Wikipedia
The Free Encyclopedia

Quasi-Newton method

Quasi-Newton methods are methods used to either find zeroes or local maxima and minima of functions, as an alternative to Newton's method. They can be used if the Jacobian or Hessian is unavailable or is too expensive to compute at every iteration. The "full" Newton's method requires the Jacobian in order to search for zeros, or the Hessian for finding extrema. Some iterative methods that reduce to Newton's method, such as SLSQP, may be considered quasi-Newtonian.

Search for zeros: root finding

Newton's method to find zeroes of a function g of multiple variables is given by $x_{n+1} = x_n - [J_g(x_n)]^{-1} g(x_n)$, where $[J_g(x_n)]^{-1}$ is the left inverse of the Jacobian matrix $J_g(x_n)$ of g evaluated for x_n.

Strictly speaking, any method that replaces the exact Jacobian $J_g(x_n)$ with an approximation is a quasi-Newton method.[1] For instance, the chord method (where $J_g(x_n)$ is replaced by $J_g(x_0)$ for all iterations) is a simple example. The methods given below for optimization refer to an important subclass of quasi-Newton methods, secant methods.[2]

Using methods developed to find extrema in order to find zeroes is not always a good idea, as the majority of the methods used to find extrema require that the matrix that is used is symmetrical. While this holds in the context of the search for extrema, it rarely holds when searching for zeroes. Broyden's "good" and "bad" methods are two methods commonly used to find extrema that can also be applied to find zeroes. Other methods that can be used are the column-updating method, the inverse column-updating method, the quasi-Newton least squares method and the quasi-Newton inverse least squares method.

More recently quasi-Newton methods have been applied to find the solution of multiple coupled systems of equations (e.g. fluid–structure interaction problems or interaction problems in physics). They allow the solution to be found by solving each constituent system separately (which is simpler than the global system) in a cyclic, iterative fashion until the solution of the global system is found.[2][3]

Search for extrema: optimization

The search for a minimum or maximum of a scalar-valued function is nothing else than the search for the zeroes of the gradient of that function. Therefore, quasi-Newton methods can be readily applied to find extrema of a function. In other words, if g is the gradient of f, then searching for the zeroes of the vector-valued function g corresponds to the search for the extrema of the scalar-valued function f; the Jacobian of g now becomes the Hessian of f. The main difference is that the Hessian matrix is a symmetric matrix, unlike the Jacobian when searching for zeroes. Most quasi-Newton methods used in optimization exploit this property.

In optimization, **quasi-Newton methods** (a special case of **variable-metric methods**) are algorithms for finding local maxima and minima of functions. Quasi-Newton methods are based on Newton's method to find the stationary point of a function, where the gradient is 0. Newton's method assumes that the function can be locally approximated as a quadratic in the region around the optimum, and uses the first and second derivatives to find the stationary point. In higher dimensions, Newton's method uses the gradient and the Hessian matrix of second derivatives of the function to be minimized.

In quasi-Newton methods the Hessian matrix does not need to be computed. The Hessian is updated by analyzing successive gradient vectors instead. Quasi-Newton methods are a generalization of the secant method to find the root of the first derivative for multidimensional problems. In multiple dimensions the secant equation is under-determined, and quasi-Newton methods differ in how they constrain the solution, typically by adding a simple low-rank update to the current estimate of the Hessian.

The first quasi-Newton algorithm was proposed by William C. Davidon, a physicist working at Argonne National Laboratory. He developed the first quasi-Newton algorithm in 1959: the DFP updating formula, which was later popularized by Fletcher and Powell in 1963, but is rarely used today. The most common quasi-Newton algorithms are currently the SR1 formula (for "symmetric rank-one"), the BHHH method, the widespread BFGS method (suggested independently by Broyden, Fletcher, Goldfarb, and Shanno, in 1970), and its low-memory extension L-BFGS. The Broyden's class is a linear combination of the DFP and BFGS methods.

