standard C++ wrapper is to use UserParam to pass the original R functions on to the C++ callback function.

Usage

```
knitroCpp(fcts, startValues, num_equality_constraints,
  num_inequality_constraints, nnzJ, RjacIndexCons, RjacIndexVars, ub, lb,
  optionsFile)
```

Arguments

fcts	is an R list of functions that includes the objFun, objGrad, c, and jac.
startValues	is a vector of start values
num_equality_constraints	is an integer with the number of equality constraints in c
num_inequality_constraints	is an integer with the number of inequality constraints in c
nnzJ	is an integer with the number of non-zero objects in the Jacobian

RjacIndexCons	is a vector of length nnzJ. Each element contains the index of a particular constraint (i.e. the index of a row in the jacobian).
RjacIndexVars	is a vector of length nnzJ. Each element contains the index of a particular variable (i.e. the index of a column in the jacobian).
ub	a vector of upper bounds for each element in x0
lb	a vector lower bounds for each element in x0
optionsFile	the location of the options file

Value

A list with the vector that minimizes the objective function, the final function value, and Knitro's exit status

See Also

http://www.artelys.com/tools/knitro_doc/2_userGuide/gettingStarted/startCallableLibrary.html

---

knitro_ms *Call the knitro C++ interface using multiple start values*

Description

This function passes user defined R functions on to the C++ interface. In contrast to knitro() knitro uses a matrix of startvalues as input, where each row corresponds to one vector of start values that knitro will attempt to optimize the objective function. The function returns the solution for the set of start values that resulted in the lowest objective function.

Usage

```
knitro_ms(objFun = NULL, objGrad = NULL, c_equality = NULL,
  c_inequality = NULL, jac = NULL, jacIndexCons = NULL,
  jacIndexVars = NULL, x0 = NA, lb = NULL, ub = NULL,
  optionsFile = "options.opt")
```

Arguments

objFun	is a scalar valued R function that returns the objective function
objGrad	is a vector-valued R function with the gradient
c_equality	is a vector-valued R function with equality constraints
c_inequality	is a vector-valued R function with inequality constraints
jac	is a vector with the content of the Jacobian (sparse)
jacIndexCons	refers to each element of jac and contains the number of the constraint it refers to. Indexing is C++ compatible, i.e. the first constraint has index 0
x0	is a matrix with starting values

`optionsFile`	is the path and filename of the options file. If it does not exist, the function will create it
`jacIndexCons`	refers to each element of jac and contains the number of the variable it refers to. Indexing is C++ compatible, i.e. the first variable has index 0

Value

a list with the final estimates, the function value, and Knitro's exit status

Index

knitro, 1
knitro_ms, 3
knitroCpp, 2