Line Search Concepts

The Line Search strategy for finding local optimal minimizers of $f : \mathbb{R}^n \to \mathbb{R}$ is based on the idea that if $\nabla f(x) \neq 0$ then some descent direction exists which we can use to find an improved point y satisfying $f(y) < f(x)$. The basic algorithm is as follows.

1. Given: initial point $x_0 \in \mathbb{R}^n$.
2. Set: iteration counter $k = 0$.
3. Find descent direction $p_k \in \mathbb{R}^n$.
4. Solve $\alpha = \arg\min_{\beta>0} \ f(x_k + \beta p_k)$.
5. Set $x_{k+1} = x_k + \alpha p_k$.
6. Set $k \leftarrow k + 1$, either go to step 3 or stop.

The algorithm is deceptively simple. There are some interesting points to discuss that make even this algorithm challenging. We will need to clarify steps 3, 4 and 6, as well as think about efficiency.

Choosing a Descent Direction

We can find descent directions by considering Taylor's Theorem applied at current point x_k:

$$f(x_k + \alpha p) = f(x_k) + \alpha p^\mathsf{T} \nabla f(x_k) + \tfrac{1}{2}\alpha^2 p^\mathsf{T} \nabla^2 f(x_k + tp)p, \quad \text{for some } t \in (0, \alpha).$$

The initial rate of change of f at x_k, relative to step size α is

$$p^\mathsf{T} \nabla f(x_k) = \|p\| \, \|\nabla f(x_k)\| \cos\theta$$

where θ is the angle between vectors p and $\nabla f(x_k)$. We want this rate of change to be as large (negatively) as possible for a fixed vector length say $\|p\| = 1$. That is, we want $\cos(\theta) = -1$ and therefore $p \propto -\nabla f(x_k)$. The particular choice $p = -\nabla f(x_k)$ is the strategy of the method of **gradient descent**.

We will explore other options soon. But for now consider one other. We could use a second order Taylor approximation of the function at x_k:

$$m(p) := f(x_k) + p^\mathsf{T} \nabla f(x_k) + \tfrac{1}{2} p^\mathsf{T} \nabla^2 f(x_k) p \approx f(x_k + p),$$

and try to find the step that minimizes this quadratic model function. The stationary points are the solutions to $\nabla m(p) = 0$. We have

$$\nabla m(p) = \nabla f(x_k) + \nabla^2 f(x_k) p \stackrel{set}{=} 0$$

with solution

$$p = -\left[\nabla^2 f(x_k)\right]^{-1} \nabla f(x_k)$$

if $\nabla f(x_k)$ is invertible. This choice of p is the strategy of the **Newton's Method**. As a general rule, for problems which are locally convex, Newton's Method outperforms gradient descent in terms of number of iterations (number of p's computed). However, calculating the step is far more computationally intensive. Newton's Method can also provide a direction p which is *not* a descent direction! Instead, it seeks to step to a stationary point, which could be a local maximizer, a saddle point, or a local minimizer. As such, some care is needed when using this method.

A general, gradient-based, form of a direction of descent is $P_k = -B_k^{-1}\nabla f(x_k)$ where the choice of B_k depends on the method. For gradient descent, $B_k = I_n$. For Newton's Method, $B_k = \nabla^2 f(x_k)$ as long as $\nabla^2 f(x_k)$ is positive definite. Later, we will explore other very useful options: conjugate gradient and quasi-Newton Methods.

Solving the Subproblem

Once we have a descent direction, we must then solve the subproblem

$$\alpha = \arg\min_{\beta>0} f(x_k + \beta p_k)$$

to determine the step size α. This is the central problem of the Line Search Strategy. This subproblem is a one-variable problem - we are searching along a line through x_k in the direction p and looking for the minimizer along this particular path (a ray actually since $\alpha > 0$). However, even this problem can be very difficult to solve exactly. Fortunately, we rarely need an exact solve. What we need is a *sufficiently improved point*, so we can solve the subproblem approximately with little concern for how accurate we are. This is because the new iterate x_{k+1} is only a step towards the optimal point.

The notion of *sufficient improvement* is illustrated by two examples. First consider the problem

$$\min_{x\in\mathbb{R}} \quad z = f(x) = x^2$$

with solution $x^* = 0$, $f^* = f(x^*) = 0$. The sequence of points $\{x_0 = 2, x_1 = 1.5, x_2 = 1.25, \ldots, x_k = 1 + 2^{-k}, \ldots\}$ satisfies a decrease condition $f(x_{k+1}) < f(x_k)$ for all k, but fails to converge to the minimizer (it converges to $x = 1$). The problem here is that the step sizes become too small too soon. In this case, $\alpha_k = 2^{-(k+1)}$ and $\sum_{k=0}^{\infty} \alpha_k = 1$, so this particular search could never reach beyond 1 unit from the starting point $x = 2$.

For the second example, we consider the similar problem

$$\min_{x\in\mathbb{R}^2} \quad z = f(x) = \|x\|$$

with solution $x^* = (0, 0)$, $z^* = 0$. The sequence of points

$$x_k = \left(1 + 2^{-k}\right) \begin{bmatrix} \cos k \\ \sin k \end{bmatrix}$$

satisfies a decrease condition $f(x_{k+1}) < f(x_k)$ for all k, but fails to converge to the minimizer. In fact, the sequence does not converge. The problem is not the step size, which is bounded below by 1, but rather a poor choice of descent directions. The problem here is that along the descent path, possible objective changes get successively smaller too quickly. That is, the descent direction is almost orthogonal to the gradient and the step size stays too large.

