Lecture-8

Conjugate Direction Algorithm
(Solution of Linear System or
Minimization of Quadratic Function)

Conjugate Gradient

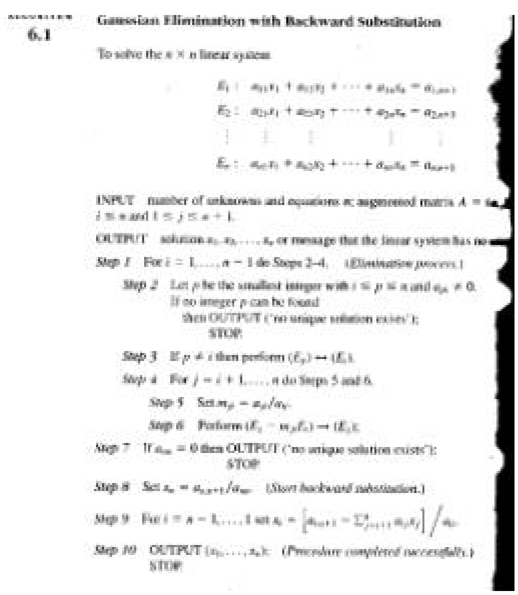
- Linear conjugate gradient: for solving linear systems Ax=b with PD matrix, A.
  - Hestenes & Stiefel, 1950s
- Non-linear conjugate gradient: for solving large-scale non-linear optimization problems.
  - Fletcher and Reeves, 1960s

Solution of A linear System

- Gaussian Elimination, Backward Substitution
- Matrix Factorization
- Iterative Techniques

$$Ax = b$$

$$\sum_{j=1}^{n} (a_{ij} x_j) = b_i \quad \text{for } i = 1, 2, \ldots, n$$



ALGORITHM 6.1

Gaussian Elimination with Backward Substitution

To solve the $n \times n$ linear system

$$\begin{aligned} E_1: &\quad a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = a_{1,n+1} \\ E_2: &\quad a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = a_{2,n+1} \\ &\quad \vdots \\ E_n: &\quad a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = a_{n,n+1} \end{aligned}$$

INPUT number of unknowns and equations n; augmented matrix $A =$ a
$i \le n$ and $1 \le j \le n + 1$.

OUTPUT solution $x_1, x_2, \ldots, x_n$ or message that the linear system has no

Step 1 For $i = 1, \ldots, n - 1$ do Steps 2–4. (*Elimination process.*)

Step 2 Let p be the smallest integer with $i \le p \le n$ and $a_{pi} \ne 0$.
If no integer p can be found
then OUTPUT ('no unique solution exists');
STOP.

Step 3 If $p \ne i$ then perform $(E_p) \leftrightarrow (E_i)$.

Step 4 For $j = i + 1, \ldots, n$ do Steps 5 and 6.

Step 5 Set $m_{ji} = a_{ji}/a_{ii}$.

Step 6 Perform $(E_j - m_{ji}E_i) \rightarrow (E_j)$;

Step 7 If $a_{nn} = 0$ then OUTPUT ('no unique solution exists');
STOP.

Step 8 Set $x_n = a_{n,n+1}/a_{nn}$. (*Start backward substitution.*)

Step 9 For $i = n - 1, \ldots, 1$ set $x_i = \left[a_{i,n+1} - \sum_{j=i+1}^{n} a_{ij}x_j\right] \Big/ a_{ii}$.

Step 10 OUTPUT $(x_1, \ldots, x_n)$; (*Procedure completed successfully.*)
STOP.

