Introduction to Local and Global Optimization for NLP

Marco Trubian

Dipartimento di Scienze dell'Informazione (DSI)
Università degli Studi di Milano

Line Search algorithms

Yesterday

Descent Method;
{
    Choose $\boldsymbol{x}_0 \in \mathbb{R}^n$; $k := 0$;
    While $\nabla f(\boldsymbol{x}_k) \neq \emptyset$;
    {
        compute $\boldsymbol{d}_k \in \mathbb{R}^n$; /* descent direction */
        compute $\alpha_k \in \mathbb{R}$; /* step along $\boldsymbol{d}_k$ */
        $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha \boldsymbol{d}_k$;
        $k := k + 1$;
    }
}

Line Search algorithms

Descent Method;
{
    Choose $\mathbf{x}_0 \in \mathbb{R}^n$; $k := 0$;
    While $\nabla f(\mathbf{x}_k) \neq \emptyset$;
    {
        compute $\mathbf{d}_k \in \mathbb{R}^n$; /* descent direction */
        compute $\alpha_k \in \mathbb{R}$; /* step along $\mathbf{d}_k$ */
        $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{d}_k$;
        $k := k + 1$;
    }
}

Obtaining a Direction

Consider Taylor's approximation near $\boldsymbol{x}_k$ along $\boldsymbol{d}$:

$$f(\boldsymbol{x}_k + \alpha \boldsymbol{d}) = f(\boldsymbol{x}_k) + \alpha \nabla f(\boldsymbol{x}_k)^T \boldsymbol{d} + \frac{1}{2}\alpha^2 \boldsymbol{d}^T H(\boldsymbol{x}_k)\boldsymbol{d} + o(\|\boldsymbol{d}\|^3)$$

- One possibility: $\min_d \{\nabla f_k^T \boldsymbol{d} : ||\boldsymbol{d}|| = 1\}$
- The objective is $||\nabla f_k|| ||\boldsymbol{d}|| cos(\theta)$ which is minmized if $\theta = \pi$
- In other words, $\boldsymbol{d}$ is along $-\nabla f_k$ and is normalized to

$$\boldsymbol{d} = -\frac{\nabla f_k}{||\nabla f_k||}$$

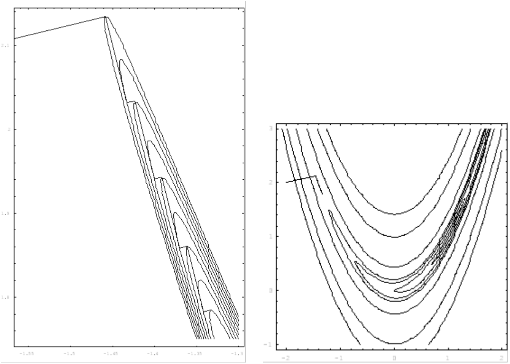
- This is one type of steepest descent direction (since it is along $-\nabla f_k$). In particular, it has a stepsize of 1 at every iteration.

Steepest Descent Methods

- Steepest descent methods: $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \alpha_k \nabla f_k$
- Directions are orthogonal to contours of function
- Low computational effort (does not need to calculate the hessian matrix $H(\boldsymbol{x}_k)$)
- Globally convergent but...

Steepest Descent Methods

- Globally convergent but...
- Painfully slow if function is ill-conditioned



(a) Contours (b) Ten iterations

Figura: Rosenbrock function: $f(x, y) = 100(y - x^2)^2 + (1 - x)^2$.

Local Convergence Rate of Steepest Descent

- Suppose $f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T Q\boldsymbol{x} - \boldsymbol{b}^T\boldsymbol{x}$, with pos. def. Q
- $\nabla f_k = Q\boldsymbol{x}_k - b$, $\alpha^* = \frac{\nabla f_k^T \nabla f_k}{\nabla f_k^T Q \nabla f_k}$ and $\boldsymbol{x}^* = Q^{-1}b$
- $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{\nabla f_k^T \nabla f_k}{\nabla f_k^T Q \nabla f_k}\nabla f_k$
- Local convergence rate is given by

$$\frac{\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_Q}{\|\boldsymbol{x}_k - \boldsymbol{x}^*\|_Q} = \left\{1 - \frac{(\nabla f_k^T \nabla f_k)^2}{(\nabla f_k^T Q \nabla f_k)(\nabla f_k^T Q^{-1} \nabla f_k)}\right\}^{\frac{1}{2}}$$

