Higher-Order Methods

Stephen J. Wright[1]

[2]Computer Sciences Department,
University of Wisconsin-Madison.

PCMI, July 2016

Smooth Functions

Consider $\min_{x \in \mathbb{R}^n} f(x)$, with f smooth.

Usually assume f twice continuously differentiable.

(Sometimes assume convexity too.)

- Newton's method
- Enhancing Newton's method for "global convergence"
  - Line Search
  - Trust Regions
- Third-order regularization, and complexity estimates.
- Quasi-Newton Methods.

Newton's Method

Assume that $\nabla^2 f$ is Lipschitz continuous:

$$\|\nabla^2 f(x') - \nabla^2 f(x'')\| \le M\|x' - x''\|. \tag{1}$$

Second-order Taylor-series approximation is

$$f(x^k + p) = f(x^k) + \nabla f(x^k)^T p + \frac{1}{2} p^T \nabla^2 f(x^k) p + O(\|p\|^3). \tag{2}$$

When $\nabla^2 f(x^k)$ is positive definite, can choose p to minimize the quadratic

$$p^k = \arg\min_p \; f(x^k) + \nabla f(x^k)^T p + \frac{1}{2} p^T \nabla^2 f(x^k) p,$$

which is

$$p^k = -\nabla^2 f(x^k)^{-1} \nabla f(x^k) \quad \text{Newton step!}$$

Thus, basic form of Newton's method is

$$x^{k+1} = x^k - \nabla^2 f(x^k)^{-1} \nabla f(x^k). \tag{3}$$

Local Convergence of Newton's Method

Assume solution x^* satisfying second-order sufficient conditions:

$$\nabla f(x^*) = 0, \quad \nabla^2 f(x^*) \text{ positive definite.}$$

For f strongly convex at solution x^*, can prove local quadratic convergence.

Theorem

If $\|x^0 - x^*\| \le \frac{m}{2M}$, we have

$$x^k \to x^* \text{ and } \|x^{k+1} - x^*\| \le \frac{M}{m}\|x^k - x^*\|^2, \quad k = 0, 1, 2, \ldots .$$

Get ϵ reduction in $\log \log \epsilon$ iterations!

($\log \log \epsilon$ is bounded by 5 for all interesting ϵ!).

Proof

$$
\begin{aligned}
x^{k+1} - x^* &= x^k - x^* - \nabla^2 f(x^k)^{-1} \nabla f(x^k) \\
&= \nabla^2 f(x^k)^{-1} [\nabla^2 f(x^k)(x^k - x^*) - (\nabla f(x^k) - \nabla f(x^*))].
\end{aligned}
$$

so that

$$
\|x^{k+1} - x^*\| \le \|\nabla f(x^k)^{-1}\| \|\nabla^2 f(x^k)(x^k - x^*) - (\nabla f(x^k) - \nabla f(x^*))\|.
$$

From Taylor's theorem:

$$
\nabla f(x^k) - \nabla f(x^*) = \int_0^1 \nabla^2 f(x^k + t(x^* - x^k))(x^k - x^*) \, dt.
$$

From Lipschitz continuity of $\nabla^2 f$, we have

$$\begin{aligned}
&\|\nabla^2 f(x^k)(x^k - x^*) - (\nabla f(x^k) - \nabla f(x^*))\| \\
&\quad = \left\| \int_0^1 [\nabla^2 f(x^k) - \nabla^2 f(x^k + t(x^* - x^k))](x^k - x^*)\, dt \right\| \\
&\quad \le \int_0^1 \|\nabla^2 f(x^k) - \nabla^2 f(x^k + t(x^* - x^k))\| \|x^k - x^*\| \, dt \\
&\quad \le \left(\int_0^1 Mt\, dt \right) \|x^k - x^*\|^2 = \tfrac{1}{2} M \|x^k - x^*\|^2. \qquad (4)
\end{aligned}$$

From Weilandt-Hoffman inequality: that

$$|\lambda_{\min}(\nabla^2 f(x^k)) - \lambda_{\min}(\nabla^2 f(x^*))| \le \|\nabla^2 f(x^k) - \nabla^2 f(x^*)\| \le M\|x^k - x^*\|,$$

