Package 'minqa'

August 17, 2024

Type Package

Title Derivative-Free Optimization Algorithms by Quadratic Approximation

Version 1.2.8

Maintainer Katharine M. Mullen <katharine.mullen@stat.ucla.edu>

Description Derivative-free optimization by quadratic approximation based on an interface to Fortran implementations by M. J. D. Powell.

License GPL-2

URL http://optimizer.r-forge.r-project.org

Imports Rcpp (>= 0.9.10)

LinkingTo Rcpp

SystemRequirements GNU make

NeedsCompilation yes

Repository CRAN

Date/Publication 2024-08-17 21:50:06 UTC

Author Douglas Bates [aut],
Katharine M. Mullen [aut, cre],
John C. Nash [aut],
Ravi Varadhan [aut]

Contents

bobyqa *An R interface to the bobyqa implementation of Powell*

Description

The purpose of bobyqa is to minimize a function of many variables by a trust region method that forms quadratic models by interpolation. Box constraints (bounds) on the parameters are permitted.

Usage

```
bobyqa(par, fn, lower = -Inf, upper = Inf, control = list(), ...)
```

Arguments

par	A numeric vector of starting estimates of the parameters of the objective function.
fn	A function that returns the value of the objective at the supplied set of parameters par using auxiliary data in .... The first argument of fn must be par.
lower	A numeric vector of lower bounds on the parameters. If the length is 1 the single lower bound is applied to all parameters.
upper	A numeric vector of upper bounds on the parameters. If the length is 1 the single upper bound is applied to all parameters.
control	An optional list of control settings. See the details section for the names of the settable control values and their effect.
...	Further arguments to be passed to fn.

Details

The function fn must return a scalar numeric value.

The control argument is a list. Possible named values in the list and their defaults are:

npt The number of points used to approximate the objective function via a quadratic approximation. The value of npt must be in the interval $[n+2, (n+1)(n+2)/2]$ where n is the number of parameters in par. Choices that exceed $2 * n + 1$ are not recommended. If not defined, it will be set to $\min(n * 2, n + 2)$.

rhobeg rhobeg and rhoend must be set to the initial and final values of a trust region radius, so both must be positive with 0 < rhoend < rhobeg. Typically rhobeg should be about one tenth of the greatest expected change to a variable. If the user does not provide a value, this will be set to min(0.95, 0.2 * max(abs(par))). Note also that smallest difference abs(upper-lower) should be greater than or equal to rhobeg*2. If this is not the case then rhobeg will be adjusted.

rhoend The smallest value of the trust region radius that is allowed. If not defined, then 1e-6 times the value set for rhobeg will be used.

iprint The value of `iprint` should be set to an integer value in `0, 1, 2, 3, ...`, which controls the amount of printing. Specifically, there is no output if `iprint=0` and there is output only at the start and the return if `iprint=1`. Otherwise, each new value of `rho` is printed, with the best vector of variables so far and the corresponding value of the objective function. Further, each new value of the objective function with its variables are output if `iprint=3`. If `iprint > 3`, the objective function value and corresponding variables are output every `iprint` evaluations. Default value is `0`.

maxfun The maximum allowed number of function evaluations. If this is exceeded, the method will terminate.

Value

A list with components:

`par`	The best set of parameters found.
`fval`	The value of the objective at the best set of parameters found.
`feval`	The number of function evaluations used.
`ierr`	An integer error code. A value of zero indicates success. Other values are **1** maximum number of function evaluations exceeded **2** NPT, the number of approximation points, is not in the required interval **3** a trust region step failed to reduce q (Consult Powell for explanation.) **4** one of the box constraint ranges is too small (< 2*RHOBEG) **5** bobyqa detected too much cancellation in denominator (We have not fully understood Powell's code to explain this.)
`msg`	A message describing the outcome of UOBYQA

References

M. J. D. Powell (2007) "Developments of NEWUOA for unconstrained minimization without derivatives", Cambridge University, Department of Applied Mathematics and Theoretical Physics, Numerical Analysis Group, Report NA2007/05, http://www.damtp.cam.ac.uk/user/na/NA_papers/NA2007_05.pdf.

M. J. D. Powell (2009), "The BOBYQA algorithm for bound constrained optimization without derivatives", Report No. DAMTP 2009/NA06, Centre for Mathematical Sciences, University of Cambridge, UK. http://www.damtp.cam.ac.uk/user/na/NA_papers/NA2009_06.pdf.

