Global Optimization in R

Katharine Mullen
kmmullen@ucla.edu

November 27, 2012
Introduction to R

What is R?

- a freely available language and environment for statistical computing and graphics
- free software, a GNU project distributed under the General Public License
- runs on a wide variety of platforms: Linux, Solaris, Mac OS, MS Windows, etc.
- 4150 user-contributed packages now available
- distributed by the Comprehensive R Archive Network (CRAN)
What's it like to use? An Example

```r
> ## simulate a matrix A
> ## with 3 columns, each containing
> ## an exponential decay
> t <- seq(0, 2, by = .04)
> k <- c(.5, .6, 1)
> A <- matrix(nrow = 51, ncol = 3)
> Acolfunc <- function(k, t) exp(-k*t)
> for(i in 1:3) A[,i] <- Acolfunc(k[i],t)
> jpeg("exp.jpg")
> par(cex.lab=1.5, cex.axis=2)
> matplot(t, A, type="l", lty=1, lwd=5)
> dev.off()
```
What's it like to use? An Example

```r
## simulate a matrix X
## with 3 columns, each containing a Gaussian shape
## the Gaussian shapes are non-negative
X <- matrix(nrow = 51, ncol = 3)
wavenum <- seq(18000, 28000, by=200)
location <- c(25000, 22000, 20000)
delta <- c(3000, 3000, 3000)
Xcolfunc <- function(wavenum, location, delta)
+   exp( - log(2) * (2 * (wavenum - location)/delta)^2)
> for(i in 1:3) X[,i] <- Xcolfunc(wavenum, location[i], delta[i])
>
jpeg("gaus.jpg")
> par(cex.lab=1.5, cex.axis=2)
> matplot(wavenum, X, type="l", lty=1, lwd=5, ylab="amplitude",
+     xlab=expression(paste("wavenumber (nm"^"{-1},")")))
> dev.off()
```

![Graph showing three Gaussian shapes with different wavenumbers.](gaus.jpg)
What's it like to use? An Example

```r
## install and load
## add-on packages
>
> install.packages("fields")
> library("fields")
>
> install.packages("nnls")
> library("nnls")

> ## set seed for reproducibility
> set.seed(3300)
>
> ## simulated data is the matrix product of A and X
> ## with spherical Gaussian noise added
> matdat <- A %*% t(X) + .005 * rnorm(nrow(A) * nrow(X))
>
> jpeg("data.jpg")
> par(cex.lab=1.5, cex.axis=2, mar=c(5, 5, 4, 2))
> image.plot(t, wavenum, matdat,
+ ylab=expression(paste("wavenumber (nm"^-1,"),"))),
+ xlab="time (ns)"
> dev.off()
```
What's it like to use? An Example

```r
> ## estimate the rows of X using NNLS criteria
> nnls_sol <- function(matdat, A) {
+   X <- matrix(0, nrow = 51, ncol = 3)
+   for(i in 1:ncol(matdat))
+     X[i,] <- coef(nnls(A,matdat[,i]))
+   X
+ } 
> X_nnls <- nnls_sol(matdat,A)  
> jpeg("result.jpg") 
> par(cex.lab=1.5, cex.axis=2) 
> matplot(wavenum, X_nnls,type="b",pch=20,
+     lwd=5, ylab="amplitude",
+     xlab=expression(paste("wavenumber (nm"^{-1},
+     "
+     ")))
+ )
> abline(0,0, col=grey(.6))
> dev.off()
```
Past: Roots in S

- S began at Bell Labs by John Chambers, Rick Becker, Douglas Dunn, Paul Tukey, Graham Wilkinson...
- AT&T releases Unix in portable form in 1978
- S Version 1 (1976-1978) implementation nearly all Fortran-based, only for Bell-proprietary operating system
- S Version 2 released outside of Bell Labs from ~1981, with books appearing in 1984/85
- S software licensed (1993) and eventually sold to Insightful Corp. (2004)
Past: Roots in S


- goal of the S language, as stated in Chambers' Green Book: To turn ideas into software, quickly and faithfully

- Chambers wins 1998 Association for Computing Machinery (ACM) Software System Award for creating the “S System” software

- The ACM's citation notes that Chambers' work "will forever alter the way people analyze, visualize, and manipulate data"

- S is an interactive environment for data analysis and graphics, a system within which to do statistics, not 'just' a language

