How I Learned to Stop Worrying and Love optim

Dale W.R. Rosenthal

23 Jan 2019

Q36

Introduction

Me:

- Head partner, Q_{36}: quant finance firm. (Here! Welcome!)
- PhD, Statistics, U. Chicago; BS, Elec. Eng., Cornell.
- Was: finance professor at UIC; now teach finance at UIUC.

What I want to talk about:

- Economics studies *rational, optimizing agents*.
- Statisticians maximize likelihood functions to fit models.
- Operations researchers optimize portfolios, routes, inventory, etc.
- But. . . how well do you know how to optimize?

Q_{36}

Optimizing: Terminology

- In optimization we first define an *objective function* f:
  - Some metric we want to minimize (to maximize: $-1 \times f$).
- Goal: find values of *decision variables* which minimize f.
- We may have *constraints* we impose on decision variables.
- We then use a *solver* to handle our type of problem.
- It turns out *R* has a nice function to do optimization.
- `optim()` implements a few handy solvers
- Some solvers handle *box constraints* (*e.g.* $a \leq x \leq b$, $y \geq c$)

Q_{36}

Optimus Prime: Preparing for optim()

- Using optim() is very easy!
- The hardest work: creating an objective function.
- OLS, for example: fit $y = \alpha + \beta x + \epsilon$.

```
linear.regression.sse <- function(theta) {
  alpha <- theta[1]
  beta <- theta[2]
  residuals <- y - (alpha + beta*x)
  sum(residuals^2)
}
```

- Then just call optim() like so:

```
start.val <- c(0,0)
fit.ols <- optim(start.val, linear.regression.sse)
fit.ols$par  # coefficient estimates
```

Q_{36}

Getting Your `optim()` Mojo Working

- Note that we can specify other objectives. For example:
  - Optimizing `max(residuals^2)` tries to make all errors equal.
- We can use solvers other than the default (Nelder-Mead):
  - BFGS estimates curvature to speed solution.
  - CG uses approximate gradients; good for larger problems; can be fussy.
- L-BFGS-B solver allows constraints. *e.g.* LS but $\hat{\beta} \geq 0$:

```
fit.cons <- optim(start.val, linear.regression.sse,
                  lower=c(-Inf,0))
```

- Can also get Hessian (Fisher info!) to find estimate standard errors:

```
fit.ols <- optim(start.val, linear.regression.sse, hessian=TRUE)
std.errors <- sqrt(diag(solve(fit.ols$hessian)))
```

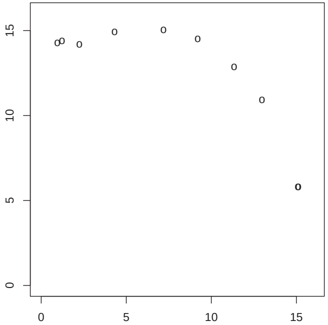
Q_{36}

Example: Center of an Imperfect Circle

- Suppose: find 10 stones near a cliff; think they formed a circle.

x	0.95	12.98	15.12	4.32	2.25	1.23	15.09	7.19	9.20	11.34
y	14.27	10.91	5.79	14.91	14.18	14.40	5.81	15.03	14.52	12.85

- Want to find the center of the circle and its radius.
- No idea about circumferential errors; assume only errors in radius.



Q_{36}

Solving: Center of an Imperfect Circle

Now we see how to set up and use optim().

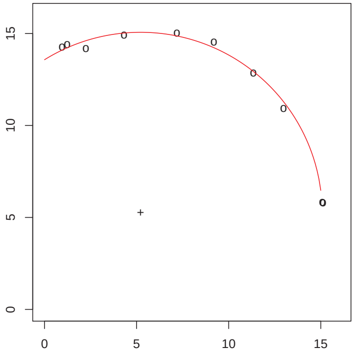
```
x <- c(0.95,12.98,15.12,4.32,2.25,1.23,15.09,7.19,9.20,11.34)
y <- c(14.27,10.91,5.79,14.91,14.18,14.40,5.81,15.03,14.52,12.85)
radialSSE <- function(theta) {
  c.x <- theta[1]
  c.y <- theta[2]
  radius <- theta[3]
  radial.error <- sqrt((x-c.x)^2 + (y-c.y)^2) - radius
  sum(radial.error^2)
}
start.values <- c(7, 7, 8)  # not sensitive to these
fit.circle <- optim(start.values, radialSSE, hessian=TRUE)
fit.circle$par  # estimated center (x,y) and radius
sqrt(diag(solve(fit.circle$hessian)))  # estimate standard errors
```

Q_{36}

Estimated Center of an Imperfect Circle

Using `optim()` gives us these estimates and standard errors:

Variable	x	y	radius
Estimate	5.22	5.20	9.86
Std Error	0.70	1.09	0.97



- As *always*: should look into model diagnostics, residuals.
  - The doubled observations near $x = 1$ and $x = 15$ seem odd.
  - Repeated measurements? Problems near site edges?

Q36

Going Further

- For more information on optimization:
  - Nocedal and Wright, *Numerical Optimization*
  - Cornuéjols, Peña, & Tütüncü, *Optimization Methods in Finance*
- Can use `rneos` package for more solvers, bigger problems.
- Intrepidly quantitative? Interested in finance?
  - My book: *A Quantitative Primer on Investments with R*