Lecture-15

Homework, Rate of Convergence of CG, preconditioning, FR-GC, PR-GC

Homework (Due April 17)

- 5.1
- 5.9
- Proof for Theorem 5.5 (see the slides)

Theorem 5.4

If A has only r distinct eigenvalues, then the CG iteration will terminate at the solution in at most r iterations.

Theorem 5.5

If A has eigenvalues $l_1 \leq l_2 \leq \ldots \leq l_n$ we have

$$\| x_{k+1} - x^* \|_A^2 \leq \left(\frac{l_{n-k} - l_1}{l_{n-k} + l_1} \right)^2 \| x_0 - x^* \|_A^2$$

Eigenvalues

$l_1, \ldots, l_{n-k}, l_{n-k+1}, \ldots, l_n$

Proof

Eigenvalues

$l_1,\ldots, l_{n-k}, l_{n-k+1},\ldots, l_n$

Select polynomial $\overline{P}_k(l)$ of degree k such that

Q has roots at k largest eigenvalues

$l_n, l_{n-1},\ldots, l_{n-k+1}$

As well as at mid point l_1 and l_{n-k}

$$Q_{k+1}(l) = 1 + l\overline{P}_k(l)$$

Maximum value attained by Q on the remaining eigenvalues is precisely

$$\left(\frac{l_{n-k} - l_1}{l_{n-k} + l_1}\right)$$

Homework: show this

(C) $\| x_{k+1} - x^* \|_A^2 \le \min_{P_k} \max_{1\le i\le n} [1 + l_i P_k(l_i)]^2 \| x_0 - x^* \|_A^2$

$$\| x_{k+1} - x^* \|_A^2 \le \left(\frac{l_{n-k} - l_1}{l_{n-k} + l_1}\right)^2 \| x_0 - x^* \|_A^2$$

Proof

Assume eigenvalues $l_{n-k+1},,\ldots, l_n$ take k distinct values:

$t_1 < t_2,\ldots,< t_k$ and $t_{k+1} = \dfrac{l_{n-k} + l_1}{2}$

Define polynomial:

$$Q_{k+1}(l) = \frac{(-1)^{k+1}}{t_1 t_2 \ldots t_k t_{k+1}} (l - t_1)(l - t_2)\ldots(l - t_k)(l - t_{k+1})$$

$Q_{k+1}(l_i) = 0$ for $i = n-k+1,\ldots,n$

$Q_{k+1}(0) = 1$

$Q_{k+1}(l) - 1$ Is polynomial of degree $k+1$ with root at

$\overline{P}_k = \dfrac{(Q_{k+1}(l) - 1)}{l}$ Degree k

$\min_{P_k} \max_{1\le i\le n} [1 + l_i P_k(l_i)]^2$ (B)

$$0 \le \min_{P_k} \max_{1\le i\le n} [1 + l_i P_k(l_i)]^2 \le \max_{1\le i\le n} [1 + l_i \overline{P}_k(l_i)]^2 = \left(\frac{l_{n-k} - l_1}{l_{n-k} + l_1}\right)^2$$

Example

$$\{l_1,\ldots,l_{n-m},\}\{l_{n-m+1},\ldots,l_n\}$$

m largest eigenvalues

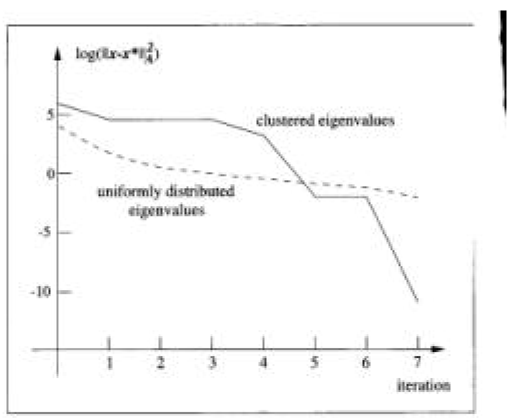
Clustered around 1

$$\| x_{m+1} - x^* \|_A \approx e \| x_0 - x^* \|_A$$

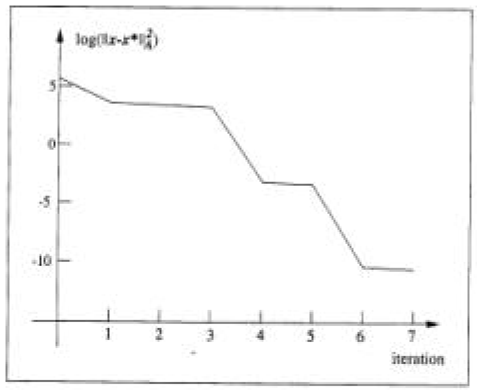
For small value of
CG will converge in only
$m+1$ steps.

$$\| x_{k+1} - x^* \|_A^2 \leq \left(\frac{l_{n-k} - l_1}{l_{n-k} + l_1} \right)^2 \| x_0 - x^* \|_A^2$$

Example



The matrix has five large eigenvalues with all smaller eigenvalues clustered around .95 and 1.05



N=14, has four clusters of eigenvalues: single eigenvalues at 140, 120, a cluster of 10 eigenvalues very close to 10 with the remaining eigenvalues clustered between .95 and 1.05.

