Conjugate Gradient Method

Com S 477/577

Nov 6, 2007

1 Introduction

Recall that in steepest descent of nonlinear optimization the steps are along directions that undo some of the progress of the others. The basic idea of the conjugate gradient method is to move in non-interfering directions.

Suppose we have just performed a line minimization along the direction $\boldsymbol{u}$. Then the gradient ∇f at the current point is perpendicular to $\boldsymbol{u}$, because otherwise we would have been able to move further along $\boldsymbol{u}$. Next, we should move along some direction $\boldsymbol{v}$. In steepest descent we let $\boldsymbol{v} = -\nabla f$. In the conjugate gradient method we perturb $-\nabla f$ by adding to it some direction to become $\boldsymbol{v}$.

We want to choose $\boldsymbol{v}$ in such a way that it does not undo our minimization along $\boldsymbol{u}$. In other words, we want ∇f to be perpendicular to $\boldsymbol{u}$ before and after we move along $\boldsymbol{v}$. At least locally we want that *the change in* ∇f *be perpendicular to* $\boldsymbol{u}$.

Now observe that a small change $\delta\boldsymbol{x}$ in $\boldsymbol{x}$ will produce a small change in ∇f given by

$$\delta\Big(\nabla f\Big) \approx Hf \cdot \delta\boldsymbol{x}.$$

Our idea of moving along non-interfering directions leads to the condition

$$\boldsymbol{u}^T \delta\Big(\nabla f\Big) = 0,$$

And the next move should be along the direction $\boldsymbol{v}$ such that

$$\boldsymbol{u}^T Hf\boldsymbol{v} = 0. \tag{1}$$

Even though $\boldsymbol{v}$ is not orthogonal to $\boldsymbol{u}$, it is Hf-orthogonal to $\boldsymbol{u}$.

Of course, we must worry about a slight technicality. The connection between $\delta\boldsymbol{x}$ and $\delta(\nabla f)$ in terms of the Hessian Hf is a differential relationship. We here use it for finite motions to the extent that Taylor's approximation of order 2 is valid. Suppose we expand f around a point $\boldsymbol{y}$:

$$f(\boldsymbol{x}+\boldsymbol{y}) \;\approx\; f(\boldsymbol{y}) + \nabla f(\boldsymbol{y})^T\boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^T Hf\boldsymbol{x}.$$

Thus f locally looks like a quadratic. If we focus on quadratics, then the Hessian Hf does not vary as we move along directions $\boldsymbol{u}$ and $\boldsymbol{v}$. Thus the condition (1) makes sense.

With this reasoning as background, one develops the conjugate gradient method for quadratic functions formed from symmetric positive definite matrices. For such quadratic functions, the conjugate gradient method converges to the unique global minimum in at most n steps, by moving along successive non-interfering directions.

For general functions, the conjugate gradient method repeatedly executes "packages" of n steps. Once near a local minimum, the algorithm converges quadratically.

2 Conjugate Direction

Given a symmetric matrix Q, two vectors $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ are said to be *Q-orthogonal*, or *conjugate with respect to Q*, if $\boldsymbol{d}_1^T Q \boldsymbol{d}_2 = 0$. A finite set of vectors $\boldsymbol{d}_0, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_k$ is said to be a *Q-orthogonal set* if $\boldsymbol{d}_i^T Q \boldsymbol{d}_j = 0$ for all $i \neq j$.

Proposition 1 *If Q is symmetric positive definite and the vectors $\boldsymbol{d}_0, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_k$ are Q-orthogonal to each other, then they are linearly independent.*

Proof Suppose there exist constants α_i, $i = 0, 1, \ldots, k$ such that

$$\alpha_0 \boldsymbol{d}_0 + \cdots + \alpha_k \boldsymbol{d}_k = 0.$$

Multiplying by Q and taking the scalar product with $\boldsymbol{d}_i$ yields

$$\alpha_i \boldsymbol{d}_i^T Q \boldsymbol{d}_i = 0, \qquad \text{for } i = 0, 1, \ldots, k.$$

But $\boldsymbol{d}_i^T Q \boldsymbol{d}_i > 0$ given the positive definiteness of Q, we have $\alpha_i = 0$ for $i = 0, \ldots, k$. □

Let us investigate just why the notion of Q-orthogonality is useful in the solution of the following problem

$$\min \quad \frac{1}{2}\boldsymbol{x}^T Q \boldsymbol{x} + \boldsymbol{b}^T \boldsymbol{x}, \tag{2}$$

where Q is symmetric positive definite. The unique solution to this problem is also the unique solution to the equation

$$Q\boldsymbol{x} + \boldsymbol{b} = 0. \tag{3}$$

Suppose that $\boldsymbol{d}_0, \ldots, \boldsymbol{d}_{n-1}$ are n non-zero Q-orthogonal vectors. By the previous proposition, these vectors are independent. Therefore they form a Q-orthogonal basis for $\mathbb{R}^n$.