The SR1 formula does not guarantee the update matrix to maintain positive-definiteness and can be used for indefinite problems. The Broyden's method does not require the update matrix to be symmetric and is used to find the root of a general system of equations (rather than the gradient) by updating the Jacobian (rather than the Hessian).

One of the chief advantages of quasi-Newton methods over Newton's method is that the Hessian matrix (or, in the case of quasi-Newton methods, its approximation) B does not need to be inverted. Newton's method, and its derivatives such as interior point methods, require the Hessian to be inverted, which is typically implemented by solving a system of linear equations and is often quite costly. In contrast, quasi-Newton methods usually generate an estimate of B^{-1} directly.

As in Newton's method, one uses a second-order approximation to find the minimum of a function $f(x)$. The Taylor series of $f(x)$ around an iterate is

$$f(x_k + \Delta x) \approx f(x_k) + \nabla f(x_k)^{\mathrm{T}}\, \Delta x + \frac{1}{2} \Delta x^{\mathrm{T}} B \, \Delta x,$$

where (∇f) is the gradient, and B an approximation to the Hessian matrix.[4] The gradient of this approximation (with respect to Δx) is

$$\nabla f(x_k + \Delta x) \approx \nabla f(x_k) + B \, \Delta x,$$

and setting this gradient to zero (which is the goal of optimization) provides the Newton step:

$$\Delta x = -B^{-1} \nabla f(x_k).$$

The Hessian approximation B is chosen to satisfy

$$\nabla f(x_k + \Delta x) = \nabla f(x_k) + B \, \Delta x,$$

which is called the *secant equation* (the Taylor series of the gradient itself). In more than one dimension B is underdetermined. In one dimension, solving for B and applying the Newton's step with the updated value is equivalent to the secant method. The various quasi-Newton methods differ in their choice of the solution to the secant equation (in one dimension, all the variants are equivalent). Most methods (but with exceptions, such as Broyden's method) seek a symmetric solution ($B^T = B$); furthermore, the variants listed below can be motivated by finding an update B_{k+1} that is as close as possible to B_k in some norm; that is, $B_{k+1} = \mathbf{argmin}_B \, \|B - B_k\|_V$, where V is some positive-definite matrix that defines the norm. An approximate initial value $B_0 = \beta I$ is often sufficient to achieve rapid convergence, although there is no general strategy to choose β.[5] Note that B_0 should be positive-definite. The unknown x_k is updated applying the Newton's step calculated using the current approximate Hessian matrix B_k:

- $\Delta x_k = -\alpha_k B_k^{-1} \nabla f(x_k)$, with α chosen to satisfy the Wolfe conditions;
- $x_{k+1} = x_k + \Delta x_k$;
- The gradient computed at the new point $\nabla f(x_{k+1})$, and

$$y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$$

is used to update the approximate Hessian B_{k+1}, or directly its inverse $H_{k+1} = B_{k+1}^{-1}$ using the Sherman–Morrison formula.

- A key property of the BFGS and DFP updates is that if B_k is positive-definite, and α_k is chosen to satisfy the Wolfe conditions, then B_{k+1} is also positive-definite.

The most popular update formulas are:

