Line search methods

From optimization

Author names: Elizabeth Conger
Steward: Dajun Yue and Fengqi You

Contents

Introduction

An algorithm is a line search method if it seeks the minimum of a defined nonlinear function by selecting a reasonable direction vector that, when computed iteratively with a reasonable step size, will provide a function value closer to the absolute minimum of the function. Varying these will change the "tightness" of the optimization. For example, given the function $f(x)$, an initial x_k is chosen. To find a lower value of $f(x)$, the value of x_{k+1} is increased by the following iteration scheme

$$x_{k+1} = x_k + \alpha_k p_k,$$

in which α_k is a positive scalar known as the step length and p_k defines the step direction. Figure 1 gives a clear flow chart to indicate the iteration scheme.

Step Length

Choosing an appropriate step length has a large impact on the robustness of a line search method. To select the ideal step length, the following function could be minimized:

$$\phi(\alpha) = f(x_k + \alpha p_k), \alpha > 0,$$

but this is not used in practical settings generally. This may give the most accurate minimum, but it would be very computationally expensive if the function ϕ has multiple local minima or stationary points, as shown in Figure 2.

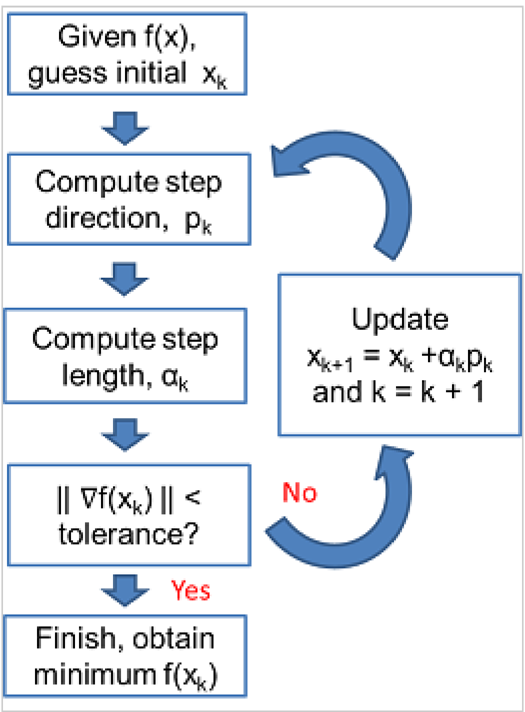

Figure 1: Algorithm flow chart of line search methods (Conger, adapted from Line Search wikipedia page)

A common and practical method for finding a suitable step length that is not too near to the global minimum of the ϕ function is to require that the step length of α_k reduces the value of the target function, or that

$$f(x_k + \alpha p_k < f(x_k)$$

This method does not ensure a convergence to the function's minimum, and so two conditions are employed to require a *significant decrease condition* during every iteration.

Wolfe Conditions

These conditions, developed in 1969 by Philip Wolfe, are an inexact line search stipulation that requires α_k to decreased the objective function by significant amount. This amount is defined by

$$f(x_k + \alpha p_k) < f(x_k) + c_1 \alpha \nabla f_k^T p_k$$

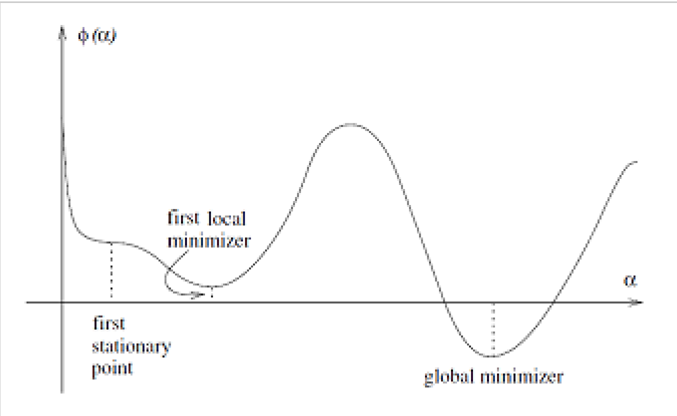

Figure 2: Complexity of finding ideal step length (Nocedal & Wright)

where c_1 is between 0 and 1. Another way of describing this condition is to say that the decrease in the objective function should be proportional to both the step length and the directional derivative of the function and step direction. This inequality is also known as the *Armijo condition*. In general, c_1 is a very small value, ~10^{-4}.

