Numerical Optimization 05: 1st order methods

Qiang Zhu

University of Nevada Las Vegas

May 20, 2020

Overview

1. In choosing the direction
2. Gradient Descent
3. Conjugate gradient
4. Summary

The choice of descent direction

In the previous chapter, we have talked about the general strategy for optimization is to decide a direction and then use the line search method to obtain a sufficient decrease. Repeating it for many time, we expect to arrive at the local minimum.

$$x^{k+1} = x^k + \alpha^k d^k$$

The search direction often has the form

$$d^k = -(B^k)^{-1}\nabla f(x^k) \tag{1}$$

where B^k is a symmetric and nonsingular matrix. In some method (e.g., steepest descent), B^k is the identify matrix, while in (quasi-) Newton's method, B^k is the approximate or exact Hessian.
In this lecture, we will cover the first-order methods which purely rely on the gradient information.

Gradient descent

An intuitive choice for the descent direction is the direction of steepest descent ($g^k = \nabla f(x^k)$).

$$d^k = -\frac{g^k}{||g^k||}$$

If we optimize the step size at each step, we have

$$\alpha^k = \arg\min_{\alpha} f(x^k + \alpha d^k)$$

Since

$$\nabla f(x^k + \alpha d^k)^T d^k = 0$$

We know

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

It is obvious that the two consecutive directions are orthogonal.

$$(d^{k+1})^T d^k = 0$$

Conjugate gradient

Gradient descent can perform poorly in narrow valleys. The conjugate gradient method overcomes this issue by doing a small transformation. When minimizing the quadratic functions:

$$\underset{\alpha}{\text{minimize}} : f(x) = \frac{1}{2}x^T Ax - b^T x$$

is equivalent to solving the linear equation

$$Ax = b$$

where A is $N \times N$ symmetric and positive definite, and thus f has a unique local minimum.

When solving $Ax = b$, a powerful method is to find a sequence of N conjugate directions satisfying

$$(d^i)^T Ad^j = 0 \quad (i \neq j)$$

To find the successive conjugate directions

One can start with the direction of steepest descent

$$d^1 = -g^1$$

We then use line search to find the next design point. For quadratic functions $f = \frac{1}{2}x^T Ax - b^T x$, the step factor α can be computed as

$$\begin{aligned}
\frac{\partial f(x+\alpha d)}{\partial \alpha} &= \frac{\partial}{\partial \alpha}\left[\frac{1}{2}(x+\alpha d)^T A(x+\alpha d) + b^T(x+\alpha d) + c\right] \\
&= d^T A(x+\alpha d) + d^T b \\
&= d^T(Ax+b) + \alpha d^T A d
\end{aligned}$$

Let the gradient be zero,

$$\alpha = -\frac{d^T(Ax+b)}{d^T A d}$$

Then the update is

$$x^2 = x^1 + \alpha d^1$$

To find the successive conjugate directions (continued)

For the next step

$$d^{k+1} = -g^{k+1} + \beta^k d^k$$

where β^k is a series of scalar parameters. Larger values of β indicate that the previous descent direction contributes strongly.
We solve β, from the followings

$$d^{(k+1)T} A d^k = 0$$
$$(-g^{k+1} + \beta^k d^{(k)})^T A d^{(k)} = 0$$
$$-g^{k+1} A d^{(k)} + \beta^k d^{(k)T} A d^{(k)} = 0$$
$$\beta^k = \frac{g^{(k+1)T} A d^{(k)}}{d^{(k)T} A d^{(k)}}$$

The conjugate method is exact for quadratic functions. But it can be applied to non quadractic functions as well when the quadratic function is a good approximation.

To Approximate A and β

Unfortunately, we don't know the value of A that best approximate f around x^k. So we choose some way to compute β.

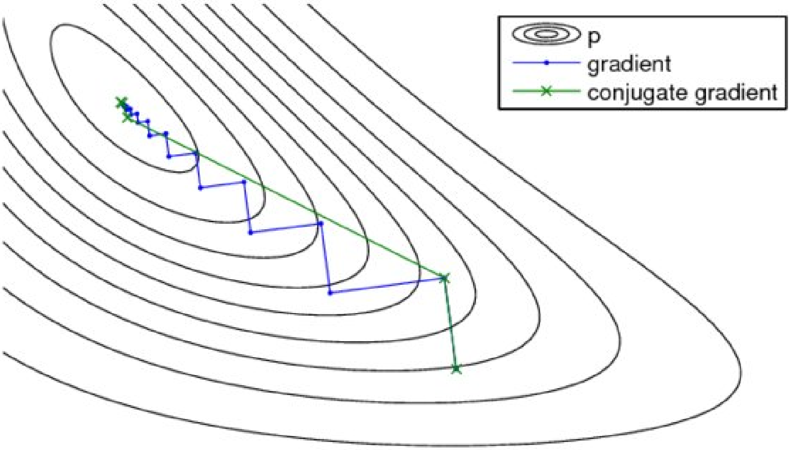
Fletcher-Reeves

$$\beta^k = \frac{g^{(k)T} g^{(k)}}{g^{(k-1)T} g^{(k-1)}}$$

Polak-Ribiere

$$\beta^k = \frac{g^{(k)T} (g^{(k)} - g^{(k-1)})}{g^{(k-1)T} g^{(k-1)}}$$

Comparison between Conjugate Gradient and Steepest Descent



Summary

- Gradient descent follows the direction of steepest descent
- Two consecutive search directions in gradient descent are orthogonal
- In conjugate gradient, the search directions are conjugate with respect to an approximate hessian.
- Both SD and CG work with the line search method