Unconstrained Optimization

This Hypercard stack was prepared by:
Dennis L. Bricker,
Dept. of Industrial Engineering,
University of Iowa,
Iowa City, Iowa 52242
e-mail: dbricker@icaen.uiowa.edu

Generic Search Method

Given the current approximation x^k to the solution, consider the questions:

1) Is x^k optimal (or sufficiently near-optimal)? If so, STOP.
2) In what direction d^k should the next approximate solution x^{k+1} lie?
3) How far should we go in the direction d^k to find x^{k+1}?

Search methods usually differ in their choice of the SEARCH DIRECTION, e.g.,

Newton's Method: $d^k = -\left[\nabla^2 f(x^k)\right]^{-1} \nabla f(x^k)$

Quasi-Newton Method: $d^k = -B^{-1}\nabla f(x^k)$ or $-Q\nabla f(x^k)$

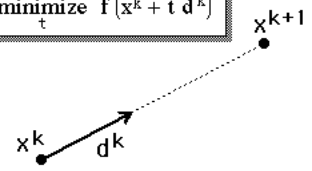
(where B approximates $\nabla^2 f(x^k)$ or Q approximates $[\nabla^2 f(x^k)]^{-1}$)

Steepest Descent: $d^k = -\nabla f(x^k)$

Conjugate Gradient: $d^k = -\nabla f(x^k) + \dfrac{\|\nabla f(x^k)\|^2}{\|\nabla f(x^{k-1})\|^2} d^{k-1}$

Any of the one-dimensional search methods may be used to choose the stepsize, once the step direction is chosen.

$\underset{t}{\text{minimize}}\ f(x^k + t\,d^k)$



Golden Section search
Fibonacci search
Quadratic interpolation
Cubic interpolation
etc.

Conjugate Directions

Most of the better descent methods make use of conjugate search directions.

Two directions $\hat{d}$ and $\bar{d}$ are *conjugate with respect to a* (square) *matrix* H if

$$\hat{d}^T H \bar{d} = 0$$

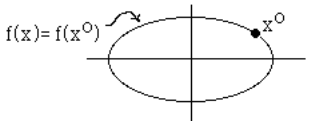
Note: If H is the identity matrix, then this property reduces to $\hat{d}^T \bar{d} = 0$ i.e., orthogonality.

Geometric Interpretation of Conjugate Directions

Let f be the quadratic form $f(x) = \frac{1}{2} x^T H x$

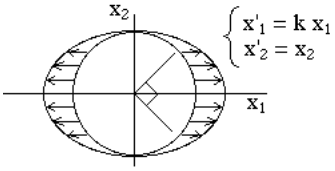
where H is a (constant) n×n matrix ($H = \nabla^2 f(x)$)

If n=2 (and H is positive definite), the contours of the function f are ellipses centered at the origin.

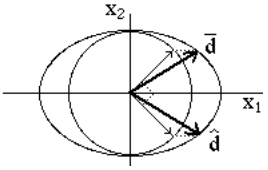


Geometric Interpretation of Conjugate Directions

An ellipse may be generated by "stretching" a circle along the direction of one diameter (the major axis).



Two orthogonal radii vectors will be correspondingly "stretched" into a pair of vectors conjugate w.r.t. H



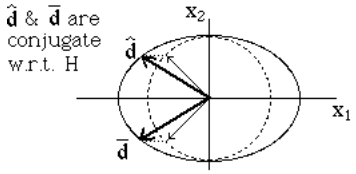
Conjugate Directions

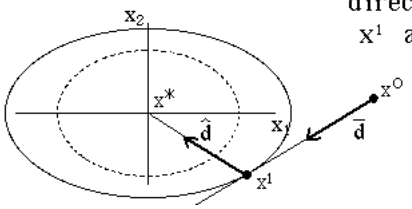
Theorem

The minimum of a convex quadratic function of n variables,

$$f(x_1,x_2,\cdots x_n) = \tfrac{1}{2} x^T H x + c^T x + k$$

can be found in at most n steps from any initial point by doing a series of exact linesearches along each of a set of n directions which are conjugate with respect to H, the Hessian matrix.





Suppose that we start from x^0, and search first in the direction $\bar{d}$ to obtain x^1 and then in the direction $\hat{d}$ to obtain x^*. Then x^* is optimal!

In practice,

- a set of directions $\{d^1, d^2, \ldots d^n\}$ which are mutually conjugate w.r.t. $\nabla^2 f(x)$ is seldom known beforehand, but is generated by a scheme which depends upon available information (Hessian matrix, gradients, function values at previous points in the search, etc.)

