Package 'knitroR'

October 28, 2015

Type Package

Title R Interface for Non-Linear Constraint Optimizer Knitro

Version 1.0

Date 2014-12-21

Author Jan Tilly

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

URL https://github.com/jtilly/knitroR

Description Provides an R interface for the non-linear constraint optimizer KNITRO. This package passes user-defined R functions on to KNITRO's C++ interface. This package does not include KNITRO. To use this package you need to install KNITRO and own a valid license.

License GPL (>= 2)

Imports Rcpp (>= 0.11.3)

LinkingTo Rcpp

Suggests testthat

R topics documented:

knitro 1
knitroCpp 2
knitro_ms 3

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

Description

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

Usage

```
knitro(objFun, 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
jacIndexVars	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
x0	is a vector with starting values
lb	is a vector of lower bounds
ub	is a vector of upper bounds
optionsFile	is the path and filename of the options file. If it does not exist, the function will create it

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 for 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 is a multi start version of knitro(). Uses a matrix as startvalues where each row corresponds to one set of startvalues to be used. This version of multi-start gives the user more control over the start values than KNITRO's built-in version of multi-start. If you want to use the built-in version of multi-start instead, you can do so via the options file.

Usage

```
knitro_ms(objFun, 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
jacIndexVars	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
x0	is a matrix with starting values
lb	is a vector of lower bounds
ub	is a vector of upper bounds
optionsFile	is the path and filename of the options file. If it does not exist, the function will create it

Value

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