
Subject: Re: Bootstrapping / Resampling

Posted by [Amara Graps](#) on Mon, 11 Sep 2000 11:39:33 GMT

[View Forum Message](#) <> [Reply to Message](#)

Dear Fraser,

- > I am using a routine to fit a circle to a set of edge detected points.
- > This gives me the radius R0 and the centre point (X0,Y0).
- >
- > I would like to obtain an estimate of the error/uncertainty in these fit
- > parameters. I understand that Bootstrapping or resampling may be away to
- > estimate these uncertainties.
- >
- > 1. Are Bootstrapping or Resampling the techniques to use ?

There's a large body of literature, especially in the computer graphics field and astronomy field, for fitting circles (ellipses) to points.

I think that bootstrapping is overkill for what you want. I suggest a linear or nonlinear fit, and using the error that naturally pops out of the least-squares fit.

For example, let's say that you have 5 points to which you want fit a circle.

Main Equation to Use:

Equation for a circle:

$$(x-h)^2 + (y-k)^2 = r^2$$

So, calculate the radius of the circle from:

$$r = \text{SQRT}[(x-h)^2 + (y-k)^2] \quad \text{Eqn(1)}$$

where:

(h,k) = center of circle

(x,y) = point coordinates

r = radius

Given:

5 Points: (x1,y1); (x2,y2); (x3,y3); (x4,y4); (x5,y5)

Want to Find for Best-Fit Circle:

(h,k), r

Procedure:

1) Get first estimate for (h,k,r).

(Find some points to make first estimates- either solve the circle exactly (3 equations, 3 unknowns) to get a first estimate of the center and radius, or just do a kind of centroid calculation on all 5 points- to get a rough estimate of the center and radius.)

2) Calculate (from Eqn(1)) r1, r2, r3, r4, r5 for each of your five points.

3) Calculate the root-mean-squared error:

$$\text{RMS error} = \text{SQRT}[[(r1-r)^2 + (r2-r)^2 + (r3-r)^2 + (r4-r)^2 + (r5-r)^2] / 3]$$

(dividing by "3" because we have 3 free parameters.)

4) Change (h,k,r) slightly in your minimization algorithm to try a new (h,k,r).

5) Calculate r1, r2, r3, r4, r5 again from new (h,k,r) above.

6) Calculate RMS error again.

7) Compare current RMS error with previous RMS error. If it doesn't vary by more some small amount, say 10^{-3} then you're done, otherwise continue steps 4 -- 7.

BTW, I would look for a "canned" minimization routine. Some commercial computer programs for plotting and spreadsheets do this sort of thing, such as Excel. The Excel spreadsheet has a built-in "solver" that will perform minimization. I've used it, and it works pretty well for simple problems- it used Newton's method to search, tangents for estimates, and forward differences for derivatives.

Some years ago I wrote the FAQ entry for the sci.math.numerical-analysis newsgroup for "Fitting a circle to a set of points", because it is a question that occurs several times a year on that newsgroup, and I had a good collection of answers from various people for the question.

Since my old FAQ entry needed to be updated, I used this opportunity (your question) to update it, and condense my haphazardly collected notes from the last few years. This way I can give something back to the newsgroup too. This updated entry will (eventually!) appear at

<http://www.mathcom.com/nafaq/index.html>
q230.8

Hope it's useful,

Amara

=====FAQ entry=====

Problem: Given N Points: (x1,y1); (x2,y2).. (xN,yN), we want to find for best-fit circle: (X0,Y0), R. (Note: for fitting an *ellipse*, substitute the equation for an ellipse for the equation for a circle in the "Brute Force Approach").

Brute Force Approach (leads to a non-linear system): [Amara Graps]

Idea: Minimize by least squares the root-mean-squared error of the radius in the equation for a circle. In this method, one minimizes the sum of the squares of the *length* differences, which gives you a non-linear problem with all the associated pitfalls.

$(x-X0)^2 + (y-Y0)^2 = R^2$ Equation for a circle
 $R = \text{SQRT}[(x-X0)^2 + (y-Y0)^2]$ Radius of the circle

where:

(X0,Y0) = center of circle
(x,y) = point coordinates
R = radius

1) Get first estimate for (X0,Y0,R).