In practice,

- Unless f is actually quadratic, the Hessian matrix is not constant, but changes as the search proceeds... thus the "conjugate" directions which may be computed are seldom actually conjugate with respect to any fixed matrix H.

  Despite this, even for non-quadratic functions, the so-called "conjugate" directions generated by the various algorithms prove to be "good" search directions for most problems.

⏎

Assorted Search Algorithms

- ☞ Newton's Method
- ☞ Newton's Method with Linesearch
- ☞ Steepest Descent Method
- ☞ Fletcher-Reeves Conjugate Gradient Method
- ☞ Quasi-Newton Methods (DFP & BFGS)
- ☞ Powell's Method

⏎

Newton's Algorithm

In Newton's method, the search direction is

$$d^k = -\left[\nabla^2 f(x^k)\right]^{-1} \nabla f(x^k)$$

No line search is performed, i.e., a step size

$$t = 1$$

is used.

$$x^{k+1} = x^k - \left[\nabla^2 f(x^k)\right]^{-1} \nabla f(x^k)$$

⏎ Numerical Example

Newton's Method with Linesearch

In Newton's method, the search direction is

$$d^k = -\left[\nabla^2 f(x^k)\right]^{-1} \nabla f(x^k)$$

In its original form, the step size $t=1$, i.e., no linesearch is performed. Including a line search

$$\underset{t}{\text{minimize}}\ f(x^k + t\, d^k)$$

will improve its performance when far from the optimal, and prevent convergence to a stationary point which is not a minimum.

⏎ Numerical Example

Steepest Descent

Consider the function

$$\phi(t) = f(x^k + t\, d^k)$$

of the stepsize t.

The *directional derivative* of f in the direction d^k at the point x^k is

$$\phi'(t) = \left[\nabla f(x^k)\right]^T d^k$$

⏎

The direction which minimizes the directional derivative at x is simply

$$d^k = -\nabla f(x^k)$$

That is:

> At a given point x^k, the negative of the gradient vector, $-\nabla f(x^k)$, points in the direction of most rapid decrease for f(x) and the rate of decrease of f(x) at x^k in this direction is
>
> $$-\left\|\nabla f(x^k)\right\|$$

The Steepest Descent Method generates search directions, each of which is perpendicular to the previous (assuming exact minimization in computing the optimal stepsize).
That is,

$$x^{k+1} = x^k + t^* d^k \quad \Rightarrow \nabla f(x^{k+1}) \perp [x^{k+1} - x^k]$$

This feature of Steepest Descent tends to result in zig-zagging paths to the optimum.

Numerical Example

Fletcher-Reeves Algorithm

(Also known as "Conjugate Gradient" method)

At the initial iteration, choose $d^0 = -\nabla f(x^0)$
i.e., the steepest descent direction.
The next search directions are then computed by

$$d^k = -\nabla f(x^k) + \frac{\left\|\nabla f(x^k)\right\|^2}{\left\|\nabla f(x^{k-1})\right\|^2} d^{k-1}$$

That is, the next search direction is a combination of the steepest descent direction and the previous search direction.

If the previous step significantly reduced the gradient, then the factor for d^{k-1} will be small, and the next direction will be $\approx -\nabla f(x^k)$

The directions generated by the Fletcher-Reeves algorithm will, if f(x) is quadratic, be conjugate with respect to the Hessian matrix!

Numerical Example

Quasi-Newton Algorithms

"Quasi-Newton" Method

$$d^k = -\left[\nabla^2 f(x^k)\right]^{-1} \nabla f(x^k)$$

Newton's step direction

The "Quasi-Newton" methods compute the step direction based upon the formula above, but approximating the Hessian without computing it.

- Davidon-Fletcher-Powell (DFP) Method
- Broyden-Fletcher-Goldfarb-Shanno (BFGS) Method

Davidon-Fletcher-Powell Algorithm

"Quasi-Newton" Method

$$d^k = -\left[\nabla^2 f(x^k)\right]^{-1} \nabla f(x^k)$$

Newton's step direction

The "DFP" algorithm builds up approximations to the inverse of the Hessian matrix

$$Q_k \approx \left[\nabla^2 f(x^k)\right]^{-1}$$

without actually computing it. The search directions are then generated by

$$d^k = -Q_k \nabla f(x^k)$$

At iteration k, suppose that we have performed an exact minimization in the direction d^k to obtain x^{k+1}. The approximation Q is updated by:

1) $p^k = x^{k+1} - x^k$ *change in x*

2) $y^k = \nabla f(x^{k+1}) - \nabla f(x^k)$ *change in gradient*

3) $A_k = \dfrac{p^k \otimes p^k}{(p^k)^T y^k}$, $B_k = -\dfrac{(Q_k y^k) \otimes (Q_k y^k)}{(y^k)^T Q_k y^k}$ ← *denominators are scalars*

4) $Q_{k+1} = Q_k + A_k + B_k$

⊗ *denotes "outer" product of 2 vectors, i.e., the matrix*
$u \otimes v \equiv u\, v^T$

Usually, $Q_o = I$, the identity matrix, is used to begin the algorithm.