Local Convergence Rate of Steepest Descent

Theorem Given a positive definite matrix Q the following relationship holds for each $\boldsymbol{x} \in \mathbb{R}^n$

$$\frac{(\boldsymbol{x}^T\boldsymbol{x})^2}{(\boldsymbol{x}^T Q\boldsymbol{x})(\boldsymbol{x}^T Q^{-1}\boldsymbol{x})} \geq \frac{4\lambda_m\lambda_M}{(\lambda_m + \lambda_M)^2}$$

where λ_m and λ_M are the min and max eigenvalue of Q.

Hence

$$\frac{\|\boldsymbol{x}_{k+1} - \boldsymbol{x}^*\|_Q}{\|\boldsymbol{x}_k - \boldsymbol{x}^*\|_Q} \leq \left(\frac{\lambda_M - \lambda_m}{\lambda_M + \lambda_m}\right)$$

On quadratic models the rate of convergence of Steepest Descent is **linear**

Newton Methods

We start with Taylor's approximation

$$f(\boldsymbol{x}+\boldsymbol{d}) \approx f(\boldsymbol{x}) + \nabla f^T \boldsymbol{d} + \frac{1}{2}\boldsymbol{d}^T H \boldsymbol{d}.$$

- Minimizing $f(\boldsymbol{x}+\boldsymbol{d})$ implies minimizing $\nabla f^T \boldsymbol{d} + \frac{1}{2}\boldsymbol{d}^T H \boldsymbol{d}$
- The resulting direction is called a Newton direction: $\boldsymbol{d}^N = -(H)^{-1}\nabla f$
- Two requirements:
    a. H positive definite
    b. function is well approximated by a quadratic model
- when H is positive definite we have

$$\nabla f^T \boldsymbol{d}^N = -\boldsymbol{d}^N H \boldsymbol{d}^N \leq -\sigma ||\boldsymbol{d}^N||^2, \quad \sigma > 0$$

  i.e. when H is positive definite Newton direction is a descent direction.
- Natural stepsize of 1 when quadratic approximation is good else do a search

Newton Methods on QP

How good is Newton method on quadratic models

$$f(\boldsymbol{x}) = \frac{1}{2}\boldsymbol{x}^T Q \boldsymbol{x} - b^T \boldsymbol{x}$$

with Q positive definite ?

$$\boldsymbol{x}_1 = \boldsymbol{x}_0 - H(\boldsymbol{x}_0)^{-1}\nabla f(\boldsymbol{x}_0) = \boldsymbol{x}_0 - Q^{-1}(Q\boldsymbol{x}_0 - \boldsymbol{b}) = \boldsymbol{x}_0 - \boldsymbol{x}_0 + Q^{-1}\boldsymbol{b} = Q^{-1}\boldsymbol{b}.$$

If Q is positive definite Newton Method converges in one step otherwise id does not even converge.

On generic function the quality of the direction depends on the definite positiveness of the hessian matrix H

Local Convergence of Newton Methods

Theorem Suppose that f is C^2 and that $H(\boldsymbol{x})$ is Lipschitz continuous in a neighborhood of x^* at which sufficient conditions hold. Assume that the unit step is admissible implying that $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \boldsymbol{d}_k^N$. Then

- if $\boldsymbol{x}_0$ is sufficiently close to $\boldsymbol{x}^*$, then $\{\boldsymbol{x}_k\} \to \boldsymbol{x}^*$
- $\{\boldsymbol{x}_k\}$ converges quadratically
- $\{||\nabla f(\boldsymbol{x}_k)||\}$ converges quadratically to zero.