Let $\boldsymbol{x}^*$ be the unique solution to (2) or (3). We can write

$$\boldsymbol{x}^* = \alpha_0 \boldsymbol{d}_0 + \cdots + \alpha_{n-1} \boldsymbol{d}_{n-1},$$

for some real numbers $\alpha_0, \ldots, \alpha_{n-1}$. Plugging the above into (3) yields

$$Q(\alpha_0 \boldsymbol{d}_0 + \cdots + \alpha_{n-1} \boldsymbol{d}_{n-1}) + \boldsymbol{b} = 0$$

and

$$\boldsymbol{d}_i^T Q(\alpha_0 \boldsymbol{d}_0 + \cdots + \alpha_{n-1} \boldsymbol{d}_{n-1}) + \boldsymbol{d}_i^T \boldsymbol{b} = 0.$$

Due to the Q-orthogonality of the $\boldsymbol{d}_i$'s, we can solve for these coefficients

$$\alpha_i = -\frac{\boldsymbol{d}_i^T \boldsymbol{b}}{\boldsymbol{d}_i^T Q \boldsymbol{d}_i}.$$

Thus we obtain the explicit formula

$$\boldsymbol{x}^* \quad = \quad -\sum_{i=0}^{n-1} \frac{\boldsymbol{d}_i^T \boldsymbol{b}}{\boldsymbol{d}_i^T Q \boldsymbol{d}_i} \boldsymbol{d}_i.$$

Notice two important facts:

1. By choosing $\boldsymbol{d}_1, \ldots, \boldsymbol{d}_n$ to be Q-orthogonal we can determine the coefficients $\alpha_1, \ldots, \alpha_n$ easily, using inner products.

2. The approach is possible for any positive-definite matrix. In particular, we could simply have chosen the $\boldsymbol{d}_i$'s to be orthogonal (i.e., I-orthogonal). Then $\boldsymbol{x}^* = \sum_{i=0}^{n-1} (\boldsymbol{d}_i^T \boldsymbol{x}^* / \boldsymbol{d}_i^T \boldsymbol{d}_i) \boldsymbol{d}_i$. However, by choosing the $\boldsymbol{d}_i$'s to be Q-orthogonal we can *determine the coefficients α_i's in terms of the known quantity $\boldsymbol{b}$, not the unknown quantity $\boldsymbol{x}^*$.*

How does this generate an algorithm? One view is purely algebraic, namely, we compute $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$. Another view is to think of these computations as an n-step search. We start the search at the origin. On the ith iteration we move in the direction $\boldsymbol{d}_i$ by α_i. After n iterations, we have found the unique minimum $\boldsymbol{x}^*$, as we will see shortly.

But two important issues remain:

1. How do we construct the Q-orthogonal vectors $\boldsymbol{d}_0, \ldots, \boldsymbol{d}_{n-1}$?

2. How do we deal with the reality that the matrix $Q = Hf$ is often unknown?

3 Properties of Descent

Let Q be a symmetric and positive definite matrix. We define $\mathcal{B}_k$ as the subspace of $\mathbb{R}^n$ spanned by a set of Q-orthogonal vectors $\boldsymbol{d}_0, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_{k-1}$; or for short,

$$\mathcal{B}_k = \operatorname{span}\{\boldsymbol{d}_0, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_{k-1}\}.$$

Theorem 2 (Expanding Subspace) *Let $\{\boldsymbol{d}_0, \ldots, \boldsymbol{d}_{n-1}\}$ be a set of nonzero Q-orthogonal vectors in $\mathbb{R}^n$. For any $\boldsymbol{x}_0 \in \mathbb{R}^n$, consider the sequence $\{\boldsymbol{x}_k\}$ generated by the rule*