(Details: Find some points to make first estimates- either solve the circle exactly (3 equations, 3 unknowns) to get a first estimate of the center and radius, or just do a kind of centroid calculation on all points- to get a rough estimate of the center and radius.)

2) Calculate r (r1, r2,.. rN) for each of your N points from the equation

for a radius of a circle.

3) Calculate the root-mean-squared error

For example, for 5 given points on the circle:

$$\text{RMS error} = \text{SQRT} [[(r1-R)^2 + (r2-R)^2 + (r3-R)^2 + (r4-R)^2 + (r5-R)^2] / 5]$$

(dividing by "5" because we have 5 free parameters.)

4) Change (X0,Y0,R) slightly in your minimization algorithm to try a new (X0,Y0,R).

MINIMIZATION DETAILS.

Because minimization algorithms can get very computationally intensive, if one's circle-fitting problem is simple, I would look for a "canned" minimization routine. Some commercial computer programs for plotting and spreadsheets do this sort of thing. For example, the Excel spreadsheet has the built-in "Solver" that will perform minimization.

Other possibilities for software.

Matlab with the optimization toolbox.

IDL with Markwardt's MPFIT

(<http://cow.physics.wisc.edu/~craigm/idl/idl.html>)

or MACSYMA. [Amara Graps]

ODRPACK from Netlib is an excellent library for a production mode approach to fitting these data, with a graphical user interface at <http://plato.la.asu.edu/guide.html> At the very beginning of the User's Guide there is a description of how to fit data to an ellipse or circle. [Don Wells, Hans D. Mittelmann]
<http://plato.la.asu.edu/topics/problems/nlolsq.html>

Gaussfit is a superb response to the whole family of problems of adjusting parameters to fit functions to data; it is especially effective and easy to use for one-time problems and for exploratory work [Don Wells].

The L-BFGS-B Nonlinear Optimization package allows one to specify bounds on the variables, from:
<http://www.ece.nwu.edu/~nosedal/lbfgsb.html>
[Helmut Jarausch].

The Fortran programs `fcircle.f`, `fellipse.f` and the Lawson & Hanson

least-squares routines ls.f shows how to implement these
least-squares problems: <ftp://obsftp.unige.ch/fit>

Alan Miller has updated the above for Fortran 90.
The least-squares software is at
<http://www.ozemail.com.au/~milleraj/>
with a working example:
<http://www.ozemail.com.au/~milleraj/lsq/fellipse.f90>

Some Web Guides

GraphPad Guid to Nonlinear Regression
<http://www.graphpad.com/www/nonling1.htm>
[Amara Graps]

User's Manual -- GaussFit: {A} system for least squares and robust
estimation", William H. Jefferys and Michael J. Fitzpatrick and
Barbara E. McArthur and James E. McCartney",
<ftp://clyde.as.utexas.edu/pub/gaussfit/> [Don Wells]

Other Papers

I. Kasa, "A Curve Fitting Procedure and Its Error Analysis", IEEE
Trans. on Inst. Meas., 3/96, describes an algorithm that is a
modified least squares analysis that uses the difference of the
square of the radii rather than the absolute difference.
[Rod Loewen]

M. Pitu, A. Fitzgibbon, R. Fisher, "Ellipse-specific Direct
least-square Fitting", IEEE International Conference on Image
Processing, Lausanne, September 1996.
<http://vision.dai.ed.ac.uk/maurizp/EllFitDemo/Paper/icip.html> A
Java demo is at:
<http://hplbwww.hpl.hp.com/people/mp/research/ellipse.htm>
[Amara Graps]

Moura, L. and Kitney, R. I. (1991). A direct method for least
-squares circle fitting. Computer Physics Communications 64, 57-63.
[Colin B. Desmond]

"A Buyers Guide to Conic Fitting", A.W. Fitzgibbon, British Machine
Vision Conference, 1995.

Cunningharn, Robert W.: Comparison of three methods for determining
fit parameter uncertainties for the Marquardt Compromise, Computers
in Physics 7(5), 570, (1993) [Amara Graps]

W. Gander, G. H. Golub and R. Strebel Least Squares Fitting of

Circles and Ellipses BIT vol.34, 1994, pgs. 558-578 [Suhail A Islam].