Thus for

$$\|x^k - x^*\| \le \frac{m}{2M}, \qquad (5)$$

we have

$$\lambda_{\min}(\nabla^2 f(x^k)) \ge \lambda_{\min}(\nabla^2 f(x^*)) - M\|x^k - x^*\| \ge m - M\frac{m}{2M} \ge \frac{m}{2},$$

so that $\|\nabla^2 f(x^k)^{-1}\| \le 2/m$. Thus

$$\|x^{k+1} - x^*\| \le \frac{2}{m}\frac{M}{2}\|x^k - x^*\|^2 = \frac{M}{m}\|x^k - x^*\|^2,$$

verifying the locally quadratic convergence rate. By applying (5) again, we have

$$\|x^{x+1} - x^*\| \le \left(\frac{M}{m}\|x^k - x^*\|\right)\|x^k - x^*\| \le \frac{1}{2}\|x^k - x^*\|,$$

so, by arguing inductively, we see that the sequence converges to x^* provided that x^0 satisfies (5), as claimed.

Enhancing Newton's Method

Newton's method converges rapidly once the iterates enter the neighborhood of a point x^* satisfying second-order optimality conditions. But what happens when we start far from such a point?

For nonconvex f, $\nabla^2 f(x^k)$ may be indefinite, so the Newton direction $p^k = -\nabla^2 f(x^k)^{-1} \nabla f(x^k)$ may not even be a descent direction

Some modifications ensure that Newton directions are descent directions, so when embedded in a line-search framework (with e.g. Wolfe conditions) we can get the same guarantees as for general line-search methods.

- convex f: Modified search direction + line search yields $1/k$ rate.
- strongly convex f: No modification needed to direction. Addition of line search yields global linear rate.
- nonconvex f: Modified search direction + line search yields $\|\nabla f(x^k)\| \to 0$.

Newton on strongly convex f

Eigenvalues of $\nabla^2 f(x^k)$ uniformly in the interval $[m, L]$, with $m > 0$.

Newton direction is a descent direction. Proof: Note first that

$$\|p^k\| \le \|\nabla^2 f(x^k)^{-1}\| \|\nabla f(x^k)\| \le \frac{1}{m} \|\nabla f(x^k)\|.$$

Then

$$\begin{aligned} -(p^k)^T \nabla f(x^k) &= \nabla f(x^k)^T \nabla^2 f(x^k)^{-1} \nabla f(x^k) \\ &\ge \frac{1}{L} \|\nabla f(x^k)\|^2 \\ &\ge \frac{m}{L} \|\nabla f(x^k)\| \|p^k\|. \end{aligned}$$

Apply line-search techniques described earlier (e.g. with weak Wolfe conditions) to get $1/T$ convergence.

Want to ensure that the convergence rate becomes quadratic near the solution x^*. Can do this by always trying $\alpha_k = 1$ first, as a candidate step length, and accepting if it satisfies the sufficient decrease condition.

Newton on weakly convex f

When $m = 0$, the Newton direction may not exist. But can modify by

- Adding elements to the diagonal of $\nabla^2 f(x^k)$ while factorizing it during the calculation of p^k;
- Defining search direction to be

$$d^k = -[\nabla^2 f(x^k) + \lambda_k I]^{-1} \nabla f(x^k),$$

  for some $\lambda_k > 0$.

These strategies ensure that d^k is a descent direction, so line search strategy can be applied to get $\|\nabla f(x^k)\| \to 0$.

If $\nabla^2 f(x^*)$ is nonsingular, can recover local quadratic convergence if the algorithm ensures $\lambda_k \to 0$ and $\alpha_k \to 1$.

Newton on nonconvex f

Use similar strategies as for weakly convex to obtain a modified Newton descent direction d^k.

Use line searches to ensure that $\|\nabla f(x^k)\| \to 0$. This implies that all accumulation points are stationary, that is, have $\nabla f(x^*) = 0$.

The modification scheme should recognize when $\nabla^2 f(x^k)$ is positive definite, and try to take pure Newton steps (with $\alpha_k = 1$) at such points, to try for quadratic convergence to a local minimizer.

But in general can only guarantee stationarity of accumulation points, which may be saddle points.

