David F. Gleich

February 25, 2025

Consider the unconstrained optimization problem:

$$\text{minimize} \quad f(\mathbf{x})$$

where $f : \mathbb{R}^n \to \mathbb{R}$ is twice continuously differentiable.

The second order necessary conditions of a minimizer are:

$$\mathbf{g}(\mathbf{x}) = 0, \boldsymbol{H}(\mathbf{x}) \succeq 0$$

where $\mathbf{g}(\mathbf{x})$ and $\boldsymbol{H}(\mathbf{x})$ are the gradient and Hessian, respectively.

The second order sufficient conditions of a minimizer are:

$$\mathbf{g}(\mathbf{x}) = 0, \boldsymbol{H}(\mathbf{x}) \succ 0.$$

If you don't know the difference between these, take a moment to think about the question: How can a piece of software with access to the gradient and Hessian of a function guarantee to the user that it's at a minimizer?

In this class, we'll study two types of optimization algorithms:[1]

line search methods   trust region methods.

[1] We'll see a third type too while studying these: exact algorithms for simple quadratics!

Both start from a given point $\mathbf{x}^{(0)}$ and are iterative in nature. That is, they try to find a point $\mathbf{x}^{(k+1)}$ "nearby" $\mathbf{x}^{(k)}$ such that $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$. Because writing

$$f(\mathbf{x}^{(k+1)}), \mathbf{g}(\mathbf{x}^{(k)}), \mathbf{g}(\mathbf{x}^{(k+1)}), \text{etc.}$$

quickly becomes tiring, we use the following shorthand at a point $\mathbf{x}^{(k)}$:

$$\begin{aligned} \mathbf{x} &= \mathbf{x}^{(k)} \\ \mathbf{x}^+ &= \mathbf{x}^{(k+1)} \\ \mathbf{g} &= \mathbf{g}(\mathbf{x}^{(k)}) \\ \mathbf{g}^+ &= \mathbf{g}(\mathbf{x}^{(k+1)}) \\ \boldsymbol{H} &= \boldsymbol{H}(\mathbf{x}^{(k)}) \\ f_k &= f(\mathbf{x}^{(k)}) \\ f^+ &= f(\mathbf{x}^{(k+1)}) \\ f_{k+1} &= f(\mathbf{x}^{(k+1)}) \end{aligned}$$

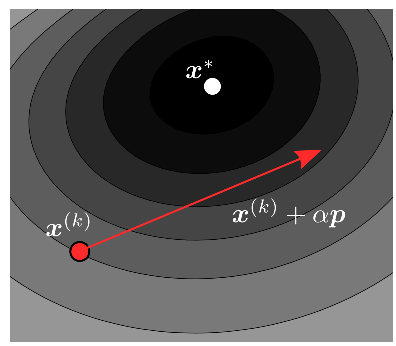

FIGURE 1 – A line search method starts at a point and picks a direction $\mathbf{p}$. It then chooses a step length α to determine how far to go along that direction.

Line search At a point $\mathbf{x}$, a line search method finds a direction $\mathbf{p}$ that ought to improve the value of the objective function f, it then considers the "line" of points:

$$\mathbf{x}^+ = \mathbf{x} + \alpha \mathbf{p}.$$

The key question with a line search method is how to pick $\mathbf{p}$ and α.

Trust region At a point $\mathbf{x}$, a trust region method fits a quadratic model around $\mathbf{x}$ and then minimizes a quadratic model exactly without moving too far:

$$f(\mathbf{x} + \mathbf{p}) \approx f(\mathbf{x}) + \mathbf{p}^T \mathbf{g} + \frac{1}{2} \mathbf{p}^T \boldsymbol{H} \mathbf{p}$$

with $\|\mathbf{p}\|$ not to big.
The key question with a trust region method is how to pick the model and maximum distance $\|\mathbf{p}\|$.

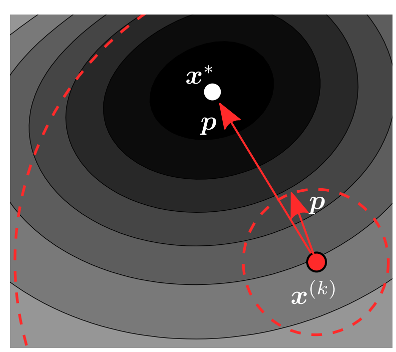

FIGURE 2 – A trust region method picks a distance (the two dashed red circles). It then tries to find the best point within the circle. Hopefully, once it gets close to a minimizer, it'll pick it out directly (e.g. the larger circle).

Try thinking about the difference this way: in a line search method, you first pick a direction, and then determine how far to go. In a trust region method, you first pick how far you are willing to go, and then pick the best direction given that distance constraint.