Lecture 4: Newton's Method and its Modifications

Prof. Krishna R. Pattipati
Dept. of Electrical and Computer Engineering
University of Connecticut
Contact: krishna@engr.uconn.edu **(860) 486-2890**

ECE 6437
Computational Methods for Optimization

Fall 2009
September 22, 2009

Outline of Lecture 4

- **Newton's Method and Quadratic Convergence**
- **How to address Indefinite Hessian case?**
  - Modified Cholesky Decomposition
  - Trust Region Approach
    - Hook step
    - Double dogleg step
- **Least Squares Problem and Gauss-Newton Method**

Newton's Method

- **Idea is to approximate f at point $\underline{x}_k$ by a quadratic surface**

$$f(\underline{x}) \approx f(\underline{x}_k) + \nabla \underline{f}^T(\underline{x}_k)(\underline{x} - \underline{x}_k) + \tfrac{1}{2}(\underline{x} - \underline{x}_k)^T \nabla^2 f(\underline{x}_k)(\underline{x} - \underline{x}_k)$$

- Find next point $\underline{x}_{k+1}$ to minimize quadratic approximation of $f(\underline{x})$ so $\nabla f(\underline{x}) = 0$

$$\Rightarrow \nabla f(\underline{x}_k) + \nabla^2 f(\underline{x}_k)\underbrace{(\underline{x} - \underline{x}_k)}_{\Delta \underline{x}} = 0$$ The solution of this equation is $\underline{x}_{k+1}$

$$\Rightarrow \quad \underline{x}_{k+1} = \underline{x}_k - [\nabla^2 f(\underline{x}_k)]^{-1} \nabla \underline{f}(\underline{x}_k)$$

$$= \underline{x}_k + \underline{d}_k; \quad \text{Note } \alpha_k = 1$$

$\underline{d}_k$ is the solution of $\nabla^2 f(\underline{x}_k)\, \underline{d}_k = -\nabla \underline{f}(\underline{x}_k) = -\underline{g}_k$

- Algorithm: Given $\underline{x}_0$

  Do while not converged

  Solve $\nabla^2 f(\underline{x}_k)\, \underline{d}_k = -\underline{g}_k$

  $\underline{x}_{k+1} = \underline{x}_k + \underline{d}_k$

  End do

- When $\underline{x}_k \approx \underline{x}^*$, method works well since surface is nearly quadratic & $\nabla^2 f > 0$. Indeed, for quadratic PD surfaces, Newton's method converges in one step $\Rightarrow$ can expect fast convergence near a minimum. Infact, Newton's method has quadratic convergence rate near $\underline{x}^*$.

Quadratic Convergence (Locally) - 1

- **Def^n: Assume for some scalars ε, *L* and *M***

$$\left.\begin{aligned} &\left\|\nabla^2 f(\underline{x}) - \nabla^2 f(\underline{y})\right\| \le L\left\|(\underline{x} - \underline{y})\right\| \quad \forall \underline{x}, \underline{y} \\ &\text{with} \quad \left\|\underline{x} - \underline{x}^*\right\| < \varepsilon \;;\; \left\|\underline{y} - \underline{x}^*\right\| < \varepsilon \\ &\left\|\left[\nabla^2 f(\underline{x})\right]^{-1}\right\| \le M \quad \forall\ \underline{x} \ni \left\|\underline{x} - \underline{x}^*\right\| < \varepsilon \end{aligned}\right\} \Rightarrow$$

Near minimum $\nabla^2 f(\underline{x})$ & $[\nabla^2 f(\underline{x})]^{-1}$ are bounded

- **Theorem**:
  - If use Newton's method, $\underline{x}_{k+1} = \underline{x}_k - \left[\nabla^2 f(\underline{x}_k)\right]^{-1} \nabla f(\underline{x}_k) \quad \forall \left\|\underline{x}_k - \underline{x}^*\right\| < \varepsilon$

$$\text{then} \quad e_{k+1} \le \frac{LM}{2} e_k^2$$

$$\text{where} \quad e_k = \left\|\underline{x}_k - \underline{x}^*\right\|$$

  - Proof :

$$\underline{x}_{k+1} - \underline{x}^* = \underline{x}_k - \underline{x}^* - \left[\nabla^2 f(\underline{x}_k)\right]^{-1} \nabla \underline{f}(\underline{x}_k)$$

$$= -\left[\nabla^2 f(\underline{x}_k)\right]^{-1} \left\{\nabla^2 f(\underline{x}_k)(\underline{x}^* - \underline{x}_k) + \nabla f(\underline{x}_k)\right\}$$

$$e_{k+1} \le \left\|\left[\nabla^2 f(\underline{x}_k)\right]^{-1}\right\| \left\|\nabla^2 f(\underline{x}_k)(\underline{x}^* - \underline{x}_k) + \nabla f(\underline{x}_k)\right\|$$

Quadratic Convergence (Locally) - 2

- **Recall mean value theorem in integral form:**

$$f(y) = f(x) + \int_0^1 f'\big(x + \alpha(y-x)\big)(y-x)\,d\alpha$$

- Proof:

$$f(y) - f(x) = \int_x^y f'(t)dt \qquad \alpha = \frac{t-x}{y-x} \quad \text{result follows}$$

$$0 = \nabla f(\underline{x}^*) = \nabla \underline{f}(\underline{x}_k) + \int_0^1 \nabla^2 f(\underline{x}_k + \alpha(\underline{x}^* - \underline{x}_k))(\underline{x}^* - \underline{x}_k)d\alpha$$

$$= \nabla \underline{f}(\underline{x}_k) + \nabla^2 f(\underline{x}_k)(\underline{x}^* - \underline{x}_k)$$

$$+ \int_0^1 \Big[\nabla^2 f(\underline{x}_k + \alpha(\underline{x}^* - \underline{x}_k)) - \nabla^2 f(\underline{x}_k)\Big](\underline{x}^* - \underline{x}_k)d\alpha$$

$$\Rightarrow \Big\|\nabla f(\underline{x}_k) + \nabla^2 f(\underline{x}_k)(\underline{x}^* - \underline{x}_k)\Big\| \qquad \boxed{2^{\text{nd}}\ \text{term}}$$

$$= \Big\|\int_0^1 \Big[\nabla^2 f(\underline{x}_k + \alpha(\underline{x}^* - \underline{x}_k)) - \nabla^2 f(\underline{x}_k)\Big](\underline{x}^* - \underline{x}_k)d\alpha\Big\|$$

$$\le L\Big\|\underline{x}^* - \underline{x}_k\Big\|^2 \int_0^1 \alpha d\alpha = \tfrac{L}{2} e_k^2$$

$$\boxed{e_{k+1} \le \frac{LM}{2} e_k^2}$$

Local quadratic convergence

Newton's Method with simple Line Search

- **Problems:**

1. Computation of $\nabla^2 f(\underline{x}_k) \Rightarrow$ use $\nabla^2 f(\underline{x}_0)$ or $\nabla^2 f(\underline{x}_{kp}) \Rightarrow$ periodic evaluation of the Hessian
2. Need not be a descent method in the sense that $f(\underline{x}_{k+1}) \nleq f(\underline{x}_k)$ if $\alpha_k = 1$
   - Solution: Modify

$$\underline{x}_{k+1} = \underline{x}_k - \alpha_k [\nabla^2 f(\underline{x}_k)]^{-1} \nabla \underline{f}(\underline{x}_k); \quad \alpha < 1$$