Convergence using Condition number

$$\| x_{k+1} - x^* \|_A^2 \leq \left(\frac{\sqrt{k(A)} - 1}{\sqrt{k(A)} + 1} \right)^2 \| x_0 - x^* \|_A^2$$

$$k(A) = \| A \|_2 \| A^{-1} \|_2 = \frac{l_1}{l_n}$$

$$l_1 > l_n$$

Convergence Rate of Steepest Descent: Quadratic Function

$$\| x_{k+1} - x^* \|_Q^2 \leq \left(\frac{l_n - l_1}{l_n + l_1} \right)^2 \| x_k - x^* \|_Q^2$$

Theorem 3.3

$l_1 < l_n$

What is desirable?

- Matrix *A* should have either:
  - Few distinct eigenvalues
  - Few distinct eigenvalues, and few clusters of eigenvalues
  - The condition number of *A* is small

Preconditioning

- If the matrix A dose not have favorable eigenvalues, we can transform the problem such that eigenvalue distribution of a matrix in the transformed problem improves.

Preconditioning

Original problem:

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

Transformation:

$$\hat{x} = Cx \qquad C^{-1}\hat{x} = x$$

Transformed problem:

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

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

Preconditioning

$$(C^{-T}AC^{-1})\hat{x} = (C^{-T}b)$$

Select C such that:

condition number of $C^{-T}AC^{-1}$ is much smaller than the original matrix A.

The eigenvalues of $C^{-T}AC^{-1}$ are clustered

One possible preconditioner is $C = L^T$, such that $A = LL^T$

$$C^{-T}AC^{-1} = L^{-1}AL^{-T} = L^{-1}LL^{T}L^{-T} = I$$

Algorithm 5.3 (Preconditioned CG)

Given x_0, preconditioner M;

set $r_0 \leftarrow Ax_0 - b$,

solve $My_0 = r_0$, for y_0;

$p_0 \leftarrow -r_0,\ k \leftarrow 0$

While $r_k \neq 0$

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

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

$$r_{k+1} \leftarrow r_k + \alpha_k A p_k;$$

$$My_{k+1} = r_{k+1}$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T y_{k+1}}{r_k^T y_k};$$

$$p_{k+1} \leftarrow -y_{k+1} + \beta_{k+1} p_k;$$

$$k \leftarrow k+1;$$

end(*while*)

$$M = C^T C$$

Homework:convert 5.2 to 5.3 using preconditioning

Non-linear CG

- Two changes in linear GC
  - Perform line search for step length
  - Replace residual r by the gradient of the function
- Two algorithms:
  - FR (Fletcher-Reves) (1964)
  - PR (Polak-Rebiere) (1969)
- The difference is only in b

Algorithm 5.4 (FR-CG)

Given x_0;
evaluate $f_0 = f(x_0), \nabla f_0 = \nabla f(x_0)$
set $p_0 \leftarrow -\nabla f_0, k \leftarrow 0$
While $\nabla f_k \neq 0$
  compute a_k;
  $x_{k+1} \leftarrow x_k + a_k p_k$;
  evaluate ∇f_{k+1};
  $b_{k+1}^{FR} \leftarrow \dfrac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k}$;
  $p_{k+1} \leftarrow -\nabla f_{k+1} + b_{k+1}^{FR} p_k$;
  $k \leftarrow k+1$;
end(*while*)

5.4

Given x_0;
set $r_0 \leftarrow Ax_0 - b,\ p_0 \leftarrow -r_0, k \leftarrow 0$
While $r_k \neq 0$
  $a_k \leftarrow -\dfrac{r_k^T r_k}{p_k^T A p_k}$;
  $x_{k+1} \leftarrow x_k + a_k p_k$;
  $r_{k+1} \leftarrow r_k + a_k A p_k$;
  $b_{k+1} \leftarrow \dfrac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$;
  $p_{k+1} \leftarrow -r_{k+1} + b_{k+1} p_k$;
  $k \leftarrow k+1$;
end(*while*)

5.2

Question

- How do we guarantee that the search direction is a descent direction for any arbitrary non-linear function?

Choice of step length

$$p_{k+1} \leftarrow -\nabla f_{k+1} + \boldsymbol{b}_{k+1}^{FR} p_k$$

The search direction p_k may fail to be a descent direction, unless step length satisfies certain conditions.

$$p_k = -\nabla f_k + \boldsymbol{b}_k^{FR} p_{k-1}$$
$$\nabla f_k^T p_k = -\nabla f_k^T \nabla f_k + \boldsymbol{b}_k^{FR} \nabla f_k^T p_{k-1}$$
$$\nabla f_k^T p_k = -\| \nabla f_k \|^2 + \boldsymbol{b}_k^{FR} \nabla f_k^T p_{k-1}$$

If $\nabla f_k^T p_{k-1} = 0$, then $\nabla f_k^T p_k < 0$, therefore p_k is a descent direction (Theorem 5.2 for quadratic functions).

