Customization of the optimization algorithm

Marie Laure Delignette Muller, Christophe Dutang

November 2015

Contents

1 Customization of the optimization method used in `optim`

Each time a numerical minimization is carried out in the `fitdistrplus` package, the `optim` function of the **stats** package is used by default with the `"Nelder-Mead"` method for distributions characterized by more than one parameter and the `"BFGS"` method for distributions characterized by only one parameter. Sometimes the default algorithm fails to converge. It is then interesting to change some options of the `optim` function or to use another optimization function than `optim` to minimize the objective function. The argument `optim.method` can be used in the call to `fitdist` or `fitdistcens`. It will internally be passed to `mledist`, `mmedist`, `mgedist` or `qmedist`, and to `optim` (see `?optim` for details about the different algorithms available).

Even if no error is raised when computing the optimization, changing the algorithm is of particular interest to enforce bounds on some parameters. For instance, a volatility parameter σ is strictly positive $\sigma > 0$ and a probability parameter p lies in $p \in [0, 1]$. This is possible by using arguments `lower` and/or `upper`, for which their use automatically forces `optim.method="L-BFGS-B"`.

Below are examples of fits of a gamma distribution $\mathcal{G}(\alpha, \lambda)$ to the `groundbeef` data set with various algorithms. Note that the conjugate gradient algorithm (`"CG"`) needs far more iterations to converge (around 2500 iterations) compared to other algorithms (converging in less than 100 iterations).

```
library(fitdistrplus)
data("groundbeef")
fNM <- fitdist(groundbeef$serving, "gamma", optim.method = "Nelder-Mead")
fBFGS <- fitdist(groundbeef$serving, "gamma", optim.method = "BFGS")
fSANN <- fitdist(groundbeef$serving, "gamma", optim.method = "SANN")
fCG <- try(fitdist(groundbeef$serving, "gamma", optim.method = "CG",
  control = list(maxit = 10000)))
if(class(fCG) == "try-error")
  fCG <- list(estimate = NA)
```

2 User-supplied optimization algorithm

2.1 Naming convention of optimization arguments

It is also possible to use another function than `optim` to minimize the objective function by specifying by the argument `custom.optim` in the call to `fitdist`. It may be necessary to customize this optimization function to meet the following requirements. (1) `custom.optim` function must have the following arguments: `fn} for the function to be optimized and`par`for the initialized parameters.` (2)custom.optim`should carry out a MINIMIZATION and must return the following components:`par`for the estimate,`convergence`for the convergence code,`value=fn(par)`and`hessian.` `Below is an example of code written to wrap the`genoud‘ function from the **rgenoud** package in order to respect our optimization “template”.

2.2 Example with `genoud`

The **rgenoud** package implements the genetic (stochastic) algorithm.

```
mygenoud <- function(fn, par, ...)
{
   require(rgenoud)
   res <- genoud(fn, starting.values = par, ...)
   standardres <- c(res, convergence = 0)
   return(standardres)
}
```

The customized optimization function can then be passed as the argument `custom.optim` in the call to `fitdist` or `fitdistcens`. The following code can for example be used to fit a gamma distribution to the `groundbeef` data set. Note that in this example various arguments are also passed from `fitdist` to `genoud` : `nvars`, `Domains`, `boundary.enforcement`, `print.level` and `hessian`. The code below compares all the parameter estimates ($\hat{\alpha}$, $\hat{\lambda}$) by the different algorithms: shape α and rate λ parameters are relatively similar on this example, roughly 4.00 and 0.05, respectively.

```
fgenoud <- mledist(groundbeef$serving, "gamma", custom.optim = mygenoud,
  nvars = 2, max.generations = 10, Domains = cbind(c(0,0), c(10,10)),
  boundary.enforcement = 1, hessian = TRUE, print.level = 0, P9 = 10)
```

```
## Loading required package: rgenoud
```

```
## ##  rgenoud (Version 5.7-12.4, Build Date: 2015-07-19)
## ##  See http://sekhon.berkeley.edu/rgenoud for additional documentation.
## ##  Please cite software as:
## ##   Walter Mebane, Jr. and Jasjeet S. Sekhon. 2011.
## ##   ``Genetic Optimization Using Derivatives: The rgenoud package for R.''
## ##   Journal of Statistical Software, 42(11): 1-26.
## ##
```

```
cbind(NM = fNM$estimate,
  BFGS = fBFGS$estimate,
  SANN = fSANN$estimate,
  CG = fCG$estimate,
  fgenoud = fgenoud$estimate)
```

```
##               NM       BFGS       SANN         CG    fgenoud
## shape 4.00825257 4.22848161 4.04084868 4.12850423 4.00834314
## rate  0.05441911 0.05742147 0.05476672 0.05606164 0.05442741
```

References