

Numerical Optimization 04: Local Descent

Qiang Zhu

University of Nevada Las Vegas

May 20, 2020

Overview

- 1 A general model for optimization
- 2 Line Search
- 3 A practical line search
- 4 Summary

Optimization involving multivariate functions

Similar to the single variable function, a common approach to optimization is to incrementally improve a design point x by taking a step that minimizes the objective value based on a local model. The local model may be obtained, for example, from a first- or second-order Taylor approximation.

- Check whether x_k satisfies the termination conditions. If it does, terminate; otherwise proceed to the next step.
- Determine the descent direction d_k using local information such as the gradient or Hessian.
- Determine the step size or learning rate α_k .
- Compute the next design point according to:

$$x_{k+1} = x_k + \alpha_k d_k$$

Line Search

Assuming that we have chosen a descent direction d . We need to choose the step factor α to obtain our next design point. One approach is to use **line search**, which selects the step factor that minimizes the one-dimensional function:

$$\underset{\alpha}{\text{minimize}} : f(x + \alpha d)$$

Line search is a univariate optimization problem, which was covered in the previous lecture. We can apply the univariate optimization method of our choice. To inform the search, we can use the **derivative** of the line search objective, which is simply the directional derivative along d at $x + \alpha d$. One needs to be cautious in choosing α . Large steps will result in faster convergence but risk overshooting the minimum. Smaller steps is more stable but very slow. A fixed step factor α is sometimes referred to as a learning rate.

Approximate line search

It is often more computationally efficient to perform more iterations of a descent method than to do exact line search at each iteration. In this case, the goal is to **find a suitable step size with a small number of evaluations**. Ideally, it needs to satisfy the following

- Sufficient decrease

$$f(x^{k+1}) \leq f(x^k) + \beta\alpha \nabla_{d^k} f(x^k)$$

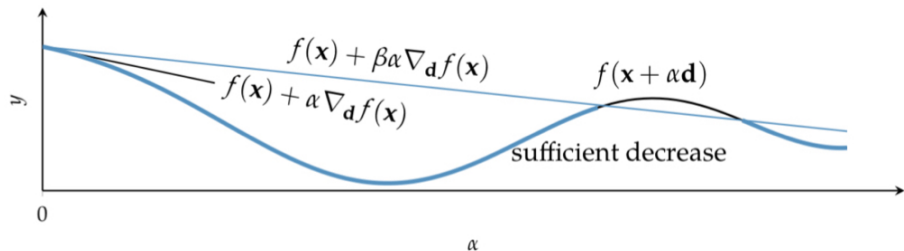
- Curvature condition

$$\nabla_{d^k} f(x^{k+1}) \geq \sigma \nabla_{d^k} f(x^k)$$

Sufficient decrease

$$f(x^{k+1}) \leq f(x^k) + \beta\alpha\nabla_{\mathbf{d}^k} f(x^k)$$

where $\beta \in [0, 1]$. A common choice is $1e-4$.

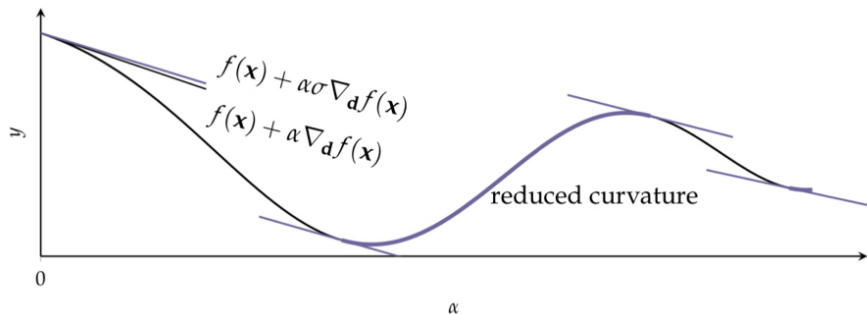


Question: what will happen if you adjust β ?