DAMPED NEWTON'S METHOD

Near $\underline{x}^*$, $\alpha \rightarrow 1$ & the algorithm still has quadratic convergence

Pick α_k via Armijo + quadratic interpolation

Given $\sigma \in (0, 0.5)$, $l = 0.1$

$\alpha_k = 1$

Do while $f(\underline{x}_{k+1}) > f(\underline{x}_k) + \sigma \alpha_k \nabla \underline{f}^T(\underline{x}_k)\, \underline{d}_k$

$$\gamma_k = \frac{-\alpha_k^2 \nabla \underline{f}^T(\underline{x}_k)\underline{d}_k}{2[f(\underline{x}_k + \alpha_k \underline{d}_k) - \alpha_k \nabla \underline{f}^T(\underline{x}_k)\underline{d}_k - f(\underline{x}_k)]}$$

Evaluate $f(\underline{x}_k + \gamma_k \underline{d}_k)$

$$g(\alpha) = a\alpha^2 + b\alpha + c$$
$$\alpha = 0 \Rightarrow g(0) = f(\underline{x}_k) = c$$
$$g'(0) = \nabla \underline{f}^T(\underline{x}_k)\underline{d}_k = b$$
$$g(\alpha_k) = f(\underline{x}_k + \alpha_k \underline{d}_k) = a\alpha_k^2 + b\alpha_k + c$$
$$\Rightarrow a = \frac{f(\underline{x}_k + \alpha_k \underline{d}_k) - \alpha_k \nabla \underline{f}^T(\underline{x}_k)\underline{d}_k - f(\underline{x}_k)}{\alpha_k^2}$$

$$\frac{-b}{2a}$$

Handling Indefinite Hessians

Set

$$\alpha_{k+1} = \begin{cases} \frac{\alpha_k}{2} & \text{if } f(\underline{x}_k + \gamma_k \underline{d}_k) \geq f(\underline{x}_k + \alpha_k \underline{d}_k) \\ \gamma_k & \text{if } f(\underline{x}_k + \gamma_k \underline{d}_k) < f(\underline{x}_k + \alpha_k \underline{d}_k) \;\&\; \gamma_k \geq l\alpha_k \\ l\alpha_k & \text{otherwise} \end{cases}$$

$$\underline{x}_{k+1} = \underline{x}_k + \alpha_k \underline{d}_k$$

$$k = k + 1$$

End do

3. $\nabla^2 f(\underline{x}_k)$ need not be PD

Solutions:

(1) Add a diagonal PD matrix E э $(\nabla^2 f(\underline{x}_k) + E)$ is PD and pick step size $\alpha_k \Rightarrow H_k = [\nabla^2 f(\underline{x}_k) + E]^{-1}$ is PD $\Rightarrow$ Descent Direction

(2) Decide on $\|\underline{x}_{k+1} - \underline{x}_k\| \leq \Delta_k$, then pick μ э

$$\left\| \left(\nabla^2 f(\underline{x}_k) + \mu I \right)^{-1} \nabla \underline{f}(\underline{x}_k) \right\| \leq \Delta_k$$

TRUST REGION APPROACH

Modified Cholesky Decomposition - 1

Method 1 for handling indefinite Hessians

- Suppose found an E э $\nabla^2 f(\underline{x}_k) + E$ is PD

$$\nabla^2 f(\underline{x}_k) + E = L_k L_k^T, \quad L_k \quad \text{lower triangular}$$

$\underline{d}_k$ is the solution of $L_k \underbrace{L_k^T \underline{d}_k}_{\underline{y}_k} = -\underline{g}_k$

$$L_k \underline{y}_k = -\underline{g}_k \quad \text{(i)}$$

$$L_k^T \underline{d}_k = \underline{y}_k \quad \text{(ii)}$$

From (i): $$y_k^{(j)} = (-g_k^{(j)} - \sum_{i=1}^{j-1} L_k^{(j,i)} y_k^{(i)}) / L_k^{(j,j)}; \quad j = 1, 2, \ldots, n$$

From (ii): $$d_k^{(j)} = (y_k^{(j)} - \sum_{i=j+1}^{n} L_k^{(i,j)} y_i) / L_k^{(j,j)}; \quad j = n, n-1, \ldots, 1$$

- We will approach the problem of computing L_k in two steps:
  a. How to compute the Cholesky (square root) factor for a PD matrix?
  b. How to detect indefiniteness and modify factors to make $\nabla^2 f(\underline{x}_k) + E$ PD (or) equivalently how to find E?

Cholesky Algorithm

Cholesky (Square Root) Decomposition of a PD matrix, *F*

$$\begin{bmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & 0 \cdots & 0 \\ \vdots & \vdots & & \\ l_{i1} & l_{i2} & \cdots & l_{ii} \\ \vdots & \vdots & & 0 \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} & \cdots & l_{k1} & \cdots & l_{n1} \\ 0 & l_{22} & \cdots & l_{k2} & \cdots & l_{n2} \\ & 0 & & \vdots & & \\ \vdots & & & l_{kk} & & \vdots \\ & \vdots & & & & \\ 0 & 0 & & \cdots & 0 & l_{nn} \end{bmatrix} = \begin{bmatrix} f_{11} & f_{21} & \cdots & f_{n1} \\ f_{21} & f_{22} & \cdots & f_{n2} \\ . & & \cdots & . \\ f_{i1} & f_{i2} & f_{ii} & \vdots \\ \vdots & \vdots & & \\ f_{n1} & f_{n2} & \cdots & f_{nn} \end{bmatrix}$$

We generate one column at a time: $k = 1,2,..,n$

$$f_{ik} = \sum_{p=1}^{k} l_{ip} l_{kp} = \sum_{p=1}^{k-1} l_{ip} l_{kp} + l_{ik} l_{kk}; \quad i > k$$

$$\Rightarrow l_{ik} = (f_{ik} - \sum_{p=1}^{k-1} l_{ip} l_{kp}) / l_{kk}$$

Also $$l_{kk} = (f_{kk} - \sum_{p=1}^{k-1} l_{kp}^2)^{\frac{1}{2}}$$

All computations can be done in place
$\Rightarrow L$ can replace F
$\Rightarrow$ No extra storage

