Package ‘ECOSolveR’

February 18, 2018

Type Package
Title Embedded Conic Solver in R
Version 0.4
Date 2018-02-16
VignetteBuilder knitr
SystemRequirements GNU make

URL https://github.com/bnaras/ECOSolveR
BugReports https://github.com/bnaras/ECOSolveR/issues
Suggests knitr, rmarkdown, testthat
Imports Matrix
Description R interface to the Embedded COnic Solver (ECOS), an efficient
and robust C library for convex problems. Conic and equality
constraints can be specified in addition to integer and
boolean variable constraints for mixed-integer problems. This
R interface is inspired by the python interface and has
similar calling conventions.
License GPL (>= 3)
RoxygenNote 6.0.1
NeedsCompilation yes
Author Anqi Fu [aut],
Balasubramanian Narasimhan [aut, cre]
Maintainer Balasubramanian Narasimhan <naras@stat.Stanford.EDU>
Repository CRAN
Date/Publication 2018-02-18 17:28:17 UTC

R topics documented:

  ecos.control .......................................................... 2
  ECOSolveR ............................................................ 3
  ECOS_csolve .......................................................... 3
ecos.control

Index

ecos.control Return the default optimization parameters for ECOS

Description

This is used to control the behavior of the underlying optimization code.

Usage

ecos.control(maxit = 100L, feastol = 1e-08, reltol = 1e-08,
abstol = 1e-08, feastol_inacc = 1e-04, abstol_inacc = 5e-05,
reltol_inacc = 5e-05, verbose = 0L, mi_max_iters = 1000L,
mi_int_tol = 1e-04, mi_abs_eps = 1e-06, mi_rel_eps = 1e-06)

Arguments

maxit the maximum number of iterations for ecos, default 100L
feastol the tolerance on the primal and dual residual, default 1e-8
reltol the relative tolerance on the duality gap, default 1e-8
abstol the absolute tolerance on the duality gap, default 1e-8
feastol_inacc the tolerance on the primal and dual residual if reduced precisions, default 1e-4
abstol_inacc the absolute tolerance on the duality gap if reduced precision, default 5e-5
reltol_inacc the relative tolerance on the duality gap if reduced precision, default 5e-5
verbose verbosity level, default 0L. A verbosity level of 1L will show more detail, but clutter session transcript.
mi_max_iters the maximum number of branch and bound iterations (mixed integer problems only), default 1000L
mi_int_tol the integer tolerance (mixed integer problems only), default 1e-4
mi_abs_eps the absolute tolerance between upper and lower bounds (mixed integer problems only), default 1e-6
mi_rel_eps the relative tolerance, \((U - L) / L\), between upper and lower bounds (mixed integer problems only), default 1e-6

Value

a list with the following elements:

FEASTOL the tolerance on the primal and dual residual, parameter feastol
ABSTOL the absolute tolerance on the duality gap, parameter abstol
RELTOL the relative tolerance on the duality gap, parameter reltol
FEASTOL_INACC the tolerance on the primal and dual residual if reduced precisions, parameter feastol_inacc
**ABSTOL_INACC** the absolute tolerance on the duality gap if reduced precision, parameter abstol_inacc

**RELTOL_INACC** the relative tolerance on the duality gap if reduced precision, parameter reltol_inacc

**MAXIT** the maximum number of iterations for ecos, parameter maxit

**MI_MAX_ITERS** the maximum number of branch and bound iterations (mixed integer problems only), parameter mi_max_iters

**MI_INT_TOL** the integer tolerance (mixed integer problems only), parameter mi_int_tol

**MI_ABS_EPS** the absolute tolerance between upper and lower bounds (mixed integer problems only), parameter mi_abs_eps

**MI_REL_EPS** the relative tolerance, \( (U - L)/L \), between upper and lower bounds (mixed integer problems only), parameter mi_rel_eps

**VERBOSE** verbosity level, parameter verbose

---

**ECOSolveR**  
**ECOSolveR: Embedded Conic Solver in R**

**Description**

ECOSolveR is a wrapper around the ecos library. Please see the examples and documentation for the function ECOS_csolve.

**References**

[https://github.com/embotech/ecos](https://github.com/embotech/ecos)

---

**ECOS_csolve**  
**Solve a conic optimization problem**

**Description**

The function ECOS_csolve is a wrapper around the ecos csolve C function. Conic constraints are specified using the \( G \) and \( h \) parameters and can be NULL and zero length vector respectively indicating an absence of conic constraints. Similarly, equality constraints are specified via \( A \) and \( b \) parameters with NULL and empty vector values representing a lack of such constraints. At most one of the pair \((G, h)\) or \((A, b)\) is allowed to be absent.

