Line Search

annxhe

June 2021

Introduction

- Most line search algorithms require p_k to be a descent direction, one for which $p_k^T \nabla f_k < 0$, because this property guarantees that the function f can be reduced along the direction
- The search direction often has the form $p_k = -B_k^{-1} \nabla f_k$
- Where B_k is a symmetric and nonsingular matrix
- In the steepest descent method, B_k is the identity matrix I

Step Length

- We face a tradeoff in computing the step length α_k
- Want: global minimizer of the univariate
- $\phi(\alpha) = f(x_k + \alpha p_k)$
- But to even find a local minimizer to moderate precision requires too many evaluations of the objective function
- More practical strategies perform an inexact line search to identify a step length that achieves adequate reductions
- Typical line search algorithms try out a sequence of candidate values for α, stopping to accept one of these values when certain conditions are satisfied
- The line search is done in two stages
- (1) Bracketing phase: find an interval to find an interval containing desirable step lengths
- (2) Bisection (or interpolation) phase: compute a good step length within this interval

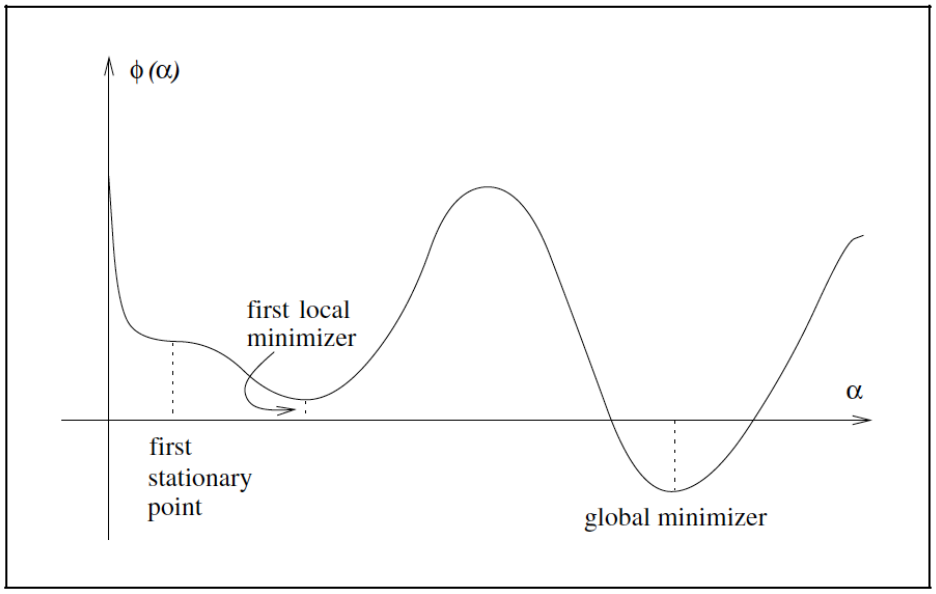


Figure 3.1 The ideal step length is the global minimizer.

- We now discuss various termination conditions for line search algorithms and show that effective step lengths need not lie near minimizers of the univariate function
- A simple condition we could impose on a_k is that $f(x_k + \alpha_k p_k) < f(x_k)$
- This requirement is not enguoh to produce convergence to x^*
- We need to enforce a sufficient decrease condition

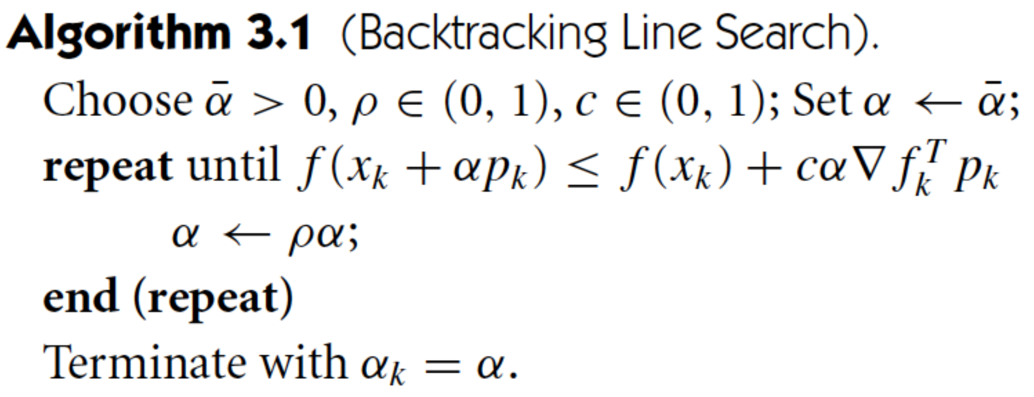
The Wolfe Conditions

- A popular inexact line search condition stipulates that α_k should first of all give sufficient decrease in the objective function f, as measured by teh following inequality (Armijo condition)
- $f(x_k + \alpha p_k) \leq f(x_k) + c_1 \alpha \nabla f_k^T p_k$
- with $0 < c < 1$
- I.e., the reduction in f should be proportional to both the step length and the directional derivative $\nabla f_k^T p_k$
- the RHS is a linear function that can be denoted by $l(\alpha)$, which has negative slope

- In practice, c_1 is chosen to be quite small, like $c_1 = 10^{-4}$
- The sufficient decrease condition is not enough by itself to ensure that the algorithm makes reasonable progress, because, as we can see, it is satisfied for all sufficiently small values of α
- To rule out unacceptably short steps we introduce a second requirement, the curvature condition
- $\nabla f(x_k + \alpha_k p_k)^T p_k \geq c_2 \nabla f_k^T p_k$
- The curvature condition ensures that the slope of ϕ at α_k is greater than c_2 times the initial slope $\phi'(0)$
- If the slope $\phi'(\alpha)$ is strongly negative, we have an indication that we can reduce f significantly my moving further along the chosen direction
- On the other hand, if $\phi'(\alpha)$ is only slightly negative or even positive, it is a sign that we cannot expect much more decrease, so we terminate the line search
- $c_2 \in (c_1, 1)$
- The sufficient decrease and curvature conditions are known collectively as the Wolf Conditions
- A step length may satisfy the Wolfe conditions without being particularly close to a minimizer of ϕ
- The Wolfe conditions are particularly important in the implementation of quasi-Newton methods

Backtracking Line Search

- We mentioned that the sufficient decrease condition alone is not sufficient to ensure the algorithm makes reasonable progress along the given search direction
- However, if the line search algorithm chooses its candidate step lengths appropriately by using backtracking, we can dispense with teh extra Wolfe condition



Algorithm 3.1 (Backtracking Line Search).

```
Choose $\bar{\alpha} > 0$, $\rho \in (0, 1)$, $c \in (0, 1)$; Set $\alpha \leftarrow \bar{\alpha}$;
repeat until $f(x_k + \alpha p_k) \leq f(x_k) + c\alpha \nabla f_k^T p_k$
        $\alpha \leftarrow \rho\alpha$;
end (repeat)
Terminate with $\alpha_k = \alpha$.
```