Example of Cholesky Decomposition

- **Example**

$$F = \begin{bmatrix} 4 & 6 & -2 \\ 6 & 10 & 1 \\ -2 & 1 & 21 \end{bmatrix} \qquad L = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix}$$

$$l_{11} = 2$$

$$l_{21} = 3 \qquad l_{22} = \sqrt{f_{22} - l_{21}^{\ 2}} = \sqrt{10 - 9} = 1$$

$$l_{31} = -1 \qquad l_{32} = \frac{1 - l_{31}.\, l_{21}}{l_{22}} = \frac{1 - (-1)(3)}{1} = 4$$

$$l_{33} = f_{33} - l_{31}^2 - l_{32}^2 = 21 - 1 - 16 = 4$$

$$l_{33} = 2$$

$$\underbrace{\begin{bmatrix} 2 & 0 & 0 \\ 3 & 1 & 0 \\ -1 & 4 & 2 \end{bmatrix}}_{L} \underbrace{\begin{bmatrix} 2 & 3 & -1 \\ 0 & 1 & 4 \\ 0 & 0 & 2 \end{bmatrix}}_{L^T} = \underbrace{\begin{bmatrix} 4 & 6 & -2 \\ 6 & 10 & 1 \\ -2 & 1 & 21 \end{bmatrix}}_{F}$$

Modified Cholesky Algorithm - 1

Algorithm Cholesky:

For $k = 1, 2, \ldots, n$ Do

$$f_{kk} \leftarrow (f_{kk} - \sum_{p=1}^{k-1} f_{kp}^2)^{\frac{1}{2}}$$

For $i = k+1, \ldots, n$ Do

$$f_{ik} \leftarrow (f_{ik} - \sum_{p=1}^{k-1} f_{ip} f_{kp}) / f_{kk}$$

end

end

Computations: n square roots +

$$\sum_{k=1}^{n}[(k-1)+k(n-k)]$$

$$= \frac{n(n-1)}{2} + \frac{n^2(n+1)}{2} - \frac{n(n+1)(2n+1)}{6}$$

$$= O(\frac{n^3}{6})$$

- Key: If F is indefinite, it manifests as negative $(f_{kk} - \sum_{p=1}^{k-1} f_{kp}^2)$

  $\Rightarrow$ need to take square root of a negative number!!

- We will modify factors as follows:

$$f_{kk} = \begin{cases} \sqrt{f_{kk} - \sum_{p=1}^{k-1} f_{kp}^2} & \text{if} \quad f_{kk} - \sum_{p=1}^{k-1} f_{kp}^{\,2} > 0 \\ \sqrt{\mu} & \text{otherwise} \end{cases}$$

Modified Cholesky Algorithm -2

- How to select μ?

$\mu = r\omega_k$ where $\omega_k = \max_k \left[\left| f_{kk} \right| \right] = \max_k \left\{ \left| \frac{\partial^2 f}{\partial x_k^2} \right| \right\}$; Initial $r = 10^{-4}$

Update r depending on step size during Newton iterations

$$r = \begin{cases} 5r & \text{if } \alpha < .2 \Rightarrow \text{ not close to quadratic; so, increase } \mu \\ r & \text{if } .2 \le \alpha \le .9 \Rightarrow \text{ leave it alone} \\ \frac{r}{5} & \text{if } \alpha > .9 \Rightarrow \text{ close to quadratic, } F \text{ is close to Hessian} \end{cases}$$

- **Modified Cholesky:** all computations in place!!

For $k = 1, 2, \ldots, n$ Do

if $(f_{kk} - \sum_{p=1}^{k-1} f_{kp}^2) > 0$

$f_{kk} \leftarrow (f_{kk} - \sum_{p=1}^{k-1} f_{kp}^2)^{\frac{1}{2}}$

else

$f_{kk} \leftarrow (\mu)^{1/2}$

end if

For $i = k+1, \ldots, n$ Do

$f_{ik} \leftarrow (f_{ik} - \sum_{p=1}^{k-1} f_{ip} f_{kp}) / f_{kk}$

end

end

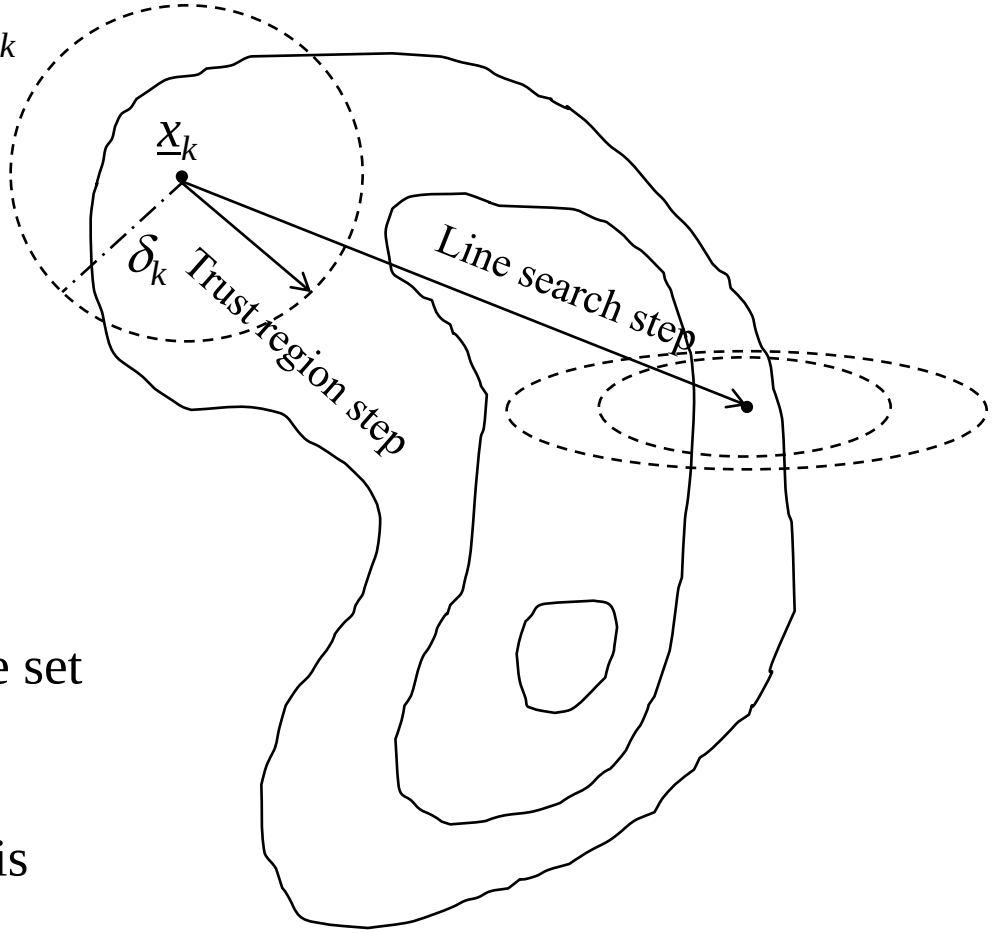
Trust Region Approach – 1

Method 2 for obtaining a globally convergent Newton iteration

- Previous approach fixed $\underline{d}_k$ and found shorter step length α_k (<1). An alternate approach is to fix the step length ("trust region" denoting the region in R^n around $\underline{x}_k$ in which a quadratic approximation is ok) and then choose a search direction $\underline{d}_k$. Basically, we set

$$\underline{x}_{k+1} = \underline{x}_k + \underline{d}_k$$

where $\underline{d}_k$ is ə a quadratic approximation around $\underline{x}_k$ is a minimum and $\|\underline{d}_k\|_2 \le \delta_k$