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k,$$

where, writing $\boldsymbol{g}_k = Q\boldsymbol{x}_k + \boldsymbol{b}$,

$$\alpha_k = -\frac{\boldsymbol{g}_k^T \boldsymbol{d}_k}{\boldsymbol{d}_k^T Q \boldsymbol{d}_k}.$$

The following statements hold:

(i) the sequence $\{\boldsymbol{x}_k\}$ converges to the unique solution $\boldsymbol{x}^*$ of $Q\boldsymbol{x} + \boldsymbol{b} = \boldsymbol{0}$ after n steps. In other words, $\boldsymbol{x}_n = \boldsymbol{x}^*$ minimizes the function $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T Q \boldsymbol{x} + \boldsymbol{b}^T \boldsymbol{x}$.

(ii) $\boldsymbol{x}_{k+1}$ minimizes the same function $f(\boldsymbol{x})$ on the line $\boldsymbol{x} = \boldsymbol{x}_k + \alpha \boldsymbol{d}_k$, $-\infty < \alpha < \infty$ as well as on the linear variety $\boldsymbol{x}_0 + \mathcal{B}_{k+1}$.

Proof To prove (i), we make use of the linear independence of the $\boldsymbol{d}_j$'s. Notice that

$$\boldsymbol{x}^* - \boldsymbol{x}_0 = \alpha_0 \boldsymbol{d}_0 + \cdots + \alpha_{n-1} \boldsymbol{d}_{n-1}$$

for some $\alpha_0, \ldots, \alpha_{n-1}$. We multiply both sides of the equation by Q and take the inner product with $\boldsymbol{d}_k$, yielding

$$\alpha_k = \frac{\boldsymbol{d}_k^T Q (\boldsymbol{x}^* - \boldsymbol{x}_0)}{\boldsymbol{d}_k^T Q \boldsymbol{d}_k}. \tag{4}$$

Now we use induction to show that α_k defined in (4) equals $-\boldsymbol{g}_k^T\boldsymbol{d}_k/\boldsymbol{d}_k^TQ\boldsymbol{d}_k$. Suppose this is true for $\alpha_0,\dots,\alpha_{k-1}$. Following the iterative steps from $\boldsymbol{x}_0$ up to $\boldsymbol{x}_k$ we have

$$\boldsymbol{x}_k - \boldsymbol{x}_0 = \alpha_0\boldsymbol{d}_0 + \cdots + \alpha_{k-1}\boldsymbol{d}_{k-1}.$$

By the Q-orthogonality of the $\boldsymbol{d}_j$'s it follows that

$$\boldsymbol{d}_k^TQ(\boldsymbol{x}_k - \boldsymbol{x}_0) = 0.$$

Substituting the above into (4) we obtain that

$$\begin{aligned}
\alpha_k &= \frac{\boldsymbol{d}_k^TQ(\boldsymbol{x}^* - \boldsymbol{x}_k + \boldsymbol{x}_k - \boldsymbol{x}_0)}{\boldsymbol{d}_k^TQ\boldsymbol{d}_k} \\
&= \frac{\boldsymbol{d}_k^TQ(\boldsymbol{x}^* - \boldsymbol{x}_k)}{\boldsymbol{d}_k^TQ\boldsymbol{d}_k} \\
&= \frac{\boldsymbol{d}_k^T(Q\boldsymbol{x}^* - Q\boldsymbol{x}_k)}{\boldsymbol{d}_k^TQ\boldsymbol{d}_k} \\
&= \frac{\boldsymbol{d}_k^T(-\boldsymbol{b} - Q\boldsymbol{x}_k)}{\boldsymbol{d}_k^TQ\boldsymbol{d}_k} \\
&= -\frac{\boldsymbol{g}_k^T\boldsymbol{d}_k}{\boldsymbol{d}_k^TQ\boldsymbol{d}_k}.
\end{aligned}$$