Description was taken from comments in the Fortran code of M. J. D. Powell on which **minqa** is based.

See Also

`optim`, `nlminb`

Examples

```
fr <- function(x) {   ## Rosenbrock Banana function
    100 * (x[2] - x[1]^2)^2 + (1 - x[1])^2
}
(x1 <- bobyqa(c(1, 2), fr, lower = c(0, 0), upper = c(4, 4)))
## => optimum at c(1, 1) with fval = 0

str(x1)  # see that the error code and msg are returned

# check the error exits
# too many iterations
x1e<-bobyqa(c(1, 2), fr, lower = c(0, 0), upper = c(4, 4), control = list(maxfun=50))
str(x1e)

# Throw an error because bounds too tight
x1b<-bobyqa(c(4,4), fr, lower = c(0, 3.9999999), upper = c(4, 4))
str(x1b)

# Throw an error because npt is too small -- does NOT work as of 2010-8-10 as
#    minqa.R seems to force a reset.
x1n<-bobyqa(c(2,2), fr, lower = c(0, 0), upper = c(4, 4), control=list(npt=1))
str(x1n)

# To add if we can find them -- examples of ierr = 3 and ierr = 5.
```

---

newuoa *An R interface to the NEWUOA implementation of Powell*

Description

The purpose of newuoa is to minimize a function of many variables by a trust region method that forms quadratic models by interpolation.

Usage

```
newuoa(par, fn, control = list(), ...)
```

Arguments

par	A numeric vector of starting estimates.
fn	A function that returns the value of the objective at the supplied set of parameters par using auxiliary data in .... The first argument of fn must be par.
control	An optional list of control settings. See the details section for the names of the settable control values and their effect.
...	Further arguments to be passed to fn.

Details

Functions `fn` must return a numeric value. The `control` argument is a list; possible named values in the list and their defaults are:

npt The number of points used to approximate the objective function via a quadratic approximation. The value of npt must be in the interval $[n+2, (n+1)(n+2)/2]$ where n is the number of parameters in `par`. Choices that exceed $2 * n + 1$ are not recommended. If not defined, it will be set to $\min(n * 2, n + 2)$.

rhobeg `rhobeg` and `rhoend` must be set to the initial and final values of a trust region radius, so both must be positive with `0 < rhoend < rhobeg`. Typically `rhobeg` should be about one tenth of the greatest expected change to a variable. If the user does not provide a value, this will be set to `max(par) / 2)`

rhoend The smallest value of the trust region radius that is allowed. If not defined, then 1e-6 times the value set for `rhobeg` will be used.

iprint The value of `iprint` should be set to an integer value in `0, 1, 2, 3, ...`, which controls the amount of printing. Specifically, there is no output if `iprint=0` and there is output only at the start and the return if `iprint=1`. Otherwise, each new value of `rho` is printed, with the best vector of variables so far and the corresponding value of the objective function. Further, each new value of the objective function with its variables are output if `iprint=3`. If `iprint > 3`, the objective function value and corresponding variables are output every `iprint` evaluations. Default value is `0`.

maxfun The maximum allowed number of function evaluations. If this is exceeded, the method will terminate.

Value

A list with components:

`par`	The best set of parameters found.
`fval`	The value of the objective at the best set of parameters found.
`feval`	Number of function evaluations to determine the optimum
`ierr`	An integer error code. A value of zero indicates success. Other values (consistent with BOBYQA values) are **1** maximum number of function evaluations exceeded **2** NPT, the number of approximation points, is not in the required interval **3** a trust region step failed to reduce q (Consult Powell for explanation.) **5** newuoa detected too much cancellation in denominator (We have not fully understood Powell's code to explain this.)
`msg`	A message describing the outcome of UOBYQA

References

M. J. D. Powell, "The NEWUOA software for unconstrained optimization without derivatives", in **Large-Scale Nonlinear Optimization**, Series: Nonconvex Optimization and Its Applications , Vol. 83, Di Pillo, Gianni; Roma, Massimo (Eds.) 2006, New York: Springer US.

M. J. D. Powell, "Developments of NEWUOA for minimization without derivatives" IMA Journal of Numerical Analysis, 2008; 28: 649-664.