Convergence of Newton Methods

Newton Methods are locally convergent

$$\boldsymbol{d}^N = -(H)^{-1}\nabla f$$

- Since $H(\boldsymbol{x})$ need not always be positive definite (unless f is a convex function), the directional derivative $\nabla f^T \boldsymbol{d}^N$ may not be negative
- However, we only consider the local convergence behavior of Newton's method
- In other words, if $H(\boldsymbol{x})$ is continuous, then it will be positive definite in some neighborhood of the solution
- In such a neighborhood, Newton's method converges **quadratically**

Problems/Issues

- Newton Method requires computing the true Hessian: $O(n^3)$ - costly
- $H(\mathbf{x}_k)$ may not be nonsingular, let alone positive definite
- Indefinite $H(\mathbf{x}_k)$ implies one is near a saddle point/maximizer
- Modified Newton methods:
  - either modify the Hessian matrix while ensuring descent. $H(\mathbf{x}_k) \rightsquigarrow (H(\mathbf{x}_k) + \gamma I)$, with $\gamma > 0$ big enough to guarantee positiveness
  - or select a Steepest Descent direction when required

Modified Newton Method

Modified Newton Method;
{
    Choose $\mathbf{x}_0 \in \mathbb{R}^n$; $k := 0$;
    While $\nabla f(\mathbf{x}_k) \neq \emptyset$;
    {
        if $H(\mathbf{x}_k)$ is singular **then** $\mathbf{d}_k := -\nabla f(\mathbf{x}_k)$;
        else
        {
            $\mathbf{s} := -H(\mathbf{x}_k)^{-1}\nabla f(\mathbf{x}_k)$;
            if $|\nabla f(\mathbf{x}_k)^T \mathbf{s}| < \varepsilon ||\nabla f(\mathbf{x}_k)|| \cdot ||\mathbf{s}||$ **then** $\mathbf{d}_k := -\nabla f(\mathbf{x}_k)$;
            else
                if $\mathbf{s}$ is descent direction **then** $\mathbf{d}_k := \mathbf{s}$;
                else $\mathbf{d}_k := -\mathbf{s}$;
        }
        compute $\alpha_k \in \mathbb{R}$; /*step along $\mathbf{d}_k$*/
        $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{d}_k$;
        $k := k + 1$;
    }
}

Quasi-Newton Methods

Alternative to Newton methods not requiring costly computation of H_k. They use an approximation G_k of H_k^{-1}.

(a) $\boldsymbol{d}_k := -G_k \nabla f(\boldsymbol{x}_k)$

(b) compute α_k with a line search technique (e.g. Armijo)

(c) $\boldsymbol{x}_{k+1} := \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$

Let approximate $f(\boldsymbol{x}_k + \boldsymbol{h}_k)$ with a quadratic model $q(\boldsymbol{h}_k)$, where $\boldsymbol{h}_k = \alpha_k \boldsymbol{d}_k = \boldsymbol{x}_{k+1} - \boldsymbol{x}_k$:

$$f(\boldsymbol{x}_k + \boldsymbol{h}_k) \approx q(\boldsymbol{h}_k) = f(\boldsymbol{x}_k) + \boldsymbol{h}_k^T \nabla f(\boldsymbol{x}_k) + \frac{1}{2} \boldsymbol{h}_k^T H(\boldsymbol{x}_k) \boldsymbol{h}_k$$

from which we derive

$$\nabla f(\boldsymbol{x}_k + \boldsymbol{h}_k) = \nabla f(\boldsymbol{x}_{k+1}) \approx \nabla q(\boldsymbol{h}_k) = \nabla f(\boldsymbol{x}_k) + H(\boldsymbol{x}_k) \boldsymbol{h}_k$$

Quasi-Newton Methods

$$\nabla f(\boldsymbol{x}_k + \boldsymbol{h}_k) = \nabla f(\boldsymbol{x}_{k+1}) \approx \nabla q(\boldsymbol{h}_k) = \nabla f(\boldsymbol{x}_k) + H(\boldsymbol{x}_k)\boldsymbol{h}_k$$

By defining $\boldsymbol{p}_k := \nabla f(\boldsymbol{x}_{k+1}) - \nabla f(\boldsymbol{x}_k)$ we can write the following

Secant relation

$$H(\boldsymbol{x}_k)\boldsymbol{h}_k \approx \boldsymbol{p}_k, \quad or \quad (H(\boldsymbol{x}_k)^{-1}\boldsymbol{p}_k \approx \boldsymbol{h}_k)$$