Method	$B_{k+1} =$	$H_{k+1} = B_{k+1}^{-1} =$
BFGS	$B_k + \frac{y_k y_k^{\mathrm T}}{y_k^{\mathrm T} \Delta x_k} - \frac{B_k \Delta x_k (B_k \Delta x_k)^{\mathrm T}}{\Delta x_k^{\mathrm T} B_k \, \Delta x_k}$	$\left(I - \frac{\Delta x_k y_k^{\mathrm T}}{y_k^{\mathrm T} \Delta x_k}\right) H_k \left(I - \frac{y_k \Delta x_k^{\mathrm T}}{y_k^{\mathrm T} \Delta x_k}\right) + \frac{\Delta x_k \Delta x_k^{\mathrm T}}{y_k^{\mathrm T} \, \Delta x_k}$
Broyden	$B_k + \frac{y_k - B_k \Delta x_k}{\Delta x_k^{\mathrm T} \, \Delta x_k} \, \Delta x_k^{\mathrm T}$	$H_k + \frac{(\Delta x_k - H_k y_k) \Delta x_k^{\mathrm T} H_k}{\Delta x_k^{\mathrm T} H_k \, y_k}$
Broyden family	$(1 - \varphi_k) B_{k+1}^{\text{BFGS}} + \varphi_k B_{k+1}^{\text{DFP}}, \quad \varphi \in [0, 1]$	
DFP	$\left(I - \frac{y_k \, \Delta x_k^{\mathrm T}}{y_k^{\mathrm T} \, \Delta x_k}\right) B_k \left(I - \frac{\Delta x_k y_k^{\mathrm T}}{y_k^{\mathrm T} \, \Delta x_k}\right) + \frac{y_k y_k^{\mathrm T}}{y_k^{\mathrm T} \, \Delta x_k}$	$H_k + \frac{\Delta x_k \Delta x_k^{\mathrm T}}{\Delta x_k^{\mathrm T} \, y_k} - \frac{H_k y_k y_k^{\mathrm T} H_k}{y_k^{\mathrm T} H_k y_k}$
SR1	$B_k + \frac{(y_k - B_k \, \Delta x_k)(y_k - B_k \, \Delta x_k)^{\mathrm T}}{(y_k - B_k \, \Delta x_k)^{\mathrm T} \, \Delta x_k}$	$H_k + \frac{(\Delta x_k - H_k y_k)(\Delta x_k - H_k y_k)^{\mathrm T}}{(\Delta x_k - H_k y_k)^{\mathrm T} y_k}$

Other methods are Pearson's method, McCormick's method, the Powell symmetric Broyden (PSB) method and Greenstadt's method.[2]

Relationship to matrix inversion

When f is a convex quadratic function with positive-definite Hessian B, one would expect the matrices H_k generated by a quasi-Newton method to converge to the inverse Hessian $H = B^{-1}$. This is indeed the case for the class of quasi-Newton methods based on least-change updates.[6]

Notable implementations

Implementations of quasi-Newton methods are available in many programming languages.

Notable open source implementations include:

- GNU Octave uses a form of BFGS in its `fsolve` function, with trust region extensions.
- GNU Scientific Library implements the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm.
- ALGLIB implements (L)BFGS in C++ and C#
- R's `optim` general-purpose optimizer routine uses the BFGS method by using `method="BFGS"`.[7]
- Scipy.optimize has fmin_bfgs. In the SciPy extension to Python, the `scipy.optimize.minimize` function includes, among other methods, a BFGS implementation.[8]

Notable proprietary implementations include:

- Mathematica includes quasi-Newton solvers.[9]

- The NAG Library contains several routines[10] for minimizing or maximizing a function[11] which use quasi-Newton algorithms.
- In MATLAB's Optimization Toolbox, the `fminunc` function uses (among other methods) the BFGS quasi-Newton method.[12] Many of the constrained methods of the Optimization toolbox use BFGS and the variant L-BFGS.[13]<

See also

- BFGS method
  - L-BFGS
  - OWL-QN
- Broyden's method
- DFP updating formula
- Newton's method
- Newton's method in optimization
- SR1 formula