Newton + Trust-Regions

Trust regions gained much currency in early 1980s. Interest revived recently becase of their ability to escape from saddle points.

Basic idea: See minimum over the quadratic model for f around x^k in a ball of radius Δ_k around x^k.

The trust-region subproblem is

$$\min_d f(x^k) + \nabla f(x^k)^T d + \frac{1}{2} d^T \nabla^2 f(x^k) d \quad \text{subject to } \|d\|_2 \le \Delta_k.$$

Shocking fact: This problem is easy to solve, even when $\nabla^2 f(x^k)$ is indefinite! Its solution satisfies these equations for some $\lambda > 0$:

$$[\nabla^2 f(x^k) + \lambda I] d^k = -\nabla f(x^k), \quad \text{for some } \lambda > 0.$$

Same as for one of the modified Newton directions described above. Solve the subproblem by doing a search for λ.

- Line search approach: choose direction d^k then length α_k;
- Trust-region approach: choose length Δ_k then direction d^k.

Trust-Region

Δ_k plays a similar role to line-search parameter α_k.

- Accept step if a sufficient decrease condition is satisfied.
- Reject step otherwise, and recalculate with a smaller value of Δ_k.
- After a successful step, increase Δ_k for the next iteration.

Trust-region can escape from saddle points. If $\nabla f(x^k) \approx 0$, d^k will tend to be in the direction of "most negative curvature" for $\nabla^2 f(x^k)$.

Normal trust-region heuristics ensure that the bound is inactive in the neighborhood of a point x^* satisfying second-order sufficient conditions. Then d^k becomes the pure Newton step, and quadratic convergence ensues.

Cubic Regularization

Suppose that the Hessian is Lipschitz continuous with constant M:

$$\|\nabla^2 f(x') - \nabla^2 f(x'')\| \le M\|x' - x''\|.$$

Then the following cubic expansion yields an upper bound for f:

$$T_M(z;x) := f(x) + \nabla f(x)^T(z-x) + \frac{1}{2}(z-x)^T\nabla^2 f(x)(z-x) + \frac{M}{6}\|z-x\|^3.$$

We have $f(z) \le T_M(z;x)$ for all z, x.

The basic cubic regularization algorithm sets

$$x^{k+1} = \arg\min_z T_M(z;x^k).$$

Nesterov and Polyak (2006); see also Griewank (1981), Cartis et al. (2011a,b). This can also escape from saddle points, and comes with some complexity guarantees.

Complexity

Assume that f is bounded below by $\bar{f}$. Then cubic regularization has the following guarantees: Finds x^k for which

$$\begin{aligned} \|\nabla f(x^k)\| \le \epsilon \quad & \text{within } k = O(\epsilon^{-3/2}) \text{ iterations;} \\ \nabla^2 f(x^k) \ge -\epsilon I \quad & \text{within } k = O(\epsilon^{-3}) \text{ iterations,} \end{aligned}$$

where the constants in $O(\cdot)$ depend on $[f(x^0) - \bar{f}]$ and M.

Thus we can guarantee an "approximate second-order necessary point," which is "likely" to be a local minimizer.

We can design a very simple algorithm — a modification of steepest descent — with only slightly inferior complexity.

Steve's Brain-Dead Second-Order Necessary Solver

Given $\epsilon > 0$, and f with the following properties:

- bounded below;
- ∇f has Lipschitz constant L;
- $\nabla^2 f$ has Lipschitz constant M.

Algorithm SBDSONS:

- When $\|\nabla f(x^k)\| \geq \epsilon$, take a steepest descent step, say with steplength $\alpha_k = 1/L$. This yields a reduction in f of at least

$$\frac{1}{2L}\|\nabla f(x^k)\|^2 \geq \frac{\epsilon^2}{2L}.$$

- When $\|\nabla f(x^k)\| < \epsilon$, evaluate $\nabla^2 f(x^k)$ and its eigenvalues.
  - If the smallest eigenvalue is $\geq -\epsilon$, stop. Success!
  - If not, calculate the unit direction of most negative curvature p^k, and flip its sign if necessary to ensure that $(p^k)^T \nabla f(x^k) \leq 0$.

