# Line Search Methods

Saurav Samantaray

Department of Mathematics

Indian Institute of Technology Madras

February 5, 2024

# Line Search Method

- In each iteration of a line search method a search direction $p_k$ is computed, and
- then its decided how far to move along that direction.

An iteration is given by

$$x_{k+1} = x_k + \alpha_k p_k \tag{1}$$

where $\alpha_k > 0$ (scalar) called the step length

. The success of a line search method depends on effective choices of both:

1. the direction $p_k$
2. the step length $\alpha_k$

# The Steepest Descent Direction

- The steepest descent direction $-\nabla f_k$ is the most obvious choice for search direction for a line search method.
- Among all Directions once could from $x_k$, along $-\nabla f_k$, $f$ decreases most rapidly.

## Justification

- Consider any search direction $p$ and step-length $\alpha$, we have

$$f(x_k+\alpha p) = f_k+\alpha p^T\nabla f_k+\frac{\alpha^2 p^T\nabla^2 f(x_k+tp)p}{2} \text{ for some } t\in(0,\alpha)$$

- Let $\alpha << 1$ (small) and we consider the first-order approximation of $f$ at $x_k+\alpha p$ around $x_k$ as:

$$f(x_k+\alpha p)\approx f(x_k)+\alpha p^T\nabla f_k$$

- Change in $f$ moving from $x_k$ to $x_k+\alpha p$ is

$$f(x_k+\alpha p)-f(x_k)$$

# The Steepest Descent Direction

- As the distance moved in the direction is $\alpha$, therefore, the rate of change of $f$ along the direction $p$ at $x_k$ is

$$\frac{f(x_k + \alpha p) - f(x_k)}{\alpha}$$

- which is coefficient of $\alpha$, i.e.

$$p^T \nabla f_k$$

- This implies smaller the above value is, more descent can be achieved.

- Hence, the unit direction $p$ of most rapid decrease is the solution to the problem

$$\min_p p^T \nabla f_k, \qquad \text{subject to } ||p|| = 1.$$

# The Steepest Descent Direction

- $$p^T \nabla f_k = ||p|| \; ||\nabla f_k|| \; \cos\theta = ||\nabla f_k|| \; \cos\theta$$
  where $\theta$ is the angle between $p$ and $\nabla f_k$.
- The minimiser is attained when $\cos\theta = -1$ and
  $$p = -\frac{\nabla f_k}{||\nabla f_k||}$$

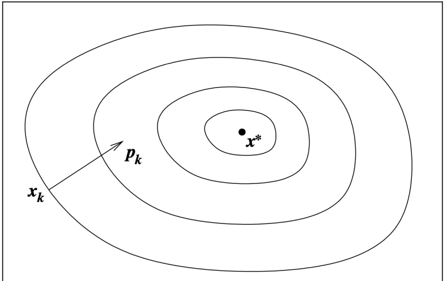


Figure illustrates this direction is orthogonal to the contours of the function

# The Steepest Descent Direction

- At every step (iteration) in the steepest descent method the search direction is chosen along

$$p = -\nabla f_k$$

- $\alpha_k$ can be chosen in a variety of ways.
- One advantage of this method is it requires only the calculation of gradient ($\nabla f_k$), but not second derivatives.
- Line search methods may use search directions other than the steepest descent direction.

# Descent Direction

## Descent Direction

Any direction that makes an angle of strictly less than $\frac{\pi}{2}$ radians with $-\nabla f_k$ is guaranteed to produce a decrease in $f$, provided the step length is sufficiently small and is called a descent direction.

- Now consider
$$f(x_k + \epsilon p_k) = f(x_k) + \epsilon p_k^T \nabla f_k + \mathcal{O}(\epsilon^2).$$
- When $p_k$ is a downhill (descent) direction, the angle $\theta_k$ between $p_k$ and $\nabla f_k$ has $\cos\theta_k < 0$, so that
$$p_k^T \nabla f_k = ||p_k||\ ||\nabla f_k||\ \cos\theta_k < 0$$
$$\implies f(x_k + \epsilon p_k) < f(x_k)$$
- Most line search algorithms require $p_k$ to be descent direction, because this property guarantees that the function

# Newton Search Direction

- This direction is derived from the second-order Taylor series approximation to $f(x_k + p)$, which is

$$f(x_{k+p}) \approx f_k + p^T \nabla f_k + \frac{1}{2} p^T \nabla^2 f_k p \stackrel{def}{=} m_k(p). \tag{2}$$

- Assuming for the moment that $\nabla^2 f_k$ is positive definite, we obtain the Newton direction by finding the vector $p$ that minimizes $m_k(p)$.

- By simply setting the derivative of $m_k(p)$ to zero, we obtain the following explicit formula:

$$p_k^N = -(\nabla^2 f_k)^{-1} \nabla f_k. \tag{3}$$

- The Newton direction is reliable when the difference between the true function $f(x_k + p)$ and its quadratic model $m_k(p)$ is not too large.