In order to avoid these two potential difficulties (disasters really), we employ so-called *sufficient decrease conditions*. Various conditions can be found in the literature, but we will focus on the **Strong Wolfe Conditions**. The first condition, also known as the **Armijo Condition**, requires that the improvement in objective value is at least some non-zero fraction of that predicted by the local gradient:

$$f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha p_k^\mathsf{T} \nabla f(x_k), \quad \text{for some } c_1 \in (0, 1).$$

As we will see later, this condition alone is sufficient to guarantee convergence to a local minimizer as long as one is careful to not choose α too small too quickly. A more robust way to enforce sufficiently large step sizes is to apply the **Curvature Condition**:

$$|p_k^\mathsf{T} \nabla f(x_k + \alpha p_k)| \geq c_2 |p_k^\mathsf{T} \nabla f(x_k)|, \quad \text{for some } c_2 \in (c_1, 1).$$

This condition enforces a step for which the slope at the new point is less steep (in the search direction p) than at the current point by at least a factor c_2.

Backtracking Line Search. Another method of enforcing sufficiently long steps is the simple method of reducing the candidate step size by a multiplicative factor in $(0, 1)$ until the Armijo condition is satisfied. This method may be efficient for some problems and is particularly suitable for descent directions derived from a full Newton Method with intial step size $\alpha = 1$.

Wolfe Condition Line Search. A more general method for finding suitable step sizes is given by Algorithms 3.5 and 3.6 in the text. Together, these procedures will find a step that satisfies the Strong Wolfe Conditions. It is also more efficient for descent directions chosen based on only gradient information. A good initial guess for a suitable step size can be crucial for efficiency. Typical strategies include local quadratic modeling of the function to estimate the distance to a minimizer in the chosen direction, choosing the same step size as the previous accepted step size, and assuming the rate of change of the function will be the same as the previous iteration. These methods are described in the text on page 59.

When to Terminate

When using an inexact subproblem solution, it is clear that we may never find a local minimizer of f, except by happenstance. All we can hope for is to converge toward a local minimizer. We might simply let the algorithm continue to run until we run out of time or patience, but this strategy is quite unsatisfying. It would be much better to know how close we are to a minimizer and decide to terminate the algorithm based on this knowledge. In a linear programming context, we can use the idea of the duality gap to directly quantify how close we are to a minimizer in terms of the objective function. We may also have some idea of a bound on the distance from our current point to the optimal point based on distances to the feasible region boundary and current reduced costs. However, in this general nonlinear context, we do not have similar tools at our disposal. We have only the idea that the algorithm seeks a stationary point, and the measure of stationarity is the magnitude of the gradient. That is, we can choose to terminate when $\|\nabla f(x_k)\|$ is less than some tolerance which we specify. Another useful measure is when the step size $\|\alpha p_k\| = \|x_{k+1} - x_k\|$ becomes small. Both of these tests indicate that we *might* be close to a stationary point.

Notes of Convergence of Line Search Algorithms

Theorem 0.1 (**Zoutendijk**). *Consider a line search iteration $x_{k+1} = x_k + \alpha p_k$ satisfying sufficient descent conditions. Suppose f is bounded below and continuously differentiable on an open set $\mathcal{N}$ containing the $f(x_0)$ sublevelset. Assume also that ∇f is Lipschitz continuous on $\mathcal{N}$. Then*

$$\sum_{k=0}^{\infty} \cos^2 \theta_k \|\nabla f(x_k)\|^2 < \infty,$$

where $\cos \theta_k = \dfrac{-p_k^{\mathsf{T}} \nabla f(x_k)}{\|p_k\| \|\nabla f(x_k)\|}$.

Zoutendijk's result shows that $\cos^2 \theta_k \|\nabla f(x_k)\|^2 \to 0$. So, if we are careful to choose θ_k bounded away from $\pi/2$, then $\|\nabla f(x_k)\| \to 0$. That is, the sequence of iterates converges to a stationary point of f. This is trivially true for the method of gradient descent and can be shown to hold for other methods which we will explore.

We can also consider the rates of convergence for various methods. The text shows that convergence for the method of gradient descent is very poor – even for the case of a strictly convex quadratic objective. In this case, we write key results in terms of the condition number of the Hessian (Q) $\kappa := \lambda_n / \lambda_1$, where the eigenvalues are $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$. We have

$$[f(x_{k+1}) - f(x^*)] \leq \left(\frac{\kappa - 1}{\kappa + 1} \right)^2 [f(x_k) - f(x^*)],$$

and

$$\|x_{k+1} - x^*\|_Q^2 \leq \left(\frac{\kappa - 1}{\kappa + 1} \right)^2 \|x_k - x^*\|_Q^2.$$

The reader should consider the implications for extreme cases $\kappa \gg 1$ and $\kappa = 1$, justifying their conclusions.

For Newton's Method, a more involved analysis shows that once the iterations are sufficiently close to x^* so that the Hessian remains positive definite (say, at iteration k), we have the stronger results (for all iterations $\ell > k$):

$$\|\nabla f(x_{\ell+1})\| \leq (const)\|\nabla f(x_\ell)\|^2,$$

and

$$f(x_\ell) - f(x^*) \leq (const)\left(\frac{1}{2}\right)^{\ell-k}.$$