Berman & Somlo: "Efficient procedures for fitting circles..." IEEE Instrum & Meas., IM-35, no.1, pp. 31-35, 1986. [Peter Somlo]

Paul T. Boggs and Richard H. Byrd and Robert B. Schnabel, "A Stable and efficient algorithm for nonlinear orthogonal distance regression, SIAM Journal on Scientific and Statistical Computing, 1987, 8, 6, 1052-1078 [Don Wells].

P. T. Boggs and J. R. Donaldson and R. H. Byrd and R. B. Snabel, "Algorithm 676, {ODRPACK} -- Software for Weighted orthogonal distance regression", ACM Transactions On Mathematical Software, 1989, 15, 348-364 [Don Wells].

5) Calculate r (r1, r2 etc.) again from new (X0,Y0,R) above.

6) Calculate RMS error again.

7) Compare current RMS error with previous RMS error. If it doesn't vary by more some small amount, say 10^{-3} then you're done, otherwise continue steps 4 -- 7.

Other (more elegant) approaches that reduce to a linear system.

If you choose to minimize the squares of the *area* differences, you get a linear problem, which is a much safer way. [Pawel Gora, Zdislav V. Kovarik, Daniel Pfenniger, Condensed by Amara Graps]

1. Function to minimize is (sum of the area differences):

$$Q = \text{Sum} \{ [(x_i - X_0)^2 + (y_i - Y_0)^2 - R^2]^2, i=1..N \}$$

2. A necessary condition is the system of 3 equations with 3 unknowns X_0, Y_0, R . Calculate the partial derivatives of Q, with respect to X_0, Y_0, R . (all d's are partial)

$$dQ / dX_0 = 0$$

$$dQ / dY_0 = 0$$

$$dQ / dR = 0$$

3. Developing we get the linear least-squares problem:

$$\begin{bmatrix} x_1 & y_1 & 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = -x_1^2 - y_1^2$$

```

| x2 y2 1 | | b | =~ | -x2^2-y2^2 |
| x3 y3 1 | | c |   | -x3^2-y3^2 |
| x4 y4 1 |         | -x4^2-y4^2 |
| x5 y5 1 |         | -x5^2-y5^2 |

```

.....
(for example, for 5 points)

where $a = -2 X_0$; $b = -2 Y_0$; $c = X_0^2 + Y_0^2 - R^2$.

Take any good least-squares algorithm to solve it, yielding a,b,c.

So the final circle solution will be given with

$X_0 = -a/2$; $Y_0 = -b/2$; $R^2 = X_0^2 + Y_0^2 - c$.

By the way, with 5 points you can also find the unique quadratic form (ellipse, hyperbola) which passes exactly through 5 points.

With more than 5 points one can do a linear least-squares as well.

The problem is then to minimize:

```

| x1^2-y1^2 x1y1 x1 y1 1 | | a |   | -x1^2-y1^2 |
| x2^2-y2^2 x2y2 x2 y2 1 | | b | =~ | -x2^2-y2^2 |
| x3^2-y3^2 x3y3 x3 y3 1 | | c |   | -x3^2-y3^2 |
| x4^2-y4^2 x4y4 x4 y4 1 | | e |   | -x4^2-y4^2 |
| x5^2-y5^2 x5y5 x5 y5 1 | | f |   | -x5^2-y5^2 |

```

.....

The robust or L1 or least-first-power approximation [Zdislav V. Kovarik].

If you try to minimize

$W(a,b,r) = \text{SUM}(j=1:N) \text{ABS}(((x_j-a)^2 + (y_j-b)^2)^{(1/2)} - r)$

all you have to do is set up the 10 (i.e. if 5, choose 3) circles passing

through every choice of 3 of 5 points, calculate $W(a,b,r)$ for each of them and pick the minimizing circle. The extra feature is that this procedure separates and discards points which are conspicuously far from the majority trend. (Of course, it becomes increasingly time-consuming when the number of given points increases.)

This method of determining the minimum bounding circle from a set of `_circles_` is solved, and with code available at:
<http://www.intergalact.com/circles.html> [Sky Coyote]

--

Amara Graps | Max-Planck-Institut fuer Kernphysik
Interplanetary Dust Group | Saupfercheckweg 1
+49-6221-516-543 | 69117 Heidelberg, GERMANY
* <http://galileo.mpi-hd.mpg.de/~graps>

"Never fight an inanimate object." - P. J. O'Rourke