The *Armijo condition* must be paired with the *curvature condition*

$$\nabla f(x_k + \alpha p_k)^T p_k \geq f(x_k) + c_2 \nabla f_k^T p_k$$

to keep the α_k value from being too short. In this condition, c_2 is greater than c_1 but less than 1. These two conditions together are the Wolfe Conditions.

Another, more stringent form of these conditions is known as the *strong Wolfe conditions*. The *Armijo condition* remains the same, but the curvature condition is restrained by taking the absolute value of the left side of the inequality.

$$|\nabla f(x_k + \alpha p_k)^T p_k| \geq f(x_k) + c_2 \nabla f_k^T p_k$$

This left hand side of the curvature condition is simply the derivative of the ϕ function, and so this constraint prevents this derivative from becoming too positive, removing points that are too far from stationary points of ϕ from consideration as viable α_k values.

These conditions are valuable for use in Newton methods.

Goldstein Conditions

Another approach to finding an appropriate step length is to use the following inequalities known as the Goldstein conditions. This condition, instead of having two constants, only employs one:

$$f(x_k) + (1-c)\alpha_k \nabla f_k^T p_k \leq f(x_k + \alpha p_k) \leq f(x_k) + c\alpha_k \nabla f_k^T p_k$$

The second equality is very similar to the Wolfe conditions in that it is simply the sufficient decrease condition. The first inequality is another way to control the step length from below. This is best seen in the Figure 3.

In comparison to the Wolfe conditions, the Goldstein conditions are better suited for quasi-Newton methods

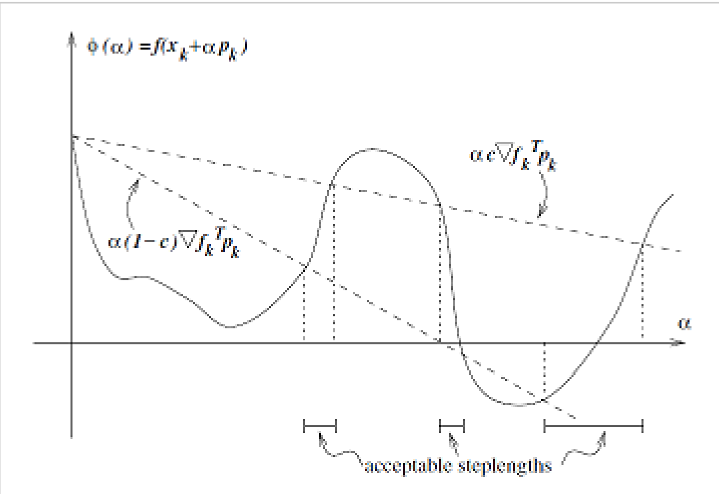

Figure 3: Application of the Goldstein Conditions (Nocedal & Wright)

(https://optimization.mccormick.northwestern.edu/index.php/Quasi-Newton_methods) than for Newton methods.

Step Direction

When using line search methods, it is important to select a search or step direction with the steepest decrease in the function. This will increase the efficiency of line search methods. To identify this steepest descent at varying points along the function, the angle θ_k between the chosen step direction and the negative gradient of the function $-\nabla f_k$, which is the steepest slope at point k. The angle is defined by

$$cos\theta_k = \frac{-\nabla f_k^T p_k}{||\nabla f_k|| ||p_k||},$$

and, as with the step length, it is not efficient to completely minimize θ_k. The amount that p_k can deviate from the steepest slope and still produce reasonable results depends on the step length conditions that are adhered to in the method. For example, if α_k satisfies the Wolfe conditions, the *Zoutendijk condition* applies:

$cos\theta_k{}^2||\nabla f_k||^2 < \infty$, which implies that $cos\theta_k{}^2||\nabla f_k||^2 \to 0$

There are various algorithms to use this angle property to converge on the function's minimum, and they each have their benefits and disadvantages depending on the application and complexity of the target function. The major algorithms available are the *steepest descent method*, the *Newton method*, and the *quasi-Newton methods*. These algorithms are explained in more depth elsewhere within this Wiki.

When using these algorithms for line searching, it is important to know their weaknessess. The steepest descent method is the "quintessential globally convergent algorithm", but because it is so robust, it has a large computation time. The Newton methods rely on choosing an initial input value that is sufficiently near to the minimum. This is because the Hessian matrix of the function may not be positive definite, and therefore using the Newton method may not converge in a descent direction. The Newton method can be modified to atone for this.

References

1. Sun, W. & Yuan, Y-X. (2006) Optimization Theory and Methods: Nonlinear Programming (Springer US) p 688.

2. Anonymous (2014) Line Search. (Wikipedia). http://en.wikipedia.org/wiki/Line_search.