Hence, after setting $G_0 = I$, we impose that at each iteration k, the matrix G_{k+1} satisfies the Secant relation as an equality:

$$G_{k+1}\boldsymbol{p}_k = \boldsymbol{h}_k$$

How to move from G_k to G_{k+1} generates different quasi-Newton methods

Quasi-Newton Methods

Quasi-Newton Method;
{
Choose $\boldsymbol{x}_0 \in \mathbb{R}^n$; $k := 0$;
$G_0 := I$;
While $\nabla f(\boldsymbol{x}_k) \neq \emptyset$;
{
$\boldsymbol{d}_k := -G_k \nabla f(\boldsymbol{x}_k)$ `/* descent direction */`
`compute` $\alpha_k \in \mathbb{R}$; `/* step along` $\boldsymbol{d}_k$ `*/`
$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha \boldsymbol{d}_k$;
`compute` G_{k+1} `from` G_k;
$k := k + 1$;
}
}

Quasi-Newton Methods

How to move from G_k to G_{k+1} generates different quasi-Newton methods

- impose $G_k = G_k^T$
- impose that $G_{k+1} - G_k$ have low rank

Symmetric Rank-One or SR1

$$G_{k+1} = G_k + \frac{(\boldsymbol{h}_k - G_k \boldsymbol{p}_k)(\boldsymbol{h}_k - G_k \boldsymbol{p}_k)^T}{(\boldsymbol{h}_k - G_k \boldsymbol{p}_k)^T \boldsymbol{p}_k}.$$

DFP (rank-two) (Davidon, Fletcher and Powell)

$$G_{k+1} = G_k + \frac{\boldsymbol{h}_k \boldsymbol{h}_k^T}{\boldsymbol{h}_k^T \boldsymbol{p}_k} - \frac{G_k \boldsymbol{p}_k \boldsymbol{p}_k^T G_k}{\boldsymbol{p}_k^T G_k \boldsymbol{p}_k}.$$

BFGS (Rank-two) (Broyden, Fletcher, Goldfarb and Shanno)

$$G_{k+1} = G_k + \left(1 + \frac{\boldsymbol{p}_k G_k \boldsymbol{p}_k^T}{\boldsymbol{h}_k^T \boldsymbol{p}_k}\right) \frac{\boldsymbol{h}_k \boldsymbol{h}_k^T}{\boldsymbol{h}_k^T \boldsymbol{p}_k} - \left(\frac{\boldsymbol{h}_k \boldsymbol{p}_k^T G_k + G_k \boldsymbol{p}_k \boldsymbol{h}_k^T}{\boldsymbol{h}_k^T \boldsymbol{p}_k}\right).$$

Quasi-Newton Methods

Some properties of Rank-two models:

- G_k converges to $H(\boldsymbol{x}_k)^{-1}$ on quadratic models,
- if G_0 is positive definite (e.g. $G_0 = I$) then all the G_k are p.d.,
- computational cost order of $O(n^2)$ in each iteration,
- superlinear-convergence rate,
- BFGS has global convergence if α_k satisfies Wolfe conditions.

Broyden Family

$$G_{k+1} = (1 - \phi)G_{k+1}^{DFP} + \phi G_{k+1}^{BFGS},$$

where $0 \leq \phi \leq 1$.

Quick Recap

We introduced line search methods and focused on the following:

1. $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{d}_k$
2. $\boldsymbol{d}_k$ satisfies $\nabla f_k^T \boldsymbol{d}_k < 0$ (a descent condition)
3. $\boldsymbol{d}_k$ may be specified as $\boldsymbol{d}_k = -B_k^{-1} \nabla f_k$
4. Choice of B_k based on method
    - a Steepest descent: $B_k = I$
    - b Newton method: $B_k = H_k^{-1}$
    - c quasi-Newton method: $B_k = G_k$
5. Choice of α_k based on Wolfe conditions

Quick Recap (2)

Convergence Analysis:

1. we may prove that an α satisfying Wolfe conditions exists
2. we may prove $||\nabla f_k|| \to 0$ i.e. global convergence under mild hypotheses
3. we may prove that
   a. Steepest descent is globally convergent with a linear-convergence rate
   b. Newton method is locally convergent with a quadratic-convergence rate
   c. quasi-Newton method is globally convergent with a superlinear-convergence rate

Conjugate Gradient Methods

One of the most useful techniques for solving large linear systems. Can be adapted to solve nonlinear optimization problems.

- First proposed by Hestenes and Stiefel for the solution of large linear systems with positive definite matrices
- Performance intimately related to the distribution of the eigenvalues
- By transforming the system (called preconditioning), we may make improve the distribution and therefore, performance.