M. J. D. Powell (2007) "Developments of NEWUOA for unconstrained minimization without derivatives" Cambridge University, Department of Applied Mathematics and Theoretical Physics, Numerical Analysis Group, Report NA2007/05, http://www.damtp.cam.ac.uk/user/na/NA_papers/NA2007_05.pdf.

Description was taken from comments in the Fortran code of M. J. D. Powell on which **minqa** is based.

See Also

optim, nlminb

Examples

```
fr <- function(x) {   ## Rosenbrock Banana function
    100 * (x[2] - x[1]^2)^2 + (1 - x[1])^2
}
(x2 <- newuoa(c(1, 2), fr))
## => optimum at c(1, 1) with fval = 0

# check the error exits
# too many iterations
x2e<-newuoa(c(1, 2), fr, control = list(maxfun=50))
str(x2e)

# Throw an error because npt is too small -- does NOT work as of 2010-8-10 as
#    minqa.R seems to force a reset.
x2n<-newuoa(c(2,2), fr, control=list(npt=1))
str(x2n)

# To add if we can find them -- examples of ierr = 3 and ierr = 5.
```

---

uobyqa *An R interface to the uobyqa implementation of Powell*

Description

The purpose of uobyqa is to minimize a function of many variables by a trust region method that forms quadratic models by interpolation.

Usage

```
uobyqa(par, fn, control = list(), ...)
```

Arguments

par	A numeric vector of starting estimates.
fn	A function that returns the value of the objective at the supplied set of parameters par using auxiliary data in .... The first argument of fn must be par.
control	An optional list of control settings. See the details section for the names of the settable control values and their effect.
...	Further arguments to be passed to fn.

Details

Functions fn must return a numeric value.

The control argument is a list. Possible named values in the list and their defaults are:

rhobeg rhobeg and rhoend must be set to the initial and final values of a trust region radius, so both must be positive with 0 < rhoend < rhobeg. Typically rhobeg should be about one tenth of the greatest expected change to a variable.

rhoend The smallest value of the trust region radius that is allowed. If not defined, then 1e-6 times the value set for rhobeg will be used.

iprint The value of iprint should be set to an integer value in 0, 1, 2, 3, ..., which controls the amount of printing. Specifically, there is no output if iprint=0 and there is output only at the start and the return if iprint=1. Otherwise, each new value of rho is printed, with the best vector of variables so far and the corresponding value of the objective function. Further, each new value of the objective function with its variables are output if iprint=3. If iprint > 3, the objective function value and corresponding variables are output every iprint evaluations. Default value is 0.

maxfun The maximum allowed number of function evaluations. If this is exceeded, the method will terminate.

Powell's Fortran code has been slightly modified (thanks to Doug Bates for help on this) to avoid use of PRINT statements. Output is now via calls to C routines set up to work with the routines BOBYQA, NEWUOA and UOBYQA.

Value

A list with components:

par	The best set of parameters found.
fval	The value of the objective at the best set of parameters found.
feval	The number of function evaluations used.
ierr	An integer error code. A value of zero indicates success. Other values (consistent with BOBYQA values) are **1** maximum number of function evaluations exceeded **3** a trust region step failed to reduce q (Consult Powell for explanation.)
msg	A message describing the outcome of UOBYQA

References

M. J. D. Powell, "The uobyqa software for unconstrained optimization without derivatives", in **Large-Scale Nonlinear Optimization**, Series: Nonconvex Optimization and Its Applications , Vol. 83, Di Pillo, Gianni; Roma, Massimo (Eds.) 2006, New York: Springer US.

M. J. D. Powell, "Developments of uobyqa for minimization without derivatives", IMA Journal of Numerical Analysis, 2008; 28: 649-664.

Description was taken from comments in the Fortran code of M. J. D. Powell on which **minqa** is based.

See Also

optim, nlminb

Examples

```
fr <- function(x) {   ## Rosenbrock Banana function
    100 * (x[2] - x[1]^2)^2 + (1 - x[1])^2
}
(x3 <- uobyqa(c(1, 2), fr))
## => optimum at c(1, 1) with fval = 0
# check the error exits
# too many iterations
x3e<-uobyqa(c(1, 2), fr, control = list(maxfun=50))
str(x3e)


# To add if we can find them -- examples of ierr = 3.
```

Index