**Usage**

```r
ECOS_csolve(c = numeric(0), G = NULL, h = numeric(0), dims = list(l = integer(0), q = NULL, e = integer(0)), A = NULL, b = numeric(0), bool_vars = integer(0), int_vars = integer(0), control = ecos.control())
```
Arguments

c  the coefficients of the objective function; the length of this determines the number of variables \( n \) in the problem.

G  the inequality constraint sparse matrix in compressed column format, e.g. \texttt{dgCMatrix-class}. Can be \texttt{NULL}.

h  the right hand size of the inequality constraint. Can be empty numeric vector.

dims  is a list of three named elements: \texttt{dims['l']} an integer specifying the dimension of positive orthant cone, \texttt{dims['q']} an integer vector specifying dimensions of second-order cones, \texttt{dims['e']} an integer specifying the number of exponential cones.

A  the optional equality constraint sparse matrix in compressed column format, e.g. \texttt{dgCMatrix-class}. Can be \texttt{NULL}.

b  the right hand side of the equality constraint, must be specified if \( A \) is. Can be empty numeric vector.

bool_vars  the indices of the variables, 1 through \( n \), that are boolean; that is, they are either present or absent in the solution.

int_vars  the indices of the variables, 1 through \( n \), that are integers.

control  is a named list that controls various optimization parameters; see \texttt{ecos.control}.

Value

a list of 8 named items

\begin{itemize}
  \item [x] primal variables
  \item [y] dual variables for equality constraints
  \item [s] slacks for \( Gx + s \leq h, s \in K \)
  \item [z] dual variables for inequality constraints \( s \in K \)
  \item [infostring] gives information about the status of solution
  \item [retcodes] a named integer vector containing four elements
    \begin{itemize}
      \item [exitflag] \( 0 = \texttt{OPTIMAL}, 1 = \texttt{PRIMAL INFEASIBLE}, 2 = \texttt{DUAL INFEASIBLE}, -1 = \texttt{MAXIT REACHED} \)
      \item [iter] the number of iteration used
      \item [mi_iter] the number of iterations for mixed integer problems
      \item [numerr] a non-zero number if a numeric error occurred
    \end{itemize}
  \item [summary] a named numeric vector containing
    \begin{itemize}
      \item [pcost] value of primal objective
      \item [dcost] value of dual objective
      \item [pres] primal residual on inequalities and equalities
      \item [dres] dual residual
      \item [pinf] primal infeasibility measure
      \item [dinf] dual infeasibility measure
      \item [pinfres] primal infeasibility residual
      \item [dinfres] dual infeasibility residual
    \end{itemize}
\end{itemize}
ECOS_csolve

gap duality gap
relgap relative duality gap
r0 Unknown at the moment to this R package maintainer.
timing a named numeric vector of timing information consisting of
runtime the total runtime in ecos
tsetup the time for setup of the problem
tsolve the time to solve the problem

Details
A call to this function will solve the problem: minimize \( c^T x \), subject to \( Ax = b \), and \( h - G \ast x \in K \). Variables can be constrained to be boolean (1 or 0) or integers. This is indicated by specifying parameters bool_vars and/or int_vars respectively. If so indicated, the solutions will be found using a branch and bound algorithm.

Examples

```r
## githubIssue98
G <- local(
  Gpr <- c(0.416757847405471, 2.136196095668454, 1.793435585194863, -1.,
           0.05626682726329, -1.640270088404989, 0.841747365656204, -1.,
           0.416757847405471, 2.136196095668454, 1.793435585194863, -1.,
           0.05626682726329, -1.640270088404989, 0.841747365656204, -1.,
           1)
  Gjc <- as.integer(c(0, 4, 8, 12, 16, 17))
  Gir <- as.integer(c(0, 1, 2, 7, 0, 1, 2, 8, 3, 4, 5, 9, 3, 4, 5, 10, 6))
  Matrix::sparseMatrix(i = Gir, p = Gjc, x = Gpr, index1 = FALSE)
)

print(G)
c <- as.numeric(c(0, 0, 0, 0, 1))
h <- as.numeric(c(0, 0, 0, 0, 0, 0, 0, 0, 0))
dims <- list(l = 6L, q = 5L, e = 0L)
ECOS_csolve(c = c, G = G, h = h,
            dims = dims,
            A = NULL, b = numeric(0))
```

## A larger problem using saved data for the large matrices
MPC01 <- readRDS(system.file("misc", "MPC01.rds", package="ECOSolveR"))
retval <- ECOS_csolve(c = MPC01$c, G = MPC01$G, h = MPC01$h,
                      dims = MPC01$dims)
retval$retcodes
tval$infostring
tval$summary
Index

dgCMatrix-class, 4
ecos.control, 2, 4
ECOS_csolve, 3
ECOSolveR, 3
ECOSolveR-package (ECOSolveR), 3