If $\nabla f_k^T p_{k-1} \neq 0$,then the second term may dominate, and $\nabla f_k^T p_k > 0$

Choice of step length

To solve this problem, we will require step length satisfies the following Strong Wolf's conditions:

$$f(x_k + \alpha p_k) \le f(x_k) + c_1 \alpha \nabla f_k^T p_k, \quad c_1 \in (0,1)$$

$$|\nabla f(x_k + \alpha p_k)^T p_k| \le c_2 |\nabla f_k^T(x_k) p_k|, \quad 0 < c_1 < c_2 < \frac{1}{2}$$

We will show in Lemma 5.6 that the Wolf's conditions guarantee:

$$\nabla f_k^T p_k < 0$$

Polak-Ribiere

$$\beta_{k+1}^{PR} \leftarrow \frac{\nabla f_{k+1}^T(\nabla f_{k+1} - \nabla f_k)}{\nabla f_k^T \nabla f_k}$$

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

They are the same if the f is quadratic function, and line search is exact, since gradients (residuals) are mutually orthogonal by Theorem 5.3

For general non-linear functions, numerical experience indicates PR-CG tends to be more robust and efficient.

For PR-CG strong wolf conditions do not guarantee that p_k is always a descent direction.

Other Choices

$$b_{k+1}^{+} = \max(b_{k+1}^{PR}, 0)$$

This can satisfy descent property

$$b_{k+1}^{HS} \leftarrow \frac{\nabla f_{k+1}^T(\nabla f_{k+1} - \nabla f_k)}{(\nabla f_{k+1} - \nabla f_k)^T p_k}$$

Yet another choice

Quadratic Termination & Restarts

Non-linear CG methods preserves their connections to linear CG. Quadratic interpolation along p_k guarantees that for a quadratic function, the step length is exact, that is non-linear CG reduces to linear GC.

Restart non-linear GC after every n steps:

$$p_{k+1} \leftarrow -\nabla f_{k+1} + b_{k+1}^{FR} p_k$$

$$p_{k+1} \leftarrow -\nabla f_{k+1}$$

It is steepest descent. It erases the old memory,which may not be beneficial.

Quadratic Termination & Restarts

N-step Quadratic convergence can be proved with restarts

If the function is strongly quadratic in a neighborhood of a solution
Assume the algorithm is converging to solution,
the iterations will enter the quadratic region,
at some point algorithm will be restarted, that point onward the behavior will be similar to linear GC.
convergence will occur within n steps
Restart is important, because finite termination is subject to p_0 equal to the negative gradient.

Even if the function is not strongly quadratic,
it can be approximated by Taylor series, if it is smooth.
Therefore substantial progress can be made toward the solution

Restarts

Practically restarts are not implemented.
Because NGC is used for function, where n is very large
often solution is reached much before n steps.

Restarts based on other strategies

Theorem 5.3

$$\frac{|\nabla f_k^T \nabla f_{k-1}|}{\| \nabla f_k \|^2} \geq n, \quad n = .1$$

Two consecutive gradients are far from orthogonal.

$b_{k+1}^{+} = \max(b_{k+1}^{PR}, 0)$ Another restarting strategy

Results

Termination conditions: $\|\nabla f_k\|_\infty < 10^{-5}(1+|f_k|)$
Or 10, 000 iterations

Problem	n	Alg FR it/f-g	Alg PR it/f-g	Alg PR+ it/f-g	mod
CALCVAR3	200	2808/5617	2631/5263	2631/5263	0
GENROS	500	*	1068/2151	1067/2149	1
XPOWSING	1000	533/1102	212/473	97/229	3
TRIDIA1	1000	264/531	262/527	262/527	0
MSQRT1	1000	422/849	113/231	113/231	0
XPOWELL	1000	568/1175	212/473	97/229	3
TRIGON	1000	231/467	40/92	40/92	0

$c_1 = 10^{-4},\ c_2 = .1$

Given x_0;
evaluate $f_0 = f(x_0), \nabla f_0 = \nabla f(x_0)$
set $p_0 \leftarrow -\nabla f_0, k \leftarrow 0$
While $\nabla f_k \neq 0$
compute α_k;
$x_{k+1} \leftarrow x_k + \alpha_k p_k$;
evaluate ∇f_{k+1};
$\beta_{k+1}^{FR} \leftarrow \frac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k}$;
/* $\beta_{k+1}^{FR} \leftarrow \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{\nabla f_k^T \nabla f_k}$; */
$p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1}^{FR} p_k$;
$k \leftarrow k+1$;
end(*while*)

Results

- Practically PR-GC is preferred over FR-GC.
- We can prove (Theorem 5.8) the global convergence of FR-GC.
- But, we can not prove the global convergence of PR-GC.
- Not only that, but theorem by Powel (1984):
  - PR-GC can cycle infinitely without approaching a solution point, even in an ideal line search is used!

Results

- Also by Powell (1976):
  - If the algorithm enters a region in which the function is 2-D quadratic, the angle between gradient and the search direction p_k stays constant. Therefore if the this angle is close to 90 degrees, FR method can be slower than the steepest descent.
  - PR behaves differently: if a very small step is generated, the next search direction tends to be steepest descent. This feature prevents a sequence of tiny steps.