Trust Region Approach – 2

- Mathematically, the problem is

$$\min_{\underline{d}_k} \; f(\underline{x}_k) + \underline{g}_k{}^T \underline{d}_k + \frac{1}{2} \underline{d}_k{}^T F_k \underline{d}_k$$

$$s.t. \quad \|\underline{d}_k\|_2 \le \delta_k$$

$$\underline{g}_k = \nabla \underline{f}(\underline{x}_k)$$

$$F_k = \nabla^2 f(\underline{x}_k)$$

$\delta_k \sim$ radius of the trust region … how to get it & update it later

- Solution:

$$\underline{d}_k(\lambda) = \begin{cases} -F_k{}^{-1} \underline{g}_k & \text{if} \quad \underline{g}_k^T F_k^{-2} \underline{g}_k \le \delta_k{}^2 \\ -(F_k + \lambda I)^{-1} \underline{g}_k & \text{otherwise} \end{cases}$$

$$\text{where} \quad \lambda \text{ is } \ni \; \|\underline{d}_k(\lambda)\|_2 = \delta_k$$

$$\lambda \ge 0$$

Why?

A Simple Constrained Optimization Problem

- Proof:

$$L(\underline{d}_k, \lambda) = f(\underline{x}_k) + \underline{g}_k{}^T \underline{d}_k + \frac{1}{2}\underline{d}_k{}^T F_k \underline{d}_k + \frac{1}{2}\lambda(\underline{d}_k{}^T \underline{d}_k - \delta_k^2)$$

$$F_k \underline{d}_k + \lambda \underline{d}_k + \underline{g}_k = 0$$

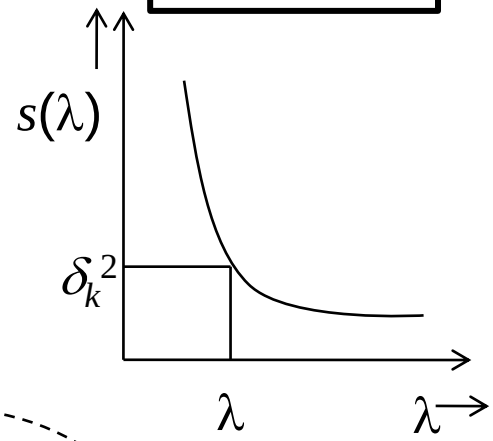
$$\Rightarrow \underline{d}_k = -(F_k + \lambda I)^{-1} \underline{g}_k$$

If when $\lambda = 0$, $\underline{g}_k{}^T F_k{}^{-2} \underline{g}_k \le \delta_k$ OK

otherwise, find λ from

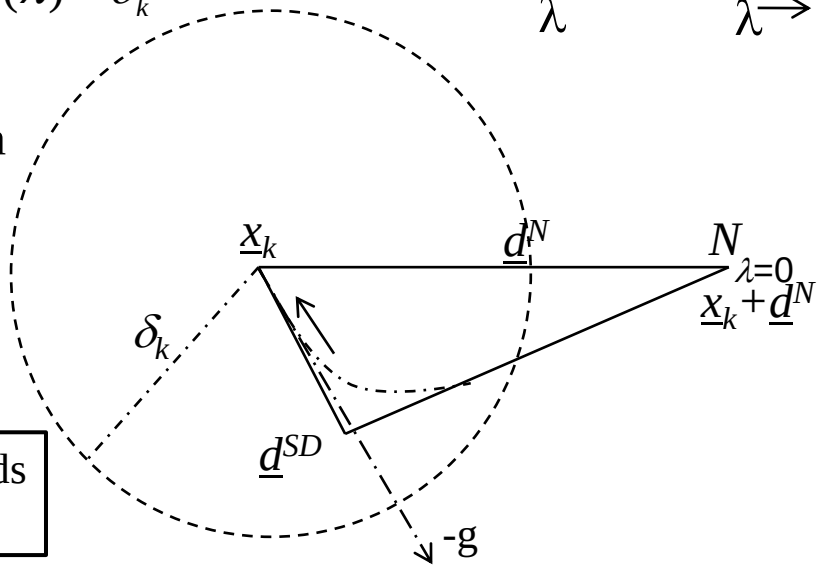
$$\underline{d}_k{}^T \underline{d}_k = \delta_k{}^2 \Rightarrow \underline{g}_k{}^T (F_k + \lambda I)^{-2} \underline{g}_k = s(\lambda) = \delta_k{}^2$$

λ= Lagrange multiplier



- Observations:
  - $\lambda = 0 \Rightarrow$ Newton direction
    $\lambda = \infty \Rightarrow$ steepest descent
  - find $\lambda \ni \|\underline{d}_k\|_2 = \delta_k$

$$\Rightarrow \underline{g}_k{}^T (F_k + \lambda I)^{-2} \underline{g}_k = \delta_k{}^2 = s(\lambda)$$



Can solve for $\underline{d}_k$ via Conjugate Gradient methods of Lecture 5 *(see Nocedal and Wright*, pp. 75)

How to Get λ ?

- Can show $S(\lambda)$ is convex, in fact

$$\frac{ds(\lambda)}{d\lambda} = -2\underline{g}_k^T (F_k + \lambda I)^{-3} \underline{g}_k < 0 \quad \forall \lambda > 0$$

$$\frac{d^2 s(\lambda)}{d\lambda} = 6\underline{g}_k^T (F_k + \lambda I)^{-4} \underline{g}_k > 0$$

$$AA^{-1} = I$$

$$\frac{dA}{d\alpha} A^{-1} + A\frac{dA^{-1}}{d\alpha} = 0$$

$$A^2 A^{-2} = I$$

$$2A\frac{dA}{d\alpha} A^{-2} + A^2 \frac{dA^{-2}}{d\alpha} = 0$$

- Mechanization

Initial $\underline{x}_0,\ \delta_0$

Do until convergence

Find $\lambda \ni \underline{d}_k = -(F_k + \lambda I)^{-1} \underline{g}_k \quad \& \quad \|\underline{d}_k\| = \delta_k$

$\underline{x}_{k+1} = \underline{x}_k + \underline{d}_k$

If $f(\underline{x}_{k+1}) > f(\underline{x}_k)$

calculate new value of δ_k

end if

end do

Newton's Method for Updating λ

- **Two questions remain:** (1) computation of $\underline{d}_k$
  (2) update of $\underline{\delta}_k$