To prove (ii), we show that $\boldsymbol{x}_{k+1}$ minimizes f over the linear variety $\boldsymbol{x}_0 + \mathcal{B}_{k+1}$, which contains the line $\boldsymbol{x} = \boldsymbol{x}_k + \alpha\boldsymbol{d}_k$. Since the quadratic function f is strictly convex, a local minimum is also a global one. So the conclusion will hold if it can be shown that the gradient $\boldsymbol{g}_{k+1}$ is orthogonal to $\mathcal{B}_{k+1}$, that is, if the gradient is orthogonal to $\boldsymbol{d}_0, \boldsymbol{d}_1, \dots, \boldsymbol{d}_k$.[1] We prove this by induction. The hypothesis is true for $k = 0$ since $\mathcal{B}_0$ is empty. Assume that $\boldsymbol{g}_k \perp \mathcal{B}_k$. We have

$$\begin{aligned}
\boldsymbol{g}_{k+1} &= Q\boldsymbol{x}_{k+1} + \boldsymbol{b} \\
&= Q(\boldsymbol{x}_k + \alpha_k\boldsymbol{d}_k) + \boldsymbol{b} \\
&= Q\boldsymbol{x}_k + \boldsymbol{b} + \alpha_kQ\boldsymbol{d}_k \\
&= \boldsymbol{g}_k + \alpha_kQ\boldsymbol{d}_k,
\end{aligned}$$

and hence by definition of α_k

$$\begin{aligned}
\boldsymbol{d}_k^T\boldsymbol{g}_{k+1} &= \boldsymbol{d}_k^T\boldsymbol{g}_k + \alpha_k\boldsymbol{d}_k^TQ\boldsymbol{d}_k \\
&= \boldsymbol{d}_k^T\boldsymbol{g}_k - \frac{\boldsymbol{g}_k^T\boldsymbol{d}_k}{\boldsymbol{d}_k^TQ\boldsymbol{d}_k}\boldsymbol{d}_k^TQ\boldsymbol{d}_k \\
&= \boldsymbol{d}_k^T\boldsymbol{g}_k - \boldsymbol{g}_k^T\boldsymbol{d}_k \\
&= 0.
\end{aligned}$$

Also it holds that

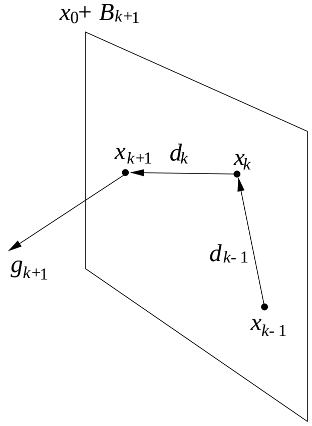
$$\boldsymbol{d}_i^T\boldsymbol{g}_{k+1} = \boldsymbol{d}_i^T\boldsymbol{g}_k + \alpha_k\boldsymbol{d}_i^TQ\boldsymbol{d}_k, \qquad \text{for } i < k.$$

The first term on the right hand side of the above equation vanishes due to the induction hypothesis, while the second term vanishes by the Q-orthogonality of the $\boldsymbol{d}_i$'s. Thus $\boldsymbol{g}_{k+1} \perp \mathcal{B}_{k+1}$. □

[1] Otherwise, we can always move along some direction in $\mathcal{B}_{k+1}$ to decrease f.

Corollary 3 *The gradients $\boldsymbol{g}_k$, $k = 0, 1, \ldots, n$, satisfy $\boldsymbol{g}_k \perp \mathcal{B}_k$.*

This theorem tells us that the conjugate gradient algorithm really is a generalization of steepest descent. Each step of adding $\alpha_k \boldsymbol{d}_k$ to the previous estimate is the same as doing a line minimization along the direction of $\boldsymbol{d}_k$. Furthermore, the offset $\alpha_k \boldsymbol{d}_k$ does not undo previous progress, that is, the minimization is in fact a minimization over $\boldsymbol{x}_0 + \mathcal{B}_{k+1}$.



So the $\mathcal{B}_k$'s form a sequence of subspace with $\mathcal{B}_k \subset \mathcal{B}_{k+1}$. Because $\boldsymbol{x}_k$ minimizes f over $\boldsymbol{x}_0 + \mathcal{B}_k$, it is clear that $\boldsymbol{x}_n$ minimizes f over the entire space $\mathbb{R}^n = \mathcal{B}_n$.

4 Conjugate Gradient Algorithm

The conjugate gradient algorithm selects the successive direction vectors as a conjugate version of the successive gradients obtained as the method progresses. Thus, the directions are not specified beforehand, but rather are determined sequentially at each step of the iteration. At step k one evaluates the current negative gradient vector and adds to it a linear combination of the previous direction vectors to obtain a new conjugate direction vector along which to move.