...meanwhile, in Auckland
Past: R is Born

- Ross Ihaka & Robert Gentleman of Auckland University wrote an experimental R, “not unlike S”, and announced R on the S-news mailing list in 1993
- Martin Maechler convinced them to release it as an open-source project under the GNU General Public License in 1995
- ‘R Core’, a group of people with write-access to the base R code is established in 1997
- Directions in Statistical Computing (DSC) meeting in Vienna, first R core meeting in 1999
Past: R becomes 'lingua franca'

- R 1.0.0 is released in 2000; 2 major releases per-year becomes standard (current version is 2.1.9)
- R Newsletter launched in 2001; becomes the R Journal in 2009
- R Foundation established in 2002 (non-profit based in Vienna, Austria)
- First useR! conference in Vienna, Austria in 2004
- >115 books published on R circa 2012
- Springer UseR Series has 39 titles circa 2012
- R-help and R-devel mailing lists regularly get >100 messages per-day
- Statistics being taught in R in large number of departments
- R implements much of modern statistics
“R is really important to the point that it’s hard to overvalue it,” said Daryl Pregibon, a research scientist at Google, which uses the software widely. “It allows statisticians to do very intricate and complicated analyses without knowing the blood and guts of computing systems.”

--- New York Times, January 6, 2009
Present: R is 'lingua franca'

Speaking of R, Mr. Chambers said, “It’s way beyond anything we could have imagined at Bell Labs.”

--- New York Times, January 8, 2009
The R User Conference, useR!

- useR! 2004 in Vienna, Austria
- useR! 2006 in Vienna, Austria
- useR! 2007 in Ames, Iowa, USA
- useR! 2008 in Dortmund, Germany
- useR! 2009 in Rennes, France
- useR! 2010 in Gaithersburg, Maryland, USA (at NIST)
- useR! 2011 in Warwick, UK
- useR! 2012 in Nashville, USA
- useR! 2013 to be held at Albacete, Spain
JSS was established in 1996 by Jan de Leeuw

- presents research that demonstrates the joint evolution of computational and statistical methods and techniques
- articles are available along with the code they discuss
- is an open-access
- 556 articles, 37 code snippets, 106 book reviews, 6 software reviews, and 14 special volumes published to date
- is premier place to publish descriptions of new R packages

Katharine Mullen  kmullen@ucla.edu

Global Optimization in R
Sources for information on optimization in R:

- the CRAN Task View on Optimization and Mathematical Programming, which lists packages available for optimization in R
- a variety of JSS and R Journal papers
- soon, a new special volume of JSS to which I am contributing a paper titled 'Global Optimization in R'

Katharine Mullen  kmmullen@ucla.edu  Global Optimization in R
linear and nonlinear regression (that is, minimization of the sum of the squared difference between model and data) has always been the bread and butter of R

regression typically relies on gradient-based local algorithms

even five years ago the number of global optimization algorithms in R was rather limited

now, a garden of different algorithms have sprouted up
Global optimization problems

- occur in a wide variety of applications, from fitting models to scientific datasets to finance
- canonical example of a *combinatorial* global optimization problem: traveling salesman problem
- *combinatorial problems* have a finite number of possible solutions
- *continuous problems* have the infinite set of vectors $\mathbf{x} \in \mathbb{R}^n$ (or a subset thereof) as possible solutions. The optimization problem is then often formulated as

$$\min_{\mathbf{x}} \ f(\mathbf{x})$$

where $f$ is a $\mathbb{R}^n \to \mathbb{R}$ *objective function*
Global optimization

- properties of the objective function $f$ may allow efficient solution of global optimization problem, e.g.,
  - \textit{continuity}: small changes in $\mathbf{x}$ translate to small changes in $f(\mathbf{x})$.
  - \textit{differentiability}: at every point $x \in \mathbb{R}^n$ the partial derivatives of $f(\mathbf{x})$ exist.
  - a \textit{limited number of local optima}
Constrained global optimization problems seek a solution $\mathbf{x}$ that minimizes an objective function $f$ such that $\mathbf{x}$ satisfies a set of constraints, typically inequalities or equalities.

- **Box constraints** provide lower and upper bounds $\mathbf{l} \in \mathbb{R}^n$ and $\mathbf{u} \in \mathbb{R}^n$ on $\mathbf{x}$.