Iterative Methods for Solving Linear Systems

- For large sparse system Gaussian Elimination and Backward substitution is not suitable.
- Approximate solution using iterative methods

Jacobi

$$\sum_{j=1}^{n} (a_{ij} x_j) = b_i \quad \text{for } i = 1,2,\dots,n$$

$$x_i^k = \frac{\sum_{j=1,\, j\neq i}^{n} (-a_{ij} x_j^{k-1}) + b_i}{a_{ii}} \quad \text{for } i = 1,2,\dots,n$$

$$X = TX + C$$

Gauss-Seidel

$$x_i^k = x_i^{k-1} + \frac{r_{ii}^k}{a_{ii}}$$

$$x_i^k = x_i^{k-1} + w\,\frac{r_{ii}^k}{a_{ii}}$$

$$\sum_{j=1}^{n} (a_{ij} x_j) = b_i \quad \text{for } i = 1,2,\dots,n$$

$$x_i^k = \frac{-\sum_{j=1}^{i-1} (a_{ij} x_j^k) - \sum_{j=i+1}^{n} (a_{ij} x_j^{k-1}) + b_i}{a_{ii}} \quad \text{for } i = 1,2,\dots,n$$

$$X = TX + C$$

Interpretation of Gauss-Seidel

$$x_i^k = \frac{-\sum_{j=1}^{i-1} (a_{ij} x_j^k) - \sum_{j=i+1}^{n} (a_{ij} x_j^{k-1}) + b_i}{a_{ii}} \quad \text{for } i = 1,2,\dots,n$$

$$r = b - A\tilde{x}$$

$$r_{ii}^k + a_{ii} x_i^{k-1} = a_{ii} x_i^k$$

$$r_{ii} = b_i - \sum_{j=1}^{i-1} (a_{ij} x_j^k) - \sum_{j=i+1}^{n} (a_{ij} x_j^{k-1}) - a_{ii} x_i^{k-1}$$

$$r_{ii}^k + a_{ii} x_i^{k-1} = a_{ii} x_i^k$$

Interpretation of Gauss-Seidel

$$r_{ii}^{k} + a_{ii} x_i^{k-1} = a_{ii} x_i^{k}$$

$$x_i^k = x_i^{k-1} + \frac{r_{ii}^k}{a_{ii}}$$

$$x_i^k = x_i^{k-1} + w\,\frac{r_{ii}^k}{a_{ii}}$$

$$x_i^k = x_i^{k-1} + \frac{w}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^k - \sum_{j=i+1}^{n} (a_{ij} x_j^{k-1}) - a_{ii} x_i^{k-1} \right]$$

$$x_i^k = (1-w) x_i^{k-1} + \frac{w}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^k - \sum_{j=i+1}^{n} (a_{ij} x_j^{k-1}) \right]$$

SOR (Successive Over Relaxation)

$$\sum_{j=1}^{n} (a_{ij} x_j) = b_i \quad \text{for } i = 1,2,\ldots,n$$

$$x_i^k = (1-w) x_i^{k-1} + \frac{w}{a_{ii}}\left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^k - \sum_{j=i+1}^{n} (a_{ij} x_j^{k-1}) \right]$$

$$w > 1$$

Theorem

If A is a PD and $0 < w < 2$ then SOR method converges for Any choice of initial approximation of solution x^{0}.

Theorem

If A is a PD and tri-diagonal, then

$$r(T_g) = r(T_j) < 1$$

Then optimal choice of w

$$w = \frac{2}{1 + \sqrt{1 - r(T_g)}}$$

$$w = \frac{2}{1 + \sqrt{1 - r(T_j)^2}}$$

Maximum eigen value

Conjugate Gradient

$$Ax = b$$ A is symmetric PD. (1)

Or minimize the following function:

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

$$\nabla f(x) = Ax - b = r(x)$$ $r(x)$ is the residual

$S = \{p_0, p_1, \ldots, p_{n-1}\}$ The set S is conjugate wrt A if

$$x_{k+1} = x_k + a_k p_k$$

$$p_i^T A p_j = 0 \qquad \forall \mathrm{i} \neq \mathrm{j}$$ $$a_k = -\frac{\nabla f_k^T p_k}{p_k^T A p_k}$$