- Newton's method for finding λ

$$\lambda_{l+1} = \lambda_l - \frac{(s(\lambda_l) - \delta_k^{\,2})}{s'(\lambda_l)}$$

Hook step

$$\underline{d}_k(\lambda) = \begin{cases} -F_k^{-1}\underline{g}_k & \text{if} \quad \underline{g}_k^T F_k^2 \underline{g}_k \le \delta_k^{\,2} \\ -(F_k + \lambda I)^{-1}\underline{g}_k & \text{otherwise} \end{cases}$$

$$= \lambda_l + \frac{1}{2}\frac{(\underline{d}_k^{\,T}(\lambda_l)\,\underline{d}_k(\lambda_l) - \delta_k^{\,2})}{\underline{d}_k^{\,T}(\lambda_l)(F_k + \lambda_l I)^{-1}\underline{d}_k(\lambda_l)}$$

Usually 1-2 Iterations suffice

use λ_{k-1}^{p} from previous iteration as λ_0

$$\lambda_0 = \lambda_{k-1}^{p} + \frac{1}{2}\frac{(\underline{d}_k^{\,T}(\lambda_{k-1}^{p})\underline{d}_k(\lambda_{k-1}^{p}) - \delta_k^{\,2})}{\underline{d}_k^{\,T}(\lambda_l)(F_k + \lambda_{k-1}^{p} I)^{-1}\underline{d}_k(\lambda_{k-1}^{p})}$$

- Second approach: Double dogleg step

Recall

$\lambda \to 0 \quad \Rightarrow \quad$ Newton direction

$\lambda \to \infty \quad \Rightarrow \quad$ steepest descent

Double Dogleg Step - 1

- Note also that
  - as $\lambda \uparrow \|\underline{d}_k(\lambda)\| \downarrow \Rightarrow \delta_k$ must have been small. So, we must take a safe course, viz., steepest descent.
  - as $\lambda \downarrow \|\underline{d}_k(\lambda)\| \uparrow \Rightarrow \delta_k$ must have been large $\Rightarrow$ Newton is ok, since we are confident that quadratic approximation to $f(\underline{x})$ at $\underline{x}_k$ is valid over a sphere of radius $\delta_{k,}$

Therefore, one strategy is to approximate the trajectory by a weighted sum of $\underline{d}^{SD}$ and $\underline{d}^{N}$.

$$\underline{d}_k = (1-\beta)\underline{d}^{SD} + \beta \underline{d}^{N} \quad \text{for appropriate choice of } \beta$$
$$= \underline{d}^{SD} + \beta(\underline{d}^{N} - \underline{d}^{SD})$$

we find $\underline{x}^{SD}$ э $\underline{d}^{SD} = -\alpha^{*}\underline{g}_k$

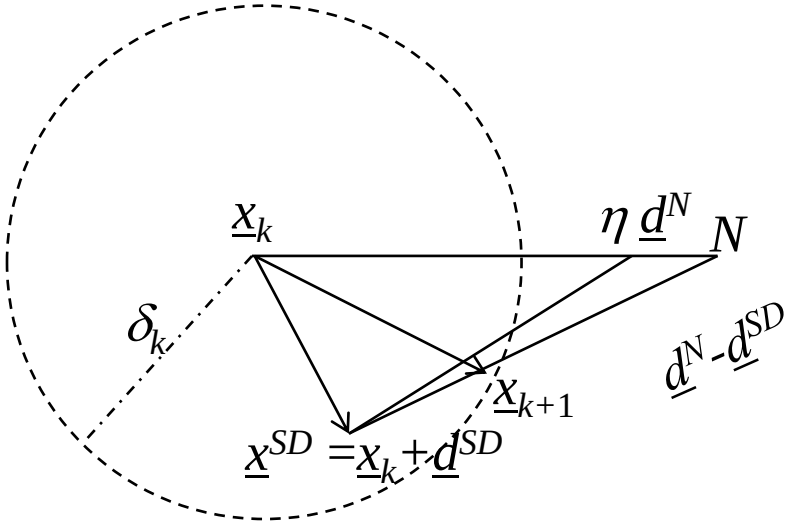
$$\alpha^{*} = \frac{\underline{g}_k{}^{T}\underline{g}_k}{\underline{g}_k{}^{T} F_k \underline{g}_k}$$

Double Dogleg Step - 2

- **Two possibilities :**
  - If $\delta_k \le \alpha^* \left\| \underline{g}_k \right\| = \dfrac{(\underline{g}_k{}^T \underline{g}_k)^{\frac{3}{2}}}{\underline{g}_k{}^T F_k \underline{g}_k}$

    then δ_k must have been small enough

    set $\underline{x}_{k+1} = \underline{x}_k - \delta_k \dfrac{\underline{g}_k}{\left\| \underline{g}_k \right\|_2}$



  - If $\delta_k > \alpha^* \left\| \underline{g}_k \right\|_2$, take steepest descent step to $\underline{x}^{SD}$ and move along *SD-N* until $\underline{x}_{k+1}$

$$\Rightarrow \underline{x}_k + \underline{d}^{SD} + \beta(\underline{d}^N - \underline{d}^{SD}) = \underline{x}_{k+1} = (1-\beta)\underline{x}_{SD} + \beta \underline{x}_N$$

$$\ni \left\| \underline{d}^{SD} + \beta(\underline{d}^N - \underline{d}^{SD}) \right\|_2 = \delta_k$$

Some times, one uses $\eta\, \underline{d}^N$ in place of $\underline{d}^N$. $\eta \approx .8 \dfrac{\left\| \underline{d}^{SD} \right\|}{\left\| \underline{d}^N \right\|} + .2$ works good

Why?

Practical Issues of Dogleg - 1

- **Computation of β:**

$$(1-\beta)^2(\underline{d}^{SD})^T\underline{d}^{SD}+\beta^2(\underline{d}^{N})^T\underline{d}^{N}+2\beta(1-\beta)(\underline{d}^{N})^T\underline{d}^{SD}=\delta_k^2$$

$$\beta^2(\underline{d}^{N}-\underline{d}^{SD})^T(\underline{d}^{N}-\underline{d}^{SD})+2\beta(\underline{d}^{N}-\underline{d}^{SD})^T\underline{d}^{SD}+(\underline{d}^{SD})^T\underline{d}^{SD}=\delta_k^2$$

$$\text{or } \beta=\frac{(\underline{d}^{SD}-\underline{d}^{N})^T\underline{d}^{SD}+\sqrt{((\underline{d}^{N}-\underline{d}^{SD})^T\underline{d}^{SD})^2+(\delta_k^2-(\underline{d}^{SD})^T\underline{d}^{SD})\bullet(\underline{d}^{N}-\underline{d}^{SD})^T(\underline{d}^{N}-\underline{d}^{SD})}}{(\underline{d}^{N}-\underline{d}^{SD})^T(\underline{d}^{N}-\underline{d}^{SD})}$$