If the function f(x) is quadratic, so that the Hessian matrix is constant, then the search methods generated by this method will be conjugate with respect to the Hessian.

Numerical Example

Broyden-Fletcher-Goldfarb-Shanno (BFGS) Method

"Quasi-Newton" Method

$$d^k = -\left[\nabla^2 f(x^k)\right]^{-1} \nabla f(x^k)$$

Newton's step direction

The "BFGS" algorithm builds up approximations to the Hessian matrix

$$B_k \approx \nabla^2 f(x^k)$$

without actually computing it. The search directions are then generated by

$$d^k = -B_k^{-1} \nabla f(x^k) \quad \text{or solving} \quad B_k d^k = -\nabla f(x^k)$$

At iteration k, suppose that we have performed an exact minimization in the direction d^k to obtain x^{k+1}. The approximation Q is updated by:

1) $p^k = x^{k+1} - x^k$ *change in x*

2) $y^k = \nabla f(x^{k+1}) - \nabla f(x^k)$ *change in gradient*

3) $$B_{k+1} = B_k - \frac{(B_k p^k) \otimes (B_k p^k)}{(p^k)^T B_k p^k} + \frac{y^k \otimes y^k}{(y^k)^T p^k}$$

denominators are scalars

$\otimes$ denotes "outer" product of 2 vectors, i.e., the matrix

$$u \otimes v \equiv u\, v^T$$

BFGS Method

Notes:

- B_o is typically the identity matrix
- B_k is guaranteed in theory to be positive definite & symmetric
- BFGS requires solving a system of linear equations at each iteration:

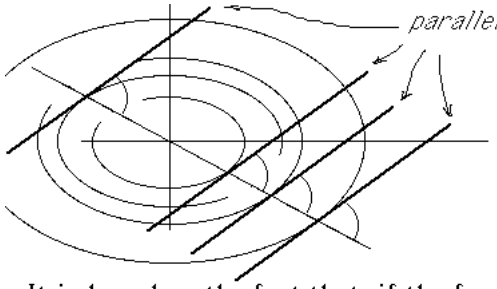
$$B_k d^k = -\nabla f(x^k)$$

This can be done by updating a Cholesky factorization at each iteration, rather than updating B itself.

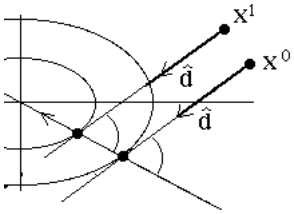
Powell's Method

This algorithm requires no computation of derivatives, but when applied to a quadratic function will generate conjugate search directions.

It is based on the fact that, if the function f is convex & quadratic, any line passing through the minimum will intersect the contour curves at equal angles.



It is based on the fact that, if the function f is convex & quadratic, any line passing through the minimum will intersect the contour curves at equal angles.



As a result of this property, if exact linesearches are performed in the same direction from two different points, the line joining the minima will be conjugate to (& pass through the optimum if n=2).

Powell's Method

Begin with x^0 and a set of n linearly independent directions $\{d^1, d^2, \ldots d^n\}$ (e.g., unit coordinate directions.)

1. For each of i=1,2,... n, perform a linesearch in the direction d^i to minimize $f(x^{i-1} + t d^i)$. and let $x = x^{i-1} + t^* d^i$ where t^* is the optimal stepsize.
2. Let $d^{n+1} = x^n - x^0$ be a new direction, and minimize $f(x^n + t d^{n+1})$ to get t^* and redefine $x^0 = x^n + t^* d^{n+1}$

Powell's Method

3. Stop if a termination criterion is satisfied. Otherwise, let $d^i = d^{i+1}$, $i=1,2,...n$ (discarding d^1) and return to step 1.

Note that (after the first iteration, both x^0 *&* x^n *are obtained by a linesearch in the direction* d^n*, and so the new direction*

$$d^{n+1} = x^n - x^0$$

should be conjugate to d^n

Numerical Example