Linear Independence

S is linearly independent

if $s_0 p_0 + s_1 p_1 + \ldots s_{n-1} p_{n-1} = 0$
then $s_0 = s_1 = s_2 = \ldots s_{n-1} = 0$

Conjugate set is also linearly independent.

$$p_i^T A p_j = 0 \qquad \forall \mathrm{i} \neq \mathrm{j}$$

Conjugate Direction Method

$$x_{k+1} = x_k + a_k p_k$$ Line search

$$p_i^T A p_j = 0 \qquad \forall i \neq j$$

$$a_k = -\frac{\nabla f_k^T p_k}{p_k^T A p_k}$$ 1D minimizer of a quadratic function

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

Convergence Rate of Steepest Descent

$$\frac{d}{da} f(x_k - a g_k) = \frac{d}{da}\left(\frac{1}{2}(x_k - a g_k)^T Q (x_k - a g_k) - b^T (x_k - a g_k)\right) = 0$$

$$= -(x_k - a g_k)^T Q g_k + b^T g_k = 0$$

$$-x_k^T Q g_k + a g_k^T Q g_k + b^T g_k = 0$$

$$a g_k^T Q g_k = x_k^T Q g_k - b^T g_k$$

$$a = \frac{x_k^T Q g_k - b^T g_k}{g_k^T Q g_k}$$

$$a = \frac{(x_k^T Q - b^T) g_k}{g_k Q g_k} \qquad \nabla f(x) = Qx - b$$

From Lecture-5

$$a = \frac{\nabla f_k^T \nabla f_k}{\nabla f_k^T Q \nabla f_k}$$

$$x_{k+1} = x_k - a_k \nabla f_k \qquad x_{k+1} = x_k - \frac{\nabla f_k^T \nabla f_k}{\nabla f_k^T Q \nabla f_k} \nabla f_k$$

Conjugate Direction Method

$$a = \frac{x_k{}^T Q g_k - b^T g_k}{g_k{}^T Q g_k}$$

$$a = \frac{(x_k{}^T A - b^T)(-p_k)}{(-p_k) A (-p_k)}$$

$$a_k = -\frac{\nabla f_k^T p_k}{p_k^T A p_k} \qquad \nabla f(x) = Ax - b = r(x)$$

$$a_k = -\frac{r_k^T p_k}{p_k^T A p_k} \qquad p_i^T A p_j = 0 \qquad \forall \mathrm{i} \neq \mathrm{j}$$

Theorem 5.1

For any x^0 the sequence $\{x_k\}$ generated by the conjugate direction algorithm, converges to the solution x^* of the linear system in at most n steps.

Proof

$$x_{k+1} = x_k + a_k p_k \qquad a_k = -\frac{r_k^T p_k}{p_k^T A p_k}$$

$$x_k = x_0 + a_0 p_0 + a_1 p_1 + \ldots + a_{k-1} p_{k-1}$$

$$x_k - x_0 = a_0 p_0 + a_1 p_1 + \ldots + a_{k-1} p_{k-1}$$

Proof

S is linearly independent

Therefore:

$$x^* - x_0 = s_0 p_0 + s_1 p_1 + \ldots s_{n-1} p_{n-1}$$

$$p_k^T A(x^* - x_0) = p_k^T A(s_0 p_0 + s_1 p_1 + \ldots s_{n-1} p_{n-1})$$

$$p_k^T A(x^* - x_0) = (0 \ + 0 + \ldots + s_k p_k^T A p_k + \ldots + 0)$$ conjugate

