Package ‘powell’

February 20, 2015

Title  Powell's UObyQA algorithm
Version  1.0-0
Author  Sundar Dorai-Raj <sdorairaj@gmail.com>, original Fortran from Mike Powell <mjdp@cam.ac.uk>
Description  Optimizes a function using Powell's UObyQA algorithm.
Maintainer  Sundar Dorai-Raj <sdorairaj@gmail.com>
License  GPL
NeedsCompilation  yes
Repository  CRAN
Date/Publication  2006-08-19 09:05:52

R topics documented:

  powell  ................................................................. 1
  powell.control  ....................................................... 3

Index  5

 powell   powell

Description

Optimizes a function using Powell's UObyQA algorithm.

Usage

  powell(par, fn, control = powell.control(), check.hessian = TRUE, ...)

1
Arguments
par Starting values for objective function
fn A function to be optimized. The function takes the parameters (par) as its first argument.
control A list of control parameters
check.hessian logical; if TRUE an eigenvalue decomposition is used to check the hessian for positive definiteness.
... Additional arguments to be passed to fn

Details
This function seeks the least value of a function of many variables, by a trust region method that forms quadratic models by interpolation. The algorithm is described in "UOBYQA: unconstrained optimization by quadratic approximation" by M.J.D. Powell, Report DAMTP 2000/NA14, University of Cambridge.

Value
A list with components
par The final values of the parameters.
value The final value of the function being optimized.
counts The number of times the function is called.
hessian A symmetric matrix of the estimated Hessian.
eigen.hessian If check.hessian is TRUE the eigenvalues and eigenvectors; otherwise NULL.
convergence 0 if converged, 1 otherwise.
control The input control parameters.
message Information about the model fit. This will be non-null only if check.hessian is TRUE and the resulting hessian is not positive definite.
call The original call to the optimizer.

Author(s)
Sundar Dorai-Raj

References
http://plato.asu.edu/topics/problems/nlounres.html

See Also
optim
Examples

```r
set.seed(1)
fn <- function(beta, y, x, w) {
  # binomial deviance using double log link
  mu <- exp(x %*% beta)
  loglik <- - y * mu + (w - y) * log(1 - exp(-mu))
  -2 * sum(loglik)
}
n <- 1000
beta <- c(-1, 0.5)
w <- rpois(n, 100)
x <- rep(c("A", "B"), length = n)
X <- model.matrix(~ x, data.frame(x))
y <- rbinom(n, w, exp(-exp(X %*% beta)))
x1 <- powell(beta, fn, y = y, x = X, w = w)
x2 <- optim(beta, fn, y = y, x = X, w = w, hessian = TRUE)
x3 <- glm(1 - y/w ~ x, data = data.frame(x, y, w),
  family = binomial("cloglog"), weights = w)
# compare coefficients from 3 fits
rbind(powell = x1$par, optim = x2$par, glm = coef(x3))
# compare standard errors from 3 fits
rbind(powell = sqrt(diag(2 * solve(x1$hessian)))/
  sqrt(diag(vcov(x3)))),
  optim = sqrt(diag(2 * solve(x2$hessian))/
  sqrt(diag(vcov(x3))))
```

Description

Controls convergence in `powell`.

Usage

```r
powell.control(trace = 0L, rhobeg = log(100)/10L, rhoend = 1e-04L,
  maxit = 1e+05L, fnscale = 1L, parscale = 1L, ...)```

Arguments

- **trace** an integer that determines the level of information produced at each step of the optimizer. The value of `trace` should be set to 0, 1, 2 or 3, which controls the amount of printing. Specifically, there is no output if `trace=0` and there is output only at the return if `trace=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 F with its variables are output if `trace=3`. 


* **rhobeg**
  About one tenth of the greatest expected change to a variable.

* **rhoend**
  Indicates the accuracy that is required in the final values of the variables.

* **maxit**
  The upper bound on the number of times \( fn \) is to be called.

* **fnscale**
  A scalar to divide the function value by. \( fnscale \) less than zero implies a maximization rather than a minimization.

* **parscale**
  A vector to divide the parameter values by. If only one value is given it is replicated to the length of \( par \).

* ... Additional arguments that will be ignored!

**Details**

* **rhobeg** and **rhoend** must be set to the initial and final values of a trust region radius, so both must be positive with \( \text{rhoend} \leq \text{rhobeg} \).

**Author(s)**

Sundar Dorai-Raj

**See Also**

* `powell`
Index

*Topic optimize
   powell, 1
   powell.control, 3

optim, 2

powell, 1, 4
powell.control, 3