The Linear CG Method

- The CG Method is an iterative method for solving systems of the form $A\boldsymbol{x} = b$ where A pos. def. and $A = A^T$.
- This is equivalent to minimizing

$$f(x) = \frac{1}{2}\boldsymbol{x}^T A\boldsymbol{x} - b^T\boldsymbol{x}$$

  We denote $r(x) := \nabla f(\boldsymbol{x}) = A\boldsymbol{x} - b$
- A set of vectors $p_0, p_1, \ldots p_h$ is said to be conjugate with respect to A if $p_i^T A p_j = 0$, for all $i \neq j$
- Given a starting point $\boldsymbol{x}_0$ and conjugate directions $p_0, \ldots p_{n-1}$, we generate sequence $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k p_k$
- α_k is the one dimensional minimizer along $\boldsymbol{x}_k + \alpha_k p_k$ and is given by

$$\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k}$$

Conjugate Direction Method

Theorem For any $\mathbf{x}_0$, the sequence generated by $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k p_k$ converges to $\mathbf{x}^*$ in at most n steps.

Interpretation of conjugate directions:

- Consider $\min \frac{1}{2}\mathbf{x}^T A\mathbf{x} - b^T\mathbf{x}$
- If A is diagonal, problem is separable and it may be solved by successive minimization along coordinate directions $e_1, \ldots, e_n$
- Solution in n iterations
- If A is not diagonal
  - We transform the problem: $\mathbf{y} = S^{-1}\mathbf{x}$ where $S = (p_0 \cdots p_{n-1})$
  - We have by conjugacy property that $S^T AS$ is diagonal and $\min y^T S^T ASy + (S^T b)^T y$ is solvable in n iterations

Conjugate Gradient Method

CG Method is a coniugate direction method with an important property: p_k can be obtained by knowing only p_{k-1} and p_k is conjugate to all previous directions.

- Choice of p_k = linear combination of ∇f_k and p_{k-1}

$$p_k = -\nabla f_k + \beta_k p_{k-1}$$

- β_k defined by conjugacy between p_{k-1} and p_k, i.e. $p_k^T A p_{k-1} = 0$
- This implies that $\beta_k = \frac{\nabla f_k^T A p_{k-1}}{p_{k-1}^T A p_{k-1}}$
- $p_0 = -\nabla f_0$

Conjugate Gradient Method

Conjugate Gradient Method

1. $r_0 = A\boldsymbol{x}_0 - b_0, \quad p_0 = -r_0, \quad k = 0$
2. $\alpha = -\frac{r_k^T p_k}{p_k^T A p_k}$
3. $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k p_k$
4. $r_{k+1} = A\boldsymbol{x}_{k+1} - b$ and $\beta_{k+1} = \frac{r_{k+1}^T A p_k}{p_k^T A p_k}$
5. $p_{k+1} = -r_{k+1} + \beta_{k+1} p_k$
6. $k = k + 1$

Let observe that β_{k+1} can be computen also as

$$\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$$

CG Method for general non linear f

Fletcher-Reeves (1964)

$$\beta_{k+1} = \frac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k}$$

with α chosen s.t. strong Wolfe conditions are satisfied

Polak-Ribiére (1969)

$$\beta_{k+1} = \frac{(\nabla f_{k+1} - \nabla f_k)^T \nabla f_{k+1}}{\|\nabla f_k\|^2}$$

with α chosen s.t. modified strong Wolfe conditions are satisfied

Dai-Yuan (1999)

$$\beta_{k+1} = \frac{\|\nabla f_{k+1}\|^2}{(\nabla f_{k+1} - \nabla f_k)^T p_k}$$

with α chosen s.t. Wolfe conditions are satisfied

Why CG Method ?

