Package ‘RcppNumerical’

Type Package
Title 'Rcpp' Integration for Numerical Computing Libraries
Version 0.4-0
Date 2019-12-01
Maintainer Yixuan Qiu <yixuan.qiu@cos.name>
Description A collection of open source libraries for numerical computing (numerical integration, optimization, etc.) and their integration with 'Rcpp'.
License GPL (>= 2)
Copyright See file COPYRIGHTS
URL https://github.com/yixuan/RcppNumerical
BugReports https://github.com/yixuan/RcppNumerical/issues
LazyData TRUE
Imports Rcpp
LinkingTo Rcpp, RcppEigen
Suggests knitr, rmarkdown, prettydoc, mvt, RcppEigen
VignetteBuilder knitr, rmarkdown
RoxygenNote 7.0.1
NeedsCompilation yes
Author Yixuan Qiu [aut, cre],
Ralf Stubner [ctb] (Integration on infinite intervals),
Sreekumar Balan [aut] (Numerical integration library),
Matt Beall [aut] (Numerical integration library),
Mark Sauder [aut] (Numerical integration library),
Naoaki Okazaki [aut] (The libLBFGS library),
Thomas Hahn [aut] (The Cuba library)
Repository CRAN
Date/Publication 2019-12-02 20:20:07 UTC

1
Description

`fastLR()` uses the L-BFGS algorithm to efficiently fit logistic regression. It is in fact an application of the C++ function `optim_lbfgs()` provided by `RcppNumerical` to perform L-BFGS optimization.

Usage

```r
fastLR(
  x, y,
  start = rep(0, ncol(x)),
  eps_f = 1e-08,
  eps_g = 1e-05,
  maxit = 300
)
```

Arguments

- `x` The model matrix.
- `y` The response vector.
- `start` The initial guess of the coefficient vector.
- `eps_f` Iteration stops if $|f - f'|/|f| < \epsilon_f$, where $f$ and $f'$ are the current and previous value of the objective function (negative log likelihood) respectively.
- `eps_g` Iteration stops if $||g|| < \epsilon_g \times \max(1, ||\beta||)$, where $\beta$ is the current coefficient vector and $g$ is the gradient.
- `maxit` Maximum number of iterations.

Value

`fastLR()` returns a list with the following components:

- `coefficients` Coefficient vector
- `fitted.values` The fitted probability values
- `linear.predictors` The fitted values of the linear part, i.e., $X\hat{\beta}$
- `loglikelihood` The maximized log likelihood
- `converged` Whether the optimization algorithm has converged
fastLR

Author(s)

Yixuan Qiu https://statr.me

See Also

glm.fit()

Examples

set.seed(123)
n = 1000p = 100x = matrix(rnorm(n * p), n)beta = runif(p)xβ = c(x %*% beta)p = 1 / (1 + exp(-xβ))y = rbinom(n, 1, p)

system.time(res1 <- glm.fit(x, y, family = binomial()))system.time(res2 <- fastLR(x, y))max(abs(res1$coefficients - res2$coefficients))
Index

* Topic models
  fastLR, 2
* Topic regression
  fastLR, 2

fastLR, 2

glm.fit, 3