There are three primary advantages to this method of direction selection. First, unless the solution is attained in less than n steps, the gradient is always nonzero and linearly independent of all previous direction vectors. Indeed, as the corollary states, the gradient $\boldsymbol{g}_k$ is orthogonal to the subspace $\mathcal{B}_k$ generated by $\boldsymbol{d}_0, \boldsymbol{d}_1, \ldots, \boldsymbol{d}_{k-1}$. If the solution is reached before n steps are taken, the gradient vanishes and the process terminates.

Second, a more important advantage of the conjugate gradient method is the especially simple formula that is used to determine the new direction vector. This simplicity makes the method only slightly more complicated than steepest descent.

Third, because the directions are based on the gradients, the process makes good uniform progress toward the solution at every step. This is in contrast to the situation for arbitrary sequences of conjugate directions in which progress may be slight until the final few steps. Although for the pure quadratic problem uniform progress is of no great importance, it is important for generalizations to nonquadratic problems.

Conjugate Gradient Algorithm

```
1. $\boldsymbol{g}_0 \leftarrow Q\boldsymbol{x}_0 + \boldsymbol{b}$
2. $\boldsymbol{d}_0 \leftarrow -\boldsymbol{g}_0$
3. for $k = 0, \ldots, n-1$ do
       a) $\alpha_k \leftarrow -\frac{\boldsymbol{g}_k^T \boldsymbol{d}_k}{\boldsymbol{d}_k^T Q \boldsymbol{d}_k}$
       b) $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$
       c) $\boldsymbol{g}_{k+1} \leftarrow Q\boldsymbol{x}_{k+1} + \boldsymbol{b}$
       d) $\beta_k \leftarrow \frac{\boldsymbol{g}_{k+1}^T Q \boldsymbol{d}_k}{\boldsymbol{d}_k^T Q \boldsymbol{d}_k}$
       e) $\boldsymbol{d}_{k+1} \leftarrow -\boldsymbol{g}_{k+1} + \beta_k \boldsymbol{d}_k$
4. return $\boldsymbol{x}_n$
```

Step 3b) when $k = 0$ is a steepest descent. Each subsequent step moves in a direction that modifies the opposite of the current gradient by a factor of the previous direction. Step 3a)–e) gives us the Q-orthogonality of the descent vectors $\boldsymbol{d}_0, \ldots, \boldsymbol{d}_{n-1}$.

Theorem 4 (Conjugate Gradient Theorem) *In the conjugate gradient algorithm, we have that*

a) $\mathrm{span}\{\boldsymbol{g}_0, \ldots, \boldsymbol{g}_k\} = \mathrm{span}\{\boldsymbol{g}_0, Q\boldsymbol{g}_0, \ldots, Q^k\boldsymbol{g}_0\}$

b) $\mathrm{span}\{\boldsymbol{d}_0, \ldots, \boldsymbol{d}_k\} = \mathrm{span}\{\boldsymbol{g}_0, Q\boldsymbol{g}_0, \ldots, Q^k\boldsymbol{g}_0\}$

c) $\boldsymbol{d}_k^T Q \boldsymbol{d}_i = 0$ *for all* $i < k$

d) $\alpha_k = \frac{\boldsymbol{g}_k^T \boldsymbol{g}_k}{\boldsymbol{d}_k^T Q \boldsymbol{d}_k}$

e) $\beta_k = \frac{\boldsymbol{g}_{k+1}^T \boldsymbol{g}_{k+1}}{\boldsymbol{g}_k^T \boldsymbol{g}_k}$

For proof of the theorem we refer to [2, pp. 245–246]. Part c) of the above theorem states that the $\boldsymbol{d}_i$'s are Q-orthogonal to each other. Part e) is very important, because it provides us a way to compute β_k without knowing Q.

5 Extension to Nonquadratic Problems

How do we compute α_k without knowing Q? The Expanding Subspace Theorem already gave us the answer — a line search. This agrees with the formula in the quadratic case.

We can generalize the conjugate gradient algorithm to devise a numerical routine to minimize an arbitrary function f. Here the Hessian of f plays the role of Q. The algorithm executes groups of n search steps. Each step builds a coordinate $\alpha_i \boldsymbol{d}_i$ in a search for the minimum $\boldsymbol{x}^*$. After n steps, the algorithm resets, using its current $\boldsymbol{x}$ location as a new "origin" from which to start another n-step search.