CG is particularly useful for several reasons:

1. Low storage
2. Compute $p_k^T A p_k$ and $r_{k+1}^T A p_k$ and two vector sums at each iteration
3. Preferable if problem is large else Gaussian elimination is preferred (less sensitive to rounding errors)
4. CG also converges fast

Two Main Strategies

- Line search: Given an iterate $\boldsymbol{x}_k$, we first determine a direction $\boldsymbol{h}_k$. Then we determine a stepsize α_k given by

  $$\min_{\alpha_k} f(\boldsymbol{x}_k + \alpha_k \boldsymbol{h}_k)$$

  The new iterate is given by $\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{h}_k$.
- Trust region: Construct a model function m_k using the information at $\boldsymbol{x}_k$. A trust-region radius Δ_k (stepsize) is selected and we obtain a $\boldsymbol{h}_k$ such that

  $$\min_{\|\boldsymbol{h}_k\| \leq \Delta_k} m_k(\boldsymbol{h}_k)$$

  If $\boldsymbol{h}_k$ does not produce sufficient descent, we shrink Δ_k and resolve.

Trust Region Methods

Construct a model function m_k using the information at $\boldsymbol{x}_k$.

$$m_k(\boldsymbol{p}) = f(\boldsymbol{x}_k) + \nabla f(\boldsymbol{x}_k)^T \boldsymbol{p} + \frac{1}{2}\boldsymbol{p}^T B_k \boldsymbol{p},$$

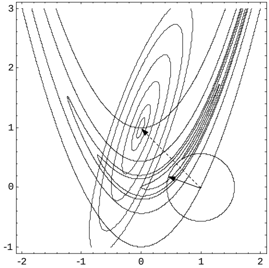
A trust-region radius Δ_k is selected and we obtain a $\boldsymbol{p}$ such that

$$\min_{\|\boldsymbol{p}\| \leq \Delta_k} m_k(\boldsymbol{p})$$

If $\boldsymbol{p}$ does not produce sufficient descent, we shrink Δ_k and resolve.
B_k can be an approximation to the Hessian or the true Hessian.
If B_k is the Hessian matrix H_k then for Δ_k large enough and H_k p.d. we have the Newton direction $\boldsymbol{p} = -B_k^{-1} \nabla f_k$.

Trust Region Methods

Quadratic model of the Rosenbrock function in $\boldsymbol{x} = (1, 0)$,
$f(x, y) = 100(y - x^2)^2 + (1 - x)^2$
$m_k(\boldsymbol{p}) = 601p_1^2 + 100p_2^2 - 400p_1p_2 + 400p_1 - 200p_2 + 100.$
The minimum of the unconstrained quadratic model is $\boldsymbol{x}^* = (0, 1)$ (dashed line), the minimum of TR model with $\Delta = 0.4$ is pointed by the full line.



Trust Region Methods

The trust-region approach requires us to solve a sequence of subproblems in which the objective function and constraint are both quadratic.

When B_k is positive definite and $||B_k^{-1}\nabla f_k|| \leq \Delta_k$ the solution is simply the unconstrained minimum $\boldsymbol{p}_k^B = -B_k^{-1}\nabla f_k$ of the quadratic model $m_k(\boldsymbol{p})$. In this case, we call $\boldsymbol{p}_k^B$ the full step.

The choice of Δ_k is based on the ratio between the *actual* and the *predicted* reduction

$$\rho_k = \frac{f(\boldsymbol{x}_k) - f(\boldsymbol{x}_{k+1})}{m_k(\boldsymbol{0}) - m_k(\boldsymbol{p}_k)}.$$

Trust Region Methods

Trust Region Method;

Choose $\mathbf{x}_0 \in \mathbb{R}^n$; $k := 0$; $\overline{\Delta} > 0$, $\Delta_0 \in (0, \overline{\Delta})$; $\eta \in [0, \frac{1}{4}]$;

While $\nabla f(\mathbf{x}_k) \neq \emptyset$;

{

$\mathbf{p}_k := \mathtt{argmin}\{m_k(\mathbf{p}),\ \mathtt{s.t.}\ \ ||\mathbf{p}|| \leq \Delta_k\}$;

$\rho_k := \frac{f(x_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)}$;

if $\rho_k < \frac{1}{4}$ **then**

$\Delta_{k+1} := \frac{1}{4}\Delta_k$

else

if $\rho_k > \frac{3}{4}$ e $||\mathbf{p}_k|| = \Delta_k$ **then**

$\Delta_{k+1} := \min\{2\Delta_k, \overline{\Delta}\}$

else

$\Delta_{k+1} := \Delta_k$

if $\rho_k > \eta$ **then**

$\mathbf{x}_{k+1} := \mathbf{x}_k + \mathbf{p}_k$

else

$\mathbf{x}_{k+1} := \mathbf{x}_k$

$k := k + 1$;

}

Trust Region Methods

To have a practical algorithm, we need to focus on solving the trust-region quadratic subproblem.