References

1. Broyden, C. G. (1972). "Quasi-Newton Methods". In Murray, W. (ed.). *Numerical Methods for Unconstrained Optimization*. London: Academic Press. pp. 87–106. ISBN 0-12-512250-0.
2. Haelterman, Rob (2009). "Analytical study of the Least Squares Quasi-Newton method for interaction problems" (https://lib.ugent.be/catalog/rug01:001333190). *PhD Thesis, Ghent University*. Retrieved 2014-08-14.
3. Rob Haelterman; Dirk Van Eester; Daan Verleyen (2015). "Accelerating the solution of a physics model inside a tokamak using the (Inverse) Column Updating Method" (https://doi.org/10.1016%2Fj.cam.2014.11.005). *Journal of Computational and Applied Mathematics*. **279**: 133–144. doi:10.1016/j.cam.2014.11.005 (https://doi.org/10.1016%2Fj.cam.2014.11.005).
4. "Introduction to Taylor's theorem for multivariable functions - Math Insight" (https://mathinsight.org/taylors_theorem_multivariable_introduction). *mathinsight.org*. Retrieved November 11, 2021.
5. Nocedal, Jorge; Wright, Stephen J. (2006). *Numerical Optimization* (https://archive.org/details/numericaloptimiz00noce_990). New York: Springer. pp. 142 (https://archive.org/details/numericaloptimiz00noce_990/page/n161). ISBN 0-387-98793-2.
6. Robert Mansel Gower; Peter Richtarik (2015). "Randomized Quasi-Newton Updates are Linearly Convergent Matrix Inversion Algorithms". arXiv:1602.01768 (https://arxiv.org/abs/1602.01768) [math.NA (https://arxiv.org/archive/math.NA)].
7. "optim function - RDocumentation" (https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/optim). *www.rdocumentation.org*. Retrieved 2022-02-21.
8. "Scipy.optimize.minimize — SciPy v1.7.1 Manual" (http://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html).
9. "Unconstrained Optimization: Methods for Local Minimization—Wolfram Language Documentation" (https://reference.wolfram.com/language/tutorial/UnconstrainedOptimizationMethodsForLocalMinimization.html.en). *reference.wolfram.com*. Retrieved 2022-02-21.
10. The Numerical Algorithms Group. "Keyword Index: Quasi-Newton" (http://www.nag.co.uk/numeric/fl/nagdoc_fl23/html/INDEXES/KWIC/quasi-newton.html). *NAG Library Manual, Mark 23*. Retrieved 2012-02-09.
11. The Numerical Algorithms Group. "E04 – Minimizing or Maximizing a Function" (http://www.nag.co.uk/numeric/fl/nagdoc_fl23/pdf/E04/e04intro.pdf) (PDF). *NAG Library Manual, Mark 23*. Retrieved 2012-02-09.
12. "Find minimum of unconstrained multivariable function - MATLAB fminunc" (http://www.mathworks.com/help/toolbox/optim/ug/fminunc.html).
13. "Constrained Nonlinear Optimization Algorithms - MATLAB & Simulink" (https://www.mathworks.com/help/optim/ug/constrained-nonlinear-optimization-algorithms.html). *www.mathworks.com*. Retrieved 2022-02-21.

Further reading

- Bonnans, J. F.; Gilbert, J. Ch.; Lemaréchal, C.; Sagastizábal, C. A. (2006). *Numerical Optimization : Theoretical and Numerical Aspects* (Second ed.). Springer. ISBN 3-540-35445-X.
- Fletcher, Roger (1987), *Practical methods of optimization* (https://archive.org/details/practicalmethods0000flet) (2nd ed.), New York: John Wiley & Sons, ISBN 978-0-471-91547-8.
- Nocedal, Jorge; Wright, Stephen J. (1999). "Quasi-Newton Methods" (https://books.google.com/books?id=7wDpBwAAQBAJ&pg=PA192). *Numerical Optimization*. New York: Springer. pp. 192–221. ISBN 0-387-98793-2.
- Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. (2007). "Section 10.9. Quasi-Newton or Variable Metric Methods in Multidimensions" (http://apps.nrbook.com/empanel/index.html#pg=521). *Numerical Recipes: The Art of Scientific Computing* (3rd ed.). New York: Cambridge University Press. ISBN 978-0-521-88068-8.
- Scales, L. E. (1985). *Introduction to Non-Linear Optimization* (https://books.google.com/books?id=AEJdDwAAQBAJ&pg=PA84). New York: MacMillan. pp. 84–106. ISBN 0-333-32552-4.

Retrieved from "https://en.wikipedia.org/w/index.php?title=Quasi-Newton_method&oldid=1174813119"

-