Fletcher-Reeves Algorithm

```
1. start at some $\boldsymbol{x}_0$
2. $\boldsymbol{d}_0 \leftarrow -\nabla f(\boldsymbol{x}_0)$
3. for $k = 0, 1, \ldots, n-1$ do
       a) obtain $\alpha_k$ that minimizes $g(\alpha) = f(\boldsymbol{x}_k + \alpha \boldsymbol{d}_k)$
       b) $\boldsymbol{x}_{k+1} \leftarrow \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$
       c) $\beta_k \leftarrow \dfrac{\|\nabla f(\boldsymbol{x}_{k+1})\|^2}{\|\nabla f(\boldsymbol{x}_k)\|^2}$
       d) $\boldsymbol{d}_{k+1} \leftarrow -\nabla f(\boldsymbol{x}_{k+1}) + \beta_k \boldsymbol{d}_k$
4. $\boldsymbol{x}_0 \leftarrow \boldsymbol{x}_n$
5. go back to step 2 until satisfied with the results.
```

To determine β_k the algorithm employs part e) of Theorem 4. Step 2 ensures that there is at least one descent direction in every n iterations. Steps 3a) and 3b) ensure that no step increases f.

Global convergence of the line search methods is established by noting that a pure steepest descent step is taken every n steps and serves as a spacer step. Since the other steps do not increase the objective, and in fact hopefully they decrease it, global convergence is assured. The restarting aspect of the algorithm is important for global convergence analysis, since in general one cannot guarantee that the directions $\boldsymbol{d}_k$ generated by the method are descent directions.

The local convergence properties of nonquadratic extensions of the conjugate gradient method can be inferred from the quadratic analysis. Assuming that at the solution, $\boldsymbol{x}^*$, the Hessian ∇f is positive definite, we expect the asymptotic convergence rate per step to be at least as good as steepest descent, since this is true in the quadratic case. In addition to this bound on the single step rate we expect that the method is of order two with respect to each complete cycle of n step. In other words, since one complete cycle solves a quadratic problem exactly just as Newton's method does in one step, we expect that for general nonquadratic problems there will hold

$$\|\boldsymbol{x}_{k+n} - \boldsymbol{x}^*\| \leq c\|\boldsymbol{x}_k - \boldsymbol{x}^*\|^2$$

for some c and $k = 0, n, 2n, \ldots$. This can indeed be proved, and of course underlies the original motivation for the method.

6 Conclusion

Recall that in finding a minimum of f of n variables, we may wish to consider the set of zeros of $f' = \nabla f$. In principle, we could apply Newton's method to ∇f, resulting in the following iteration formula:

$$\boldsymbol{x}^{(m+1)} \quad = \quad \boldsymbol{x}^{(m)} - \left(Hf\left(\boldsymbol{x}^{(m)}\right)\right)^{-1} \nabla f\left(\boldsymbol{x}^{(m)}\right).$$

Suppose f is a quadratic function with the form

$$f(\boldsymbol{x}) = c + \boldsymbol{b}^T \boldsymbol{x} + \frac{1}{2}\boldsymbol{x}^T A \boldsymbol{x},$$

where A is symmetric positive definite. Then $\nabla f = \boldsymbol{b} + A\boldsymbol{x}$ and $Hf = A$, and the global minimum of f satisfies $A\boldsymbol{x} = -\boldsymbol{b}$. In this case, Newton's method converges in a single step. But for general f, the Hessian Hf often is unknown. To remedy this, there exist methods called Quasi-Newton methods that build $(Hf)^{-1}$ iteratively as they move.

Conjugate Gradient is an intermediate between steepest descent and Newton's method. It tries to achieve the quadratic convergence of Newton's method without incurring the cost of computing Hf. At the same time, Conjugate Gradient will execute at least one gradient descent step per n steps. It has proved to be extremely effective in dealing with general objective functions and is considered among the best general purpose methods presently available.

References

[1] M. Erdmann. Lecture notes for *16-811 Mathematical Fundamentals for Robotics.* The Robotics Institute, Carnegie Mellon University, 1998.

[2] D. G. Luenberger. *Linear and Nonlinear Programming.* Addison-Wesley, 2nd edition, 1984.