- **Practicalities: Why $\eta\, d_N$?**
  - Note:

$$\left\|\underline{d}^{SD}\right\|_2=\|\alpha^*\underline{g}_k)\|_2=\frac{\|\underline{g}_k\|_2^3}{\underline{g}_k^T F_k\underline{g}_k}=\frac{\|\underline{g}_k\|_2^3(\underline{g}_k^T F_k^{-1}\underline{g}_k)}{(\underline{g}_k^T F_k\underline{g}_k)(\underline{g}_k^T F_k^{-1}\underline{g}_k)}$$

$$\le\frac{\|\underline{g}_k\|_2^3\|\underline{g}_k\|_2\|F_k^{-1}\underline{g}_k\|_2}{(\underline{g}_k^T F_k\underline{g}_k)(\underline{g}_k^T F_k^{-1}\underline{g}_k)}=\frac{[\underline{g}_k^T\underline{g}_k]^2\|\underline{d}^{N}\|_2}{(\underline{g}_k^T F_k\underline{g}_k)(\underline{g}_k^T F_k^{-1}\underline{g}_k)}$$

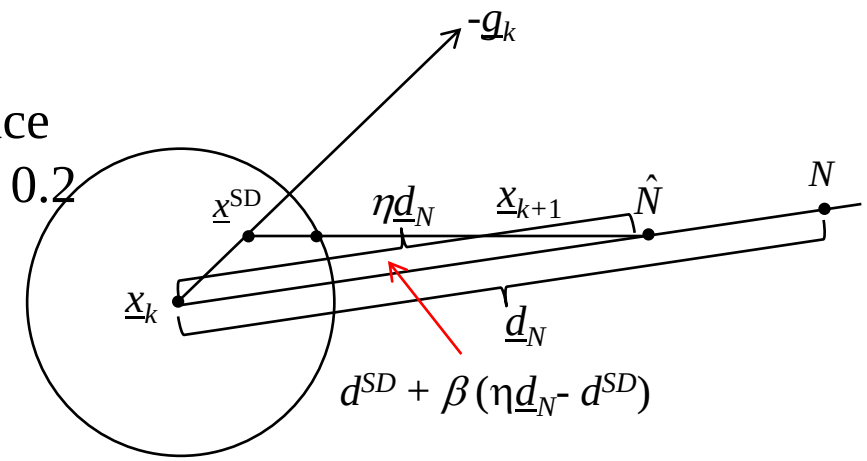
Practical Issues of Dogleg - 2

- Let $\underline{\mu} = F_k^{\frac{1}{2}} \underline{g}_k$ and $\underline{\nu} = F_k^{-\frac{1}{2}} \underline{g}_k$

$$\Rightarrow \| \underline{d}^{SD} \|_2 \leq \frac{[\underline{u}^T \underline{\nu}]^2}{(\underline{u}^T \underline{u})(\underline{\nu}^T \underline{\nu})} \| \underline{d}^N \|_2 = \gamma \| \underline{d}^N \|_2; \quad \gamma = \cos^2 \theta \leq 1$$

$$\| \underline{d}^{SD} \|_2 \leq \gamma \| \underline{d}^N \|_2 \leq \| \underline{d}^N \|_2$$

- Sometimes, instead of taking N, we take $\hat{N} = \underline{x}_k + \eta \underline{d}^N \quad (\eta \leq 1)$ for double dogleg step
  - $\Rightarrow$ Replace $\underline{d}^N$ by $\eta \underline{d}^N$ in computing β
  - $\Rightarrow$ Computational experience suggests that $\eta = 0.8\gamma + 0.2$ works good
- Computational load per step: $O(n^2)$



Illustrative Example

- **Example:** $f(\underline{x}) = \frac{1}{2}x_1^2 + x_2^2 \qquad \underline{x}_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \qquad \delta = 1.3$

$$\underline{g}_0 = \begin{pmatrix} x_1 \\ 2x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \end{pmatrix} \qquad F_0 = \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}$$

$$\underline{x}^{SD} = \underline{x}_0 - \alpha^* \underline{g}_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \frac{5}{9}\begin{pmatrix} 1 \\ 2 \end{pmatrix} = \begin{pmatrix} \frac{4}{9} \\ -\frac{1}{9} \end{pmatrix} \qquad \underline{d}^{SD} = \underline{x}^{SD} - \underline{x}_0 = \begin{pmatrix} -\frac{5}{9} \\ -\frac{10}{9} \end{pmatrix}$$

- Newton Step

$$\underline{d}^N = -F_0^{-1}\underline{g}_0 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}, \qquad \left\|\underline{d}^N\right\|_2 = \sqrt{2} > 1.3$$

$$\gamma = \frac{\left\|\underline{d}^{SD}\right\|}{\left\|\underline{d}^N\right\|} = \frac{\sqrt{\frac{125}{81}}}{\sqrt{2}} = .921 \qquad \eta = .2 + .8\gamma = .9368$$

$$\text{want} \quad \left\| \underbrace{\underline{d}^{SD} + \beta(\eta \underline{d}^N - \underline{d}^{SD})}_{\underline{d}_k} \right\|_2 = 1.3 \qquad \beta \approx .44$$

$$\underline{x}_{k+1} = \underline{x}_k + \underline{d}^{SD} + \beta(\eta \underline{d}^N - \underline{d}^{SD}) = \begin{pmatrix} .28 \\ -.19 \end{pmatrix}$$

How to Update δ_k ? - 1

- **How to update δ: similar to Armijo + quadratic interpolation**

Suppose $f(\underline{x}_{\delta_k}) > f(\underline{x}_k) + \sigma \nabla \underline{f}^T(\underline{x}_k)(\underline{x}_{\delta_k} - \underline{x}_k), \quad \sigma \sim 10^{-4}$

