Optimization

Lusine Poghosyan

AUA

March 1, 2019

Conjugate Gradient Methods

The class of conjugate direction methods can be viewed as being intermediate between the method of steepest descent and Newton's method. The conjugate direction methods have the following properties:

- Finds the minimizer of $f(x) = \frac{1}{2}x^T Qx - bx$, $x \in \mathbb{R}^n$, $Q = Q^T \succ 0$ in n steps;
- Requires no Hessian matrix evaluations.
- No matrix inversion and no storage of an $n \times n$ matrix are required.

Definition

Let Q be a real symmetric $n \times n$ matrix. The directions $d^{(0)}$, $d^{(1)}$, $\dots$, $d^{(m)}$ are Q-conjugate if for all $i \neq j$, we have $d^{(i)T} Q d^{(j)} = 0$.

Example

Let

$$Q = \begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix}.$$

Show that the directions $d^{(0)} = [1, 0]^T$, $d^{(1)} = [-3/8, 3/4]^T$ are Q-conjugate directions.

The Conjugate Direction Algorithm

Assume

$$f(x) = \frac{1}{2}x^T Qx - bx, \quad x \in \mathbb{R}^n, \quad Q = Q^T \succ 0.$$

As $Q \succ 0$, $f(x)$ has a global minimizer x^* which can be found by solving $Qx = b$.

Basic Conjugate Direction Algorithm. Given a starting point $x^{(0)}$ and Q-conjugate directions $d^{(0)}, d^{(1)}, \dots, d^{(n-1)}$; for $k \geq 0$,

$$g^{(k)} = \nabla f\left(x^{(k)}\right) = Qx^{(k)} - b,$$

$$\alpha_k = -\frac{g^{(k)T} d^{(k)}}{d^{(k)T} Q d^{(k)}},$$

$$x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}.$$

Note. $\alpha_k = \arg\min \Phi_k(\alpha) = \arg\min f\left(x^{(k)} + \alpha d^{(k)}\right)$.

Theorem

For any starting point $x^{(0)}$ the basic conjugate direction algorithm converges to the unique x^* in n steps; that is, $x^{(n)} = x^*$.

Example

Find the minimizer of

$$f(x) = \frac{1}{2}x^T \begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix} x - x^T \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \quad x \in \mathbb{R}^2,$$

using the conjugate direction method with the initial point $x^{(0)} = [0, 0]^T$, and Q-conjugate directions $d^{(0)} = [1, 0]^T$, $d^{(1)} = [-3/8, 3/4]^T$.

The Conjugate Gradient Algorithm for Quadratic Functions

The conjugate gradient algorithm does not use prespecified conjugate directions, but instead computes the directions as the algorithm progresses.

As before, we assume that

$$f(x) = \frac{1}{2}x^T Qx - bx, \quad x \in \mathbb{R}^n, \quad Q = Q^T \succ 0.$$

1. Set $k := 0$; select the initial point $x^{(0)}$.
2. $g^{(0)} = \nabla f\left(x^{(0)}\right)$. If $g^{(0)} = 0$, stop; else, set $d^{(0)} = -g^{(0)}$.
3. $\alpha_k = -\frac{g^{(k)T} d^{(k)}}{d^{(k)T} Q d^{(k)}} = \arg\min_{\alpha \geq 0} f\left(x^{(k)} + \alpha d^{(k)}\right)$.
4. $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$.
5. $g^{(k+1)} = \nabla f\left(x^{(k+1)}\right)$. If $g^{(k+1)} = 0$, stop.
6. $\beta_k = \frac{g^{(k+1)T} Q d^{(k)}}{d^{(k)T} Q d^{(k)}}$.
7. $d^{(k+1)} = -g^{(k+1)} + \beta_k d^{(k)}$.
8. Set $k = k + 1$; go to step 3.

Example

Find the minimizer of

$$f(x_1, x_2) = x_1^2 + 4x_2^2,$$

using the Conjugate Gradient Algorithm with the initial point $x^{(0)} = [4, 1]^T$.

The Conjugate Gradient Algorithm for Nonquadratic Problems

In The Conjugate Gradient Algorithm we want to avoid Hessian evaluation. In the algorithm for quadratic functions we saw that the Hessian is used in the calculation of α_k and β_k.

In order to avoid Hessian evaluation we can calculate α_k using the following formula

$$\alpha_k = \arg\min_{\alpha \geq 0} f\left(x^{(k)} + \alpha d^{(k)}\right).$$

Recall that

$$\beta_k = \frac{g^{(k+1)T} Q d^{(k)}}{d^{(k)T} Q d^{(k)}}.$$

We will consider the following modification of β_k.

Hestenes-Stiefel Formula.

$$\beta_k = \frac{g^{(k+1)T} \left(g^{(k+1)} - g^{(k)}\right)}{d^{(k)T} \left(g^{(k+1)} - g^{(k)}\right)}.$$

The Conjugate Gradient Algorithm will be written in the following way

1. Set $k := 0$; select the initial point $x^{(0)}$.
2. $g^{(0)} = \nabla f\left(x^{(0)}\right)$. If $g^{(0)} = 0$, stop; else, set $d^{(0)} = -g^{(0)}$.
3. $\alpha_k = \arg\min_{\alpha \geq 0} f\left(x^{(k)} + \alpha d^{(k)}\right)$.
4. $x^{(k+1)} = x^{(k)} + \alpha_k d^{(k)}$.
5. $g^{(k+1)} = \nabla f\left(x^{(k+1)}\right)$. If $g^{(k+1)} = 0$, stop.
6. $\beta_k = \frac{g^{(k+1)T}\left(g^{(k+1)} - g^{(k)}\right)}{d^{(k)T}\left(g^{(k+1)} - g^{(k)}\right)}$.
7. $d^{(k+1)} = -g^{(k+1)} + \beta_k d^{(k)}$.
8. Set $k = k + 1$; go to step 3.

Stopping conditions

- $||\nabla f\left(x^{(k)}\right)|| < \varepsilon$
- $||x^{(k+1)} - x^{(k)}|| < \varepsilon$ or $\frac{||x^{(k+1)} - x^{(k)}||}{||x^{(k)}||} < \varepsilon$ if $||x^{(k)}|| \neq 0$
- $\left|f\left(x^{(k+1)}\right) - f\left(x^{(k)}\right)\right| < \varepsilon$ or $\frac{\left|f\left(x^{(k+1)}\right) - f\left(x^{(k)}\right)\right|}{\left|f\left(x^{(k)}\right)\right|} < \varepsilon$ if $f\left(x^{(k)}\right) \neq 0$.

Note. A common practice is to reinitialize the direction vector to the negative gradient after every few iterations (e.g., n or $n+1$) and continue until the algorithm satisfies the stopping criterion.