Steplength for the negative-curvature direction

From the cubic upper bound, we find the steplength α_k to move along p^k.

$$
\begin{aligned}
f(x^k + \alpha p^k) &\le f(x^k) + \alpha \nabla f(x^k)^T p^k + \frac{1}{2}\alpha^2 (p^k)^T \nabla^2 f(x^k) p^k + \frac{M}{6}\alpha^3 \|p^k\|^3 \\
&\le f(x^k) - \frac{1}{2}\alpha^2 \epsilon + \frac{M}{6}\alpha^3.
\end{aligned}
$$

By minimizing the right hand side, we obtain $\alpha_k = 2\epsilon/M$, for which

$$
f(x^k + \alpha_k p^k) \le f(x^k) - \frac{2}{3}\frac{\epsilon^3}{M^2}.
$$

Complexity analysis: The algorithm can encounter at most

$$
\frac{2L}{\epsilon^2}(f(x^0) - \bar{f}) = O(\epsilon^{-2})
$$

iterates with $\|\nabla f(x^k)\| \ge \epsilon$, and at most

$$
\frac{3M^2}{2\epsilon^3}(f(x^0) - \bar{f}) = O(\epsilon^{-3})
$$

iterates with $\nabla^2 f(x^k)$ having eigenvalues less than $-\epsilon$.

Quasi-Newton Methods

Idea: Build up approximation B_k to the Hessian $\nabla^2 f(x^k)$, using information gathered during the iterations.

Key Observation: If we define

$$s^k = x^{k+1} - x^k, \quad y^k = \nabla f(x^{k+1}) - \nabla f(x^k),$$

then Taylor's theorem implies that

$$\nabla^2 f(x^{k+1}) s^k \approx y^k.$$

We require B_{k+1} to satisfy this secant equation:

$$B_{k+1} s^k = y^k, \quad \text{where } \; s^k = x^{k+1} - x^k, \;\; y^k = \nabla f(x^{k+1}) - \nabla f(x^k).$$

Derive formulae for updating B_k, $k = 0, 1, 2, \ldots$, to satisfy this property, and several other desirable properties.

Desirable Properties

Use B_k as a proxy for $\nabla^2 f(x^k)$ in computation of the step:

$$p^k = -B_k^{-1} \nabla f(x^k).$$

Other desirable properties of B_k:

- Simple, low-cost update formulas $B_k \to B_{k+1}$;
- Symmetric (like the true Hessian);
- Positive definiteness (so that p^k is guaranteed to be a descent direction).

A necessary condition for positive definiteness is that $(y^k)^T s^k > 0$.
(Proof: If $(y^k)^T s^k \le 0$ we have from secant equation that

$$0 \ge (y^k)^T s^k = (s^k)^T B_k^T s^k,$$

so that B_k is not positive definite.) However we can guarantee positive definiteness if the Wolfe conditions hold for α_k:

$$\nabla f(x^k + \alpha_k p^k)^T p^k \ge c_2 \nabla f(x^k)^T p^k, \quad \text{for some } c_2 \in (0,1).$$

It follows that $(y^k)^T s^k \ge (c_2 - 1)\alpha_k \nabla f(x^k)^T p^k > 0$.

DFP and BFGS

DFP (1960s) and BFGS (1970) methods use rank-2 updates that satisfy the secant equation and maintain positive definiteness and symmetry.

Defining $\rho_k = (y^k)^T s^k) > 0$, we have:

$$DFP: \quad B_{k+1} = (I - \rho_k y^k (s^k)^T) B_k (I - \rho_k s^k (y^k)^T) + \rho_k (y^k)(y^k)^T$$

$$BFGS: \quad B_{k+1} = B_k - \frac{B_k s^k (s^k)^T B_k}{(s^k)^T B_k s^k} + \frac{y^k (y^k)^T}{(y^k)^T s^k}.$$

These two formulae are closely related. Suppose that instead of maintaining $B_k \approx \nabla^2 f(x^k)$ we maintain instead an inverse approximation $H_k \approx \nabla^2 f(x^k)^{-1}$. (This has the advantage that step compuation is a simple matrix-vector multiplication: $p^k = -H_k \nabla f(x^k)$.)