Curvature condition

$$\nabla_{d^k} f(x^{k+1}) \geq \sigma \nabla_{d^k} f(x^k)$$

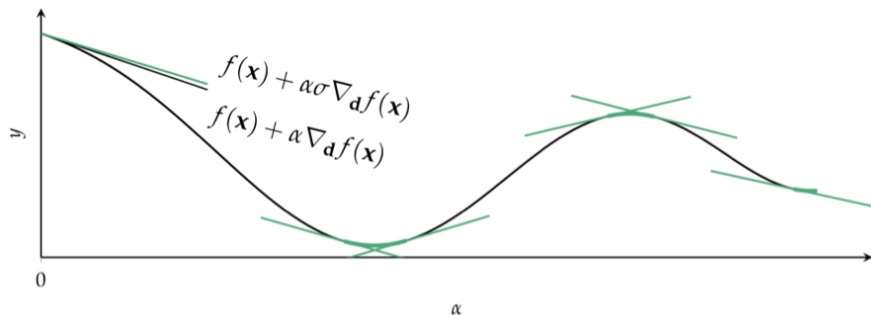
where σ controls how shallow the next directional derivative must be. It is common to set $\beta < \sigma < 1$ with $\sigma = 0.1$ in the conjugate gradient method and 0.9 in Newton's method.



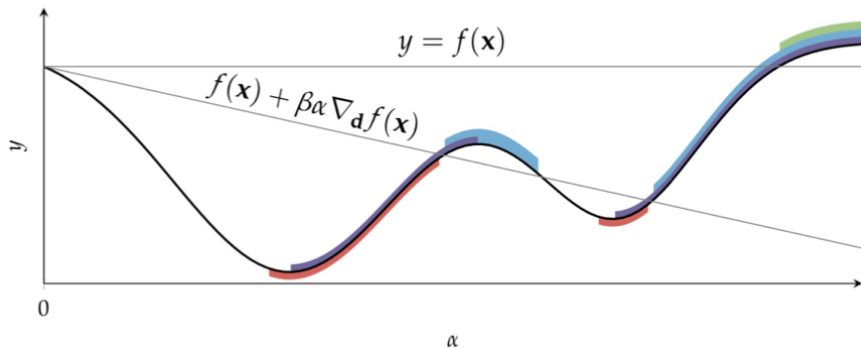
More restrictive curvature condition (strong Wolfe)

$$|\nabla_{d^k} f(x^{k+1})| \leq -\sigma \nabla_{d^k} f(x^k)$$

where σ controls how shallow the next directional derivative must be. It is common to set $\beta < \sigma < 1$ with $\sigma = 0.1$ in the conjugate gradient method and 0.9 in Newton's method.

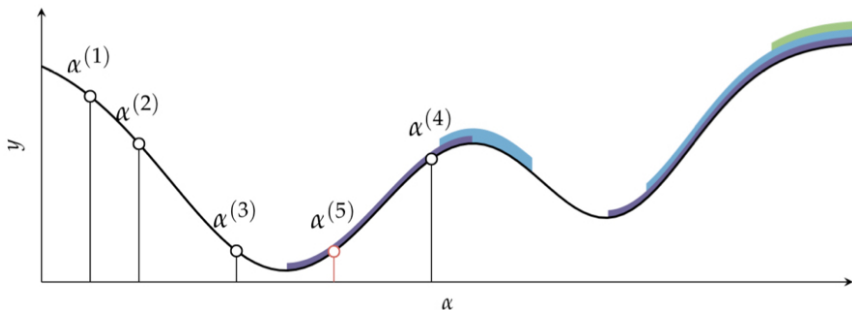


When both conditions are applied



Graphical illustration of line search

- Initial Bracket
- Fibonacci/0.618/bisection until it satisfies the conditions



Terminations conditions

- Maximum iterations.
- Absolute improvement. If the change is smaller than a given threshold, it will terminate:

$$f(x_k) - f(x_{k+1}) < \epsilon_a$$

- Relative improvement. If the change is smaller than a given threshold, it will terminate:

$$f(x_k) - f(x_{k+1}) < \epsilon_r |f(x_k)|$$

- Gradient magnitude. We can also terminate based on the magnitude of the gradient:

$$|\nabla f(x_{k+1})| < \epsilon_g$$

Summary

- Descent direction methods incrementally descend toward a local optimum.
- Univariate optimization can be applied during line search.
- Approximate line search can be used to identify appropriate descent step sizes.
- Termination conditions for descent methods can be based on multiple criteria