Theorem The vector $\boldsymbol{p}^*$ is a global solution of the trust-region problem

$$\min_{\boldsymbol{p}\in\mathbb{R}^{\boldsymbol{n}}} m(\boldsymbol{p}) = f(\boldsymbol{x}) + \nabla f^T \boldsymbol{p} + \frac{1}{2}\boldsymbol{p}^T B \boldsymbol{p}, \text{ s.t } ||\boldsymbol{p}|| \leq \Delta.$$

if and only if $\boldsymbol{p}^*$ is feasible and there is a scalar $\lambda \geq 0$ such that the following conditions are satisfied:

$$\begin{aligned} (B + \lambda I)\boldsymbol{p}^* &= -\nabla f, \\ \lambda(\Delta - ||\boldsymbol{p}^*||) &= 0, \\ (B + \lambda I) & \quad \text{is positive semidefinite} \end{aligned}$$

Cauchy point

- Line search methods can be globally convergent even when the step length α_k satisfies fairly loose criteria.
- Likewise trust-region methods require to find an approximate solution $\boldsymbol{p}_k$ that lies within the trust region and gives a sufficient reduction in the model to ensure global convergence.
- The sufficient reduction can be quantified in terms of the Cauchy point, $\boldsymbol{p}^C$.
- The Cauchy point is the minimum of the quadratic model along the $-\nabla f_k$ direction.
- $$\boldsymbol{p}_k^C = -\tau_k \frac{\Delta_k}{||\nabla f_k||} \nabla f_k$$

  where

  $$\tau_k = \begin{cases} 1 & \text{if } \nabla f_k^T B_k \nabla f_k \leq 0 \\ \min\{\tau^*, 1\} & \text{otherwise} \end{cases}$$

Cauchy point

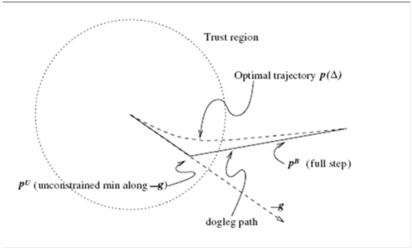
- The Cauchy step $\boldsymbol{p}_k^C$ can be computed in order of $O(n^2)$
- it is of crucial importance in deciding if an approximate solution of the trust-region subproblem is acceptable
- A trust-region method will be globally convergent if its steps $\boldsymbol{p}_k$ give a reduction in the model m_k that is at least some fixed positive multiple of the decrease attained by the Cauchy step
- by always taking the Cauchy point as our step, we are simply implementing the steepest descent method with a particular choice of step length and steepest descent performs poorly
- we need to use information from the matrix B_k.

The Dogleg Method

We apply this method when B is p.d.

- compute $\boldsymbol{p}^B = -B^{-1}\nabla f$
- if $||\boldsymbol{p}^B|| \leq \Delta$ then $\boldsymbol{p}^* = \boldsymbol{p}^B$
- else compute $\boldsymbol{p}^U = -\frac{\nabla f^{\boldsymbol{T}} \nabla f}{\nabla f^{\boldsymbol{T}} B \nabla f} \nabla f$
- build the path $\tilde{\boldsymbol{p}}(\tau)$ for $\tau \in [0, 2]$

$$\tilde{\boldsymbol{p}}(\tau) = \begin{cases} \tau \boldsymbol{p}^U, & \tau \in [0, 1] \\ \boldsymbol{p}^U + (\tau - 1)(\boldsymbol{p}^B - \boldsymbol{p}^U), & \tau \in [1, 2] \end{cases}$$



The Dogleg Method

Property If B is p.d. then

- $\|\tilde{\boldsymbol{p}}(\tau)\|$ is an increasing function of τ, and
- $m(\tilde{\boldsymbol{p}}(\tau))$ is a decreasing function of τ.

- τ can be compute by solving the scalar quadratic equation

$$||\boldsymbol{p}^U + (\tau - 1)(\boldsymbol{p}^B - \boldsymbol{p}^U)||^2 = \Delta^2$$

- there are iterative method for solving the quadratic model
- superlinear convergence rate can be achieved