If we make the replacements:

$$H_k \leftrightarrow B_k, \quad s^k \leftrightarrow y^k,$$

then the DFP updated applied to H_k corresponds to the BFGS update applied to B_k, and vice versa.

Other Motivations and Properties

These updates are least-change updates in certain norms:

$$B_{k+1} := \arg\min_B \|B - B_k\| \;\; s.t. \;\; B = B^T, \;\; Bs^k = y^k.$$

They generalize to the Broyden class, which is the convex combination of DFP and BFGS.

BFGS performs significantly better in practice. WHY??? Some explanations were given by Powell in the mid-1980s.

Remarkably, these methods converge superlinearly to solutions satisfying second-order sufficient conditions:

$$\|x^{k+1} - x^*\| = o(\|x^k - x^*\|).$$

(Analysis is quite technical.)

See (Nocedal and Wright, 2006, Chapter 6).

Limited-Memory BFGS

An important issue with DFP and BFGS for large-scale problems is that the matrices B_k and H_k are $n \times n$ dense, even if the true Hessian $\nabla^2 f(x^k)$ is sparse. Hence require $O(n^2)$ storage and cost per iteration — could be prohibitive.

But note that we can store B_k (or H_k) implicitly, by storing B_0, and $s^0, s^1, \dots, s^k$ and $y^0, y^1, \dots, y^k$. If B_0 is a multiple of the identity, have

- total storage is about $2kn$ (ccan be reduced to kn if careful);
- Work to compute p^k is also $O(kn)$. Can design a simple recursive scheme based on the update formula.

This is all fine provided that $k \ll n$, but we typically need more iterations than this.

Solution: Don't store all k updates so far, just the last m updates, for small m. ($m = 3, 5, 10, 20$ are values that I have seen in applications)

L-BFGS

L-BFGS has become standard method in large-scale smooth nonlinear optimization.

- Rotate storage — at each iterations, latest s^k, y^k replaces oldest stored values s^{k-m}, y^{k-m},
- No convergence rate guarantee beyond the linear rate associated with descent methods.
- Can be viewed as an extension of nonlinear conjugate gradient, with more memory.
- Can rescale the choice of B_0 at each iteration. Often use a Barzilai-Borwein type scaling, e.g. $B_0 = (s^k)^T y^k / (s^k)^T s^k$.

Liu and Nocedal (1989), (Nocedal and Wright, 2006, Chapter 7).

L-BFGS Details

Use the inverse form: $H_k \approx \nabla^2 f(x^k)^{-1}$.

$$x^{k+1} = x^k - \alpha_k H_k \nabla f(x^k).$$

Update formula:

$$H_{k+1} = V_k^T H_k V_k + \rho_k s^k (s^k)^T,$$

where

$$\rho_k = 1/(y^k)^T s^k, \quad V_k = I - \rho_k y^k (s^k)^T.$$

Uses

$$H_0 = \gamma_k I, \quad \gamma_k = (s^{k-1})^T y^{k-1} / (y^{k-1})^T y^{k-1}.$$

See (Nocedal and Wright, 2006, p. 178) for a two-loop recursion to compute $H_k \nabla f(x^k)$.

References I

Cartis, C., Gould, N. I. M., and Toint, P. L. (2011a). Adaptive cubic regularisation methods for unconstrained optimization. Part I: motivation, convergence and numerical results. *Mathematical Programming, Series A*, 127:245–295.

Cartis, C., Gould, N. I. M., and Toint, P. L. (2011b). Adaptive cubic regularisation methods for unconstrained optimization. part ii: worst-case function-and derivative-evaluation complexity. *Mathematical Programming, Series A*, 130(2):295–319.

Griewank, A. (1981). The modification of Newton's method for unconstrained optimization by bounding cubic terms. Technical Report NA/12, DAMTP, Cambridge University.

Liu, D. C. and Nocedal, J. (1989). On the limited-memory BFGS method for large scale optimization. *Mathematical Programming, Series A*, 45:503–528.

Nesterov, Y. and Polyak, B. T. (2006). Cubic regularization of Newton method and its global performance. *Mathematical Programming, Series A*, 108:177–205.

Nocedal, J. and Wright, S. J. (2006). *Numerical Optimization*. Springer, New York.