- Constrained global optimization programs are often formulated as *mathematical programming problems*, often stated in the standard form

\[
\begin{align*}
\text{minimize} & \quad f(\mathbf{x}) \\
\text{subject to} & \quad h_i(\mathbf{x}) = 0, \ i = 1, \ldots, p \\
& \quad g_j(\mathbf{x}) \leq 0, \ j = 1, \ldots, q
\end{align*}
\]

where $h$ and $g$ are functions that may be nonlinear.
Integer programming problems

- Integer programming problems

\[
\min \limits_{\bar{x}} \quad f(\bar{x}) = c^T x \\
\text{subject to} \quad Ax \leq b \\
E x = d
\]

with a parameter space comprised of integer vectors \( \bar{x} \in I \), require exponential time algorithms to solve.

- the traveling salesman problem and other difficult combinatorial optimization problems are possible to translate into equivalent integer programming problems.

- mixed integer programming problem: some elements of the parameter vector must be integer while other elements are reals.
Categories of implementations in R

- Annealing methods
- Evolutionary methods
- Particle swarm optimization methods
- Combinatorial and integer optimization methods
- Branch and bound methods
- Miscellaneous stochastic methods

Some of these methods are *heuristic*, meaning that they are likely but not guaranteed to find the optimal solution. Nonheuristic methods, in contrast, may maintain a provable upper and lower bound on the globally optimal objective value.
Annealing methods

Stochastic, heuristic methods that starts at a randomized point $\tilde{x}$ in the parameter space, and then evaluates a neighboring point $(\tilde{x})'$, usually chosen at random. If the value of the objective function $f((\tilde{x})') < f(\tilde{x})$, the new point $x'$ is chosen with some acceptance probability $P$ less than unity.

Implementations in R include:

- optim function
- rneos package: adaptive simulated annealing
- GenSA package
Evolutionary methods

Evolutionary algorithms are stochastic, heuristic methods inspired by the process of natural selection. They maintain a population of candidate solutions, and generate successive generations of the population using operations with an analogue to sexual recombination and mutation.

Implementations in R include:
- rgenoud package
- DEoptim
- NMOF package: DEopt
- soma package
- Rmalschains package
- cmaes package
- genopt function
- GA package
- nloptr package: ISRES
- rneos package: PGAPack
Particle swarm optimization methods

Particle swarm optimization is a stochastic, heuristic method similar to an evolutionary method in that a population (swarm) of candidate solutions (particles) is moved through search space. The movement of particles during each generation is influenced by the state of the rest of the swarm. Implementations in R include:

- pso package
- ppso package
- rneos package: PSwarm
Combinatorial and integer optimization methods apply to objective functions $f : I^n \to \mathbb{R}$ and $f : I^n \to I$.

Implementations in R include:

- TSP package
- rneos package: BARON
Branch and bound is a systematic, nonheuristic method for solving optimization problems, and is the most widely used type of algorithm for solving difficult combinatorial optimization problems. It often leads to exponential time complexities in the worst case. Most algorithms in this category apply a breadth-first search for the optimal solution, but not all nodes get expanded. Implementations in R include:

- rneos package: BARON
- rneos package: ICOS
- rneos package: Lindo Global
- nloptr package: StoGo
- nloptr package: DIRECT
These are stochastic, heuristic methods that do not fall into the preceding categories.

Implementations in R include:

- nloptr package: CRS
- NMOF package: TAopt
- NMOF package: LSopt
- nloptr package: MLSL
For the purposes of computation comparison, selected 13 functions for continuous global optimization. These methods:

- work out-of-the-box (that is, without installing anything other than R packages)
- do not require anything more than (possibly) a starting vector, bound constraints, and the objective function
- allow passing arguments in addition to the parameter vector to the objective function

Performance was benchmarked in terms of the quality of the solution reached after a set number of evaluations of the objective function.
Benchmarking performance

For comparison study:

- created a new package `globalOptTests`
2-parameter Dekkers and Aarts Problem

Katharine Mullen  kmmullen@ucla.edu  Global Optimization in R
10-parameter Ackley's Problem

Katharine Mullen  kmmullen@ucla.edu
Global Optimization in R
5–parameter Epistatic Michalewicz Problem
if you have a global optimization problem, use R!

look for my paper ’Global Optimization in R’ sometime next year in JSS