# Newton Search Direction

- If $\nabla^2 f$ is sufficiently smooth, this difference introduces a perturbation of only $\mathcal{O}(||p||^3)$.
- Therefore, when $||p||$ is small

$$f(x_k + p) \approx m_k(p) \qquad \text{quite accurately}$$

- The Newton direction can be used in a line search method when $\nabla^2 f_k$ is positive definite, as in this case we have

$$\begin{aligned} \nabla f_k^T p_k^N &= -{p_k^N}^T \nabla^2 f_k p_k^N \\ &< -\sigma_k ||p_k^N||^2 \end{aligned}$$

for some $\sigma_k > 0$ (+ve definiteness of $\nabla^2 f_k$)

- Unless the gradient $\nabla f_k$ (and therefore the step $p_k^N$) is zero, we have

$$\nabla f^T p_k^N < 0$$

# Newton Search Direction

- Unlike the steepest descent direction, there is a "natural" step length of 1 associated with the Newton direction.
- Adjust $\alpha$ only when it does not produce a satisfactory reduction in the value of $f$ .
- Note that when $\nabla^2 f_k$ is not positive definite the Newton direction may not exist, since $(\nabla^2 f_k)^{-1}$ may not exist.
- Even when it is defined, it may not satisfy the descent property, and therefore is unsuitable.
- Methods that use Newton direction have fast rate of local convergence (more on this later).
- After a neighbourhood of the solution is reached, convergence to high accuracy often occurs in just a few iterations.

# Newton Search Direction

- The main drawback is the need to calculate the Hessian $\nabla^2 f(x)$.
- Explicit computation of this matrix of second derivatives can sometimes be a cumbersome, error-prone, and expensive process.
- Finite-difference and automatic differentiation techniques come useful in avoiding the need to calculate second derivatives by hand.

# Quasi-Newton Search Direction

- Quasi-Newton search directions provide an attractive alternative to Newton's method.
- They do not require computation of the Hessian and yet still attain a superlinear rate of convergence.
- In place of the true Hessian $\nabla^2 f_k$, they use an approximation $B_k$, which is updated after each step to take account of the additional knowledge gained during the step.
- The updates make use of the fact that changes in the gradient $g$ provide information about the second derivative of $f$ along the search direction.

# Quasi-Newton Search Direction

- Using the Taylor's expansion we have

$$\nabla f(x+p) = \nabla f(x) + p^T \nabla^2 f(x) + \mathcal{O}(||p||^2)$$

- By setting $x = x_k$ and $p = x_{k+1} - x_k$, we obtain

$$\nabla f_{k+1} = \nabla f_k + \nabla^2 f_k (x_{k+1} - x_k) + \mathcal{O}(||x_{k+1} - x_k||^2).$$

# Quasi-Newton Search Direction

- When $x_k$ and $x_{k+1}$ lie in a region near the solution $x^*$, within which $\nabla^2 f$ is positive definite, the final term in this expansion is eventually dominated by the $\nabla^2 f_k(x_{k+1} - x_k)$ term, and we an write.

$$\nabla^2 f_k(x_{k+1} - x_k) \approx \nabla f_{k+1} - \nabla f_k. \tag{4}$$

- We choose the new Hessian approximation $B_{k+1}$ so that it mimics the property (4) of the true Hessian.

- That is we require it to satisfy the following condition, known as the secant equation:

$$B_{k+1} s_k = y_k, \tag{5}$$

where $s_k = x_{k+1} - x_k$, $y_k = \nabla f_{k+1} - \nabla f_k$.

# Quasi-Newton Search Direction

Typically, additional conditions are imposed on $B_{k+1}$, such as

- Symmetry (Motivated by symmetry of the exact Hessian).
- a requirement that the difference between successive approximations $B_k$ and $B_{k+1}$ have low rank.

Two of the most popular formulae for updating the Hessian approximation $B_k$ are:

## Symmetric-rank-one (SR1) formula:

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}. \tag{6}$$

# Quasi-Newton Search Direction

## BFGS Formula:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}. \tag{7}$$

- The difference between the matrices $B_k$ and $B_{k+1}$ is
  - rank-one for SR1.
  - rank-two in case of BFGS.
- Both updates satisfy the secant equation and both maintain symmetry.

# Quasi-Newton Search Direction

- It can be shown that the BFGS formula generates positive definite approximations whenever the initial approximation $B_o$ is positive definite and

$$s_k^T y_k > 0$$

- The quasi-Netwon search direction is obtained by using $B_k$ instead of the exact Hessian $\nabla^2 f_k$ in the Newton direction

$$p_k = -B_k^{-1} \nabla f_k.$$

- Some practical implementations of quasi-Newton methods avoid the need to factorise $B_k$ at each iteration by updating inverse of $B_k$, instead of $B_k$ itself.

# Quasi-Newton Search Direction

- The equivalent formula for (SR1), applied to the inverse approximation

$$H_k \overset{defn}{:=} B_k^{-1} \quad \text{is}$$

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T$$

where $\rho_k = \frac{1}{y_k^T s_k}$

- Then $p_k$ is given by

$$p_k = -H_k \nabla f_k$$

- This matrix-vector multiplication is simpler than the factorisation / back-substitution produce.

# Non-linear Conjugate Gradient Methods

- The direction here are generated by non-linear conjugate gradient methods.
- They have the form

$$p_K = -\nabla f(x_k) + B_k p_{k-1}$$

  where $B_k$ is a scalar that ensures that $p_k$ and $p_{k-1}$ are conjugate (to be defined later).
- Conjugate gradient methods were originally designed to solve systems of linear equations:

$$Ax = b$$

  where the coefficient matrix $A$ is symmetric and positive definite.

# Non-linear Conjugate Gradient Methods

- The problem of solving this linear system is equivalent to the problem of minimising the convex quadratic function defined by
$$\phi(x) = \frac{1}{2}x^T Ax - b^T x.$$
- In general, non-linear conjugate gradient directions are much more effective than the steepest descent direction and are almost as simple to compute.
- These methods do not attain fast convergence rates of Newton or quasi-Newton methods.
- But, they have the advantage of not requiring storage of matrices.