$$s_k = \frac{p_k^T A(x^* - x_0)}{p_k^T A p_k}$$

Proof

$$x_{k+1} = x_k + a_k p_k \qquad a_k = -\frac{r_k^T p_k}{p_k^T A p_k}$$

$$x_k = x_0 + a_0 p_0 + a_1 p_1 + \ldots + a_{k-1} p_{k-1}$$

$$x_k - x_0 = a_0 p_0 + a_1 p_1 + \ldots + a_{k-1} p_{k-1}$$

$$p_k^T A(x_k - x_0) = 0$$

$$p_k^T A x_k = p_k^T A x_0$$

$$p_k^T A(x^* - x_0) = p_k^T A(x^* - x_k) = p_k^T (b - A x_k) = -p_k^T r_k$$

$$p_k^T A(x^* - x_0) = -p_k^T r_k$$

Proof

$$p_k^T A(x^* - x_0) = -p_k^T r_k$$

$$s_k = \frac{p_k^T A(x^* - x_0)}{p_k^T A p_k} \qquad a_k = -\frac{r_k^T p_k}{p_k^T A p_k}$$

Therefore:

$$s_k = a_k$$

QED

Interpretation of Theorem 5.1

If A is a diagonal matrix, then we can minimize (1-D) the function along coordinate axes in n iterations.

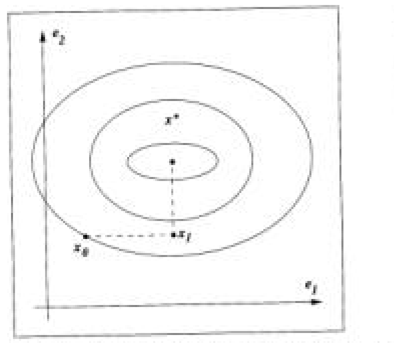
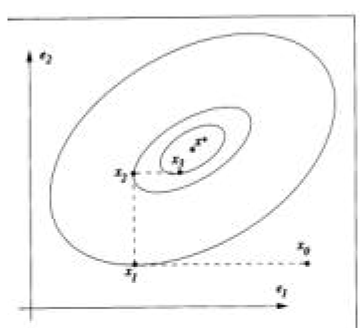


Figure 5.1 Successive minimizations along the coordinate directions
minimizer of a quadratic with a diagonal Hessian in n iterations.

Interpretation of Theorem 5.1

If A is not a diagonal matrix, then we can not minimize the function along Coordinate axes in n iterations.



igure 5.2 Successive minimization along coordinate axes does not find the solution
n iterations, for a general convex quadratic.

Transformed Problem

Let

$$\hat{x} = S^{-1}x \qquad \text{where} \qquad S = [p_0, p_1, \ldots, p_{n-1}]$$

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

$$\hat{j}(\hat{x}) = f(S\hat{x}) = \frac{1}{2}\hat{x}^T (S^T AS)\hat{x} - (S^T b)^T \hat{x}$$

By conjugacy $S^T AS$ is a diagonal matrix.

Now we can minimize along coordinate directions in transformed space.

However, each coordinate direction in transformed space correspond to the conjugate direction in the original space due to $\hat{x} = S^{-1}x$

Therefore, we conclude the conjugate direction algorithm converges in *n* steps.

Basic Properties of the CG: How do we select conjugate directions

Each direction is chosen to be a linear combination of the steepest descent direction and the previous direction.

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

$$p_k = -r_k + b_k p_{k-1}$$

$$p_{k-1}^T A p_k = -r_k p_{k-1}^T A + b_k p_{k-1}^T A p_{k-1}$$

$$b_k = \frac{r_k^T A p_{k-1}}{p_{k-1}^T A p_{k-1}}$$

Algorithm 5.1

Given x_0;

set $r_0 \leftarrow Ax_0 - b,\ \ p_0 \leftarrow -r_0, k \leftarrow 0$

While $r_k \neq 0$

$$a_k \leftarrow -\frac{r_k^T p_k}{p_k^T A p_k};$$

$$x_{k+1} \leftarrow x_k + a_k p_k;$$

$$r_{k+1} \leftarrow Ax_{k+1} - b;$$

$$b_{k+1} \leftarrow \frac{r_{k+1}^T A p_k}{p_k^T A p_k};$$

$$p_{k+1} \leftarrow -r_{k+1} + b_{k+1} p_k;$$

$$k \leftarrow k + 1;$$

end (*while*)

p_0 is steepest descent