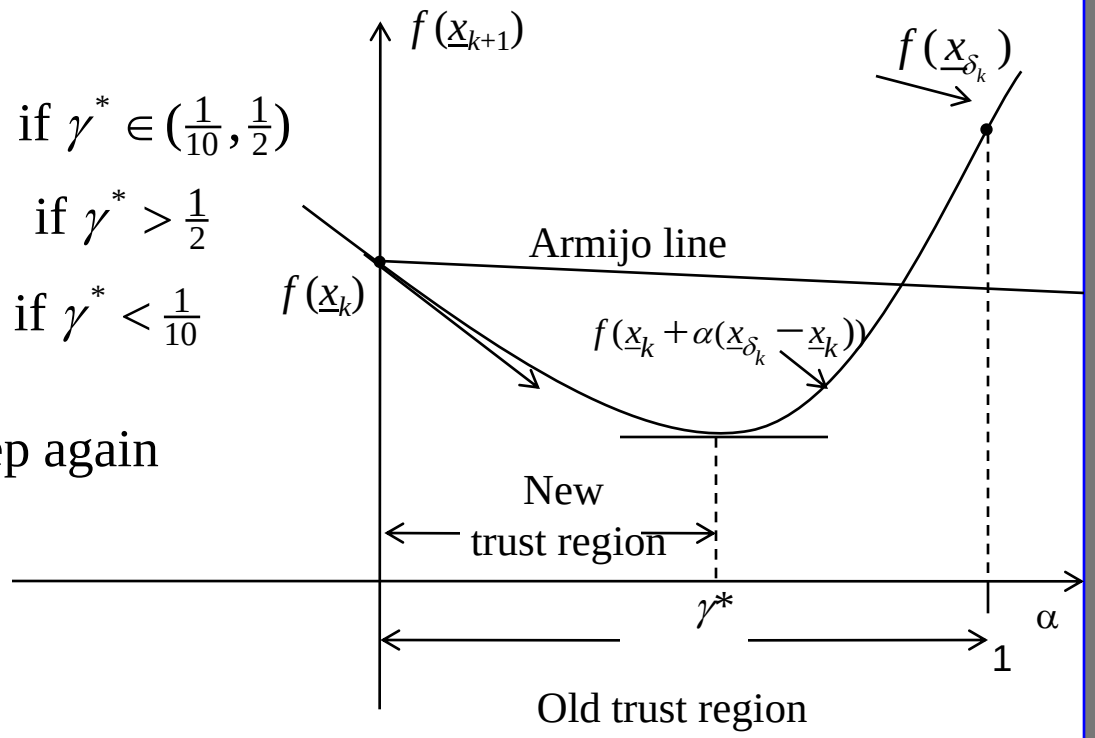
$$\underline{x}_{\delta_k} - \underline{x}_k = \begin{cases} \underline{d}_k \text{ if Hook step} \\ \underline{d}_{SD} + \beta(\eta \underline{d}_N - \underline{d}_{SD}) \text{ if Dogleg} \end{cases}$$

$$\gamma^* = \frac{-\nabla \underline{f}^T(\underline{x}_k)(\underline{x}_{\delta_k} - \underline{x}_k)}{2[f(\underline{x}_{\delta_k}) - f(\underline{x}_k) - \nabla \underline{f}^T(\underline{x}_k)(\underline{x}_{\delta_k} - \underline{x}_k)]}$$

use

$$\delta_{k+1} = \begin{cases} \gamma^* \| \underline{x}_{\delta_k} - \underline{x}^* \|_2 & \text{if } \gamma^* \in (\frac{1}{10}, \frac{1}{2}) \\ 2\delta_k & \text{if } \gamma^* > \frac{1}{2} \\ \frac{\delta_k}{10} & \text{if } \gamma^* < \frac{1}{10} \end{cases}$$

Do the double dogleg step again



How to Update δ_k ? - 2

- On the other hand, if

$$f(\underline{x}_{\delta_k}) \le f(\underline{x}_k) + \sigma \nabla \underline{f}^T(\underline{x}_k)(\underline{x}_{\delta_k} - \underline{x}_k)$$

If $\underline{x}_{\delta_k}$ is a Newton step (Hook step with λ=0), let $\underline{x}_{k+1} = \underline{x}_{\delta_k}$, update $\delta_k = \| \underline{d}_N \|_2$, next dogleg step etc.

If $\underline{x}_{\delta_k}$ is not a Newton step (Double dogleg or λ>0), how to update δ_k ?

$$f(\underline{x}_{k+1}) - f(\underline{x}_k) = \Delta f$$

$r \approx 0.75$

$$(\Delta f)_{pred} = \underline{g}_k^T(\underline{x}_{k+1} - \underline{x}_k) + \tfrac{1}{2}(\underline{x}_{k+1} - \underline{x}_k)^T F_k(\underline{x}_{k+1} - \underline{x}_k)$$

$$|\Delta f| \ge r\,|\Delta f_{pred}| \Rightarrow \text{Quadratic approx. is very good} \Rightarrow \delta_{k+1} = 2\delta_k$$

$$|\Delta f| < (1-r)\,|\Delta f_{pred}| \Rightarrow \text{approx. not good} \Rightarrow \delta_{k+1} = \delta_k / 4$$

$$(1-r)\,|\Delta f_{pred}| \le |\Delta f| \le r\,|\Delta f_{pred}| \Rightarrow \text{ok approx.} \Rightarrow \delta_{k+1} = \delta_k$$

Hook step Implementation

Initial $\underline{x}_0$, δ

Do until convergence

(1) Find λ ə $\underline{d}_k = -(F_k + \lambda I)^{-1}\underline{g}_k$ & $\| \underline{d}_k \| \le \delta_k$

(2) If $f(\underline{x}_k + \underline{d}_k) < f(\underline{x}_k)$

$$\underline{x}_{k+1} = \underline{x}_k + \underline{d}_k$$

$$ratio = \frac{f(\underline{x}_k) - f(\underline{x}_{k+1})}{-\underline{g}_k^T \underline{d}_k - \frac{1}{2}\underline{d}_k^T F_k \underline{d}_k} = \frac{(-\Delta f)}{(-\Delta f_{pred})}$$

go to step 3

else $\delta_k = .25 \| \underline{d}_k \|$

go to step1

end if

(3)

$$\delta_{k+1} = \begin{cases} .25 \| \underline{d}_k \| & \text{if} \quad ratio \le .25 \\ 2\delta_k & \text{if} \quad ratio \ge .75 \ \& \ \| \underline{d}_k \| = \delta_k \\ \delta_k & \text{otherwise} \end{cases}$$

End do

Double Dogleg Step Implementation -1

Initial $\underline{x}_0$, δ_0

Do until convergence

Compute $\underline{d}^{SD} = -\alpha_k \underline{g}_k$; $\underline{d}^{N} = -F_k^{-1} \underline{g}_k$;

(1) If $\| \underline{d}^{N} \| < \delta_k$

$$\underline{d}_k = \underline{d}_N$$

$$\tilde{\underline{x}} = \underline{x}_k + \underline{d}_k$$

elseif $\| \underline{d}^{SD} \| > \delta_k$

$$\underline{d}_k = -\delta_k \underline{g}_k \Big/ \| \underline{g}_k \|$$

$$\tilde{\underline{x}} = \underline{x}_k + \underline{d}_k$$

else

$$\underline{d}_k = \underline{d}^{SD} + \beta(\eta \underline{d}^{N} - \underline{d}^{SD})$$

$$\tilde{\underline{x}} = \underline{x}_k + \underline{d}_k$$

end if

Double Dogleg Step Implementation -2

(2) If $f(\tilde{x}) < f(\underline{x}_k)$

$$\underline{x}_{k+1} = \tilde{x}$$

$$ratio = \frac{f(\underline{x}_{k+1}) - f(\underline{x}_k)}{-\underline{g}_k^T \underline{d}_k - \frac{1}{2}\underline{d}_k^T F_k \underline{d}_k}$$

go to step 3

else

$$\delta_k = .25 \| \underline{d}_k \|$$

go to (1)

end if

(3)

$$\delta_{k+1} = \begin{cases} .25 \| \underline{d}_k \| & \text{if } ratio \le .25 \\ 2\delta_k & \text{if } ratio \ge .75 \ \& \ \| \underline{d}_k \| = \delta_k \text{ or } \underline{d}_k = \underline{d}^N \\ \delta_k & \text{if } .25 \le ratio \le .75 \end{cases}$$

end do

Gauss-Newton Method -1

Gauss-Newton Method:

$$\min_{\underline{x}\in R^n} f(\underline{x}) = \tfrac{1}{2}\sum_{i=1}^{m} g_i^2(\underline{x}) = \tfrac{1}{2}\underline{g}^T(\underline{x})\underline{g}(\underline{x}) \quad \underline{g} = \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_m \end{bmatrix}, \quad m \gg n$$

- These problems arise in system identification, Neural Networks,…

  $\underline{x}_{k+1} = \underline{x}_k - \alpha_k [\nabla \underline{\underline{g}}(\underline{x}_k) \quad \nabla^T \underline{\underline{g}}(\underline{x}_k)]^{-1} \nabla \underline{\underline{g}}(\underline{x}_k)\underline{g}(\underline{x}_k)$

  $\nabla \underline{\underline{g}}(\underline{x}_k) = [\nabla \underline{g}_1(\underline{x}_k)\ \nabla \underline{g}_2(\underline{x}_k) \cdots \nabla \underline{g}_m(\underline{x}_k)]$ an n by m matrix

  $\nabla \underline{\underline{g}}(\underline{x}_k)$ has full row rank $\Rightarrow$ $\nabla \underline{\underline{g}}(\underline{x}_k)\nabla \underline{\underline{g}}^T(\underline{x}_k)$ is PD

- When $\nabla \underline{\underline{g}}(\underline{x}_k)$ is not full rank ($\Rightarrow$ over parameterization), then

  $D_k = [\nabla \underline{\underline{g}}(\underline{x}_k)\nabla \underline{\underline{g}}^T(\underline{x}_k) + \mu I]^{-1}; \quad \mu > 0$

  Levenberg-Marquardt algorithm

Gauss-Newton Method - 2

Remarks:

- Note the similarity to trust region approach
- When $m = n$ and $\nabla \underline{\underline{g}}(\underline{x}_k)$ is invertible, the method is equivalent to Newton's method for solving $g_i(\underline{x}) = 0 \qquad 1 \le i \le n$;

$$\underline{x}_{k+1} = \underline{x}_k - \alpha_k [\nabla \underline{\underline{g}}^T(\underline{x}_k)]^{-1} \underline{g}(\underline{x}_k); \quad \alpha_k = 1$$

- Gauss-Newton method exhibits quadratic convergence near minimum when $m = n$

If $\| \nabla \underline{\underline{g}}(\underline{x}) - \nabla \underline{\underline{g}}(\underline{y}) \| \le L \| \underline{x} - \underline{y} \| \quad \forall \underline{x}, \underline{y} \in N(\underline{x}^*, \varepsilon)$

$\| \nabla \underline{\underline{g}}^{-1}(\underline{x}) \| \le M \quad \forall \underline{x} \in N(\underline{x}^*, \varepsilon)$

Then $\underline{x}_{k+1} = \underline{x}_k - [\nabla \underline{\underline{g}}^T(\underline{x}_k)]^{-1} \underline{g}(\underline{x}_k)$

has quadratic convergence rate, i.e.,

$\| \underline{x}_{k+1} - \underline{x}^* \| \le \frac{LM}{2} \| \underline{x}_k - \underline{x}^* \|^2$

Gauss-Newton Method - 3

- Gauss-Newton method when applied to $f(\underline{x}) = \underline{g}^T(\underline{x})\underline{g}(\underline{x})$ is NOT Newton's method faraway from minimum

$$\nabla^2 f(\underline{x}) = \nabla \underline{\underline{g}}(\underline{x}) \nabla \underline{\underline{g}}^T(\underline{x}) + \underbrace{\sum_{i=1}^{m} \nabla^2 g_i(\underline{x}) g_i(\underline{x})}$$

Neglected in Gauss-Newton

Near $\underline{x}^*, g_i(\underline{x}) \cong 0 \Rightarrow$ Gauss-Newton $\approx$ Newton

and inherits properties of Newton's method

Least Squares Problem - 1

- **Linear Least Squares Problems:**

$$g_i(\underline{x}) = -y_i + \underline{a}_i^T \underline{x} \quad i = 1, 2, ..., m \quad \text{residual}$$

$$f(\underline{x}) = \tfrac{1}{2} \sum_{i=1}^{m} (-y_i + \underline{a}_i^T \underline{x})^2$$

$$\nabla \underline{f} = \sum_{i=1}^{m} (-y_i + \underline{a}_i^T \underline{x}) \underline{a}_i$$

$$F = \sum_{i=1}^{m} \underline{a}_i \underline{a}_i^T$$

From first order condition: $(\sum_{i=1}^{m} \underline{a}_i \underline{a}_i^T) \underline{x} = \sum_{i=1}^{m} \underline{a}_i y_i$

$$\Rightarrow AA^T \underline{x} = {}_n{}^m A \underline{y} \quad \Rightarrow \underline{x} = (AA^T)^{-1} A \underline{y}$$

Least Squares Problem - 2

SD: $\underline{x}_{k+1} = \underline{x}_k - \alpha_k \sum_{i=1}^{m} (\underline{a}_i^T \underline{x} - y_i)\underline{a}_i$

NM: $\underline{x}_{k+1} = \underline{x}_k + (\sum_{i=1}^{m} \underline{a}_i \underline{a}_i^T)^{-1}\{A\underline{y} - (\sum_{i=1}^{m} \underline{a}_i \underline{a}_i^T)\underline{x}_k\}$

$= (AA^T)^{-1} A\underline{y}$ in one step

- Conjugate gradient method is another way (next Lecture)
- Incremental SD (this is how neural networks are trained. Basis of so-called back propagation algorithms):

Initialize $\underline{x}_0^{(0)} = \underline{x}_0$

For k =0,1,2,... until convergence

For i =1,2,..,m

$\underline{x}_k^{(i)} = \underline{x}_k^{(i-1)} - \alpha_k (\underline{a}_i^T \underline{x}_k^{(i-1)} - y_i)\underline{a}_i$

end

$\underline{x}_{k+1}^{(0)} = \underline{x}_k^{(m)}$

end

Evidently, optimization techniques provide more sophisticated methods for Neural Network training

Summary

- **Newton's Method and Quadratic Convergence**
- **How to address Indefinite Hessian case?**
  - Modified Cholesky Decomposition
  - Trust Region Approach
    - Hook step
    - Double dogleg step
- **Least Squares Problem and Gauss-Newton Method**